Towards Authenticated and Private Computer and Wireless Communications

Basel Alomair

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

University of Washington

2011

Program Authorized to Offer Degree: Electrical Engineering

University of Washington
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Basel Alomair

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of the Supervisory Committee:

Radha Poovendran

Reading Committee:

Radha Poovendran

James Ritcey

Tadayoshi Kohno

Date:

Signature______________________________

Date__________________________________

University of Washington

Abstract

Towards Authenticated and Private Computer and Wireless Communications

Basel Alomair

Chair of the Supervisory Committee:
Professor Radha Poovendran
Electrical Engineering

In the first part of the dissertation, we investigate the problem of message authentication when the privacy of the message is also required. First, we show that there is a redundancy in the computations performed by the authentication and encryption primitives and show how to remove such redundancy to improve the overall efficiency. Then, we propose a new security model that captures essential differences between standard authentication algorithms and those coupled with encryption algorithms to compose an authenticated encryption scheme. We use the new model to design the first secure keyless message authentication algorithm in the literature. Furthermore, we utilize the underlying encryption to propose the first secure authentication technique in which only a small portion of the message needs to be processed, without affecting the integrity of the entire message. Lastly, we investigate the security of authentication based on universal hash-function families with computations performed over integer rings; we show that the probability of successful forgery is proportional to the reciprocal of the smallest prime factor of the used modulus.

In the second part, we shift the attention to radio frequency identification (RFID) systems. We propose the first authentication protocol for RFID systems that can be proven information-theoretically secure, given a reasonable relaxation of the adversary's capabilities. Then, we propose techniques that are designed specifically to take advantage of the fact that exchanged messages in RFID systems are short strings to authenticate them efficiently. Lastly, we investigate the problem of efficient identification and propose the first

privacy-preserving protocol that can identify encrypted identifiers in constant-time.

In the third part, we investigate the problem of preserving source location privacy in wireless sensor networks (WSNs). We propose a new framework for modeling, analyzing, and designing anonymous sensor networks. We use the proposed framework to map the problem of statistical source anonymity to the classic binary hypothesis testing with nuisance parameters. We show that with the appropriate data transformation, contrary to the current belief, existing statistical solutions can reveal critical information about the locations of reported events. We then investigate a new approach to improve the anonymity of existing statistical solutions.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Number Page

LIST OF TABLES

ACKNOWLEDGMENTS

First, I would like to thank my advisor Radha Poovendran for his direction, advice, and support throughout my Ph.D. studies at the University of Washington. I am deeply indebted to Radha for showing me the excitement that can be found in collaborative academic research and for allowing me to work independently. His guidance equipped me with the tools necessary to be successful in the future of my academic career. He taught me that hard work, perseverance, and believing in myself are the keys for success. I would also like to thank the members of my Ph.D. supervisory committee: Jim Ritcey, Yoshi Kohno, Tom Anderson, and Neal Koblitz.

I would like to extend my thanks to Jim Ritcey and Payman Arabshahi for their valuable support throughout my Ph.D. studies. I would also like to extend my thanks to Loukas Lazos for his guidance and collaboration during the early stage of my research. I would like to thank Jorge Cuellar at Siemens for his insight and collaboration on the problems of efficient identification in RFID systems and anonymous wireless sensor networks. I would like to thank Krishna Sampigethaya at the Boeing company for his collaboration on the work of digital signatures. Finally, I would like to acknowledge the generous support by the following funding sources: ARO PECASE, W911NF-05-1-0491; ARO MURI, W911NF-07-1-0287; and KACST Fellowship.

I would like to collectively acknowledge the current and former members of the Network Security Lab. I would like to thank the members of the previous incarnation of the NSL, including Loukas Lazos, Mingyan Li, Javier Salido, Krishna Sampigethaya, and Patrick Tague who warmly welcomed me into the group and provided unending support during the early years of my Ph.D. studies. I would also like to thank the current instance of the NSL, including Andrew Clark, Sidharth Nabar, David Slater, Phillip Lee, Tamara Bonaci, He (Tony) Wu, and Chouchang (Jack) Yang who worked with me during the later part of my time at UW.

Last, but not least, I would like to thank my family for the inspiration and unconditional support that has allowed me to become the person that I am. For my father who is the main reason behind my pursuit of Ph.D. studies, for my mom for her continuous praying, and for my wife for being by my side throughout this journey (I would not have done it without you, Rasha).

Chapter 1
INTRODUCTION

With today's technology, an immeasurable amount of data is being transmitted wirelessly and/or over the Internet. In such scenarios, users have no means of controlling the path their messages take in their routes to the intended receivers. Since some, if not all, links that messages take in their routes can be insecure, it is desirable, even necessary in many applications, to protect exchanged messages against malicious users. Message integrity and privacy are amongst the most concerning problems when communicating through insecure channels. (Message integrity and authenticity will be used synonymously throughout the dissertation; similarly, message privacy and confidentiality will be used synonymously.) In this dissertation, we analyze and propose solutions that provide authenticated and private communications over insecure channels.

We begin this chapter by giving brief descriptions of the studied problems. We then give a road map to the remainder of the dissertation.

1.1 Summary of Investigated Problems and Contributions

In this dissertation, we investigate three problems in the security of modern communication systems. Namely, we investigate the problem of authenticating messages exchanged over public computer networks, the problem of designing private and authenticated low-cost radio frequency identification systems, and the problem of preserving location privacy in wireless sensor networks.

1.1.1 Authenticated Message Exchange

With the increasing amount of critical information transmitted over the Internet, and the increasing deployment of wireless pervasive networks, comes an increasing demand for computationally-efficient and resource-efficient techniques to protect the integrity of exchanged information. A Message Authentication Code (MAC) algorithm is a cryptographic primitive that enables users to exchange messages, through public channels, in an authenticated manner. However, in most applications in which message integrity needs to be preserved, message privacy must also be preserved. In such scenarios, an encryption algorithm is combined with a MAC algorithm to compose an *authenticated encryption* scheme.

Indeed, many practical systems, such as the Secure Shell (SSH) [271], the Internet Protocol Security (IPsec) [66], the Secure Socket Layer (SSL) [84], and the Transport Layer Security (TLS) [64], provide integrity and privacy for messages exchanged over the Internet by making use of an encryption primitive (for data privacy) and a MAC primitive (for data integrity). Part I of this dissertation investigates the problem of message authentication, including three novel techniques to improve the efficiency of MACs used to authenticate private messages. Highlights of the main contributions of this dissertation in the area of message authentication are as follows.

1.1.1.1 Utilizing IND-CPA Security: Eliminating Redundant Computations

In an effort to improve the efficiency of protocols that provide integrity and privacy of exchanged messages, we propose the design of special purpose MACs that utilize the fact that the message to be authenticated must also be encrypted. We demonstrate that when messages must be both encrypted and authenticated, there can be a redundancy in the computations performed by the two algorithms. Hence, removing such redundancy can improve the efficiency of the overall construction. Furthermore, we show how such special purpose MACs can be made more secure than their general purpose counterparts by inheriting security from the coupled encryption primitive. Detailed description is given in Chapter 4.

1.1.1.2 Utilizing Block Cipher Security: Keyless MACs

We take the idea of utilizing the privacy of the message to be authenticated one step further. That is, not only we assume that the message to be authenticated is also encrypted, but it is encrypted with a secure block cipher. The novelty of the approach we propose can be highlighted by two main points. First, as opposed to analyzing MACs according to the standard security model, we propose a new model that captures essential differences between standard MACs and the ones coupled with an encryption primitive to compose an authenticated encryption system. Second, we show how to utilize the security of the underlying block cipher to significantly reduce the amount of computations performed by the MAC primitive. In particular, we show how to construct the first secure keyless MAC that is faster than the fastest MAC in the cryptographic literature. Detailed description is given in Chapter 5.

1.1.1.3 Utilizing IND-CPA Security: Randomizing the MAC's Algebraic Structure

One of the fundamental principles in cryptography is that all information about the cryptosystem should be publicly known, except for the secret key. That is, when analyzing the security of a certain MAC algorithm, all information about the mathematical structure under which the computation is performed must be known. While it is possible, in theory, to assume the MAC's operations are secret, it is impractical to assume that legitimate users have agreed on arbitrary number of different operations to compute different authentication operations. We show that this does not apply to MACs in authenticated encryption systems. We propose an authentication mechanism in which the algebraic structure of the performed computations is randomly variant. In particular, every authentication tag is computed in a randomly selected integer field. For the intended receiver, information about the integer field to be used for verification can be delivered secretly, provided the security of the underlying encryption algorithm. We show that the added dimension of uncertainty leads to authentication codes that are more efficient than existing ones. This proposed MAC is the first in the literature in which only a small portion of the message needs to be authenticated, without affecting the integrity of the entire message. Detailed description is given in Chapter 6.

1.1.1.4 Security of Authentication Based on Universal Hashing

The profound understanding of the security of a certain cryptographic primitive is essential for adopting, rejecting, and/or improving its deployment in practical systems. One particular class of MACs is the one based on universal hash-function families. A popular class of universal hash families is based on arithmetic over the finite integer field $\mathbb{Z}_p$, where p is a prime integer. When computations are performed over a finite integer field, the construction of such MACs is known to be secure. However, no previous works study the security of such MACs when computations are allowed to be performed over arbitrary finite integer rings (i.e., $\mathbb{Z}_n$ for an arbitrary, nonprime modulus, n). In an effort to give more insight to the security of universal hashing based MACs, we investigate their security under arbitrary finite integer arithmetic. We derive an important relation between the prime factorization of the modulus, n, and the security of the constructed MAC. In particular, we show that the probability of successful forgery against such MACs is proportional to the reciprocal of the smallest prime factor of the used modulus, n. Detailed description is given in Chapter 7.

1.1.2 Securing Radio Frequency Identification

Radio Frequency Identification (RFID) is an important example of wireless systems that has been increasingly used in commercial, personal, and defense sectors and is often deployed in untrustworthy environments. A typical RFID system consists of three functional components: tags, readers and a database. When an RFID reader interrogates a tag within its communication range, the tag responds with information that should allow the reader to access the database and obtain information about the tag. When deployed in untrustworthy environments, in addition to their basic objective of identification, tags must also be able to perform extra algorithmic and protocol operations to provide security and privacy properties against malicious entities. Since energy consumption is one of the critical constraints of RFID tags, energy-efficient RFID protocol design is an active area of research. Highlights of the main contributions of this dissertation in the area of RFID systems are as follows.

1.1.2.1 Securing RFID Systems: An Information-Theoretically Secure Approach

Several protocols based on randomizing tags responses in different protocol steps have been proposed to secure low-cost RFID systems. However, almost every proposed protocol that has not been based on a well-established cryptographic primitive has been analyzed and found to be vulnerable to attacks. The lack of adequate security, even with randomization, in existing protocols is due to the lack of formalization of the threat models that are applicable to resource-constrained RFID systems. A suitable security model has to incorporate RFID specific features such as physical location proximity and limited computational capabilities of low-cost tags. We develop and propose a reasonable relaxation of the adversarial capabilities to formally prove the achievable security level in authentication. We propose an information theoretic approach for designing low-cost RFID systems that can be formally proven given the developed security relaxation. As more research is directed to formally design secure, low-cost RFID systems, the goal is to eventually reach maximum security with minimum relaxation on the adversary's power. That is, given the strength of an adversary the system must be secure against, one can choose the most cost-effective system that can be shown to be secure against such adversary. Detailed description is given in Chapter 9.

1.1.2.2 Securing RFID Systems: A Computationally Secure Approach

Our investigation of message authentication algorithms led to investigating whether MACs can be designed to utilize application specific properties. For instance, unlike arbitrary messages, tag identifiers are typically short strings that belong to a certain domain. Combined with the fact that tag identities must remain private, we propose two novel computationally-efficient and resource-efficient techniques for identity authentication directed for RFID applications. The proposed techniques reduce the energy consumption as well as the circuit area required by standard MACs, while maintaining provable security. Detailed description is given in Chapter 10.

1.1.2.3 Reducing Identification Complexity

Repeated tag identification is the main objective of many RFID applications including supply chain management, inventory tracking, and access control. Hence, efficient identification is a basic performance requirement of such applications. A major shortcoming in existing secure resource-efficient RFID protocols is that privacy is achieved at the expense/loss of identification efficiency. This is because, to protect tags against adversaries querying them to obtain their identities, a tag's response to each query must be randomized in a way adversaries are unable to correlate two or more responses from the same tag. Under randomization, authorized readers can perform linear search of tag labels. Linear search, however, is a computationally cumbersome process in ultra large scale RFID systems. Recently, it was shown that allowing partial dependency among tag identifiers can reduce the identification complexity to be logarithmic in the number of tags in the system. Such dependency, however, was later shown to introduce new vulnerabilities to the RFID system as a compromised tag reveals partial secret information about other uncompromised tags.

We pose and investigate the following research question: How to develop efficient identification algorithms and protocols that preserve privacy of tags even under some tag compromises? We were able to not only break the barrier of linear search complexity, but also the logarithmic, while maintaining independent tags' information. The key insight/idea behind the result was identifying a set of non-cryptographic techniques along with specially designed data structures that lead to private and efficient identification. The proposed algorithm led to the first symmetric-key privacy-preserving protocol for RFID systems with constant-time identification. Detailed description is given in Chapter 11.

1.1.3 Anonymous Wireless Sensor Networks

Wireless Sensor Networks (WSNs) belong to another increasingly deployed technology that highlights the richness of security and privacy in ubiquitous networks. In WSNs, small devices are deployed to sense, monitor, and report events of interest. Traditionally, well established cryptographic mechanisms such as encryption have been used to hide the content of communicated messages. Now, consider a wireless network that needs to monitor events of interest and report the location and the time of occurrence of the events to authorized entities while ensuring that unauthorized entities monitoring sensor nodes do not infer the location of the origin of the event. This problem is known as the source anonymity problem in WSNs. In addition to the information conveyed by the message itself, the source anonymity problem has two added dimensions that need to be considered, namely, the spatial and timing information about the reported event. The use of cryptography can indeed protect the content of the message sent by the node that detects the event of interest. However, a set of collaborating adversaries collectively observing the wireless network can infer approximate location of the node that performed the transmission along with approximate time of origin of transmission based on the mere existence of the ciphertext itself. Hence, even without decrypting the content, collaborating adversaries can have a fairly good estimate of the locations of event of interests at the time of observed transmissions.

Existing solutions to the source anonymity problem rely on the use of statistical goodness of fit tests to design transmission algorithms that embed the reports of detected events within a series of fake transmissions, so that adversaries are unable to distinguish the reports of events of interest from fake transmissions by means of statistical tests. We propose a statistical framework for modeling, analyzing, and designing anonymous WSNs. In the proposed model, the notion of interval indistinguishability is introduced to capture all possible statistical tests an adversary might launch to infer private source information. Then, the source anonymity in WSNs is mapped to the classic statistical problem of binary hypothesis testing with nuisance information. In doing so, we introduce a mapping that converted the problem from data sample to code. Based on the new framework, we are able to first identify that the state of the art solutions had hidden vulnerabilities that were readily exploitable, and then propose remedies for the observed vulnerabilities. Detailed description is given in Chapter 12.

1.2 Organization

The dissertation is divided into three parts based on the main investigated problem. Before we start describing the three parts constituting the major contribution of this dissertation, we give relevant background material and preliminaries that will be used throughout the dissertation in Chapter 2. In the first part, we investigate the problem of authenticated message exchange in computer communication networks (Chapters 3–7). In the second part, we investigate the problem of designing low-cost RFID systems (Chapters 8–11). In the third part, we investigate the problem of location privacy in wireless sensor networks (Chapter 12). Finally, we conclude the dissertation in Chapter 13.

Chapter 2

BACKGROUND AND PRELIMINARIES

This chapter contains background material relevant to the main technical contributions discussed in the dissertation. We start by listing the notations that will be used throughout the dissertation.

2.1 Notations

The following notations will be used throughout the rest of the chapter.

- For two nonempty sets A and B, where B is a subset of A, we denote by $A\backslash B$ the set consisting of all elements of A that are not in B.

- We use the usual notation of $\mathbb{Z}_n$ to denote the integer ring defined over the set $\{0, 1, ..., n-1\}$ with addition and multiplication operations modulo n.

- We use $\mathbb{Z}_n^*$ as the usual notation of the multiplicative group modulo n. That is, $\mathbb{Z}_n^*$ consists of the set of integers relatively prime (co-prime) to n. When n is a prime integer $\mathbb{Z}_n^* = \mathbb{Z}_n\backslash\{0\} := \{1, 2, ..., n-1\}$. For the rest of the dissertation, for a prime integer p, the two notations $\mathbb{Z}_p^*$ and $\mathbb{Z}_p\backslash\{0\}$ will be used interchangeably to emphasize the co-prime property or the exclusion of the zero element, respectively.

- If S is a non-empty set, $|S|$ denotes the cardinality of the set. If s is a binary string, $|s|$ denotes the length of s in bits.

- The function $\varphi(n)$ (the *Euler totient function*) is defined to be the number of positive integers less than n that are relatively prime to n. Equivalently, $\varphi(n) = |\mathbb{Z}_n^*|$.

- For two integers a and b, we say $a \mid b$, read as a divides b, if there exists an integer c such that $b = c \times a$.

- For two integers a and b, we say $a \nmid b$, read as a does not divide b, if there is no integer c such that $b = c \times a$.

- For any two integers a and b, $\gcd(a, b)$ is the greatest common divisor of a and b.

- For an element a in a ring R, the element a^{-1} denotes the multiplicative inverse of a in R, if it exists.

- For any two strings a and b, $(a||b)$ denotes any operation that allows the reconstruction of a and b from $(a||b)$. When the lengths of a and b are known, the concatenation operation is an example of such operations.

- For a positive integer β, $\{0,1\}^{\beta}$ denotes a binary string of length β-bits, and $\{0,1\}^{*}$ denotes a binary string of arbitrary length.

- For a non-empty set $\mathcal{F}$, we denote by $f \overset{\$}{\leftarrow} \mathcal{F}$ the selection of a member of $\mathcal{F}$ uniformly at random and assigning it to f.

- For an integer p, the function $\mathsf{isprime}(p)$ is a polynomial time algorithm that returns true if p is a prime integer and false otherwise.[1]

- Throughout the rest of the dissertation, random variables will be represented by bold font symbols, whereas the corresponding non-bold font symbols represent specific values that can be taken by these random variables.

2.2 Negligible Functions

An important term that will be used in the reminder of the dissertation is the definition of negligible functions. A function $\mathsf{negl} : \mathbb{N} \to \mathbb{R}$ is said to be negligible if for any nonzero polynomial poly, there exists a natural number N_0 such that for all natural numbers $N > N_0$, $|\mathsf{negl}(N)| < \dfrac{1}{|\mathsf{poly}(N)|}$. That is, the function is said to be negligible if it converges to zero faster than the reciprocal of any polynomial function [94].

[1] Interested readers may refer to [212, 95, 4] for more discussion about polynomial-time primality testing.

2.3 Symmetric vs. Asymmetric Cryptography

A typical cryptographic system can be either symmetric-key (private-key) or asymmetric-key (public-key). In symmetric-key cryptosystems, users exchange a secret with which they secure their subsequent communications. The same secret is used for different operations (thus the name symmetric-key). For instance, in a symmetric-key encryption system, the same key is used for both the encryption and decryption operations. Asymmetric-key cryptosystems, on the other hand, use two set of keys: a private key and a public key (thus the name asymmetric-key). For instance, in an asymmetric-key encryption system, the key used for encryption is different than the key used for decryption. A user can publish his/her public key and keep the corresponding private key secret. Anyone can encrypt a message with the user's public key and only with the knowledge of the private key the message can be successfully decrypted.

Symmetric-key operations are typically orders of magnitude faster than their asymmetric-key counterparts. For example, while symmetric-key algorithms typically run in about 100 Mbit/s – 70 Gbit/s, asymmetric-key algorithms run in about 100 Kbit/s – 10 Mbit/s [208]. Furthermore, symmetric-key operations can be built using orders of magnitude less circuitry and consume orders of magnitude less energy. For instance, as symmetric-key algorithms require $3K - 100K$ gates to implement, asymmetric-key algorithms require $100K - 1M$ gates [208]. Moreover, while symmetric-key algorithms typically consume $20 - 30$ μJ/bit, asymmetric-key algorithms typically consume $1000 - 2000$ μJ/bit [208].[2]

Symmetric-key systems will be the main focus of the dissertation, with occasional references to asymmetric-key ones for comparison purposes.

2.4 Encryption Algorithms

Since the use of secure encryption algorithms is essential in this dissertation, we give in this section a brief discussion about encryption algorithms. A symmetric-key encryption consists of three algorithms: a key generation algorithm ($\mathcal{K}$), an encryption algorithm ($\mathcal{E}$), and a

[2] Note further that public-key systems require much longer keys; while 128-bit keys are considered secure for symmetric-key cryptosystems, public-key ones typically require 1024-bit keys to be secured.

decryption algorithm ($\mathcal{D}$). The key generation algorithm takes a security parameter as an input and returns a secret key. The encryption algorithm takes the secret key and a plaintext message as input and returns a ciphertext. The decryption algorithm takes the secret key and the ciphertext as input and returns a plaintext message. We start this section by formally defining the security notion that will be used throughout the dissertation. We then discuss two cryptographic primitives that can be used to construct encryption algorithms that satisfy the defined security notion.

2.4.1 Security of Encryption Algorithms

The main purpose of encryption is to protect the privacy of plaintext messages. That is, without the knowledge of the decryption key, observing a certain ciphertext does not reveal any information about its corresponding plaintext. One of the well-established security measures of an encryption algorithm is the notion of indistinguishability under chosen-plaintext attacks (IND-CPA). To formally analyze the security of different encryption algorithms, the adversary is given oracle access to the algorithm. That is, the encryption algorithm is treated as a black box that the adversary can ask to encrypt plaintexts of her choice and observe the corresponding ciphertexts.

Now, let $\mathcal{A}$ be an adversary who is given oracle access to an encryption algorithm, $\mathcal{E}$, and can ask the oracle to encrypt a polynomial number of messages to get their corresponding ciphertexts. Informally speaking, the encryption algorithm is said to be IND-CPA secure if the adversary, after calling the encryption oracle a polynomial number of times, is given a ciphertext corresponding to one of two plaintext messages of her choice cannot determine the plaintext corresponding to the given ciphertext with an advantage significantly higher than 1/2. Formally, the following standard game is used to model IND-CPA security of encryption algorithms.

Game 1 (IND-CPA game)**.**

1. *The challenger draws a key $K \overset{\$}{\leftarrow} \mathcal{K}$ uniformly at random.*

2. *$\mathcal{A}$ calls the encryption oracle a polynomial number of times on messages of its choice*

and records the corresponding ciphertexts.

3. *$\mathcal{A}$ gives the encryption oracle two messages, m_0 and m_1, of equal length.*

4. *The challenger draws a bit $b \xleftarrow{\$} \{0,1\}$ uniformly at random, encrypts m_b, and returns the resulting ciphertext to $\mathcal{A}$.*

5. *$\mathcal{A}$ can then call the encryption oracle a polynomial number of times and eventually outputs a bit, b'.*

6. *$\mathcal{A}$ wins the game if $b' = b$.*

Let $\mathsf{Adv}_{\mathcal{E}}^{\text{ind-cpa}}(\mathcal{A})$ denote adversary's $\mathcal{A}$ advantage of breaking the IND-CPA security of the encryption algorithm $\mathcal{E}$. Then, $\mathcal{E}$ is said to be IND-CPA secure if

$$\mathsf{Adv}_{\mathcal{E}}^{\text{ind-cpa}}(\mathcal{A}) \leq \frac{1}{2} + \mathsf{negl}(\kappa), \tag{2.1}$$

where $\mathsf{negl}(\kappa)$ is a negligible function in the security parameter κ (typically the length of the secret key).

Note that IND-CPA security implies that the encryption algorithm must be probabilistic [96]. That is, encrypting the same message twice yields different ciphertexts. To see that, let the adversary call the encryption oracle on a message m_1 and receiving its ciphertext c_1. The adversary now chooses two messages, m_1 and m_2, asks the encryption oracle to encrypt them and receives the ciphertext corresponding to one of them. If the encryption is deterministic, the adversary can determine, with high confidence, to which plaintext the ciphertext corresponds by comparing it to c_1.

2.4.2 Block Ciphers

Block ciphers are amongst the most important encryption primitives, if not the most important. The significance of block ciphers is that, nowadays, they are the recommended building blocks for constructing secure encryption algorithms [138]. A block cipher mapping ℓ-bit strings to ℓ-bit strings is a family of permutations $\mathcal{F}$ specified by a finite set of keys $\mathcal{K}_e$. Each key $K \in \mathcal{K}_e$ defines a member of the family $\mathcal{F}_K \in \mathcal{F}$. As opposed to thinking of $\mathcal{F}$ as a set of functions mapping elements from $\{0,1\}^\ell$ to elements in $\{0,1\}^\ell$, it can be viewed as a single function $\mathcal{F} : \mathcal{K}_e \times \{0,1\}^\ell \rightarrow \{0,1\}^\ell$, whose first argument is usually written as a subscript. A random element $f \xleftarrow{\$} \mathcal{F}$ is determined by selecting a $K \xleftarrow{\$} \mathcal{K}_e$ uniformly at random and setting $f \leftarrow \mathcal{F}_K$.

In this dissertation, we adopt the notion of security for block ciphers introduced in [169] and adopted for the concrete setting in [29]. Let $\mathcal{F} : \{0,1\}^\kappa \times \{0,1\}^\ell \rightarrow \{0,1\}^\ell$, where κ is the key length and ℓ is the block size of the block cipher, be a block cipher and let $\mathsf{Perm}(\ell)$ denote the set of all permutations on $\{0,1\}^\ell$. Let $\mathcal{A}$ be an adversary with access to an oracle and that returns a bit. Then,

$$\mathsf{Adv}_{\mathcal{F}}^{\mathrm{prp}}(\mathcal{A}) = \Pr\left[f \xleftarrow{\$} \mathcal{F} : \mathcal{A}^{f(\cdot)} = 1\right] - \Pr\left[\pi \xleftarrow{\$} \mathsf{Perm}(\ell) : \mathcal{A}^{\pi(\cdot)} = 1\right] \tag{2.2}$$

denotes the prp-advantage of $\mathcal{A}$ in distinguishing a random instance of $\mathcal{F}$ from a random permutation. Intuitively, we say that $\mathcal{F}$ is a secure prp, or a secure block cipher, if the prp-advantages of all adversaries using reasonable resources is small.

A block cipher is said to be strong pseudorandom permutation (sprp) if it is indistinguishable from a random permutation even if the adversary is given an oracle access to the inverse function. Then,

$$\mathsf{Adv}_{\mathcal{F}}^{\mathrm{sprp}}(\mathcal{A}) = \Pr\left[f \xleftarrow{\$} \mathcal{F} : \mathcal{A}^{f(\cdot),f^{-1}(\cdot)} = 1\right] - \Pr\left[\pi \xleftarrow{\$} \mathsf{Perm}(\ell) : \mathcal{A}^{\pi(\cdot),\pi^{-1}(\cdot)} = 1\right] \tag{2.3}$$

denotes the sprp-advantage of $\mathcal{A}$ in distinguishing a random instance of $\mathcal{F}$ from a random permutation.

Due to their importance, the literature is rich with variety of proposed block ciphers. The most prominent of which is the Advanced Encryption Standard (AES) [187], which replaced the previous Data Encryption Standard (DES) [186]. Note, however, that block ciphers take a fixed-length plaintext block as an input and output a ciphertext block of the same size. To build an encryption algorithm that takes an arbitrary length plaintext messages and outputs their corresponding ciphertexts, modes of encryptions based on block ciphers are used. Examples of such modes of operations include, but are not limited to, Electronic Codebook (ECB), Cipher Block Chaining (CBC), Propagating Cipher Block

Chaining (PCBC), Cipher Feedback (CFB), Output Feedback (OFB), and Counter (CTR). Note further that block ciphers are deterministic permutations. Therefore, to achieve IND-CPA security, the mode of encryption must induce some randomness, which is typically done by the use of nonces or counters.

2.4.3 Stream Ciphers

A stream cipher is another symmetric-key primitive that can be used construct encryption algorithms. Stream ciphers represent a different approach to symmetric encryption from block ciphers. While block ciphers operate on large blocks of bits with a fixed, unvarying transformation, stream ciphers operate on plaintexts bit by bit, typically by XORing plaintext bits with a pseudorandom cipher bit stream (keystream).

The distinction between block ciphers and stream ciphers is not always clear-cut: in some modes of operation, a block cipher is used in such a way that it acts effectively as a stream cipher (the Counter mode of encryption is an example [243, 138]). Stream ciphers typically are extraordinary fast and have lower hardware complexity compared to block ciphers [138]. However, stream ciphers can be susceptible to serious security problems if used incorrectly [138].

In their earliest use, stream ciphers XOR plaintexts with a random, one-time pad, keystream (Vernam's cipher [253]). Although this is the only cipher that can be information-theoretically secure (as proven in the seminal work of Shannon [228]), it is impractical to demand that they key cannot be used more than once. In modern stream ciphers, the key can be used for multiple encryptions. Linear Feedback Shift Registers (LFSR) have been widely used historically [173], but are considered insecure [138]. The mostly used stream cipher nowadays is the RC4 of Ron Rivest, which is used in the standardized Wired Equivalent Privacy (WEP) algorithm for IEEE 802.11 wireless networks [118] as well as the Secure Socket Layer (SSL) protocol [84].

2.5 Hash Functions

Hash functions belong to another class of primitives that are used extensively in cryptography. The main purpose of hash functions is message compression, which implies that they are noninvertible. Therefore, unlike block ciphers, hash functions cannot be used for message encryption. There are two types of hash functions: cryptographic hash functions and universal hash-function families.

2.5.1 Cryptographic Hash Functions

Cryptographic hash functions are arguably the most deployed cryptographic primitives. They are used in a wide variety of applications, such as message digest, message authentication, digital signatures, etc. Typically, a cryptographic hash function is an unkeyed function that takes an arbitrary length message as input and produces a fixed length output (usually called hashed value, image, fingerprint, or digest).

For any cryptographic hash function to be considered secure, the following problems must be hard to solve [243].

1. **Preimage.** Given a hash value y, find any input x such that $h(x) = y$. If, for a certain hash function, this problem cannot be solved efficiently, the hash function is said to be *one-way* or *preimage resistance.*

2. **Second preimage.** Given an input x, find another input x', where $x \neq x'$, such that $h(x) = h(x')$. If, for a certain hash function, this problem cannot be solved efficiently, the hash function is said to be *second preimage resistance.*

3. **Collision.** Find two inputs x and x', where $x \neq x'$, such that $h(x) = h(x')$. If, for a certain hash function, this problem cannot be solved efficiently, the hash function is said to be *collision resistance.*

An important observation about cryptographic hash functions can be inferred from the statements “hard to solve” and “cannot be solved efficiently”. These statements imply that there is no mathematical proof on how hard it is to solve the aforementioned problems.

This observation explains why some hash functions that are believed to be secure for long time, such as MD5 [216] and SHA-1 [70], have been severely broken [41, 259, 257, 258].

2.5.2 Universal Hash-Function Families

A critical notion that will be used repeatedly throughout this dissertation is the notion of universal hash-function families. Unlike cryptographic hash functions, universal hash-function families are keyed functions. That is, the use of a secret key is necessary for the computations of the universal hashing. While both cryptographic and universal hash functions are used for message compression, there is a fundamental difference between them: mathematical guarantee. That is, unlike cryptographic hash functions, one can derive a precise mathematical expression for the probability that two distinct inputs will collide in universal hash functions.

The computations of a universal hash function typically involve only basic algebraic arithmetic (e.g., simple integer multiplications or polynomial evaluations). This implies that the computations of universal hash functions can be performed much faster than cryptographic hash functions. On the other hand, due to the algebraic structure, the observation of multiple input-output pairs of a certain universal hash function can reveal the value of the secret key used for message compression. This implies that, since finding a collision is easy once the hashing key is exposed, universal hash families are *non-cryptographic* functions.

Formally, a family of hash functions $\mathcal{H}$ is specified by a finite set of keys $\mathcal{K}$. Each key $k \in \mathcal{K}$ defines a member of the family $\mathcal{H}_k \in \mathcal{H}$. As opposed to thinking of $\mathcal{H}$ as a set of functions from $\mathcal{D}$ to $\mathcal{R}$, it can be viewed as a single function $\mathcal{H} : \mathcal{K} \times \mathcal{D} \to \mathcal{R}$, whose first argument is usually written as a subscript. A random element $h \overset{\$}{\leftarrow} \mathcal{H}$ is determined by selecting a $k \overset{\$}{\leftarrow} \mathcal{K}$ uniformly at random and setting $h \leftarrow \mathcal{H}_k$. There are many classes of universal hash families, depending on their probability of message collision (see, e.g., [260, 55, 241, 148, 149, 104]). We give below a formal definition of one class of universal hash families called ϵ-almost universal.

Definition 1. *Let $\mathcal{H} = \{h : A \to B\}$ be a family of hash functions and let $\epsilon \geq 0$ be a real number. $\mathcal{H}$ is said to be ϵ-almost universal, denoted ϵ-AU, if for all distinct $M, M' \in A$, we* *have that* $\Pr_{h \leftarrow \mathcal{H}}\Big[h(M) = h(M')\Big] \leq \epsilon$*.* $\mathcal{H}$ *is said to be* ϵ*-almost universal on equal-length strings if for all distinct, equal-length strings* $M, M' \in A$*, we have that* $\Pr_{h \leftarrow \mathcal{H}}\Big[h(M) = h(M')\Big] \leq \epsilon$*.*

Although not as widely used as block ciphers or cryptographic hash functions, universal hash families are important primitives for building highly efficient message authentication codes.

2.6 Unconditional vs. Computational Security

There are two different notions of security in cryptographic systems: unconditional and computational security. In the former notion, one can mathematically prove that a certain system is secure, regardless of how much time and resources an adversary dedicates to break the system. The historical Vernam one-time pad cipher is an example of unconditionally secure systems [253]. In one-time pad ciphers, plaintext bits are XORed with a random keystream that is never used again. In his seminal work, Shannon proved that such one-time pad systems are unconditionally (also known as information-theoretically) secure [228]. More importantly, Shannon proved that for a cipher to be unconditionally secure, the entropy of the keystream must be at least as the entropy of the plaintext [228]. This implies that the key must be as long as the message to be encrypted.

In [142], Kerckhoffs stated, among six other, one of the most important principles in cryptography: *"The system must be practically, if not mathematically, indecipherable"*. This is the key principle behind modern cryptography. Although not explicitly stated, this principle implies that it is not necessary to have unconditional security as long as the system cannot be broken in a reasonable time with a reasonable probability of success. In Kerckhoffs words, such systems are "practically secure", which are now called "computationally" or "provably" secure.

Modern cryptographic primitives, such as public-key cryptography, block ciphers, cryptographic hash functions are all computationally secure. That is, given unlimited resources, such primitives can be broken. However, the time required to break one of the aforementioned primitives can be hundreds of years (provided a reasonable key length). In today's cryptography, it is sufficient, and even desirable, to reduce the security of a certain system to the security of a known computationally secure primitive to be considered secure. In such scenarios, the system is said to be provably secure.

Before we proceed, we give a formal definition of Shannon's information-theoretic security (also known as perfect secrecy) as it will be used in later parts of the dissertation.

Definition 2 (Perfect Secrecy)**.** *For a plaintext m and its corresponding ciphertext φ, the cipher is said to achieve perfect secrecy if* $\Pr[\boldsymbol{m} = m | \boldsymbol{\varphi} = \varphi] = \Pr[\boldsymbol{m} = m]$ *for all plaintext m and all ciphertext φ. That is, the a posteriori probability that the plaintext is m, given that the ciphertext φ is observed, is identical to the a priori probability that the plaintext is m.*

2.7 A Useful Result

We conclude this chapter by stating an important lemma, a general result known in probability and group theory [224], that will be used in multiple proofs in Part I.

Lemma 1. *Let G be a finite group and $\boldsymbol{X}$ a uniformly distributed random variable defined on G, and let $k \in G$. Let $\boldsymbol{Y} = k * \boldsymbol{X}$, where $*$ denotes the group operation. Then $\boldsymbol{Y}$ is uniformly distributed on G.*

Part I
AUTHENTICATED MESSAGE EXCHANGE

The first part of this dissertation investigates the problem of authenticated message exchange over public channels. There are four technical contributions in this part, detailed in four different chapters. Before we describe the four technical contributions of this part, we give brief background relevant to field of message authentication in Chapter 3. The first technical contribution entitled "*Utilizing IND-CPA Security: Eliminating Redundant Computations*" is given in Chapter 4. The second technical contribution entitled "*Utilizing Block Cipher Security: Keyless MACs*" is given in Chapter 5. The third technical contribution entitled "*Utilizing IND-CPA Security: Randomizing the MAC's Algebraic Structure*" is given in Chapter 6. The fourth technical contribution entitled "*Security of Authentication Based on Universal Hashing*" is given in Chapter 7.

Chapter 3

MESSAGE AUTHENTICATION TECHNIQUES

3.1 Message Authentication Code Algorithms

A message authentication code (MAC) algorithm is a cryptographic primitive used for preserving message integrity. Unlike digital signatures, MACs are symmetric-key primitives. That is, the key used for message authentication is the same key that must be used for message verification.

Formally, A message authentication scheme consists of a signing algorithm $\mathcal{S}$ and a verifying algorithm $\mathcal{V}$. The signing algorithm might be probabilistic, while the verifying one is usually not. Associated with the scheme are parameters κ and N describing the length of the shared key and the resulting authentication tag, respectively. On input an κ-bit key K and a message M, algorithm $\mathcal{S}$ outputs an N-bit string τ called the authentication tag, or the MAC of m.[1] On input an κ-bit key K, a message M, and an N-bit tag τ, algorithm $\mathcal{V}$ outputs a bit, with 1 standing for accept and 0 for reject. A basic validity requirement is that authentic tags are accepted with probability one. That is, if $\tau = \mathcal{S}(K, M)$, it must be the case that $\mathcal{V}(K, M, \tau) = 1$ for any key K, message M, and tag τ.

Based on their security, message authentication codes can be categorized into unconditionally or computationally secure MACs. Similar to the case of encryption algorithms, unconditional security in authentication codes is obtained by mandating that the secret key must be as long as the message to be authenticated. That is, after authenticating a message, a new randomly generated key must be used to authenticate the next message. Therefore, the use of unconditionally secure MACs is considered impractical for most modern applications.[2] The secret key in computationally secure MACs, on the other hand, can be used to

[1] Depending on the specific implementation, messages usually need to be pre-processed, e.g., padded and divided into blocks.

[2] Although there are unconditionally secure MACs in which the key can be shorter than the message to be authenticated, the length of the key is still proportional to the length of the message [241, 242] and

authenticate an arbitrary number of messages. That is, after agreeing on a relatively short secret key, legitimate users can authenticate an arbitrary number of exchanged messages.

MACs are the most extensively used primitives for message integrity and, consequently, the design of efficient MAC algorithms has attracted a lot of attention historically. Based on the basic building block used to construct them, computationally secure MACs can be block ciphers based, cryptographic hash functions based, or universal hash-function families based.

3.1.1 MACs Based on Block Ciphers

Earliest computationally secure MAC algorithms are typically based on block ciphers. The cipher block chaining message authentication code (CBC-MAC) is one of the most known block cipher based MACs specified in FIPS publication 113 [80] and the International Organization for Standardization ISO/IEC 9797-1 [119]. CMAC, a modified version of CBC-MAC, is presented in the National Institute of Standards and Technology (NIST) special publication 800-38B [68], which was based on OMAC of Iwata and Kurosawa[122]. Other block cipher based MACs include, but are not limited to, XOR-MAC [30] and PMAC [43]. The security of block cipher based MACs has been exhaustively studied (see, e.g., [211, 31, 124, 127, 35]).

3.1.2 MACs Based on Cryptographic Hash Functions

Since the computations of cryptographic hash functions can be faster than those of block ciphers, MACs based on cryptographic hash functions have been proposed to come up with faster MAC algorithms. The use of one-way cryptographic hash functions for message authentication was introduced by Tsudik in [246]. HMAC is a prominent example of the use of iterated cryptographic hash functions in the design of MACs [28], which was adopted as a standard in [81]. The computations of HMAC can be performed using SHA-1 [70] or MD5 [216], producing HMAC-SHA1 and HMAC-MD5, respectively. The Internet Protocol Security (IPsec) [66] and Transport Layer Security (TLS) [64] use HMAC-SHA1 and HMAC-

cannot be used for multiple authentication operations.

MD5 for message authentication.

Another cryptographic hash function based MAC is the MDx-MAC of Preneel and Oorschot [209]. HMAC and two variants of MDx-MAC are specified in the International Organization for Standardization ISO/IEC 9797-2 [120]. Bosselaers et al. described how cryptographic hash functions can be carefully coded to take advantage of the structure of the Pentium processor to speed up the authentication process [47]. Similar to the case of block cipher based MACs, the security of cryptographic hash function based MACs has been extensively studied (see, e.g., [210, 27, 144, 60, 269]).

3.1.3 MACs Based on Universal Hash-Function Families

As mentioned in Section 2.5.2, universal hash families are non-cryptographic functions. Therefore, when first introduced by Carter and Wegman [55, 56, 260, 261], universal hash families were restricted to the design of unconditionally secure authentication. That is, the hashing key used to authenticate a given message must be replaced by another, randomly generated, key to authenticate the next message. Hence, similar to the case of unconditionally secure one-time pad ciphers for encryption, the use of unconditionally secure universal hashing was considered impractical for most modern applications. This is the main reason why, although have been around for long time, universal hash families based MACs have not been as widely deployed in practical systems as their block cipher and cryptographic hashing based counterparts.

However, as opposed to transmitting the output of the universal hash function as the authentication tag, the hashed image can be processed with a cryptographic function (e.g., encryption algorithm or cryptographic hash function) before transmission. In such scenarios, the MAC is computationally secure provided the computational security of the cryptographic primitive. More importantly, since the hashed image is not transmitted in the clear, the observation of multiple message-tag pairs does not reveal the secret hashing key. Therefore, the hashing key can now be used to authenticate an arbitrary number of messages.

Computationally secure MACs based on universal hashing are the fastest method for preserving message integrity. This is due to two main facts. First, processing messages block by block using universal hash functions is much faster than processing them block by block using cryptographic hash functions or block ciphers. Second, since the compressed image is much shorter than the message itself, processing it with a cryptographic function can be performed faster. The use of universal hash functions for the design of computationally secure MACs appeared in [40, 176, 104, 73, 42, 136, 39, 11, 16].

Since the speed of a given universal hash family based MAC relies heavily on the speed of the used universal hash family, the design of fast universal hash families has been an active research area. In [148], Krawczyk introduced the cryptographic CRC, which hashes in about 6 cycles/byte, as shown by Shoup in [233]. In [217], Rogaway proposed the bucket hashing, which runs in about $1.5-2.5$ cycles/byte [42]. The bucket hashing was the first hash family explicitly targeted for fast software implementation. Since one of the known shortcomings of universal hash functions is the requirement of substantially long keys compared to block cipher and cryptographic hashing based MACs [104], in [126], Johansson described bucket hashing with smaller key size.

In [104], Halevi and Krawczyk proposed the Multilinear Modular Hashing (MMH) family, which hashes at about $1.2-3$ cycles/byte. In [73], Etzel et al. proposed the square hash, an MMH-variant that can be more efficient than MMH in certain settings [42]. In [39], Bernstein proposed floating-point arithmetic based hash function that achieves a peak speed of 2.4 cycles/byte. In [3], Afanassiev et al. described an application of hashing based on polynomial evaluation over finite fields. In [188], Nevelsteen and Preneel study the performance of several universal hash functions proposed for MACs.

The speed champion of universal hash families directed for software implementation is the NH family of Black et al. [42]. The NH family is an extension to the MMH family of [104]. The speed improvement comes from eliminating the non-trivial modular reduction required by the MMH family (the MMH family uses arithmetic over an integer field $\mathbb{Z}_p$, where p is a prime integer). The novelty of the NH family is that it uses arithmetic modulo powers of two, i.e., "computations that computers like to do [42]". The NH family hashes at about 0.34 cycles/byte for 2^{-32} probability of message collision.

With such fast MAC algorithms already available, it might be tempting to call for discontinuing the search for faster algorithms. Although it is true that the available MAC speeds are capable of keeping up with high speed network traffic, the real goal is to use the smallest possible portion of the processor's cycles in order to allow it to perform other operations. As noted in [42, 218], hardware advances have not made cryptographic efficiency any less important.

3.2 Authenticated Encryption Schemes

What is more relevant to this dissertation than standard MACs is authenticated encryption systems. The notion of authenticated encryption was introduced independently by Katz and Yung in [140], and by Bellare and Rogaway in [36]. As the name implies, authenticated encryption systems provide both integrity and privacy for messages exchanged over public channels. The motive behind the study of authenticated encryption schemes is the observation that, in almost all applications in which message privacy is required, so is message integrity. Consequently, the design of authenticated encryption schemes has attracted a lot of attention historically. Proposals that use simple checksum or manipulation detection code (MDC) have appeared in [180, 146, 92]. Such simple schemes, however, are known to be vulnerable to attacks [134]. Secure authenticated encryption systems can be constructed in one of two main approaches: generic authenticated encryption compositions and dedicated authenticated encryption primitives.

3.2.1 Generic Authenticated Encryption Compositions

In generic compositions, an encryption algorithm (for message privacy) and a MAC algorithm (for message integrity) are combined to construct an authenticated encryption system. Generic compositions can be constructed in three different ways: Encrypt-and-MAC (E&M), Encrypt-then-MAC (EtM), or MAC-then-Encrypt (MtE). In E&M compositions, the plaintext is passed to the encryption algorithm to get a corresponding ciphertext, the same plaintext is passed to the MAC algorithm to get a corresponding tag, and the resulting ciphertext-tag pair, $\big(\mathcal{E}(M), \mathrm{MAC}(M)\big)$, is transmitted to the intended receiver. In EtM compositions, the plaintext is passed to the encryption algorithm to get a ciphertext,

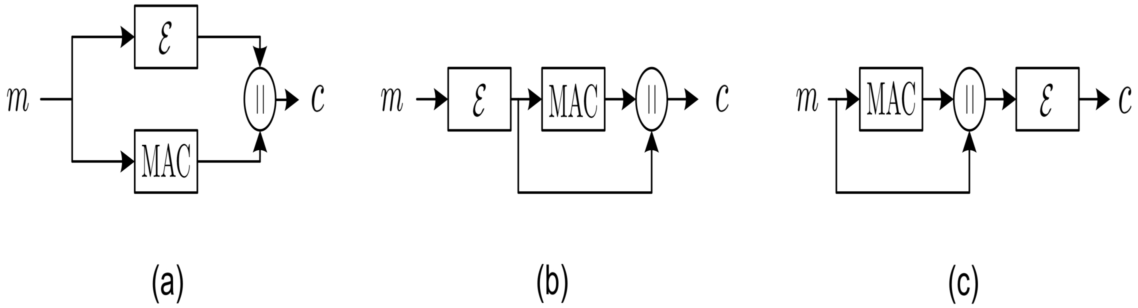


Figure 3.1: A schematic of the three generic approaches; (a) E&M, (b) EtM, and (c) MtE.

the resulting ciphertext is passed to the MAC algorithm to get a tag, and the resulting $\Big(\mathcal{E}(M), \text{MAC}\big(\mathcal{E}(M)\big)\Big)$ is transmitted to the intended receiver. In MtE compositions, the plaintext is passed to the MAC algorithm to get a tag, the resulting tag is appended to the plaintext message, the plaintext-tag concatenation is passed to the encryption algorithm, and the resulting $\Big(\mathcal{E}\big(M, \text{MAC}(M)\big)\Big)$ is transmitted to the intended receiver. Decryption and verification are performed the natural way. Figure 3.1 depicts the three generic compositions methods.

Establishing secure channels by means of generic constructions of authenticated encryption schemes was of particular interest to practical systems. The network layer of the Secure Shell (SSH) protocol [271] uses a variant of the E&M composition. The Internet Protocol Security (IPsec) [66] uses a variant of the EtM composition. The Transport Layer Security (TLS) protocol [64] and its predecessor, the Secure Sockets Layer (SSL) protocol [84], use variants of the MtE composition. The NIST standardized Galois/Counter Mode (GCM) [69] of authenticated encryption uses a variant of the EtM composition. (The GCM is based on the Carter-Wegman Counter (CWC) of Kohno et al. [147], which first encrypts the message using the counter mode and then authenticate the resulting ciphertext using a universal hash function in the Carter-Wegman style.)

However, generic compositions are more involved than just combining an encryption algorithm with a MAC algorithm. In [150, 34] the security of different generic compositions of authenticated encryption systems is analyzed. Using a secure encryption algorithm (secure in the sense that it provides privacy against chosen-plaintext attacks) and a secure MAC (secure in the sense that it provides unforgeability against chosen-message attacks), it was shown that only the EtM will guarantee the construction of secure channels [150, 34]. Therefore, special attention must be paid to the design of authenticated encryption systems if the E&M or the MtE compositions are used.

The security relations among different notions of security in authenticated encryption schemes was studied in detail by Bellare and Namprempre in [34]. Canetti and Krawczyk showed that EtM schemes build secure channels [53]. Krawczyk analyzed the security of the three generic constructions methods in [150]. Bellare et al. showed that SSH is provably secure in [33]. Maurer and Tackmann showed that the MtE can result in secure channels in [175].

3.2.2 Dedicated Authenticated Encryption Primitives

In efforts to come up with more efficient authenticated encryption systems, dedicated algorithms have been proposed. In dedicated algorithms, an authenticated encryption scheme is built as a standalone primitive. Jutla pioneered the design of dedicated authenticated encryption primitives in [134]. He proposed the block cipher based integrity aware parallelizable mode (IAPM). The authenticated encryption requires a total of $m + 2$ block cipher evaluation for a message of m blocks. That is, adding authentication to the existing encryption requires only two extra block cipher calls.

Another prominent instance of dedicated authenticated encryption algorithm is the OCB of Rogaway et al. [218]. Similar to IAPM, the OCB is block cipher based. For a message of length M-bits and an n-bit cipher block size, OCB requires $\lceil \frac{M}{n} \rceil + 2$ block cipher calls. Other block cipher based authenticated encryption include [93, 37]. Although stream cipher based authenticated encryption primitives have appeared in [79, 263], such proposals have been analyzed and shown to be vulnerable to attacks [182, 198, 197, 264].

Over dedicated primitives, generic compositions possess several design and analysis advantages due to their modularity and the fact that the encryption and authentication primitives can be designed, analyzed, and replaced independently from each other [150]. More importantly, generic compositions can allow for faster implementations of authenticated encryption when fast encryption algorithms, such as stream ciphers, are combined with fast MACs, such as universal hash functions based MACs [150]. Another observation about dedicated primitives is that, as pointed out in [147], they are all patent protected, which can potentially limit their deployment in practical systems.

In the following three chapters, we introduce three approaches to design message authentication codes that can be used to construct authenticated encryption systems. In each approach, the security of the encryption algorithm is utilized in a novel way to design authentication codes that are more efficient than existing MACs in the cryptographic literature.

Chapter 4

UTILIZING IND-CPA SECURITY: ELIMINATING REDUNDANT COMPUTATIONS

In cryptography, secure channels enable the confidential and authenticated message exchange between authorized users. As illustrated in Section 3.2.1, a generic approach of constructing such channels is by combining an encryption primitive with an authentication primitive (MAC). In this chapter, we introduce the design of a new cryptographic primitive to be used in the construction of secure channels. Instead of using general purpose MACs, we propose the deployment of special purpose MACs, named $\mathcal{E}$-MACs. The main motive behind this work was the intuition that MACs used in the generic construction of authenticated encryption systems, unlike standard MACs, can utilize the fact that messages to be authenticated must also be encrypted. That is, since both the encryption and authentication algorithms are applied to the same message, there might be some redundancy in the computations of the two primitives. If this turned out to be the case, removing such redundancy can improve the efficiency of the overall operation.

4.1 Special Purpose MACs

Although significant efforts have been devoted to the design of dedicated authenticated encryption primitives and the analysis of the generic compositions, little effort has been made to the design of new primitives in order to improve the efficiency and security of generic compositions. In this chapter, we introduce the first such work. Specifically, we introduce the design of special purpose MACs to be used in the construction of E&M and MtE compositions.

As opposed to EtM compositions, E&M and MtE compositions impose an extra requirement on the MAC algorithm. Authentication tags in an E&M or a MtE composition are functions of plaintext messages (not ciphertexts as in EtM compositions). Therefore, the tag must be at least as confidential as the ciphertext since, otherwise, the privacy of the plaintext can be compromised by an adversary observing its corresponding tag.

One class of MACs that is of a particular interest, due its fast implementation, is the class of MACs based on universal hash-function families introduced in Section 3.1.3. Recall that, in MACs based on universal hash-function families, the message to be authenticated is first compressed using a universal hash function in the Carter-Wegman style and, then, the compressed image is processed with a cryptographic function. Indeed, processing messages using universal hash functions is faster than processing them block by block using cryptographic functions. Combined with the fact that processing short strings is faster than processing longer ones, it becomes evident why universal hash functions based MACs are the fastest for message authentication. (The speed champions of MACs in the literature of cryptography are UMAC [42] and hash127 [39]; both of which are based on universal hash functions [151, 249].)

In this chapter, we lay down the foundations of a new direction in the design of symmetric-key primitives for message authentication. We propose the deployment of a new cryptographic primitive for the construction of secure channels using the E&M and MtE compositions. We introduce the design of $\mathcal{E}$-MACs, Message Authentication Codes for $\mathcal{E}$ncrypted messages. By proposing the first instance of $\mathcal{E}$-MACs, we show how the structure of the E&M and MtE systems can be utilized to increase the efficiency and security of the authentication process. In particular, we show how a universal hash function based $\mathcal{E}$-MAC can be computed with fewer operations than what standard universal hash functions based MACs require. That is, we will demonstrate that universal hash functions based $\mathcal{E}$-MACs can be implemented without the need to apply any cryptographic operation to the compressed image. Moreover, we will also demonstrate that $\mathcal{E}$-MACs can further utilize the special structures of the E&M and MtE systems to improve the security of the authentication process. That is, we will show how universal hash functions based $\mathcal{E}$-MACs can be secured against the key-recovery attack, to which standard universal hash functions based MACs are vulnerable. Finally, we will show that the extra confidentiality requirement on $\mathcal{E}$-MACs can be achieved rather easily, again, by taking advantage of the E&M and MtE structures.

The rest of the chapter is organized as follows. In Section 4.2, the security model that will be used to analyze the proposed $\mathcal{E}$-MACs is described. An instance of $\mathcal{E}$-MACs is proposed in Section 4.3. The performance of $\mathcal{E}$-MACs compared to their counterpart MACs is discussed is Section 4.4. The security analysis of the proposed $\mathcal{E}$-MAC and the security of the generic compositions constructed using the proposed $\mathcal{E}$-MAC are detailed in Section 4.5. Section 4.6 is dedicated to the discussion of the key-recovery vulnerability of universal hash functions based MACs and the description of how $\mathcal{E}$-MACs can utilize the structures of the E&M and MtE systems to overcome this vulnerability. The chapter is summarized in Section 4.7.

4.2 Security Model

In this chapter, we adopt the standard security model to analyze MACs. In general, an adversary in a message authentication scheme is a probabilistic algorithm $\mathcal{A}$, which is given oracle access to the signing and verifying algorithms $\mathcal{S}(K, \cdot)$ and $\mathcal{V}(K, \cdot, \cdot)$ for a random but hidden choice of K. $\mathcal{A}$ can query $\mathcal{S}$ to generate a tag for a plaintext of its choice and ask the verifier $\mathcal{V}$ to verify that τ is a valid tag for the plaintext. Formally, $\mathcal{A}$'s attack on the scheme is described by the following experiment:

1. A random string of length κ is selected as the shared secret.

2. Suppose $\mathcal{A}$ makes a signing query on a message M. Then the oracle computes an authentication tag $\tau = \mathcal{S}(K, M)$ and returns it to $\mathcal{A}$. (Since $\mathcal{S}$ may be probabilistic, this step requires making the necessary underlying choice of a random string for $\mathcal{S}$, anew for each signing query.)

3. Suppose $\mathcal{A}$ makes a verify query (M, τ). The oracle computes the decision $d = \mathcal{V}(K, M, \tau)$ and returns it to $\mathcal{A}$.

The verify queries are allowed because, unlike the setting in digital signatures, $\mathcal{A}$ cannot compute the verify predicate on its own (since the verify algorithm is not public). Note that $\mathcal{A}$ does not see the secret key K, nor the coin tosses of $\mathcal{S}$.

The adversary can query the signing oracle for q times before attempting the forgery attempt. The outcome of running the experiment in the presence of an adversary is used to define security. As in [34], we say that the MAC algorithm is weakly unforgeable against chosen-message attacks (WUF-CMA) if $\mathcal{A}$ cannot make a verify query (M, τ) which is accepted for an M that has not been queried to the signing oracle $\mathcal{S}$. We say that the MAC algorithm is strongly unforgeable against chosen-message attacks (SUF-CMA) if $\mathcal{A}$ cannot make a verify query (M, τ) which is accepted regardless of whether or not M is *new*, as long as the tag has not been attached to the message by the signing oracle.

4.3 The Proposed $\mathcal{E}$-MAC

In this section, we describe an instance of $\mathcal{E}$-MACs and use it to construct two generic authenticated encryption compositions: one is based on the Encrypt-and-MAC (E&M) composition and the other is based on the MAC-then-Encrypt (MtE) composition.

4.3.1 Overview of the Proposed $\mathcal{E}$-MAC

The basic goal of any encryption scheme is message privacy; that is, given the ciphertext, it must be hard for an adversary without the knowledge of the decryption key to recover the plaintext. Since the main objective of this chapter is to introduce the general idea of utilizing the encryption operation for better designs of MACs and not to target any specific application, the proposed $\mathcal{E}$-MAC is designed to work with any secure encryption scheme. Thus, the only assumption that we make on the underlying encryption scheme is indistinguishability under chosen plaintext attacks (IND-CPA), as defined in Section 2.4.1.

Just like fast MACs, the proposed $\mathcal{E}$-MAC utilizes universal hash families in the Carter-Wegman style. However, as opposed to universal hash functions based MACs, we will show that $\mathcal{E}$-MACs can be secure without any post computation performed on the compressed image. (Recall that universal hash functions based MACs have two rounds of computations: 1. message compression using universal hash functions and, 2. output transformation, which in most practical applications a pseudorandom function applied to the compressed image [42, 106].) That is, as will be shown in the remaining of this section, the structure of the authenticated encryption system can be utilized to eliminate the need to employ pseudorandom function families. Thus, improving the speed of the MAC and reducing the required amount of shared key information (the key needed to identify the pseudorandom function).

Before we proceed with the detailed description of the proposed $\mathcal{E}$-MAC, we emphasize that the universal hash family used for the implementation of the proposed $\mathcal{E}$-MAC is not the only possible solution. As mentioned earlier, the goal of this chapter is not to come up with any specific design but rather the general idea of utilizing of the structure of the authenticated encryption composition to improve the security and efficiency of message authentication. In fact, any ϵ-almost-Δ-universal (ϵ-AΔU) hash family, such as the MMH family of Halevi and Krawczyk [104] or the NH family of Black et al. [42], will satisfy the security requirements detailed in Section 4.5, as can be seen in the proof of Theorem 3 and the remark following it. (The ϵ-AΔU is a stronger notion than ϵ-AU given in Definition 1; interested readers may refer to [104] for a formal definition of ϵ-AΔU hash families.)

Furthermore, different assumptions about the underlying encryption algorithm may lead to different constructions of $\mathcal{E}$-MACs. We only show here how the IND-CPA security of the underlying encryption algorithm can be utilized to improve the efficiency and security of message authentication. In the next chapter, the assumption that the encryption algorithm is also a strong pseudorandom permutation will be utilized for further improvements in $\mathcal{E}$-MACs performance.

The only operations required to implement the proposed $\mathcal{E}$-MAC are modular addition and multiplication (i.e., operations over the integer ring $\mathbb{Z}_n$, for a finite integer n). For the proposed universal hash family to be secure against message modification, and to ensure a 2^{1-N}-AU hashing, where N is the length of the authentication tag,[1] the multiplication needs to satisfy two properties.

Property 1. *For any two integers α and β in $\mathbb{Z}_n$, if n divides $\alpha\beta$, then one of the integers α and β must be the zero element. Formally, the following one-way implication must hold.*

$$\{\alpha\beta \equiv 0 \mod n\} \Rightarrow \{\alpha \equiv 0 \ \vee \ \beta \equiv 0 \mod n\} \tag{4.1}$$

[1] In Section 4.5, a tighter bound will be derived in the proof of Theorem 3.

Property 1 is satisfied by any $\mathbb{Z}_n$ that is also an integral domain [111].

Property 2. *Given an integer $k \in \mathbb{Z}_n^*$, for an r uniformly distributed over $\mathbb{Z}_n$, the value δ given by:*

$$\delta \equiv rk \mod n \tag{4.2}$$

is uniformly distributed over $\mathbb{Z}_n$.

Property 2 is satisfied by any $\mathbb{Z}_n$ that is also a field (it is a direct consequence of the fact that every nonzero element in a field is invertible). Since every field is an integral domain, and every integer ring $\mathbb{Z}_p$, where p is prime integer, is a field, multiplication modulo p satisfies both properties.[2] Thus, the operations used for the rest of the chapter are performed over the integer field $\mathbb{Z}_p$.

4.3.2 Encrypt-and-MAC Composition

In this section, we describe a construction based on the Encrypt-and-MAC (E&M) generic composition.

4.3.2.1 Instantiation

Assume legitimate users agreed on using an encryption algorithm, $\mathcal{E}$, that provides indistinguishability under chosen plaintext attacks (IND-CPA). Based on a security parameter, N, choose p to be a largest prime integer less than 2^N (for instance, $p = 2^{32} - 5$ for $N = 32$). Define $K := (k_1, k_2, \ldots, k_B)$, for k_i's drawn uniformly and independently from the multiplicative group $\mathbb{Z}_p^*$, to be the shared secret key that will be used for message authentication. As in typical universal hash functions, depending on the values of N and B, the key might be long. One way to generate such a key is via a pseudorandom generator, e.g., [44, 108]. In such a case, only the seed of the pseudorandom generator is required to be distributed to the legitimate parties. Note further that this key generation operation is performed only once during the instantiation phase. That is, once the key is generated, it can be used to

[2] In fact, any finite integral domain is also a field [67].

authenticate an arbitrary number of messages. Thus, the key generation does not affect the complexity of the overall system.

As in symmetric-key cryptographic systems, the shared secret is distributed to the legitimate users via a secure channel. With the knowledge of the shared secret, legitimate users can exchange subsequent messages, over insecure channels, in an authenticated and confidential way. Observe, however, that the encryption key, $K_{\mathcal{E}}$, in our setup is independent of the authentication key, K. Only the shared keys are assumed to be secret; all other parameters such as N, B, and p are publicly known.

4.3.2.2 Authentication

Define $\mathsf{MaxLen} := N(B-1)-1$ to be the upper bound on the length of plaintext messages (in bits) to be authenticated. Append the bit '1' to the end of the message, M, and divide M into blocks of length N-bits; that is, $M = m_1||m_2||\dots||m_L$, where $L = \lceil |M|/N \rceil \leq B-1$ and $|m_i| = N$ for all i's except possibly m_L. (We overload m_i to denote both the binary string in the i^{th} block and the unsigned integer representation of the i^{th} block as an element of $\mathbb{Z}_p$ in a big-endian format; the distinction between the two representations will be omitted when it is clear from the context.)

Remark 1. *We emphasize that each message block, m_i, is considered an element of $\mathbb{Z}_p$ not $\mathbb{Z}_{2^N}$. That is, if two distinct N-bit integers are congruent modulo p, they are considered the same message block. Note, however, that this does not have a noticeable impact on the the performance of the system since only a negligible portion of N-bit integers will be congruent modulo the largest N-bit prime. For instance, if $N = 32$, only five 32-bit integers are congruent module $2^{32}-5$.*

Now, for every message M to be encrypted and authenticated, the sender generates an integer r drawn uniformly at random from $\mathbb{Z}_p$ (this r represents the coin tosses of $\mathcal{S}$). We emphasize that r must be independent of all r's generated to authenticate other messages. The sender encrypts (M, r) and transmits the resulting ciphertext $c = \mathcal{E}(M, r)$ to the intended receiver (recall that the encryption key is independent of the $\mathcal{E}$-MAC key). The

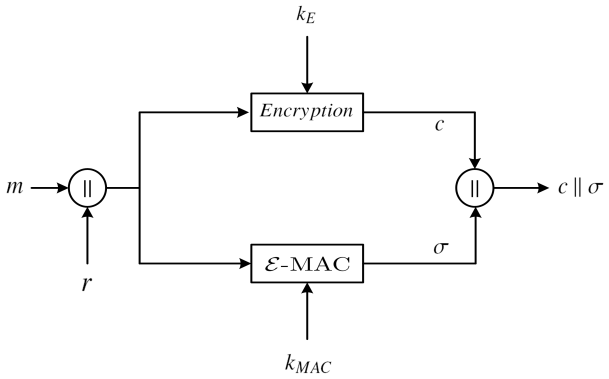


Figure 4.1: A block diagram illustrating the use of $\mathcal{E}$-MAC to construct an Encrypt-and-MAC (E&M) generic composition. A random number, r, is appended to the plaintext message. The resulting $M||r$ is compressed according to equation (4.3) and the same $M||r$ goes to the encryption algorithm. The output of the encryption algorithm and $\mathcal{E}$-MAC are concatenated and transmitted to the intended receiver.

N-bit long tag of message M is computed as:

$$\tau = \sum_{i=1}^{L} k_i m_i + k_B r \mod p, \tag{4.3}$$

where m_i denotes the i^{th} block of message M.

A block diagram depicting the use of the proposed $\mathcal{E}$-MAC for the construction of an Encrypt-and-MAC generic composition is shown in Figure 4.1.

Remark 2. *Appending a '1' at the end of the message is important to guarantee security for variable-length messages. Without the '1' at the end of the message, the authentication is only secure for equal-length messages. To see that, consider messages* $M = m_1||0$ *and* $M' = m_1||00$*, where* M *has only a single zero bit in its last block and* M' *has two zeros. Then,* M *and* M' *will have the same authentication tag, provided the coin tosses,* r*, used in both authentication is the same. Now, assume a stream cipher is used for encryption. Then, an adversary can call the oracle on* $M' = m_1||00$ *and obtain the outputted ciphertext and tag. The adversary can use the same tag to authenticate the message* $M = m_1||0$ *since the second message block does not contribute to the authentication tag. Attaching a '1' at the end of* *the last message bit will make $M = m_1||01$ and $M' = m_1||001$ and, hence, the scheme can be used to authenticate messages of different lengths (since changing the message length will change the authentication tag in an unpredictable way depending on the key corresponding to the last message block).*

Remark 3. *A misconception about universal hashing is that the key needs to be as long as the message to be authenticated, which renders them impractical to build MACs. While this was the case in the earliest use of universal hashing to construct unconditionally secure MACs, this is not the case for computationally secure MACs. Therefore, we emphasize that once the hashing key, K, is drawn, it can be used to authenticate an arbitrary number of messages. For a message that is longer than MaxLen, it can be treated as multiple messages each of length less than or equal MaxLen. For the rest of the chapter, without loss of generality, we will assume messages with $|M| \leq \mathsf{MaxLen}$. Since MaxLen is a function of the length of the shared secret key, the maximum message length is a design tradeoff. That is, as in typical MACs based on universal hash functions, the length of the tag will increase if a message exceeds the maximum length (since the message will be treated as multiple messages of lengths less than or equal to MaxLen).*

Remark 4. *As will be formally proven in Section 4.5, the bound on the probability of successful forgery is dependent on the security parameter, N. Depending on application, one might require lower bounds on probability of successful forgery. A straightforward way is to increase the security parameter to give lower probability of successful forgery. This approach is not desired, especially for software implementations as it results in performance degradation. Another method is to hash the same message multiple times with independent keys. This, however, will require a much longer key. A well-studied and more efficient method is to use the Toeplitz-extension on the hash function [172, 148] (see, e.g., [42] for a detailed use of Toeplitz-extension to increase the security of MACs based on universal hash functions). We omit describing this topic since it is out of the scope of this work and refer interested readers to [172, 148, 104, 42] for more details.*

```
Algorithm $\mathcal{S}(K, M, r)$
    if $|M| >$ MaxLen then Return 0
    $L \leftarrow \lceil |M|/N \rceil$;
    Break $K$ into $N$-bit chunks $k_i$'s;
    $M \leftarrow M||1$
    Break $M$ into $N$-bit chunks $m_i$'s;
    $\tau \leftarrow \sum_{i=1}^{L} k_i m_i + k_B r \mod p$;
    Return $\tau$
```

```
Algorithm $\mathcal{V}(K, M, r, \tau)$
    Break $K$ into $N$-bit chunks $k_i$'s;
    $L \leftarrow \lceil |M|/N \rceil$;
    Break $M$ into $N$-bit chunks $m_i$'s;
    $\tau' \leftarrow \sum_{i=1}^{L} k_i m_i + k_B r \mod p$;
    if $\tau' = \tau$ then Return 1
    Return 0
```

Figure 4.2: The signing and verifying algorithms of the Encrypt-and-MAC composition using the proposed $\mathcal{E} - MAC$.

4.3.2.3 Verification

Upon receiving a ciphertext c, the receiver calls the corresponding decryption algorithm $\mathcal{D}$ to extract the plaintext $M||r$. To verify the integrity of $M||r$, the receiver computes $\sum_{i=1}^{L} k_i m_i + k_B r$ and authenticates the message only if the computed value is congruent to the received τ modulo p. Formally, the following integrity check must be satisfied for the message to be authenticated:

$$\tau \overset{?}{\equiv} \sum_{i=1}^{L} k_i m_i + k_B r \mod p. \tag{4.4}$$

Remark 5. *We emphasize that the random nonce r requires no key management. It is generated by the sender as the coin tosses of the signing algorithm and delivered to the receiver via the ciphertext. In other words, it is not a shared secret and it needs no synchronization.*

The pseudocodes describing the signing and verification algorithms of the proposed E&M composition are shown in Figure 4.2.

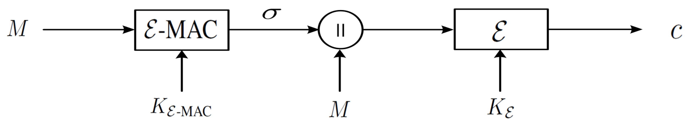


Figure 4.3: A block diagram illustrating the use of $\mathcal{E}$-MAC to construct an MAC-then-Encrypt generic composition. A compressed image of the message is first computed according to equation (4.5). The compressed image is then appended to the plaintext message and the result goes to the encryption algorithm as input. The output of the encryption algorithm is transmitted to the intended receiver.

4.3.3 MAC-then-Encrypt Composition

In this section, we describe a construction based on the MAC-then-Encrypt (MtE) generic composition.

4.3.3.1 Instantiation

As in the E&M of the Section 4.3.2, assume legitimate users agreed on using an encryption algorithm, $\mathcal{E}$, that provides indistinguishability under chosen plaintext attacks (IND-CPA). Based on a security parameter N, legitimate users choose p to be the largest N-bit long prime integer. Define $K := (k_1, k_2, \ldots, k_B)$, for k_i's drawn uniformly and independently from $\mathbb{Z}_p^*$, to be the shared secret key that will be used for message authentication.

4.3.3.2 Authentication

Define $\mathsf{MaxLen} := NB - 1$ to be the upper bound on the length of plaintext messages (in bits) to be authenticated. Append the bit '1' to the end of the message, M, and divide M into blocks of length N-bits; that is, $M = m_1||m_2||\ldots||m_L$, where $L = \lceil |M|/N \rceil \leq B$ and $|m_i| = N$ for all i's except possibly m_L. Compute the N-bit compressed image of M as

$$\sigma = \sum_{i=1}^{L} k_i m_i \mod p, \tag{4.5}$$

where m_i denotes the i^{th} block of the plaintext message, M. A block diagram depicting the use of the proposed $\mathcal{E}$-MAC to construct an MAC-then-Encrypt composition is shown

```
Algorithm $\mathcal{S}(K, M)$
    if $|M| >$ MaxLen then Return 0
    $L \leftarrow \lceil |M|/N \rceil$;
    Break $K$ into $N$-bit chunks $k_i$'s;
    $M \leftarrow M||1$
    Break $M$ into $N$-bit chunks $m_i$'s;
    $\tau \leftarrow \mathcal{E}\Big( \sum_{i=1}^{L} \; k_i m_i \mod p\Big)$;
    Return $\tau$
```

```
Algorithm $\mathcal{V}(K, M, \tau)$
    Break $K$ into $N$-bit chunks $k_i$'s;
    $L \leftarrow \lceil |M|/N \rceil$;
    Break $M$ into $N$-bit chunks $m_i$'s;
    $\tau' \leftarrow \mathcal{E}\Big( \sum_{i=1}^{L} \; k_i m_i \mod p\Big)$;
    if $\tau' = \tau$ then Return 1
    Return 0
```

Figure 4.4: The signing and verifying algorithms of the MAC-then-Encrypt composition using the proposed $\mathcal{E}$-MAC.

in Figure 4.3.

The sender encrypts (M, σ) and transmits the resulting ciphertext to the intended receiver. The ciphertext can be the encryption of the concatenation of the plaintext message and its compressed image (i.e., $c = \mathcal{E}(M, \sigma)$) or it can be the concatenation of the encryption of the plaintext and the encryption of the compressed image (i.e., $c = \mathcal{E}(M), \tau = \mathcal{E}(\sigma)$). In either scenario, the security of the system is the same and, for the rest of the chapter, we will assume the latter scenario ($c = \mathcal{E}(M)$ will denote the ciphertext and $\tau = \mathcal{E}(\sigma)$ will denote the authentication tag).

4.3.3.3 Verification

Upon receiving the ciphertext, the receiver calls the corresponding decryption algorithm $\mathcal{D}$ to extract the plaintext message, M. To verify the integrity of M, the receiver computes its N-bit long compressed image $\sum_{i=1}^{L} k_i m_i \mod p$, encrypts the resulting compressed image, and authenticates the message only if the encryption of the compressed image is equal to the received authentication tag, τ. Formally, the following integrity check must be satisfied for the message to be authenticated:

$$\tau \overset{?}{\equiv} \mathcal{E}\Big(\sum_{i=1}^{L} \; k_i m_i \mod p\Big). \tag{4.6}$$

The pseudocodes describing the signing and verification algorithms of the proposed MtE composition are shown in Figure 4.4.

4.4 Performance of $\mathcal{E}$-MACs

As discussed in Section 3.1, there are three main classes of MACs that can be used in the generic compositions of secure channels: MACs based on block ciphers, MACs based on cryptographic hash functions, and MACs based on universal hash functions. However, universal hashing is the fastest method to construct MACs; hence, we restrict the performance discussion to universal hashing based MACs used to construct secure channels.

Recall that universal hash function based MACs consist of two sequential operations: a universal hashing followed by a cryptographic operation. Observe further that universal hashing is much faster than cryptographic primitives. For instance, while universal hash functions can run in about 0.34 cycles/byte [42], the cryptographic hash functions SHA-256 and SHA-512 run in about 23.73 cycles/byte and 40.18 cycles/byte, respectively [184]. That is, universal hash computations are typically orders of magnitude faster than cryptographic computations. Therefore, it is evident how eliminating the need to post-process the compressed image with a cryptographic function will have a significant impact on the efficiency of the overall construction.

Note further that, although we require the additional encryption of the coin tosses r, this is typically performed in parallel with other plaintext blocks. That is, the encryption of the coin tosses does not affect the performance of encryption, provided parallel computing is available. Furthermore, any IND-CPA secure mode of encryption will require some randomness anyway, whether via the use of nonces or coin tosses [243, 138]. Therefore, the ciphertext block corresponding to the encryption of r in our construction does not impose extra communication overhead.

Compared to single-pass authenticated encryption algorithms, when combined with a stream cipher, the proposed compositions will be much faster (since all secure single-pass authenticated encryption methods are block cipher based[3]). This is due to the fact that

[3] Recall that stream cipher based authenticated encryption primitives are known to be vulnerable to attacks.

when combining block ciphers with universal hashing to construct secure channels, the encryption operation is the most time-consuming process [150]. Since stream ciphers are much faster than block ciphers, it is evident that using stream ciphers to build secure channels will be faster [150].

Recall that the main idea allowing for the design of $\mathcal{E}$-MACs is the fact that the authentication tag is a function of the plaintext, which must also be encrypted. In the Encrypt-then-MAC (EtM) generic composition, on the other hand, authentication tags are functions of ciphertexts. Since ciphertexts must be sent in the clear, EtM compositions cannot take advantage of $\mathcal{E}$-MACs.

4.5 Security Analysis

This section is devoted to the security analysis of the proposed compositions of Sections 4.3.2 and 4.3.3. We start be stating general lemmas.

4.5.1 General Lemmas

The following lemmas are the main ingredient for the security of the proposed $\mathcal{E}$-MAC.

Lemma 2. *Let m_i and k_i be the i^{th} message block and i^{th} key, respectively. For a modified message block $m'_i \not\equiv m_i \mod p$, the probability that $k_i m'_i \equiv k_i m_i \mod p$ is zero.*

Proof: Assume $m'_i \equiv m_i + \delta \mod p$ for some $\delta \in \mathbb{Z}_p$. Then,

$$k_i m'_i - k_i m_i = k_i(m'_i - m_i) = k_i \delta \overset{?}{\equiv} 0 \mod p. \tag{4.7}$$

Trivially, the value $\delta \equiv 0 \mod p$ satisfies the condition in equation (4.7). However, $\delta \equiv 0 \mod p$ implies that the received block is identical to the transmitted one.

For all other values of δ, the condition in equation (4.7) can never be satisfied. This is a direct consequence of Property 1, which states that for the multiplication of any two integers in $\mathbb{Z}_p$ to be congruent to *zero* modulo p, one of them *must* be zero. By design, however, the key k_i is not the zero element. Therefore, for any nonzero $\delta \in \mathbb{Z}_p$, $k_i \delta \not\equiv 0 \mod p$ and consequently, $k_i m'_i \not\equiv k_i m_i \mod p$ for all $m'_i \not\equiv m_i \mod p$. ■

Lemma 3. *Let k_1 and k_2 be two secret keys in the proposed $\mathcal{E}$-MAC. The probability to choose two nonzero integers δ_1 and δ_2 in $\mathbb{Z}_p$ such that $k_1\delta_1 \equiv k_2\delta_2 \mod p$ is at most $1/(p-1)$.*

Proof: Fix a $\delta_1 \in \mathbb{Z}_p^*$. By Property 2, the resulting ($k_1\delta_1 \mod p$) will be uniformly distributed over $\mathbb{Z}_p^*$. Similarly, the resulting ($k_2\delta_2 \mod p$) is uniformly distributed over $\mathbb{Z}_p^*$. Since k_1 and k_2 are assumed to be secret, the probability that $k_1\delta_1 \equiv k_2\delta_2 \mod p$ is $1/(p-1)$. ■

4.5.2 Privacy of Compositions

Let the privacy of the proposed authenticated encryption compositions be modeled as their indistinguishability under chosen plaintext attacks. That is, the composition is considered to preserve data privacy if the combination of the ciphertext and authentication tag provide indistinguishability under chosen plaintext attacks. In this section, we show that the privacy of the proposed compositions is provably secure assuming the underlying encryption algorithm is IND-CPA secure.

Theorem 1. *Let $\mathcal{E}$-$\mathrm{MAC}_{\mathrm{E\&M}}$ be the authenticated encryption composition described in Section 4.3.2. Then given an adversary, $\mathcal{A}$, against the privacy of $\mathcal{E}$-$\mathrm{MAC}_{\mathrm{E\&M}}$, one can construct an adversary $\mathcal{B}$ against $\mathcal{E}$ such that*

$$\mathsf{Adv}^{\mathrm{priv}}_{\mathcal{E}\text{-}\mathrm{MAC}_{\mathrm{E\&M}}}(\mathcal{A}) \leq \mathsf{Adv}^{\text{ind-cpa}}_{\mathcal{E}}(\mathcal{B}). \tag{4.8}$$

Theorem 1 states that if the adversary can expose private information from the proposed E&M composition of Section 4.3.2, she can also break the security of the underlying encryption algorithm. That is, if $\mathcal{E}$ provides IND-CPA, then the proposed authenticated encryption composition provides data privacy. Note that private information refers not only to the plaintext message, but also the $\mathcal{E}$-MAC key and the encryption key. Before we proceed with the proof of Theorem 1, we need the following lemma.

Lemma 4. *In the* E&M *composition described in Section 4.3.2, authentication tags are statistically independent of their corresponding messages, and different authentication tags are mutually independent.*

Proof: Let the secret key $K = k_1||k_2||\cdots||k_B$ be fixed. Let the plaintext message, M, consist of L blocks, where $L \leq B-1$. Then for any tag $\tau \in \mathbb{Z}_p$ computed according to equation (4.3) and any plaintext message M the following holds:

$$\begin{aligned}\Pr(\boldsymbol{\tau} = \tau | \boldsymbol{M} = M) &= \Pr\left(\boldsymbol{r} = (\tau - \sum_{i=1}^{L} k_i m_i)\ k_B^{-1}\right) &\quad (4.9)\\ &= \frac{1}{p}, &\quad (4.10)\end{aligned}$$

where m_i denotes the i^{th} block of the message M. Equation (4.10) holds by the assumption that r is drawn uniformly from $\mathbb{Z}_p$. The existence of k_B^{-1}, the multiplicative inverse of k_B in the integer field $\mathbb{Z}_p$, is a direct consequence of the fact that k_B is not the zero element.

Furthermore, by Property 2, for an r drawn uniformly at random from $\mathbb{Z}_p$, the resulting $(k_B r \mod p)$ is uniformly distributed over $\mathbb{Z}_p$. Consequently, for any plaintext message M, since the tag is a result of adding $(k_B r \mod p)$ to $(\sum_i k_i m_i \mod p)$, and since $(k_B r \mod p)$ is uniformly distributed over $\mathbb{Z}_p$, the resulting tag is uniformly distributed over $\mathbb{Z}_p$. That is, for any fixed value $\tau \in \mathbb{Z}_p$, the probability that the tag will take this specific value is given by:

$$\Pr(\boldsymbol{\tau} = \tau) = \frac{1}{p}. \quad (4.11)$$

Combining Bayes' theorem [101] with equations (4.10) and (4.11) yields:

$$\begin{aligned}\Pr(\boldsymbol{M} = M | \boldsymbol{\tau} = \tau) &= \frac{\Pr(\boldsymbol{\tau} = \tau | \boldsymbol{M} = M) \Pr(\boldsymbol{M} = M)}{\Pr(\boldsymbol{\tau} = \tau)} &\quad (4.12)\\ &= \Pr(\boldsymbol{M} = M). &\quad (4.13)\end{aligned}$$

Equation (4.13) implies that the tag τ gives no information about the plaintext M since τ is statistically independent of M. Similarly, one can show that the tag is independent of the secret key.

Now, let τ_1 through τ_ℓ represent the tags for messages M_1 through M_ℓ, respectively. Let $L_i \leq B-1$ be the number of blocks of message M_i, for $i = 1, \cdots, \ell$. Further, let r_1

through r_ℓ be the coin tosses of the signing algorithm $\mathcal{S}$ for the authentication of messages M_1 through M_ℓ, respectively. Recall that r_i's are *mutually independent* and *uniformly* distributed over $\mathbb{Z}_p$. Then, for any possible values of the messages M_1 through M_ℓ with arbitrary joint probability mass function, and all possible values of τ_1 through τ_ℓ, we get:

$$\Pr(\boldsymbol{\tau}_1 = \tau_1, \cdots, \boldsymbol{\tau}_\ell = \tau_\ell)$$

$$= \sum_{M_1,\cdots,M_\ell} \Pr(\boldsymbol{\tau}_1 = \tau_1, \cdots, \boldsymbol{\tau}_\ell = \tau_\ell | \boldsymbol{M_1} = M_1, \cdots, \boldsymbol{M_\ell} = M_\ell) \cdot \Pr(\boldsymbol{M_1} = M_1, \cdots, \boldsymbol{M_\ell} = M_\ell) \tag{4.14}$$

$$= \sum_{M_1,\cdots,M_\ell} \Pr\left(\boldsymbol{r}_1 = (\tau_1 - \sum_{i=1}^{L_1} k_i m_{1_i})\ k_B^{-1}, \cdots, \boldsymbol{r}_\ell = (\tau_\ell - \sum_{i=1}^{L_\ell} k_i m_{\ell_i})\ k_B^{-1}\right) \cdot \Pr(\boldsymbol{M_1} = M_1, \cdots, \boldsymbol{M_\ell} = M_\ell) \tag{4.15}$$

$$= \sum_{M_1,\cdots,M_\ell} \Pr\left(\boldsymbol{r}_1 = (\tau_1 - \sum_{i=1}^{L_1} k_i m_{1_i})\ k_B^{-1}\right) \cdots \Pr\left(\boldsymbol{r}_\ell = (\tau_\ell - \sum_{i=1}^{L_\ell} k_i m_{\ell_i})\ k_B^{-1}\right) \cdot \Pr(\boldsymbol{M_1} = M_1, \cdots, \boldsymbol{M_\ell} = M_\ell) \tag{4.16}$$

$$= \sum_{M_1,\cdots,M_\ell} \left(\frac{1}{p}\right)^\ell \cdot \Pr(\boldsymbol{M_1} = M_1, \cdots, \boldsymbol{M_\ell} = M_\ell) \tag{4.17}$$

$$= \Pr(\boldsymbol{\tau}_1 = \tau_1) \cdots \Pr(\boldsymbol{\tau}_\ell = \tau_\ell), \tag{4.18}$$

where m_{j_i} denotes the i^{th} block of the j^{th} message M_j. Equation (4.16) holds due to the independence of the $\boldsymbol{r}_i$'s; equation (4.17) holds due to the uniform distribution of the $\boldsymbol{r}_i$'s; and equation (4.18) holds due to the uniform distribution of the $\boldsymbol{\tau}_i$'s. Therefore, authentication tags are mutually independent, and the lemma follows. ■

We can now proceed with the proof of Theorem 1.

Proof: [Theorem 1] There are two functions of the plaintext that are transmitted to the intended receiver: the ciphertext and the authentication tag. By Lemma 4, each authentication tag is statistically independent of its corresponding message and the $\mathcal{E}$-MAC key. Therefore, no information about the encrypted message nor the $\mathcal{E}$-MAC key can be exposed by the observation of their corresponding tag. Furthermore, also by Lemma 4, different authentication tags are mutually independent. Therefore, no advantage can be gained by the observation of multiple authentication tags. Consequently, unless private information is exposed by the observed ciphertexts, no information about the encrypted messages or the $\mathcal{E}$-MAC key will be exposed by the observed authentication tags.

Now, let $\mathcal{A}$ be an adversary against the privacy of the E&M composition and let $\mathcal{B}$ be an adversary with access oracle to the encryption algorithm $\mathcal{E}$. Adversary $\mathcal{A}$ runs adversary $\mathcal{B}$ to attack the privacy of observed ciphertexts. Then,

$$\mathsf{Adv}^{\text{priv}}_{\mathcal{E}\text{-MAC}_{\text{E\&M}}}(\mathcal{A}) \leq \mathsf{Adv}^{\text{ind-cpa}}_{\mathcal{E}}(\mathcal{B})$$

as desired. ■

We now state the theorem concerning the privacy of the MAC-then-Encrypt composition of Section 4.3.3.

Theorem 2. *Let $\mathcal{E}$-MAC$_{\text{MtE}}$ be the authenticated encryption composition described in Section 4.3.3. Then given an adversary, $\mathcal{A}$, against the privacy of $\mathcal{E}$-MAC$_{\text{MtE}}$, one can construct an adversary $\mathcal{B}$ against $\mathcal{E}$ such that*

$$\mathsf{Adv}^{\text{priv}}_{\mathcal{E}\text{-MAC}_{\text{MtE}}}(\mathcal{A}) \leq \mathsf{Adv}^{\text{ind-cpa}}_{\mathcal{E}}(\mathcal{B}). \tag{4.19}$$

The proof of Theorem 2 is similar to the proof of Theorem 1 and, thus, is omitted. The only difference here is that the privacy of the authentication tag is not obtained from the coin tosses, the r's, but rather by encrypting the compressed image with the underlying encryption algorithm.

We will now state the main theorem regarding the probability of successful forgery against the proposed $\mathcal{E}$-MAC.

4.5.3 Security of Authentication

Let $\mathsf{Adv}^{\text{auth}}_{\mathcal{E}\text{-MAC}}(\mathcal{A})$ denotes adversary's $\mathcal{A}$ advantage of successful forgery against the generic compositions described in Sections 4.3.2 and 4.3.3. We give here information-theoretic bounds on the adversary's probability of successful forgery assuming the underlying encryption algorithm is information-theoretically secure (the complexity-theoretic analog is discussed after the theorem statement).

Theorem 3. *Let $\mathcal{A}$ be an adversary making a q signing queries before attempting a forgery on the proposed $\mathcal{E}$-MAC. Provided the information-theoretic security of the underlying encryption scheme, the probability that $\mathcal{A}$ is successful is at most*

$$\mathsf{Adv}^{\text{auth}}_{\mathcal{E}\text{-MAC}}(\mathcal{A}) \leq \begin{cases} \dfrac{1}{p} & \text{if } \ q = 0 \\ \\ \dfrac{1}{p-1} & \text{if } \ q > 0. \end{cases} \tag{4.20}$$

It is standard to pass to a complexity-theoretic analog of Theorem 3. One gets the following. Let $\mathcal{A}$ be an adversary with oracle access to the generic compositions of Sections 4.3.2 and 4.3.3. Then, there is an adversary $\mathcal{B}$ attacking the privacy of the underlying encryption algorithm in which

$$\mathsf{Adv}^{\text{auth}}_{\mathcal{E}\text{-MAC}}(\mathcal{A}) \leq \mathsf{Adv}^{\text{ind-cpa}}_{\mathcal{E}}(\mathcal{B}) + \frac{1}{p-1}.$$

Proof: [Theorem 3] By Lemma 4, the tag is uniformly distributed over $\mathbb{Z}_p$. Hence, if the adversary makes no signing queries, the probability of forging a valid tag is $1/p$.

Assume that the adversary has queried the signing oracle $\mathcal{S}(K, \cdot)$ for q times and recorded the sequence $(M_1, \tau_1), \cdots, (M_q, \tau_q)$.

Now, consider calling the query $\mathcal{V}(K, M', \tau')$, where M' and τ' are any message-tag pair of the adversary's choice. We aim to bound the probability of successful forgery for an M' that has not been queried to the signing oracle; that is, $M' \neq M_i$ for any $i = 1, \cdots, q$. We break the proof into two cases: queried tag and unqueried tag. (In the case of the E&M composition of Section 4.3.2, r_i will be denoted as the B^{th} block of the i^{th} message, that is, $r = m_{i_B}$.)

Queried tag $(M', \tau' = \tau_i)$: Assume that $\tau' = \tau_i$ for an $i \in \{1, \cdots, q\}$. This case represents the event that a collision in the hashing operation occurs. Then, $\mathcal{V}(k, M', \tau') = 1$ if and only if the following holds:

$$\sum_{\ell=1}^{B} k_\ell \; m'_\ell \overset{?}{\equiv} \tau' \equiv \tau_i \equiv \sum_{\ell=1}^{B} k_\ell \; m_\ell \mod p, \tag{4.21}$$

where m'_ℓ denotes the ℓ^{th} block of M' and m_ℓ denotes the ℓ^{th} block of M_i (note that we write m_ℓ instead of m_{i_ℓ} for ease of notations since no distinction between different messages is necessary). We will analyze equation (4.21) by considering the following three cases: M' and M_i differ by a single block, M' and M_i differ by two blocks, or M' and M_i differ by more than two blocks.

1. Assume that only a single message block is different. Since addition is commutative, assume without loss of generality that the first message block is different; that is, $m'_1 \not\equiv m_1 \mod p$. Since only the first message block is different, equation (4.21) is equivalent to
$$k_1 m'_1 \equiv k_1 m_1 \mod p. \tag{4.22}$$
Therefore, by Lemma 2, the probability of successful forgery given a single block difference is *zero*.

2. Assume, without loss of generality, that the first two message blocks are different; i.e., $m'_1 \equiv m_1 + \delta_1 \not\equiv m_1 \mod p$ and $m'_2 \equiv m_2 + \delta_2 \not\equiv m_2 \mod p$. Then, equation (4.21) is equivalent to
$$k_1\delta_1 + k_2\delta_2 \equiv 0 \mod p. \tag{4.23}$$
Therefore, by Lemma 3, the probability of successful forgery given that exactly two message blocks are different is at most $1/(p-1)$.

3. Assume that more than two message blocks are different, i.e., $m'_j \equiv m_j + \delta_j \not\equiv m_j \mod p$; $\forall\ j \in J \subseteq \{1, 2, \cdots, B\}$; $|J| \geq 3$. Then, equation (4.21) is equivalent to
$$k_j\delta_j + \sum_{\substack{\ell\in J\\ \ell\neq j}} k_\ell\delta_\ell \equiv 0 \mod p, \tag{4.24}$$
for some $j \in J$. Therefore, using Lemma 3 and the fact that $\sum_{\ell\in J,\ell\neq j} k_\ell\delta_\ell$ can be congruent to *zero* modulo p, the probability of success is at most $1/p$. (The difference between this case and the case of exactly two blocks is that, even if the δ's are chosen to be nonzero integers, $\sum_{\ell\in J,\ell\neq j} k_\ell\delta_\ell$ can still be congruent to zero modulo p.)

From the above three cases, the probability of successful forgery when the forged tag has been queried to the signing oracle is at most $1/(p-1)$.

Unqueried tag (M', τ'): Assume now that the tag τ' is different than all the recorded tags; that is, $\tau' \neq \tau_i$ for any $i = 1, \cdots, q$. If τ' is independent of the recorded tags, then the probability of successful forgery is $1/p$ (using the fact that the tag is uniformly distributed over $\mathbb{Z}_p$). Assume, however, that τ' is a function of τ_i, for an $i \in \{1, \cdots, q\}$. Let $\tau' \equiv \tau_i + \gamma \mod p$ for some $\gamma \in \mathbb{Z}_p \backslash \{0\}$ of the adversary's choice. (Note that, γ can be a function of any value recorded by the adversary.) Then, $\mathcal{V}(K, M', \tau') = 1$ if and only if the following congruence holds:

$$\sum_{\ell=1}^{B} k_\ell \; m'_\ell \overset{?}{\equiv} \tau' \equiv \tau_i + \gamma \equiv \sum_{\ell=1}^{B} k_\ell \; m_\ell + \gamma \mod p, \tag{4.25}$$

where m'_ℓ denotes the ℓ^{th} block of M' and m_ℓ denotes the ℓ^{th} block of M_i. Bellow we analyze equation (4.25) by considering two cases: M' and M_i differ by a single block, or M' and M_i differ by more than one block.

1. Without loss of generality, assume that M' and M_i differ in the first block only. That is $m'_1 \equiv m_1 + \delta \not\equiv m_1 \mod p$ and $m'_j \equiv m_j \mod p$ for all $j = 2, \cdots, B$. Then, equation (4.25) is equivalent to

$$k_1 \delta \equiv \gamma \mod p. \tag{4.26}$$

   Therefore, by Lemma 3, the probability of success is at most $1/(p-1)$.

2. Assume now that M' and M_i differ by more than one block. That is, $m'_j \equiv m_j + \delta_j \neq m_j \mod p$; $\forall j \in J \subseteq \{1, 2, \cdots, B\}$; $|J| \geq 2$. Then, equation (4.25) is equivalent to

$$\sum_{j \in J} k_j \delta_j \equiv \gamma \mod p. \tag{4.27}$$

   By Lemma 3 and the fact that $\sum_{j \in J} k_j \delta_j$ can be congruent to *zero* modulo p, the probability of success is at most $1/p$.

From the above two cases, the probability of successful forgery when the forged tag has not been queried is at most $1/(p-1)$.

Therefore, given that $\mathcal{A}$ has made at least one signing query, $\mathcal{A}$'s probability of successful forgery for each verify query is at most $1/(p-1)$. ■

Remark 6. *The proof of Theorem 3 gives a tighter bound on the used universal hash family. Specifically, the case of queried tag implies that the used hash family is $(\frac{1}{p-1})$-AU. Similarly, the case of unqueried tag implies that the used hash family is $(\frac{1}{p-1})$-$A\Delta U$. The proof also illustrates why any ϵ-$A\Delta U$ hash family can be used to construct the proposed $\mathcal{E}$-MAC. That is, any ϵ-$A\Delta U$ hash family will have a probability of successful forgery given an unqueried tag less than ϵ.*

We now show that the proposed $\mathcal{E}$-MACs are strongly unforgeable under chosen message attacks (SUF-CMA). Recall that SUF-CMA requires that it be computationally infeasible for the adversary to find a new message-tag pair after chosen-message attacks even if the message is not new, as long as the tag has not been attached to the message by a legitimate user [34].

Theorem 4. *The* E&M *generic composition using the $\mathcal{E}$-MAC described in Section 4.3.2 is strongly unforgeable under chosen message attacks.*

Proof: Let (M, τ) be a valid message tag pair. Assume that the adversary is attempting to authenticate the same message with a different tag τ'. Since the plaintext message is the same but the tag is different, the r corresponding to τ must be different than the r' corresponding to τ'. For the (M, τ') pair to be authenticated, $\sum_i k_i m_i + k_B r' \mod p$ must be equal to τ'. That is, given τ', r' must be set to $k_B^{-1}(\tau' - \sum_i k_i m_i) \mod p$ for the tag to be authenticated. By Theorems 1, however, the adversary cannot expose the $\mathcal{E}$-MAC's key. Therefore, Theorem 3 holds whether or not the message is new, as long as the tag has not been attached to the message by the signing oracle. ■

The MtE composition of Section 4.3.3 requires more discussion. If the encryption algorithm is deterministic, then the same message cannot be authenticated with two distinct tags. However, the use of deterministic encryption algorithm violates the assumption that the underlying encryption provides indistinguishability under chosen plaintext attacks (an encryption algorithm with IND-CPA must be probabilistic [96, 138]). Although most practical secure encryption algorithms that can be used to construct the MtE of Section 4.3.3 will result in a strongly unforgeable authentication, one can come up with an algorithm that satisfies IND-CPA but does not result in a strongly unforgeable authentication when used to compose the MtE system of Section 4.3.3. To guarantee strong unforgeability for all constructions, the last message block can be replaced by a random string, in which case the proof of strong unforgeability will be the same as the proof of Theorem 4.

4.5.4 Security of the Generic Compositions

In [34], Bellare and Namprempre defined two notions of integrity in authenticated encryption schemes, integrity of plaintexts (INT-PTXT) and integrity of ciphertexts (INT-CTXT). INT-PTXT implies that it is computationally infeasible for an adversary to produce a ciphertext decrypting to a message which the sender had never encrypted, while INT-CTXT implies that it is computationally infeasible for an adversary to produce a ciphertext not previously produced by the sender, regardless of whether or not the corresponding plaintext is *new*. By combining an encryption algorithm that provides indistinguishability under chosen-plaintext attacks (IND-CPA) and a MAC algorithm that is unforgeable under chosen-message attack, the work in [34] analyzes the security of the three generic compositions.

In [34], Bellare and Namprempre showed that the E&M and MtE generic compositions are generally insecure, the results do not apply to all variants of E&M and MtE constructions. For instance, as per [34], E&M compositions do not generally provide IND-CPA because there exist secure MACs that reveal information about the plaintext (the authors of [34] provide a detailed example). Obviously, if such a MAC is used in the construction of an E&M system, the resulting composition will not provide IND-CPA. Unlike standard MACs, however, it is a basic requirement of $\mathcal{E}$-MACs to be as secret as the used encryption algorithm. Indeed, Theorem 1 guarantees that the proposed E&M composition does not reveal any information about the plaintext that is not revealed by the ciphertext.

Another result of [34] is that generic E&M and MtE compositions do not provide INT-

CTXT. (Although the authors acknowledged that the notion of INT-PTXT is the more natural security requirement while the interest of the stronger INT-CTXT notion is more in the security implications derived in [34].) The reason why E&M and MtE compositions generally do not provide INT-CTXT is that one can come up with a secure encryption algorithm with the property that a ciphertext can be modified without changing its decryption [34]. Obviously, when such an encryption algorithm is combined with the proposed $\mathcal{E}$-MACs to construct an E&M or MtE system, since the tag is computed as a function of the plaintext, only INT-PTXT is reached.

In practice, however, it is possible to construct E&M and MtE systems that do provide INT-CTXT. For instance, a sufficient condition for the proposed $\mathcal{E}$-MAC to provide INT-CTXT for the composed system is to be used with a secure *one-to-one* encryption algorithm. To see this observe that any modification of the ciphertext will correspond to modifying the plaintext (since the encryption is one-to-one). Therefore, by Theorem 3, modified ciphertexts can only be accepted with negligible probabilities. Indeed, secure E&M and MtE systems have been constructed in practice. Popular examples of such constructions are SSH [271], SSL [84] and TLS [64], which use variants of the E&M and MtE compositions that are known to be secure [150, 33, 175, 237].

So far, we have shown that $\mathcal{E}$-MACs can be used to replace standard MACs in the construction of E&M and MtE systems with two additional properties: they can have provable confidentiality and they can be more efficient. What we will show next is that $\mathcal{E}$-MACs can have another security advantage. More specifically, we will show that $\mathcal{E}$-MACs can utilize the structure of the E&M system to achieve better resilience to a new attack on universal hash functions based MACs; namely, the key-recovery attack [106].

4.6 $\mathcal{E}$-MACs and Key Recovery

Recently, Handschuh and Preneel [106] showed that, compared to block cipher based, MACs based on universal hash functions have a key-recovery vulnerability. In principle, a small probability of successful forgery on authentication codes is always possible. However, the work in [106] demonstrates that, for universal hash functions based MACs, once a successful forgery is achieved, subsequent forgeries can succeed with high probabilities. The main idea in their attacks is to look for a collision in the message compression phase. Once a message that causes a collision is found, partial information about the hashing keys can be exposed. Using this key information an attacker can forge valid tags for fake messages. We give a detailed example below.

Example 1. *Consider the universal hash family presented in this chapter. Assume an adversary calling the signing oracle on $M = m_1||m_2$, thus obtaining its authentication tag τ. The adversary now can call the verification oracle with $M = m_2||m_1$ and the same tag τ. Obviously, the verification will pass if and only if $k_1 \equiv k_2 \mod p$ (in which case $k_1m_1 + k_2m_2 \equiv k_2m_1 + k_1m_2 \mod p$).*

Although the verification will pass with a small probability, the adversary can continuously call the verification oracle with $M = m_2||\alpha_i m_1$, for different α_i's until the message is authenticated. Let $M = m_2||\alpha m_1$ be the message that passes the verification test, for some $\alpha \in \mathbb{Z}_p^*$. Then, the relation

$$k_1 \equiv \beta k_2 \mod p, \tag{4.28}$$

where $\beta = (\alpha m_1 - m_2)(m_1 - m_2)^{-1}$ is exposed to the adversary. With this knowledge, a man in the middle can always replace the first two blocks, $m_1||m_2$, of any future message M with $\beta^{-1}m_2||\beta m_1$ without violating its tag. This is because $k_1(\beta^{-1}m_2) + k_2(\beta m_1) \equiv k_2m_2 + k_1m_1 \mod p$ regardless of values of m_1 and m_2.

Handschuh and Preneel [106] defined three classes of weak keys in universal hash functions. Each class can be exploited in a way similar to the one discussed in the above example to substantially increase the probability of successful forgery after a single collision. This attack is shared by all universal hash based MACs [106]. As per [106], the recommended mitigations to this attack are to use the less efficient block cipher based MACs, or not to reuse the same hashing key for multiple authentication.

Compared to standard MACs, however, $\mathcal{E}$-MACs can utilize the structure of the E&M and MtE systems to overcome the key-recovery problem discovered in [106]. Consider the $\mathcal{E}$-MAC proposed in Section 4.3, and recall that a random number $r \in_R \mathbb{Z}_p$ is generated internally in the E&M process. In the basic construction of Section 4.3, the goal of r

is to encrypt the authentication tag. However, the random r can play a pivotal role in key-recovery security.

In the basic construction in Section 4.3, the universal hashing key is $K = k_1||k_2||\cdots||k_B$ and the authentication tag is computed as:

$$\tau = \sum_{i=1}^{L} k_i m_i + k_B r \mod p. \tag{4.29}$$

Now, with the same shared key, consider another use of r. More specifically, let the authentication tag be computed as follows:

$$\tau = \sum_{i=1}^{L} (k_i \oplus r) m_i + k_B r \mod p. \tag{4.30}$$

In other words, r can be used to randomize the key in every authentication call.

Assume the same attack described in Example 1 and let $M = m_2||\alpha m_1$ passes the verification test, for some $\alpha \in \mathbb{Z}_p^*$. This time, however,

$$k_1' \equiv \beta k_2' \mod p, \tag{4.31}$$

where $k_1' = k_1 \oplus r$, $k_2' = k_2 \oplus r$, and $\beta = (\alpha m_1 - m_2)(m_1 - m_2)^{-1}$ is the relation revealed to the adversary. For any future authentication, the sender will generate a new random number r' that is independent of r. Thus, the keys that will be used for authentication will be k_1'' and k_2'', where $k_i'' = k_i \oplus r'$ for $i = 1, 2$. That is, from the standpoint of key-recovery attacks, by using equation (4.30) instead of equation (4.29), different authentication tags are computed with different keys. Therefore, finding a collision in the message compression phase does not lead to information leakage about the keys, as long as the same nonce does not authenticate different messages. (Note that there is no need to randomize k_B since it is independent of the message to be authenticated.)

Remark 7. *This shows how the system can be designed to utilize the authenticated encryption application to increase the robustness of universal hash functions based $\mathcal{E}$-MACs. This could not have been achieved without the use of the fresh random number r that was secretly delivered to the verifier as part of the ciphertext.*

4.7 Summary

In this chapter, we studied the generic composition of authenticated encryption systems. We introduced $\mathcal{E}$-MACs, a new symmetric-key cryptographic primitive that can be used in the construction of E&M and MtE compositions. By taking advantage of the E&M and MtE structures, the use of $\mathcal{E}$-MACs is shown to improve the efficiency and security of the authentication operation. More precisely, since the message to be authenticated is encrypted, universal hash functions based $\mathcal{E}$-MACs can designed without the need to apply cryptographic operations on the compressed image, since this can be replaced by operations performed by the encryption algorithm. Further, by appending a random string at the end of the plaintext message, $\mathcal{E}$-MAC can be secured against key-recovery attacks.

Chapter 5

UTILIZING BLOCK CIPHER SECURITY: KEYLESS MACS

In the previous chapter, we showed how the IND-CPA security of encryption algorithms can be utilized to improve the efficiency of MAC algorithms. In this chapter, we show how the pseudorandomness property of encryption algorithm can be utilized to further improve MACs efficiency. We propose a new security model that captures the differences between standard MACs and those used in the design of MtE or E&M generic compositions. We then utilize the new model to design a new MAC that is faster than the fastest MAC reported in the literature of cryptography. We show that, when coupled with a strong pseudorandom permutation (i.e., a block cipher), the first secure keyless MAC in the literature of cryptography (named "KMAC") can be constructed. KMAC utilizes the security of the encryption primitive, and the special structure of the MtE and E&M compositions, to considerably improve the efficiency of message authentication, even when compared to the fastest MAC algorithms in the literature. We also show that the proposed construction gives better performance even when compared to the most efficient dedicated authenticated encryption schemes in the literature.

5.1 The New Security Model

In this section, we will describe one of the main contributions of this chapter. Namely, the details of the new model that will be used for analyzing MACs used in the MtE and E&M generic compositions. Note that this model can be used to analyze the security of the $\mathcal{E}$-MACs introduced in Chapter 4. However, while the security of the $\mathcal{E}$-MAC proposed in Chapter 4 can be proven using the standard security model, the security of KMAC (which is an instance of $\mathcal{E}$-MACs) cannot be proven using the standard security model. Therefore, we delayed the formal description of the new model to this chapter.

Recall that, in the standard setup, a message authentication scheme consists of two algorithms: a signing algorithm $\mathcal{S}$ and a verifying algorithm $\mathcal{V}$. While the signing algorithm might be probabilistic, the verifying algorithm is usually not. On input a key K and a message M, algorithm $\mathcal{S}$ outputs a string τ called the authentication tag, or simply the tag of M. On input a key K, a message M, and an authentication tag τ, algorithm $\mathcal{V}$ outputs the bit 1 when the message is authentic, and the bit 0 when the message is not.

To model the security of a message authentication scheme, a probabilistic polynomial time adversary, $\mathcal{A}$, is given oracle access to the signing and verifying algorithms. After calling the signing and verifying algorithms a polynomial number of times on messages of its choice and observing the outputs, $\mathcal{A}$ attempts to generate a new massage-tag pair that will be accepted as valid, for a tag that has not been attached to the message by the signing oracle.

This standard model, however, does not properly address the security of message authentication codes in our setup. To see why, observe that the message to be authenticated in our setup must also be encrypted. That is, what the intended destination receives is a ciphertext-tag pair, as opposed to plaintext-tag pair in the standard model. As will be demonstrated in the remainder of the chapter, this observation is critical for the design of more efficient MACs to be used in the construction of the MtE or E&M generic compositions.

To properly model the security of MACs in our setup, we introduce the following modification. Let $\mathcal{E}$ be the underlying encryption algorithm. Depending on the mode of operation, $\mathcal{E}$ may or may not require the use of a nonce for encryption. If the encryption algorithm requires the use of nonces, the input to the algorithm is a nonce-message pair (N, M); otherwise, the input to the encryption algorithm is simply a plaintext message M. In the proposed model, the signing oracle internally calls the encryption algorithm and outputs a ciphertext-tag pair. That is, given an encryption algorithm $\mathcal{E}$, on input a key K, and a nonce-message pair (N, M), the signing algorithm $\mathcal{S}_{\mathcal{E}}(K, N, M)$ outputs (c, τ), where c is the ciphertext corresponding to (N, M) and τ is the authentication tag. (For the special case in which the MAC is keyless, as in our proposed MAC, or if the encryption algorithm does not requires the use of a nonce, the key K and the nonce N can be the empty string

λ.)

The verifying oracle must also be modified to properly model the system. That is, given the decryption algorithm $\mathcal{D}$ corresponding to $\mathcal{E}$, on input a key K, a nonce N, a ciphertext c, and an authentication tag τ, the verifying oracle $\mathcal{V}_{\mathcal{D}}$ outputs a bit, with 1 standing for accept and 0 for reject. We ask for a basic validity condition, namely that authentic tags are accepted with probability one. That is, if $(c, \tau) = \mathcal{S}_{\mathcal{E}}(K, N, M)$, it must be the case that $\mathcal{V}_{\mathcal{D}}(K, N, c, \tau) = 1$ for any encryption/decryption algorithms, key K, nonce N, ciphertext c, and authentication tag τ.

As in the standard model, an adversary is a probabilistic polynomial time algorithm, $\mathcal{A}$. The adversary is given oracle access to algorithms $\mathcal{S}_{\mathcal{E}}(K, \cdot, \cdot)$ and $\mathcal{V}_{\mathcal{D}}(K, \cdot, \cdot, \cdot)$ for a random but hidden choice of K. $\mathcal{A}$ can query $\mathcal{S}_{\mathcal{E}}$ to generate a ciphertext-tag pair for a none-message of its choice and ask the verifier $\mathcal{V}_{\mathcal{D}}$ to verify that (N, c, τ) is a valid tuple. Formally, $\mathcal{A}$'s attack on the scheme is described by the following experiment:

1. A random string is selected as the shared secret, K.

2. Suppose $\mathcal{A}$ makes a signing query (N, M). The oracle computes $(c, \tau) \leftarrow \mathcal{S}_{\mathcal{E}}(K, N, M)$, the ciphertext-tag pair, and returns it to $\mathcal{A}$. (Since $\mathcal{S}_{\mathcal{E}}$ is typically probabilistic, this step requires making the necessary coin tosses, anew for each signing query.)

3. Suppose $\mathcal{A}$ makes a verify query (N, c, τ). The oracle computes the decision $d = \mathcal{V}_{\mathcal{D}}(K, N, c, \tau)$ and returns it to $\mathcal{A}$.

Note that the encryption and decryption algorithms require a secret key, call it $K_{\mathcal{E}}$, which is independent of the MAC key K. Note also that $\mathcal{A}$ does not see the encryption-decryption key, the MAC key, nor the coin tosses of $\mathcal{S}_{\mathcal{E}}$.

An adversary is said to be nonce-respecting if she never repeats a nonce. That is, after calling the signed encryption oracle on (N, M), the adversary never asks its oracle a query (N, M'), regardless of the oracle responses. We emphasize, however, that the nonce used in the forgery attempt may coincide with a nonce used in one of the adversary's queries to the signing oracle.

$\mathcal{A}$ can query the signing oracle q number of times and record the outputs. $\mathcal{A}$ then stops and attempts its forgery. The outcome of running the experiment in the presence of an adversary is used to define security. We say that $\mathcal{A}$ is successful if it is nonce-respecting and makes a verify query (N, c, τ) which is accepted, for an (N, c, τ) tuple that has not been outputted by the signing oracle $\mathcal{S}_{\mathcal{E}}$.

5.2 The Proposed KMAC

In this section, we describe the details of the proposed keyless MAC (KMAC). Obviously, no keyless MAC can be secure in the standard settings. The novelty of this work, however, is to utilize the non-malleability [65] of encryption algorithms and the special structure of the MtE and E&M compositions (captured by the new security model of section 5.1), to come up with the first secure keyless MAC. To the best of our knowledge, the utilization of the special characteristics of the generic composition of authenticated encryption schemes, as we do in this work, has not appeared the cryptographic literature. For the rest of the chapter, we will describe only the E&M composition. However, the main concepts can also be applied to MtE. We start by specifying the necessary conditions that must be satisfied by an encryption algorithm to be used in our construction.

5.2.1 Requirements on Encryption

The main idea behind the efficiency of the proposed authentication scheme is to assume that the encryption algorithm, $\mathcal{E}$, acts as a random permutation, so that the effect of changing one plaintext bit will have an impact on all ciphertext bits. However, since building random permutations for messages of arbitrary lengths is impractical, one must resort to modes of encryptions based on pseudorandom permutations (i.e., block ciphers). To simulate a secure pseudorandom encryption, the following are necessary conditions the encryption algorithm must satisfy.

1. Encryption is probabilistic (this is required to achieve indistinguishability under chosen plaintext attacks [96]),

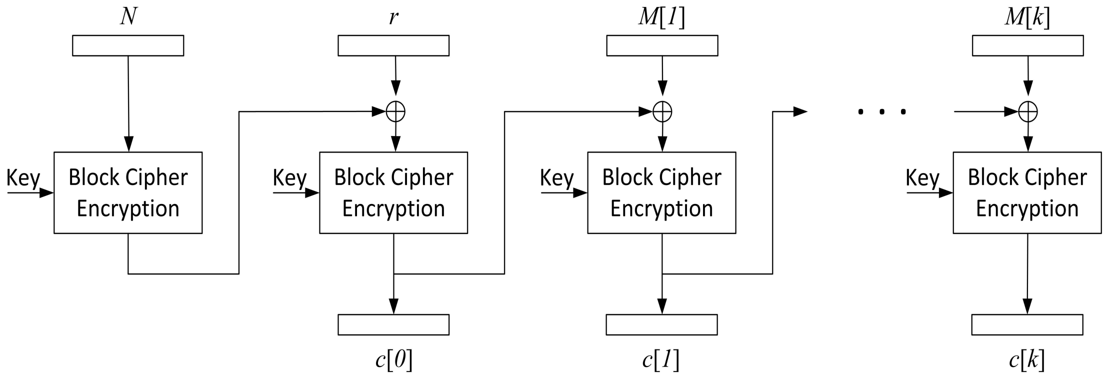


Figure 5.1: Illustration of the cipher block chaining (CBC) non-parallelizable mode of encryption. N is a nonce, r is the coin tosses of the signing algorithm, $M[i]$ is the i^{th} plaintext block of a message consisting of k blocks, and $c[i]$ is the i^{th} ciphertext block.

2. plaintext blocks must be processed by a block cipher in which its input-output relation cannot be distinguished from a random permutation using reasonable resources (most practical block ciphers are believed to be pseudorandom permutations [138]),

3. encryption is secure against cut-and-past attacks (in which two encryptions of different messages under the same key are combined to form a new ciphertext [38]); this implies that swapping two ciphertext blocks does not decrypt to swapping their corresponding plaintext blocks, except with a negligible probability; it also implies that truncating a given ciphertext does not decrypt to the truncation of its corresponding plaintext.

To support our claim of generic composition, we give in Figures 5.1 and 5.2 two different modes of encryption that can be used in our construction; other modes of encryption can also be used provided they satisfy the above necessary conditions. In Figure 5.1, we show an example based on the non-parallelizable cipher block chaining (CBC) mode of encryption (variants of the CBC mode, such as the propagating cipher block chaining (PCBC) can also be used). In the example of Figure 5.1, N is a nonce which can simply be a counter that increments every time the encryption oracle is called (similar to the initialization vector (IV) in the standard CBC mode, N will ensure that the encryption algorithm is

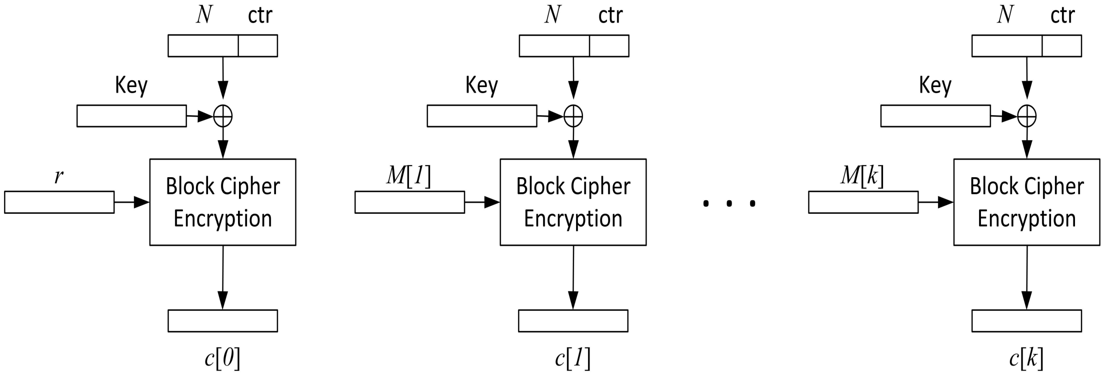


Figure 5.2: Illustration of the counter based parallelizable mode of encryption. N is a nonce, ctr is a counter that increments each block, r is the coin tosses of the signing algorithm, $M[i]$ is the i^{th} plaintext block of a message consisting of k blocks, and $c[i]$ is the i^{th} ciphertext block.

probabilistic). r is the coin tosses of the signing algorithm that plays an important role in message authentication. While r must remain secret, N is transmitted to the intended receiver in clear text format since both N and c are required for decryption.

Since non-parallelizable modes of encryption can be inefficient in scenarios where parallel computing is available, we show in Figure 5.2 a parallelizable mode that satisfies the above conditions. The mode of encryption of Figure 5.2 utilizes the use of a nonce and a counter. The concatenation of the nonce and the counter is of length equal to that of the block cipher key. The same nonce cannot be used twice for two encryption operation unless a new key for the block cipher has been randomly selected. The maximum length of plaintext messages that can be encrypted using the algorithm of Figure 5.2 is exponential in the length of the counter. Since typical key lengths are sufficiently long (e.g., 128, 192, or 256 in AES), deciding these two parameters is not a challenging issue. For example, choosing $|N| = 88$ and $|\mathsf{ctr}| = 40$ when the key length is 128 can be found in [147]. Similar to the mode of encryption in Figure 5.1, both the nonce, N, and the ciphertext, c, are required for decryption.

To see that the necessary conditions listed above are satisfied by both modes of encryption of Figures 5.1 and 5.2, observe that the first condition is satisfied via the use of the nonce. The second condition is satisfied by both modes of encryption of Figures 5.1 and 5.2 by the use of a block cipher that satisfies the strong pseudorandomness property. The fact that swapping two ciphertext blocks does not decrypt to their corresponding plaintext blocks is satisfied by the mode of encryption of Figure 5.1 via its serialized nature and satisfied by the mode of encryption of Figure 5.2 via the use of the nonce and counter. One way of guaranteeing that truncating a given ciphertext does not decrypt to the truncation of its corresponding plaintext can be achieved by encoding the message with a unique End-of-Message character.

5.2.2 KMAC Description

Let M be the plaintext message to be authenticated and n be a security parameter agreed upon by legitimate users. (For ease of notation, we will assume that n is equal to the size of the block cipher used to construct the encryption algorithm; we emphasize, however, that n can be different.) Append a unique End-of-Message character to the end of M and divide M into $k := \lceil \frac{|M|}{n} \rceil$ blocks, each of length n-bits, except possibly the k^{th} block. Append the k^{th} block of M with ℓ zeros, where $\ell = n - \Big(|M| \pmod n \Big)$, so that it becomes an n-bit long string. Let σ be the n-bit compressed image of M, evaluated as follows.

$$\sigma = \sum_{i=1}^{k} M[i] \pmod{2^n}, \tag{5.1}$$

where $M[i]$ denotes the i^{th} block of the message M. (We overload $M[i]$ to denote both the n-bit binary string in the i^{th} block and its integer representation as an element of $\mathbb{Z}_{2^n}$ in a big-endian format; the distinction between the two representations will be omitted whenever it is clear from the context.)

Given a plaintext message M, compute its compressed image according to equation (5.1). Generate a string r, drawn uniformly at random from $\{0,1\}^n$, and append it to the message M (for the rest of the chapter, r will be referred to as the coin tosses of the signing algorithm). We emphasize that the r's generated at different signing operations must be mutually independent. Using the underlying encryption algorithm $\mathcal{E}$, encrypt $r||M$ to get the corresponding ciphertext; i.e.,

$$c = \mathcal{E}(N, r||M), \tag{5.2}$$

where N is a nonce (assuming the encryption algorithm requires the use of nonces). The authentication tag of M is simply

$$\tau = \sigma + r \pmod{2^n}. \tag{5.3}$$

The tuple (N, c, τ) from equations (5.2) and (5.3) is then transmitted to the intended receiver.

Given (N, c), the intended receiver decrypts the ciphertext to obtain the plaintext message, M, and the coin tosses, r. The receiver then breaks M into its n-bit blocks (the $M[i]$'s), computes the modular summation $\sum_i M[i] + r \pmod{2^n}$, and authenticates the message if and only if the summation is congruent to the received tag τ. Formally, the following integrity check must be satisfied to validate the message

$$\tau \overset{?}{\equiv} \sum_{i=1}^{k} M[i] + r \pmod{2^n}. \tag{5.4}$$

Remark 8. *We emphasize that r requires no key management; that is, it is not a shared one-time pad. The coin tosses, r, is delivered to the receiver via the ciphertext, just like the plaintext message.*

In the security analysis of the remainder of the chapter, we assume that the parallelizable mode of encryption of Figure 5.2 is used with KMAC for the generic composition. The main ideas can also be used to analyze the construction when the mode of encryption of Figure 5.1 is used; only few modifications are needed.

5.3 Security Analysis

In this section, we will analyze the security of the composition in Section 5.2. The analysis will illustrate the importance of the new model of Section 5.1 for the design of highly efficient and highly secure MACs.

5.3.1 General Lemma

We start here by stating the a general lemma that will be used for the remaining of this section.

Lemma 5. *In the proposed* KMAC *of Section 5.2, authentication tags are statistically independent of their corresponding plaintext messages. Furthermore, authentication tags corresponding to different messages are mutually independent.*

Proof: Recall that the coin tosses, r, generated during the E&M process of Section 5.2 is uniformly distributed over the set of all possible n-bit binary strings, $\{0,1\}^n$. Then, for any possible value of τ computed according to equation (5.3), and any possible plaintext message M, the following holds:

$$\Pr\Big[\boldsymbol{\tau} = \tau | \boldsymbol{M} = M\Big] = \Pr\Big[\boldsymbol{r} = (\tau - \sum_{i=1}^{k} M[i])\Big] = 2^{-n}, \tag{5.5}$$

where $M[i]$ denotes the i^{th} block of the plaintext message M, and $\boldsymbol{\tau}$, $\boldsymbol{M}$, and $\boldsymbol{r}$, denote the random variables representing the values of τ, M, and r selected according to their respective distributions.

Furthermore, for an r drawn uniformly at random from $\{0,1\}^n$, by Lemma 1, the resulting tag is uniformly distributed over $\{0,1\}^n$. That is, for any fixed value $\tau \in \mathbb{Z}_{2^n}$, the probability that the tag will take this specific value is given by:

$$\Pr\Big[\boldsymbol{\tau} = \tau\Big] = 2^{-n}. \tag{5.6}$$

Combining Bayes' theorem [101] with equations (5.5) and (5.6) yields:

$$\begin{aligned} \Pr\Big[\boldsymbol{M} = M | \boldsymbol{\tau} = \tau\Big] &= \frac{\Pr\Big[\boldsymbol{\tau} = \tau | \boldsymbol{M} = M\Big] \Pr\Big[\boldsymbol{M} = M\Big]}{\Pr\Big[\boldsymbol{\tau} = \tau\Big]} && (5.7) \\ &= \Pr\Big[\boldsymbol{M} = M\Big], && (5.8) \end{aligned}$$

for any plaintext M and any authentication tag τ. That is, authentication tags are statistically independent of their corresponding plaintext messages. In other words, the observation of an authentication tag gives no information about its corresponding plaintext message, as required.

To show that different authentication tags are mutually independent, let $\boldsymbol{M_1}$ through $\boldsymbol{M_\ell}$ denote the random variables representing the experiments of drawing messages M_1 through M_ℓ according to any arbitrary probabilistic distribution. Similarly, let $\boldsymbol{\tau_1}$ through $\boldsymbol{\tau_\ell}$ denote the random variables representing the authentication tags corresponding to messages M_1 through M_ℓ, respectively. Further, let $\boldsymbol{r_1}$ through $\boldsymbol{r_\ell}$ be the random variables representing the coin tosses of the signing algorithm $\mathcal{S}_\mathcal{E}$ for the authentication of messages M_1 through M_ℓ, respectively. Recall that the $\boldsymbol{r_i}$'s are mutually independent and identically distributed (iid) uniform random variables drawn from $\{0,1\}^n$. Then, for any possible values of the messages M_1 through M_ℓ with arbitrary joint probability mass function, and all possible values of τ_1 through τ_ℓ, we get:

$$
\begin{aligned}
&\Pr\Big[\boldsymbol{\tau_1}=\tau_1,\cdots,\boldsymbol{\tau_\ell}=\tau_\ell\Big]\\
&=\sum_{M_1,\cdots,M_\ell}\Pr\Big[\boldsymbol{\tau_1}=\tau_1,\cdots,\boldsymbol{\tau_\ell}=\tau_\ell|\boldsymbol{M_1}=M_1,\cdots,\boldsymbol{M_\ell}=M_\ell\Big]\\
&\qquad\qquad\qquad\qquad\cdot\Pr\Big[\boldsymbol{M_1}=M_1,\cdots,\boldsymbol{M_\ell}=M_\ell\Big] \quad (5.9)\\
&=\sum_{M_1,\cdots,M_\ell}\Pr\Big[\boldsymbol{r_1}=\Big(\tau_1-\sum_{i=1}^{k}M_1[i]\Big),\cdots,\boldsymbol{r_\ell}=\Big(\tau_\ell-\sum_{i=1}^{k}M_\ell[i]\Big)\Big]\\
&\qquad\qquad\qquad\qquad\cdot\Pr\Big[\boldsymbol{M_1}=M_1,\cdots,\boldsymbol{M_\ell}=M_\ell\Big] \quad (5.10)\\
&=\sum_{M_1,\cdots,M_\ell}\Pr\Big[\boldsymbol{r_1}=\Big(\tau_1-\sum_{i=1}^{k}M_1[i]\Big)\Big]\cdots\Pr\Big[\boldsymbol{r_\ell}=\Big(\tau_\ell-\sum_{i=1}^{k}M_\ell[i]\Big)\Big]\\
&\qquad\qquad\qquad\qquad\cdot\Pr\Big[\boldsymbol{M_1}=M_1,\cdots,\boldsymbol{M_\ell}=M_\ell\Big] \quad (5.11)\\
&=\sum_{M_1,\cdots,M_\ell}2^{-n}\cdots2^{-n}\cdot\Pr\Big[\boldsymbol{M_1}=M_1,\cdots,\boldsymbol{M_\ell}=M_\ell\Big] \quad (5.12)\\
&=\Pr\Big[\boldsymbol{\tau_1}=\tau_1\Big]\cdots\Pr\Big[\boldsymbol{\tau_\ell}=\tau_\ell\Big], \quad (5.13)
\end{aligned}
$$

where $M_j[i]$ denotes the i^{th} block of the j^{th} message M_j. Equation (5.11) holds due to the mutual independence of the $\boldsymbol{r_i}$'s; equation (5.12) holds due to the uniform distribution of the $\boldsymbol{r_i}$'s; and equation (5.13) holds due to the uniform distribution of the $\boldsymbol{\tau_i}$'s. Therefore, authentication tags are mutually independent, and the lemma follows. ■

5.3.2 Authenticity of the Construction

Before we provide an upper bound on the probability of successful forgery, we give an informal discussion on how the structure of the E&M composition will be utilized. Recall that, in standard MACs, the security is modeled by the adversary's probability of predicting a valid authentication tag for a certain message. That is, given the adversary's knowledge of a polynomial number of valid message-tag pairs, the goal of the adversary is to forge a new message-tag pair that will be accepted as valid.

MACs in E&M compositions, on the other hand, are fundamentally different than standard MACs. The intended receiver in an E&M system receives a ciphertext-tag pair as opposed to message-tag pair. This implies that, even if the adversary is given the ability to launch chosen message attacks, the adversary must come up with a ciphertext-tag pair that will be accepted as valid for the forgery attempt to succeed. This was the key observation behind the design of our keyless MAC, which was captured by our security model.[1]

Consequently, MACs in the E&M composition possess a security advantage over standard MACs. That is, unlike standard MACs, MACs in the E&M composition can benefit from the security of the coupled encryption algorithm. We utilized this fact to relax the security requirement on the MAC algorithm, thus, allowing for a more efficient design. For example, observe that it is easy for an adversary to come up with two different messages that have the same compressed image in the construction of Section 5.2. Although this is sufficient to break the security of standard MACs, it can be insufficient to break the integrity of MACs in the E&M composition. This is because the adversary must also predict the correct ciphertexts of forged messages in the E&M composition.

In what follows we give a formal security treatment of the proposed scheme. Let $\mathcal{E}$ be the underlying encryption algorithm and define $\mathsf{Adv}^{\mathrm{auth}}_{\mathrm{KMAC}}(\mathcal{A}) = \Pr[\mathcal{A}^{\mathcal{S}_{\mathcal{E}}(\cdot,\cdot)} \text{ forges }]$ to be $\mathcal{A}$'s advantage in breaking the authenticity of the proposed KMAC coupled with the encryption algorithm of Figure 5.2 when given oracle access to the signing algorithm $\mathcal{S}_{\mathcal{E}}$. In the next

[1] Clearly, this does not apply to the EtM generic composition. The tag in the EtM composition is a function of the ciphertext which, obviously, must be transmitted to the intended receiver. Therefore, MACs in the EtM compositions must be analyzed under the standard model. As mentioned earlier, however, MACs in the MtE composition can utilize our model.

theorem, we provide an information-theoretic statement regarding the authenticity of the proposed scheme assuming the used block cipher, BC, is a true random permutation.

Theorem 5. *Let* $\mathsf{Perm}(n) : \{0,1\}^n \to \{0,1\}^n$ *be a true random permutation used to construct an encryption algorithm,* $\mathcal{E}$*, according to Figure 5.2 . Let* KMAC *be used to compose the* E&M *system of Section 5.2 with* $\mathcal{E}$ *as the underlying encryption algorithm. Let* $\mathcal{A}$ *be a nonce-respecting adversary making* q *signing queries before attempting its forgery. Then,* $\mathcal{A}$*'s advantage of successful forgery is at most*

$$\mathsf{Adv}^{\mathrm{auth}}_{\mathrm{KMAC}}(\mathcal{A}) \leq 2^{1-n}. \tag{5.14}$$

It is standard to pass a complexity-theoretic analog of Theorem 5, but in doing this one will need access to a BC^{-1} oracle in order to verify a forgery attempt, which translates into needing the strong pseudorandom permutation assumption. One gets the following. Fix a block cipher $\mathsf{BC} : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ that is used to construct the mode of encryption of Figure 5.2. Let $\mathcal{A}$ be a nonce-respecting adversary that asks q signing queries totaling at most λ bits of payload before attempting its forgery. Then, there is an adversary $\mathcal{B}$ attacking the sprp-security of the block cipher in which

$$\mathsf{Adv}^{\mathrm{auth}}_{\mathrm{KMAC}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathrm{sprp}}_{\mathsf{BC}}(\mathcal{B}) + 2^{1-n}, \tag{5.15}$$

where $\mathsf{Adv}^{\mathrm{sprp}}_{\mathsf{BC}}(\mathcal{B})$ is as defined in equation (2.3). Furthermore, adversary $\mathcal{B}$ takes the same time adversary $\mathcal{A}$ takes, minus the time of generating the coin tosses and the generation and authentication of tags, and makes at most $2\lceil\lambda/n\rceil + q + 1$ oracle queries.

Proof: [of Theorem 5] When $q = 0$ it is rather straightforward. It follows directly from the fact that each value of the authentication tag is equally probable (see the proof of Lemma 5).

Now, assume $\mathcal{A}$ has made q signing queries and recorded the sequence

$$\mathsf{Seq} = \Big\{(M_1, N_1, c_1, \tau_1), \cdots, (M_q, N_q, c_q, \tau_q)\Big\}, \tag{5.16}$$

where M_i, N_i, c_i, τ_i are the message, nonce, ciphertext, and tag corresponding to the i^{th} signing query, respectively. $\mathcal{A}$ then calls the verify oracle with (N, c, τ), where $(N, c, \tau) \neq$

(N_i, c_i, τ_i) for any $i = 1, \cdots, q$ since otherwise $\mathcal{A}$ does not win by definition. We aim to bound the probability that (N, c, τ) will be validated. Let M be the plaintext message corresponding to the decryption of c, the ciphertext in the forgery attempt. There are two possible strategies for forgery:

1. attempt to forge a valid ciphertext-tag pair corresponding to a specific plaintext-nonce pair of $\mathcal{A}$'s choice,

2. attempt to authenticate a ciphertext-tag pair regardless of the corresponding plaintext (i.e., modify a recorded ciphertext-tag pair in a way undetected by the legitimate receiver).

Call the former $\mathsf{forgery}_1$ and the latter $\mathsf{forgery}_2$.

To bound the probability of $\mathsf{forgery}_1$, let $\mathcal{A}$ attempt to falsely authenticate a message-nonce, (M, N) pair of its choice. The first ciphertext block plays a pivotal role in this scenario. First, observe that, for a nonce-respecting adversary, the use of the counter prevents the concatenation of the nonce-counter of the first block of the mode of encryption of Figure 5.2 to be the same as the concatenation of a nonce-counter of any other block in any signing query. Therefore, if $c[1]$, the first ciphertext block of the attempted forgery, is not equal to any of the $c_i[1]$'s, the first ciphertext blocks of the c_i's observed in the recorded sequence of equation (5.16), the resulting coin toss, r, will be a random element of $\mathbb{Z}_{2^n}$. Therefore, by Lemma 1, the resulting tag is uniformly distributed over $\mathbb{Z}_{2^n}$ and, hence, the adversary's advantage of successful forgery is 2^{-n}.

Assume now that $c[1]$ is equal to $c_i[1]$ for an $i \in \{1, \cdots, q\}$. There are two possible scenarios here: either $N = N_i$ or $N \neq N_i$. Let $N = N_i$. Then, $r = r_i$, where r represents the coin tosses of the attempted forgery and r_i represents the coin tosses of the i^{th} signing query. Therefore, if $M = M_i$ only (c_i, τ_i) will be a valid ciphertext-tag pair and the adversary does not win by definition. Assume now that $M \neq M_i$ and let $M[d]$ be a plaintext block in which $M[d] \neq M_i[d]$. Then, the adversary must predict the correct value of $c[d]$, the d^{th} ciphertext block of c, corresponding to $M[d]$. Therefore, when $(c[1], N) = (c_i[1], N_i)$ for some $i \in \{1, \cdots, q\}$, the adversary's advantage of successful forgery is bounded by 2^{-n}.

Finally, assume that $c[1]$ is equal to $c_i[1]$ for an $i \in \{1, \cdots, q\}$ but $N \neq N_i$. Then, r is uniformly distributed over $\mathbb{Z}_{2^n}$ and, similar to the case in which $c[1] \neq c_i[1]$ for any $i \in \{1, \cdots, q\}$, the adversary's advantage is 2^{-n}. Consequently, the probability of successful $\mathsf{forgery}_1$ is at most 2^{-n}.

To bound the probability of $\mathsf{forgery}_2$, denote by M the concatenation of the coin tosses, r, and the plaintext message (i.e., r becomes the first block of the plaintext message). Denote by $\mathsf{Collision}$ the event that $\sum_{j=1}^{k_1} M[j] \equiv \sum_{j=1}^{k_2} M_i[j] \pmod{2^n}$ for an $i \in \{1, \cdots, q\}$, where $k_1 = \lceil \frac{|M|}{n} \rceil$, $k_2 = \lceil \frac{|M_i|}{n} \rceil$, and $M_i[j]$ denotes the j^{th} block of the i^{th} message. That is, M collides with M_i, one of the recorded messages in the sequence of equation (5.16). Also, we use $\overline{\mathsf{Collision}}$ as the typical notation for the complement of $\mathsf{Collision}$. (Note that collision here refers to authentication tags and not ciphertexts; since $\mathcal{E}$ is a permutation, no distinct plaintext messages will have the same ciphertext.)

Obviously, there are two possible scenarios here: either M will collide with one of the M_i's or it will not. Assume that M collides with M_i for an $i \in \{1, \cdots, q\}$. Then, $(N_i, c, \tau_i) \neq (N_i, c_i, \tau_i)$ will pass the integrity check. Let $c_i[d]$ be a ciphertext block in which c and c_i differ. Recall, however, that $\mathsf{Perm}(n)$ is a true random permutation; hence, $M[d]$, the block of M corresponding to $c[d]$, cannot be correlated to $M_i[d]$. That is, from the adversary's standpoint, $M[d]$ is a random element of $\mathbb{Z}_{2^n}$. Therefore, if only one ciphertext block is modified, the probability that M, the plaintext corresponding to c (the modified version of c_i), will collide with M_i is zero (since $\mathsf{Perm}(n)$ is a permutation). Furthermore, if more than one ciphertext blocks are modified the probability of collision is

$$\Pr\left[\mathsf{Collision}\right] = \Pr\Big[\sum_{j=1}^{k_1} M[j] = \sum_{j=1}^{k_2} M_i[j]\Big] = 2^{-n}. \tag{5.17}$$

Assume now that M does not collide with any of the M_i's. If no collision has occurred, then the adversary's probability of successful forgery is bounded by the probability of predicting the plaintext message corresponding to c and modify the tag accordingly. That is, similar to the probability of $\mathsf{forgery}_1$,

$$\Pr\left[\mathsf{forgery}_2 | \overline{\mathsf{Collision}}\right] = 2^{-n}. \tag{5.18}$$

By equations (5.17) and (5.18), the probability of $\mathsf{forgery}_2$ can be boun-ded by

$$\Pr\left[\mathsf{forgery}_2\right] = \Pr\left[\mathsf{forgery}_2|\mathsf{Collision}\right] \cdot \Pr\left[\mathsf{Collision}\right] + \Pr\left[\mathsf{forgery}_2|\overline{\mathsf{Collision}}\right] \cdot \Pr\left[\overline{\mathsf{Collision}}\right] \tag{5.19}$$

$$\leq \Pr\left[\mathsf{Collision}\right] + \Pr\left[\mathsf{forgery}_2|\overline{\mathsf{Collision}}\right] \tag{5.20}$$

$$= 2^{-n} + 2^{-n}. \tag{5.21}$$

Hence, $\max\left\{\Pr\left[\mathsf{forgery}_1\right], \Pr\left[\mathsf{forgery}_2\right]\right\} = 2^{1-n}$, is $\mathcal{A}$'s maximum advantage of successful forgery, and the theorem follows. ■

5.3.3 Privacy of the Construction

There are two pieces of information sent to the intended receiver, the tag and the ciphertext. Since both are functions of the plaintext message, we must show that neither one of them reveals secret information about the confidential message. We start by giving information-theoretic analysis of the privacy of the scheme given the observation of authentication tags.

Theorem 6. *Assume that the coin tosses of $\mathcal{S}_{\mathcal{E}}$, the r's, are sha-red secrets (e.g., delivered out of band). Then, when KMAC is used to construct an* E&M *composition as in Section 5.2, no information about plaintext messages can be reveled by the authentication tags.*

Proof: By Lemma 5, each tag is independent of its corresponding message. Therefore, by only observing a single authentication tag, the adversary cannot reveal any information about the encrypted message. Assume now the adversary has observed the sequence $\mathsf{Seq} = \{\tau_1, \cdots, \tau_q\}$ of authentication tags. By Lemma 5, different authentication tags are mutually independent. Therefore, the observation of multiple tags gives the adversary no extra information than what a single tag gives individually, and the theorem follows. ■

Recall that the coin tosses, the r's, are delivered to the intended receiver by encrypting them with the underlying encryption algorithm. Theorem 6 implies that as long as the adversary cannot extract secret information about the r's from observed ciphertexts, authentication tags do not reveal any private information about encrypted messages. In other words, the only way to attack the privacy of the composition is by attacking the security of the underlying encryption algorithm. To complete the analysis of the privacy of the composition, it remains to prove the privacy of the underlying encryption algorithm. Below, we prove the privacy of the encryption algorithm depicted in Figure 5.2.

Consider an adversary $\mathcal{A}$ who has one of two types of oracles: a real encryption oracle and a fake encryption oracle. The real encryption oracle $\mathcal{E}_K(\cdot,\cdot)$ takes as input a pair (N, M) and returns a ciphertext $c \leftarrow \mathcal{E}_K(N, M)$. Assume that the length of the ciphertext depends only on the length of the plaintext, that is, $|c| = l(|M|)$. The fake encryption oracle, $\$(\cdot,\cdot)$, takes as input a pair (N, M) and returns a random string $c \overset{\$}{\leftarrow} \{0,1\}^{l(|M|)}$. Given adversary $\mathcal{A}$ and the encryption scheme $\mathcal{E}_K$, define

$$\mathsf{Adv}^{\text{priv}}_{\mathcal{E}}(\mathcal{A}) = \Pr\left[K \overset{\$}{\leftarrow} \mathcal{K} : \mathcal{A}^{\mathcal{E}_K(\cdot,\cdot)} = 1\right] - \Pr\left[\mathcal{A}^{\$(\cdot,\cdot)} = 1\right] \tag{5.22}$$

to be $\mathcal{A}$'s advantage of breaking the privacy of the encryption algorithm. That is, as in standard practice (see, e.g., [218, 147]), the model of distinguishing the ciphertext from a random string is used to model the privacy of encryption.

Theorem 7. *Let $\mathcal{E}$ be the encryption algorithm of Figure 5.2 and let* BC *be the block cipher used to construct $\mathcal{E}$. Then given a nonce-respecting adversary, $\mathcal{A}$, against $\mathcal{E}$, one can construct an adversary $\mathcal{B}$ against* BC *such that*

$$\mathsf{Adv}^{\text{priv}}_{\mathcal{E}}(\mathcal{A}) \leq \mathsf{Adv}^{\text{prp}}_{\mathsf{BC}}(\mathcal{B}).$$

Furthermore, the experiment for $\mathcal{B}$ takes the same time as the experiment for $\mathcal{A}$ and, if $\mathcal{A}$ makes at most q_e oracle queries totaling at most μ bits of payload data, then $\mathcal{B}$ makes at most $\lceil \mu/n \rceil + q_e$ oracle queries.

Theorem 7 states that, if the block cipher used to construct the encryption algorithm of Figure 5.2 is a secure pseudorandom permutation, then the resulting encryption algorithm provides data privacy.

Proof: [Theorem 7] Let $\mathcal{B}$ be an adversary against BC that uses adversary $\mathcal{A}$ and that has oracle access to a function $f \overset{\$}{\leftarrow} \mathsf{BC}$. Adversary $\mathcal{B}$ runs $\mathcal{A}$ and replies to $\mathcal{A}$'s encryption oracle queries using its own oracle $f(\cdot,\cdot)$ for the block cipher use in the mode of encryption depicted in Figure 5.2 . Adversary $\mathcal{B}$ returns the same bit that $\mathcal{A}$ returns. Then,

$$\Pr\left[K \overset{\$}{\leftarrow} \mathcal{K} : \mathcal{A}^{\mathcal{E}_K(\cdot,\cdot)} = 1\right] = \Pr\left[f \overset{\$}{\leftarrow} \mathsf{BC} : \mathcal{B}^{f(\cdot,\cdot)} = 1\right], \tag{5.23}$$

since when $\mathcal{B}$ is given a random instance of BC it runs $\mathcal{A}$ exactly as if $\mathcal{A}$ was given the real encryption oracle. Furthermore,

$$\Pr\left[\mathcal{A}^{\$(\cdot,\cdot)} = 1\right] = \Pr\left[\pi \xleftarrow{\$} \mathsf{Perm}(n) : \mathcal{B}^{\pi(\cdot)} = 1\right] \tag{5.24}$$

since $\mathcal{B}$ replies to all of $\mathcal{A}$'s oracle queries with independently selected random strings. Consequently,

$$\begin{aligned} \mathsf{Adv}^{\mathrm{priv}}_{\mathcal{E}}(\mathcal{A}) &= \Pr\left[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathcal{E}_K(\cdot,\cdot)} = 1\right] - \Pr\left[\mathcal{A}^{\$(\cdot,\cdot)} = 1\right] && (5.25)\\ &= \Pr\left[f \xleftarrow{\$} \mathsf{BC} : \mathcal{B}^{f(\cdot,\cdot)} = 1\right] - \Pr\left[\pi \xleftarrow{\$} \mathsf{Perm}(n) : \mathcal{B}^{\pi(\cdot)} = 1\right] = \mathsf{Adv}^{\mathrm{prp}}_{\mathsf{BC}}(\mathcal{B}), && (5.26) \end{aligned}$$

and the theorem follows. ■

5.3.4 Security of the Generic Composition

So far, we have shown that the proposed scheme provides secure authentication as long as the used block cipher is a strong pseudorandom permutation. What we will discuss now is the security of the overall composition. As mentioned earlier, it was shown in [34, 150] that only the Encrypt-then-MAC (EtM) composition guarantees a secure authenticated encryption. This, however, holds for the most general case. That is, as mentioned in [34], many E&M compositions can construct secure authenticated encryption systems. In fact, a secure E&M composition has been proposed by Bellare et al. in [32]. When it comes to MAC-then-Encrypt (MtE) compositions, Maurer and Tackmann recently showed that they can also be used to design secure authenticated encryption schemes [175]. The analysis provided Sections 5.3.2 and 5.3.3 implies that our construction belongs to the class of secure E&M compositions.

5.4 Performance and Security Discussions

5.4.1 Performance

First, we compare the performance of the proposed scheme with generic authenticated encryption compositions. Then, we compare the performance of the proposed scheme with dedicated authenticated encryption schemes.

Table 5.1: Performance comparison of the MMH hash family of Halevi and Krawczyk [104], the polynomial-evaluation (POLY) hash family of Bernstein [39], the NH hash family of Black et al. [42], and the proposed KMAC. (All codes are written in the C programming language using a machine with 3.00GHz Intel(R) Xeon(TM) 64-bit CPU running on UNIX operating system.)

	MMH family [104]	POLY family [39]	NH family [42]	KMAC
Collision probability	2^{-30}	2^{-96}	2^{-32}	2^{-63}
Hashed image (bits)	32	128	64	64
Speed (cycles/byte)	1.2	2.4	0.34	0.03

5.4.1.1 Comparison with Generic Authenticated Encryption Schemes

The Carter-Wegman Counter (CWC) mode of authenticated encryption is a prominent instance of generic compositions. CWC was proposed by Kohno et al. in [147] and was later adopted in the NIST standardized Galois/Counter Mode (GCM) [69]. Although we restrict the comparison to the CWC scheme, the key ideas of the comparison can also be applied to any generic composition using a standard MAC for message integrity. In the CWC mode of authenticated encryption, Kohno et al. proposed encrypting the plaintext message using the counter (CTR) mode and then authenticating the ciphertext using a universal hashing based MAC in the Carter-Wegman style. The universal hash family used in [147] is based on Bernstein's hash127 universal hash family [39].

To start, observe that the CWC scheme uses an EtM composition. Therefore, one clear advantage of the proposed scheme is that the encryption and authentication can be performed in parallel, while encryption must be completed before the authentication process can take place in CWC. The major advantage of the proposed scheme, however, is when KMAC is compared to standard universal hash-function families. In what follows, we give a detailed performance comparison between the compression phase of KMAC and the NH family of Black et al. [42], the fastest universal hash-function family reported in the literature of cryptography for software implementations. Comparison with other popular universal hash-function families, including Bernstein's polynomial evaluation based hashing, is summarized in Table 5.1.

As before, let M be a message to be authenticated and write M as a sequence of n-bit strings; i.e., $M = M[1]||\cdots||M[k]$, where $|M[i]| = n$. Similarly, let the key $x = x[1]||\cdots||x[k]$ be the hashing key of the NH family. Let NH_x be a random member of the NH family determined by the key x. Then, the compressed image of M is computed as

$$\begin{aligned}\mathsf{NH}_x(M) = \sum_{i=1}^{k/2} \Big[\Big(x[2i-1] + M[2i-1] \pmod{2^n} \Big) \\ \cdot \Big(x[2i] + M[2i] \pmod{2^n} \Big) \Big] \pmod{2^{2n}}. \quad (5.27)\end{aligned}$$

The probability that any two distinct messages will hash to the same value (i.e., collide) in the NH family is 2^{-n} [42]. On the other hand, the compressed image of M as computed by the proposed KMAC is

$$\text{KMAC}(M) = \sum_{i=1}^{k} M[i] \pmod{2^n}. \quad (5.28)$$

That is, for a message consisting of k blocks of length n-bits, both KMAC and the NH family require k modular additions over the integer ring $\mathbb{Z}_{2^n}$. However, the NH family requires an extra $k/2$ modular multiplications over the larger integer ring $\mathbb{Z}_{2^{2n}}$. Since multiplication is more time consuming than addition, it is obvious that KMAC computations will be much faster than UMAC computations. More specifically, while addition is performed in $O(n)$ time, the fastest integer multiplication algorithms typically require $O(n \ \log n \ \log\log n)$ time [85].[2] Therefore, the proposed KMAC is $O(\log n \ \log\log n)$ faster than the NH family of [42].

The fact that the NH family uses arithmetic over a larger integer ring will have other performance implications. Consider the case in which $n = 64$. When using a 64-bit machine, the 64-bit multiplication of NH must be split over multiple registers, while the 64-bit addition of KMAC can be stored in one register. Furthermore, in standard compilers, there is no integer data type of size 128-bit to accommodate the result of multiplying two 64-bit integers.

[2] A recent FFT-based algorithm reduced the complexity of integer multiplication to $n \ \log n \ 2^{O(\log^* n)}$ on a 2-tape Turing machine [85].

Therefore, to multiply two 64-bit integers, one needs to split each integer into two 32-bit parts and multiply with appropriate shifts. In the comparison of Table 5.1, n is set to 32 when performing NH computations to avoid the above performance degradation, while n is set to 64 for KMAC.

As can be seen in equation (5.27), universal hash families are not cryptographic functions. That is, the observation of multiple message-image pairs can reveal secret key information. Therefore, in any universal hash-function family based MAC, the compressed image must be processed with a cryptographic function to produce authentication tags. In the CWC scheme, after encrypting the message and universally hashing its corresponding ciphertext, one needs another block cipher call to encrypt the hashed image. In summary, in the CWC scheme (and any EtM composition based on universal hashing), the authenticated encryption is performed in three sequential steps: a message encryption, followed by a universal hashing of the resulting ciphertext, followed by one block cipher call to encrypt the hashed image. In the proposed scheme, on the other hand, encryption and authentication can be performed in parallel, and there is no need for the last block cipher call since the coin tosses, r, is used to encrypt the hashed image.

One of the main shortcomings of many universal hash families is the need for substantially long keys. Depending on the design, the length of the hashing key can be in the Kbits range [104]. When hash families with long keys are used, fetching the keys can be a bottleneck [219].[3] Therefore, in addition to its computational advantage, KMAC possess another efficiency advantage over standard universal hash-function families based MACs. Namely, the keyless property.

Note that the hash families in Table 5.1 provide a much stronger security notion than the simple computations of the proposed KMAC: they are universal hash-function families. Recall, however, that this is one of the main contributions of this work. That is, by utilizing the existence of the encryption algorithm, the security requirements on the compression function have been significantly relaxed, without jeopardizing the security of the resulting MAC.

[3] In authenticated encryption compositions, however, one can overcome this bottleneck by sharing the keys between the encryption and the hashing [37, 147].

5.4.1.2 Comparison with Dedicated Authenticated Encryption Schemes

When compared to single-pass authenticated encryption primitives the proposed scheme possesses simplicity and efficiency advantages. The simplicity advantage can be seen in both the description of the scheme and its security proofs. To give efficiency comparisons, we focus on two of the prominent dedicated authenticated encryption schemes, the IAPM of Jutla [134] and the OCB of Rogaway et al. [218].

First, observe that both IAPM and OCB require pre-processing (whitening) plaintext blocks before block cipher encryption. For instance, IAPM requires XORing plaintext blocks with pair-wise differentially-uniform sequences named s_i. Each s_i is generated by performing modular multiplication over the finite field $\mathbb{Z}_p$, similar to the multiplication of the MMH family in Table 5.1 but with larger primes (the author proposed setting $p = 2^{128} - 159$ for 128-bit block ciphers and $p = 2^{64} - 257$ for 64-bit block ciphers). In the OCB mode of operation, each message block, $M[i]$, is XORed with a string the authors denoted as $Z[i]$. The computation of each $Z[i]$ requires the generation of a Gray code γ_i (in which each γ_i and γ_{i+1} have a Hamming distance of one), multiply two polynomials over the field GF(2^n), and then take the reminder after dividing the multiplication result by a fixed irreducible polynomial. In the proposed scheme, plaintext blocks go to the block cipher without any pre-processing. To the best of our knowledge, the proposed scheme is the first scheme that does not require multiplication operations either before block cipher encryption, such as IAPM and OCB, or after block cipher encryption, such as CWC and GCM.

Note further that both the IAPM and OCB require two block cipher calls in addition to the block cipher calls required for encryption. The proposed parallelizable scheme requires only one extra block cipher call. More importantly, in both the IAPM and OCB, the first and last block cipher calls cannot be performed in parallel with other block cipher calls. That is, the first block cipher call must be completed before the parallelizable second through second-to-last block cipher calls which, in turn, must be completed before the last block cipher call. Therefore, even with the ability to perform parallel computing, both the IAPM and OCB effectively require three sequential block cipher calls. In the proposed scheme, all block cipher calls can be computed in parallel, making the total effective computation time a single block cipher call.

In summary, dedicated authenticated encryption schemes must be performed in the following sequential steps:

1. One block cipher call.
2. Parallelizable generation of whitening sequences (modular multiplications for the IAPM and polynomial evaluations for the OCB).
3. Parallelizable block cipher calls to encrypt the whitened plaintext blocks.
4. A check-sum operation.
5. A final block cipher call to encrypt the whitened check-sum.

Note that there are also three extra XOR operations (between Steps 2 and 3, between Steps 3 and 4, and between Steps 4 and 5). These operations are not included since the time to perform them is almost negligible compared to the five steps above. On the other hand, as mentioned earlier, the entire authenticated encryption scheme proposed in this paper can be performed during a single block cipher call.

5.4.2 Security

Compared to standard universal hashing based MACs (including those used to compose an EtM composition), the proposed KMAC has also a security advantage. Recall that, as illustrated in Section 4.6, MACs based on universal hash functions have a key-recovery vulnerability [106]. Obviously, being keyless, KMAC does not have the key-recovery vulnerability of standard universal hash functions based MACs.

5.5 Summary

In this chapter, another instance of $\mathcal{E}$-MACs is introduced. The fact that the ciphertext, as opposed to the plaintext, is transmitted to the intended receiver along with an authentication tag that is a function of the plaintext, not the transmitted ciphertext, is used to propose a new security model to analyze MACs in the E&M or MtE compositions. However, as opposed to assuming the encryption is only IND-CPA secure, as in Chapter 4, it is also assumed to act as a strong pseudorandom permutation (i.e., block cipher based). The security of the underlying encryption algorithm is then utilized to design the first secure keyless MAC (called KMAC)[4] in the cryptographic literature. KMAC is shown to be $O(\log n \ \log\log n)$ faster than the fastest MAC (where n is the block size). Furthermore, KMAC is demonstrated to construct a practical authenticated encryption scheme that is faster than existing ones, including the fastest dedicated primitives.

[4] Although one may argue that the proposed scheme is not keyless, since the encryption algorithm requires a secret key, the choice for the word "keyless" is tied to the fact that the proposed system is viewed as a generic composition. That is, as a part of a generic authenticated encryption composition, the proposed KMAC is indeed keyless.

Chapter 6

UTILIZING IND-CPA SECURITY: PROCESSING SMALL PORTIONS OF AUTHENTICATED MESSAGE

In the previous two chapters, we introduced two methods to design message authentication algorithms that utilize the coupled encryption algorithm to improve the efficiency of authentication. In this chapter, we introduce yet another method to utilize the security of encryption for more efficient MAC designs. The main idea of this chapter is to utilize the IND-CPA security of encryption so that only a small portion of the message needs to be authenticated, without affecting the integrity of the entire message.

6.1 Message Length and MAC Algorithms

In any secure MAC algorithm, every bit of the message to be authenticated must be processed. That is, the authentication tag of a certain message must be a function of all message bits. Intuitively, if the authentication tag is a function of all parts of the message, excluding a small portion, any bit in this excluded portion can be modified without affecting the value of the authentication tag, leading to easily successful forgeries. Consequently, for any secure authentication algorithm, the required resources to authenticate a message will increase as its length increases (resources here can be time, hardware, energy, or any combination of them).

Obviously, one way around this is to hide the specifics on how the tag is being computed (e.g., hiding which part of the message is not being authenticated), which is known in the literature as "security through obscurity". Depending on security through obscurity, however, is highly discouraged in the literature of cryptographic research [138]. This goes back to Kerckhoffs in one of his principles to design military ciphers: "a cryptosystem should be secure even if everything about the system, except the key, is public knowledge [142]". The same principle was reformulated by Shannon (perhaps independently) as: "the enemy knows the system [228]".

In this chapter, we take a fundamentally new approach to improve the computational efficiency of authenticated encryption schemes. Without adopting security through obscurity, we introduce the first MAC in which only a small portion of the message is authenticated, without affecting the integrity of the entire message. We introduce the p-var authenticated encryption. Instead of relying on security by hiding the specifications of a fixed system from adversaries, the system is randomly changing for every message to be authenticated. However, similar to the impracticality of one-time pad cryptosystems, it is impractical to assume legitimate users have agreed on an arbitrary number of different systems, which is where the security of the underlying encryption comes into play. That is, information about the randomly selected system can be delivered to the intended receiver privately via the ciphertext.

In p-var, the system is completely determined given a relatively short prime integer. That is, given a prime integer, p, the underlying algebraic structure under which the authentication tag is computed is the finite integer field $\mathbb{Z}_p$. Similar to the two variants of $\mathcal{E}$-MACs introduced in Chapters 4 and 5, the proposed p-var is provably secure. The privacy as well as the authenticity of the p-var authenticated encryption can be proven assuming only IND-CPA security of the underlying encryption algorithm. Therefore, unlike the KMAC scheme of the Chapter 5, since neither the pseudorandomness nor the strong pseudorandomness permutation assumption is required for security, the p-var scheme can be used with stream ciphers, leading to fast authenticated encryption.

The rest of the chapter is organized as follows. In Section 6.2 we describe the used model. In Section 6.3, we describe the details of the proposed p-var authenticated encryption. In Section 6.4, we state and prove the security theorems of the proposed p-var. In Section 6.5, we summarize the chapter.

6.2 Authenticated Encryption Schemes

A symmetric authenticated encryption scheme $\mathcal{AE} = (\mathcal{K}, \mathcal{SE}, \mathcal{VD})$ consists of three algorithms: the key generation algorithm ($\mathcal{K}$), the signed encryption algorithm ($\mathcal{SE}$), and the verified decryption algorithm ($\mathcal{VD}$). $\mathcal{AE}$ is defined over some key space KeySp and some message space $\mathsf{MsgSp} = \{0,1\}^*$. The randomized key generation algorithm $\mathcal{K}$ returns a key $K \in \mathsf{KeySp}$. The probabilistic signed encryption algorithm $\mathcal{SE}$ takes as input a key $K \in \mathsf{KeySp}$ and a payload message $m \in \mathsf{MsgSp}$, and returns a ciphertext $c \in \{0,1\}^*$. The deterministic verified decryption algorithm $\mathcal{VD}$ takes as input a key $K \in \mathsf{KeySp}$ and a string $c \in \{0,1\}^*$, and outputs a message $m \in \mathsf{MsgSp}$ or the bit 0 for invalid messages. We ask for the basic validity requirement that if $c = \mathcal{SE}(K, m)$ then it must be the case that $\mathcal{VD}(K, c) = m$.

6.2.1 Adversarial Model

We adopt the standard adversarial model used in authenticated encryption schemes appeared in [34]. The adversary is given oracle access to the signed encryption algorithm $\mathcal{SE}(\cdot, m)$. The adversary can call the $\mathcal{SE}$ oracle on plaintext messages of her choice and observe the outputs. After calling the $\mathcal{SE}$ oracle for q times, the adversary attempts a forgery by calling the verified decryption algorithm $\mathcal{VD}(\cdot, c)$ for a ciphertext c that has not been outputted by the signed encryption oracle. Note that the adversary does not see the secret key K nor the coin tosses of $\mathcal{SE}$. If the verified decryption oracle returns the 0 bit for invalid, the adversary is considered unsuccessful; otherwise, the forgery attempt is said to be successful.

Game 2 (forgery game)**.**

1. *The challenger draws a key $K \xleftarrow{\$} \mathcal{K}$ uniformly at random.*

2. *$\mathcal{A}$ calls the signed encryption oracle a polynomial number of times on messages of its choice and records the corresponding ciphertexts.*

3. *$\mathcal{A}$ then calls the verified decryption oracle on a ciphertext c of its choice.*

4. *$\mathcal{A}$ wins the game if $\mathcal{VD}(K, c) \neq 0$ and c has never been outputted by the signed encryption oracle.*

Let $\mathsf{Adv}^{\text{auth}}_{\mathcal{AE}}(\mathcal{A})$ denotes adversary's $\mathcal{A}$ advantage of successful forgery against the authenticated encryption scheme $\mathcal{AE}$. Then, $\mathcal{AE}$ is said to provide message integrity if $\mathsf{Adv}^{\text{auth}}_{\mathcal{AE}}(\mathcal{A}) \leq \mathsf{negl}(\kappa)$, where $\mathsf{negl}(\kappa)$ is a negligible function in the security parameter κ.

The privacy of the authenticated encryption algorithm will be modeled by its indistinguishability under chosen-plaintext attacks (IND-CPA) as defined in Section 2.4.1. Let $\mathsf{Adv}^{\text{ind-cpa}}_{\mathcal{AE}}(\mathcal{A})$ denotes adversary's $\mathcal{A}$ advantage of breaking the IND-CPA security of the signed encryption algorithm $\mathcal{AE}$. Then, $\mathcal{AE}$ is said to be IND-CPA secure if $\mathsf{Adv}^{\text{ind-cpa}}_{\mathcal{AE}}(\mathcal{A}) \leq \frac{1}{2} + \mathsf{negl}(|K|)$, where $\mathsf{negl}(|K|)$ is a negligible function in the security parameter (the length of the secret key).

6.3 The p-var Scheme

The proposed scheme is a secure instance of the MAC-then-Encrypt (MtE) generic composition. The new idea proposed in this chapter is related to the MAC part of the scheme; the encryption part can be any IND-CPA secure one. Therefore, in the rest of the chapter, p-var can refer to either the overall authenticated encryption system or the MAC part of the system, since the distinction is technically irrelevant in our discussion.

6.3.1 The Basic Idea

The main idea of the proposed scheme is to compute authentication tags over different prime fields. To do that, the transmitter generates a new prime integer p for every authenticated encryption operation, compute the authentication tag over the finite integer field $\mathbb{Z}_p$, and privately convey the used prime integer to the intended receiver via the ciphertext. That is, every authentication tag is computed modulo a prime p, for a randomly chosen prime p (thus the name p-var for varying p). Therefore, unlike standard message authentication schemes, security is not only obtained from the secrecy of the key, but also from the secrecy of the prime field under which the tag is computed.

In practical setups, it is impractical to use a different prime filed to compute different authentication tags (similar to the impracticality of managing one-time keys). Since it is an authenticated encryption system, however, the new prime can be delivered to the intended receiver via the ciphertext.

Before we describe the details of the proposed scheme, we give a brief background showing that there is a sufficient number of primes to grant security. That is, correctly guessing a prime chosen randomly from the set of equal length (in bits) primes is a negligible function in the length of the prime.

6.3.2 The Prime Number Theorem

Let n be an integer. The prime counting function $\pi(n)$ counts the number of primes less than n. Gauss first proposed that the prime counting function can be approximated by

$$\pi(n) \approx \frac{n}{\ln n} \tag{6.1}$$

and then later refined it to

$$\pi(n) \approx \mathrm{Li}(n), \tag{6.2}$$

where $\mathrm{Li}(n) = \int_2^n \frac{1}{\ln x}\,\mathrm{d}x$ is the logarithmic integral [109]. The relation in equation (6.2) is known as the prime number theorem; the theorem was independently proven by Hadamard [103] and Poussin [207].

For a large n, Chebyshev showed that $0.89\ \mathrm{Li}(n) < \frac{\pi(n)}{n/\ln n} < 1.11\ \mathrm{Li}(n)$ [71]. Chebyshev also showed that

$$0.922 < \frac{\pi(n)}{n/\ln n} < 1.105 \tag{6.3}$$

and proved that if the limit $\lim_{n\to\infty} \frac{\pi(n)}{n/\ln n}$ exists, then it is equal to 1 [109]. Rosser and Schoenfeld showed that $\pi(n) > \frac{n}{\ln n}$ for all $n \geq 17$ [220].

Consequently, for a large enough positive integer κ, the number of κ-bit primes can be approximated by

$$\frac{2^{\kappa}}{\kappa \ln 2} - \frac{2^{\kappa-1}}{(\kappa-1)\ln 2} \approx \frac{2^{\kappa-1}}{\kappa \ln 2}. \tag{6.4}$$

That is, the number of κ-bit primes is an exponentially increasing function of κ. In other words, if $\mathcal{P}_\kappa$ is the set of all κ-bit prime integers, the probability of guessing a prime number drawn uniformly at random from $\mathcal{P}_\kappa$ is a negligible function in κ. This fact is essential for the security of the proposed scheme.

6.3.3 Detailed Description

Before we describe the details of the scheme, we emphasize that we do not propose a particular mode of authenticated encryption. Rather, we investigate the general idea of utilizing the underlying encryption algorithm to allow performing authentication arithmetic over varying integer fields. Therefore, for the rest of the chapter, we will assume a generic encryption/decryption scheme that takes only a plaintext/ciphertext as an input. That is, other inputs that the encryption/decryption algorithms might take, such as nonces, will be suppressed.

Let κ be the security parameter of message authentication. The p-var authenticated encryption scheme consists of three algorithms: the key generation algorithm $\mathcal{K}$, the signed encryption algorithm $\mathcal{SE}$, and the verified decryption algorithm $\mathcal{VD}$. The key generation algorithm takes no input and returns a pair (k_e, k_a), where k_e is the encryption/decryption key and k_a is the authentication key. The length of the encryption/decryption key kl depends on the used encryption/decryption algorithms while the authentication key is κ-bit long.

On input an arbitrarily long message m, the signed encryption algorithm draws a κ-bit prime uniformly at random from $\mathcal{P}_\kappa$, the set of all κ-bit primes, and computes the authentication tag as follows

$$\tau \equiv m \cdot k_a \pmod{p}. \tag{6.5}$$

For the rest of the chapter, we overload notations such as m, k_a, τ to denote both the binary strings of their respective parameters and their integer representation in a big-endian format; the distinction between the two representations will be omitted as long as it is clear from the context.

The ciphertext is the encryption of the concatenation of the prime integer p, the authentication tag τ computed according to equation (6.5), and the plaintext message m. The three algorithms constituting the p-var authenticated encryption are shown in Figure 6.1. For the rest of the chapter, we write p-$\mathsf{var}[\mathcal{E}, \kappa]$ when the specification of the underlying encryption algorithm $\mathcal{E}$ and the length of the primes κ is relevant to the discussion; otherwise, we simply write p-var.

There is an important implication of equation (6.5) and the fact that the message m is

```
Algorithm $\mathcal{K}$
    $k_e \xleftarrow{\$} \{0,1\}^{\mathsf{kl}}$
    $k_a \xleftarrow{\$} \{0,1\}^{\kappa}$
    return $K = (k_e, k_a)$
```

```
Algorithm $\mathcal{SE}(K, m)$
    $p \xleftarrow{\$} \mathcal{P}_{\kappa}$
    $\tau \leftarrow m \cdot k_a \pmod p$
    $c \leftarrow \mathcal{E}_{k_e}(p, \tau, m)$
    return $c$
```

```
Algorithm $\mathcal{VD}(K, c)$
    $(p, \tau, m) \leftarrow \mathcal{D}_{k_e}(c)$
    if $\mathsf{isprime}(p) = \mathsf{false}$
        return $0$
    $\tau' \leftarrow m \cdot k_a \pmod p$
    if $\tau' \neq \tau$
        return $0$
    return $m$
```

Figure 6.1: The three algorithms constituting the proposed p-var authenticated encryption: the key generating algorithm $\mathcal{K}$, the signed encryption algorithm $\mathcal{SE}$, and the verified decryption algorithm $\mathcal{VD}$.

arbitrarily long while the prime p is of fixed length. Namely, if the modulus p is known, forgery can trivially succeed without any knowledge of k_a. That is, replacing m with any message that is different than m by multiples of p, i.e., $m_\ell = m + \ell p$ for any integer ℓ, will not change the value of the authentication tag. Therefore, it is obvious that p-var cannot be used as a secure, standalone, MAC algorithm. The novelty of the proposed scheme is that, by utilizing the existence of the encryption primitive, a randomly selected prime integer can be delivered privately to the intended receiver for every message to be authenticated. As will be formally proven in Section 6.4, the probability of successful forgery under this strategy is a negligible function in κ, the length of the chosen prime modulus.

6.3.4 Performance Discussion

We reemphasize that the prime p is κ-bit long and is not equal to the message length. Otherwise, the scheme is considered impractical since, depending on the message length, it might be impossible to find a prime of an equal length efficiently. In fact, this is the main reason why universal hash-function families based on integer multiplications, such as the MMH family of Halevi and Krawczyk [104] and the NH family of Black et al. [42], divide the entire message into equal-length blocks and multiply different blocks with independent and random keys. In other words, p-var can be viewed as a single block in a universal hash-function family.

There are two significant properties that standout in the proposed p-var. First, only the residue of the message, modulo p, needs to be authenticated; not the entire message as in any secure MAC in the cryptographic literature, including the most efficient dedicated authenticated encryption primitives. For instance, messages can be in the Mega or Giga bytes of size and only 64 bits of them, representing their residue modulo p (assuming p is a 64-bit prime), need to be authenticated, without affecting the integrity of the entire messages. Second, as mentioned earlier, only IND-CPA security of the underlying encryption algorithm is required. Therefore, the proposed scheme can be used even when the underlying encryption primitive is a stream cipher. This fact is specially important when comparing p-var to dedicated authenticated encryption primitives (recall that all secure dedicated primitives are block cipher based). Consequently, combining p-var with a stream cipher will yield a faster authenticated encryption scheme.

Note also that the communication overhead of p-var is similar to the communication overhead of efficient authenticated encryption primitives. For instance, in the OCB scheme of Rogaway et al. [218], the transmission is of length $|N| + |m| + |\tau|$, where N is a nonce, m is the plaintext, and τ is the authentication tag. In p-var, the transmission is of length $|p|+|m|+|\tau|$, where p is the prime modulus, m is the plaintext, and τ is the authentication tag.

Before we proceed with theorem statements and proofs, we discuss two other issues relevant to the performance of the proposed p-var. The first is the generation of the set of κ-bit prime integers $\mathcal{P}_\kappa$. The set $\mathcal{P}_\kappa$ can be generated offline so that it does not affect the performance of the signed encryption algorithm. The other issue is the check of whether the received modulus is indeed a prime. This check will affect the performance of the verified decryption algorithm as it must be performed online. Since the prime modulus is typically much shorter than the plaintext itself, this validity check can be performed efficiently. In addition, it is not obvious at this point how the removal of such primality testing will affect the security of the scheme. That is, although it was shown in [11] that the probability of successful forgery against authentication based on computations similar to equation (6.5) is proportional to the reciprocal of the smallest prime factor of the used modulus, the result holds only if the key k_a is relatively prime to the modulus. Further analysis can prove or disprove the necessity for testing the primality of the received modulus; such analysis, however, is out of the scope of this paper.

6.4 Theorem Statements and Proofs

In this section, we give a formal security analysis of the proposed message authentication mechanism, prove the confidentiality of the system, and then discuss the security of the composed authenticated encryption system.

6.4.1 Data Authenticity

In this section, we prove the authenticity of the scheme assuming the use of a stream cipher for encryption. Different analysis assuming the use of block ciphers satisfying the PRP or the SPRP notions can be obtained. However, since the literature is rich with a variety of efficient authenticated encryption schemes that are provably secure assuming PRP or SPRP block ciphers, we restrict our analysis to stream ciphers.

Let $p\text{-}\mathsf{var}[\mathcal{E}, \kappa]$ denotes the proposed authenticated encryption composition of Section 6.3 using $\mathcal{E}$ as the underlying encryption algorithm. Let $\mathsf{Adv}^{\mathrm{auth}}_{p\text{-}\mathsf{var}[\mathcal{E},\kappa]}(\mathcal{A})$ denotes adversary's $\mathcal{A}$ advantage of successful forgery against $p\text{-}\mathsf{var}[\mathcal{E}, \kappa]$. We give below an information-theoretic bound on the adversary's advantage of successful forgery assuming the plaintext is encrypted by XORing it with the output of a true random number generator; i.e., the encryption is a one-time pad (OTP) cipher.

Theorem 8. *Let κ be the bit length of the prime moduli that are used to compute authentication tags according to equation (6.5). Let $p\text{-}\mathsf{var}[\mathrm{OTP},\kappa]$ denote the proposed authenticated encryption of Section 6.3 with an information-theoretically secure one-time pad cipher as the underlying encryption algorithm. Let $\mathcal{A}$ be an adversary making q signed encryption queries before attempting its forgery. Then,*

$$\mathsf{Adv}^{\mathrm{auth}}_{p\text{-}\mathsf{var}[\mathrm{OTP},\kappa]}(\mathcal{A}) \leq \mathsf{negl}(\kappa).$$

It is standard to pass a complexity-theoretic analog of Theorem 8. Let $\mathcal{E}$ be an IND-CPA secure encryption algorithm that XORs plaintexts with a pseudorandom bit stream (e.g., a stream cipher or a block cipher based encryption using the counter (CTR) mode of operation). Given adversary $\mathcal{A}$ against the authenticity of $p\text{-}\mathsf{var}[\mathcal{E}, \kappa]$, one can construct an adversary $\mathcal{B}$ against $\mathcal{E}$ so that

$$\mathsf{Adv}^{\text{auth}}_{p\text{-}\mathsf{var}[\mathcal{E},\kappa]}(\mathcal{A}) \leq \mathsf{Adv}^{\text{ind-cpa}}_{\mathcal{E}}(\mathcal{B}) + \mathsf{negl}(\kappa). \tag{6.6}$$

Furthermore, if $\mathcal{A}$ makes q queries totaling μ bits of payload, then $\mathcal{B}$ makes q queries totaling $\mu + 2\kappa q$ bits of payload.

Equation (6.6) states that the adversary's advantage of successful forgery is provably secure provided the underlying encryption algorithm is IND-CPA secure.

Proof: [Theorem 8] Assume an adversary calling the signing oracle for q times and recording the sequence

$$\mathsf{Seq} = \Big\{(m_1, c_1), \cdots, (m_q, c_q)\Big\} \tag{6.7}$$

of message-ciphertext pairs. We aim to bound the probability that a ciphertext c of the adversary's choice will be accepted as valid, where $c \neq c_i$ for any $i \in \{1, \cdots, q\}$, since otherwise the adversary does not win by definition.

Let $c = c_i + \epsilon_c$ for any $i \in \{1, \cdots, q\}$, where ϵ_c can be any value of the adversary's choice. Since the ciphertext is an XOR of the plaintext with a random string, this implies that $m = m_i + \epsilon_m$, where m is the plaintext corresponding to c in the forgery attempt and ϵ_m can be completely determined given m_i and ϵ_c. Similarly, let the prime modulus and the tag extracted from c be $p = p_i + \epsilon_p$ and $\tau = \tau_i + \epsilon_\tau$, respectively. If p fails the primality test, the forgery attempt is unsuccessful. Assume p is indeed a prime. Then, the integrity check becomes

$$\begin{aligned} \tau' &\equiv (m_i + \epsilon_m)k_a \pmod{p} && (6.8)\\ &\equiv m_i k_a + \epsilon_m k_a \pmod{p} && (6.9)\\ &\overset{?}{\equiv} \tau \pmod{p}, && (6.10) \end{aligned}$$

where τ' is the tag computed at the receiver end.

The key idea is that, by Lemma 1, the value of $\epsilon_m k_a \pmod{p}$ is uniformly distributed over $\mathbb{Z}_p^*$ for any prime p (unless ϵ_m or k_a happen to be equal to p, which will occur with a negligible probability). Therefore, the probability of satisfying equation (6.10) for a successful forgery is at most $1/(p-1) \leq 2^{1-\kappa}$, a negligible function in κ.

In a different direction, for any prime modulus p and a valid message-tag pair (m, τ) computed according to equation (6.5), the pair $(m_\ell = m + \ell \cdot p, \tau)$ for any integer ℓ is a valid message-tag pair. Since the ciphertext is an XOR of the plaintext with a random string, the adversary can also determine c_ℓ, the ciphertext corresponding to to m_ℓ. Hence, the ciphertext c_ℓ can result in a successful forgery. However, since the prime modulus is encrypted with a random one-time string, the privacy of the used prime modulus is information-theoretically secured. Further, by the prime number theorem, the probability of guessing a randomly chosen κ-bit prime in a negligible function in κ, and the theorem follows. ■

Remark 9. *To see how equation (6.6) follows from Theorem 8, observe that, in the proof of Theorem 8, the only use of the information-theoretic security of the underlying OTP encryption algorithm is to imply that no information about the used prime modulus can be exposed by the ciphertext. Therefore, the difference between using an information-theoretically secure OTP encryption and an IND-CPA secure stream cipher is that the adversary in the latter case has a negligible advantage of exposing the value of the prime modulus by breaking the IND-CPA security of encryption. The rest is just a standard complexity reduction.*

6.4.2 Data Privacy

We show in this section that the privacy of the proposed scheme is provably secure assuming the used encryption algorithm provides indistinguishability under chosen plaintext attacks (IND-CPA). We model the privacy of the system as its indistinguishability under chosen plaintext attacks. Let $p\text{-}\mathsf{var}[\mathcal{E}, \cdot]$ denote the proposed authenticated encryption scheme described in Section 6.3 using $\mathcal{E}$ as the underlying encryption algorithm. Let $\mathcal{A}$ be an adversary against the privacy of $p\text{-}\mathsf{var}[\mathcal{E}, \cdot]$ and let $\mathsf{Adv}^{\mathrm{priv}}_{p\text{-}\mathsf{var}[\mathcal{E},\cdot]}(\mathcal{A})$ denote adversary's $\mathcal{A}$ advantage in breaking the privacy of the system. One gets the following theorem.

Theorem 9. *Let $p\text{-}\mathsf{var}[\mathcal{E},\kappa]$ be the authenticated encryption composition described in Section 6.3 using $\mathcal{E}$ as the underlying encryption algorithm. Then given any adversary $\mathcal{A}$ against the privacy of $p\text{-}\mathsf{var}[\mathcal{E},\kappa]$, one can construct an adversary $\mathcal{B}$ against $\mathcal{E}$ such that*

$$\mathsf{Adv}^{\mathrm{priv}}_{p\text{-}\mathsf{var}[\mathcal{E},\kappa]}(\mathcal{A}) \leq \mathsf{Adv}^{\mathrm{ind\text{-}cpa}}_{\mathcal{E}}(\mathcal{B}).$$

Furthermore, $\mathcal{B}$ uses the same resources as $\mathcal{A}$ except that each encryption query of $\mathcal{B}$ is 2κ bits longer than that of $\mathcal{A}$, where κ is the length of the prime modulus used to compute the tag.

Proof: Let $p\text{-}\mathsf{var}[\mathcal{E},\kappa]$ be the scheme of Section 6.3 using $\mathcal{E}$ as the underlying encryption algorithm and let $\mathcal{A}$ be an adversary against the privacy of $p\text{-}\mathsf{var}[\mathcal{E},\kappa]$. Let adversary $\mathcal{B}$ be an adversary against $\mathcal{E}$ that runs $\mathcal{A}$. $\mathcal{B}$ simulates $\mathcal{SE}$ to answer $\mathcal{A}$'s encryption queries and returns the same bit $\mathcal{A}$ returns. Then, it follows that $\mathsf{Adv}^{\mathrm{priv}}_{p\text{-}\mathsf{var}[\mathcal{E},\kappa]}(\mathcal{A}) \leq \mathsf{Adv}^{\mathrm{ind\text{-}cpa}}_{\mathcal{E}}(\mathcal{B})$, as required. ■

Theorem 9 states that an adversary breaking the privacy of the proposed system will also be able to break the IND-CPA of the underlying encryption algorithm. That is, the privacy of the proposed technique is provably secure given the IND-CPA security of the underlying encryption algorithm, as desired. This result is not surprising since the system can be viewed as a MAC-then-Encrypt authenticated encryption in which Bellare and Namprempre showed that if the encryption is IND-CPA secure, then so is the composed authenticated encryption [34].

6.4.3 Security of p-var as a Generic Authenticated Encryption Composition

In [34], Bellare and Namprempre defined two notions of integrity for generic authenticated encryption compositions: integrity of plaintext (INT-PTXT) and integrity of ciphertext (INT-CTXT). Combined with encryption algorithms that provide indistinguishability under chosen-plaintext attacks (IND-CPA), the security of different methods for constructing generic compositions is analyzed.

One result of [34] is that, in general, MtE compositions do not provide INT-CTXT. However, the authors also acknowledged that the notion of INT-PTXT is the more natural requirement, while the main purpose of introducing the stronger notion of INT-CTXT is for the security relations derived in [34]. It should be noted, however, that a sufficient condition for the proposed scheme to provide INT-CTXT is to use a one-to-one encryption algorithm. To see this, observe that, by the one-to-one property, any modification of the ciphertext will correspond to changing its corresponding plaintext and, by Theorem 8, a modified plaintext will go undetected with a negligible probability.

Another result of [34] is that MtE compositions do not generally provide indistinguishability under chosen-ciphertext attacks (IND-CCA) nor non-malleability under chosen-plaintext attacks (NM-CPA). As can be seen in the proofs of [34], however, this is true due to the fact that the underlying encryption algorithm may not provide such security notions. That is, the results of [34] hold only if the encryption algorithm is not IND-CCA nor NM-CPA secure. In fact, Maurer and Tackmann recently proved the soundness of MtE compositions for all encryption schemes with suitably restricted malleability, including stream ciphers [175].

6.5 Summary

In this chapter, we proposed the p-var authenticated encryption scheme. The authentication tag in p-var is computed using modular multiplication. However, as opposed to relying solely on the secret key to grant message integrity, we utilize the existence of the encryption algorithm to change the prime field under which the computation is performed. By doing so, we show how to construct the first authentication mechanism in which, regardless of the length of the message to be authenticated, only the residue of the message (in its integer representation) modulo the used prime needs to be authenticated, while maintaining the integrity of the entire message. Furthermore, unlike efficient dedicated authenticated encryption primitives, neither the pseudorandom permutation (PRP) nor the strong pseudorandom permutation (SPRP) are needed to prove the integrity of the scheme; only indistinguishability under chosen-plaintext attacks (IND-CPA) is needed. Therefore, the proposed scheme can be used to construct secure authenticated encryption systems even if the underlying encryption primitive is a stream cipher, leading to faster designs.

Chapter 7

SECURITY OF AUTHENTICATION BASED ON UNIVERSAL HASHING

Unlike the previous three chapters, the main focus of this chapter is analyzing the security of MACs based on universal hash-function families, no new scheme is proposed.

7.1 The Power of Primes

The security of many universal hash-function families rely on the fact that computations are performed over finite fields (see, e.g., [88, 261, 234, 62, 110, 149, 104, 73, 7, 17, 16]). In this chapter, we investigate a universal hash-function family that belongs to this class of universal hash families. Unlike previous analysis, however, we will consider the effect of performing operations over finite integer rings, as opposed to fields, on the security of authentication codes based on this universal hash family.

To give an example of the universal hash family under study, let a message m be divided into equal-length blocks $m_i \in \mathbb{Z}_p$, where p is a pre-specified prime integer. Given the secret hashing keys $k_i \in \mathbb{Z}_p$, compute the hashed image of the message m as $h(m) = \sum_i k_i m_i \pmod{p}$. Then, the authentication tag of m is simply an encryption of its hashed image. There have been multiple proposals in the literature of message authentication that were based on variants of this approach (see, e.g., [234, 104, 73, 7, 17]). When the multiplication is performed modulo a prime integer, it has been proven that such proposals provide message integrity. However, the effect of using non-prime moduli on the security of such proposals has not been previously investigated.

In this chapter, we investigate the relation between the underlying integer ring and the security of authentication based on the class of universal hash families described above. We derive tight bounds on the probabilities of successful forgeries for all choices of finite integer rings $\mathbb{Z}_n$. We show the direct relation between the prime factorization of the modulus n and the security of authentication. More precisely, we prove that the probability of successful forgery is proportional to the reciprocal of the smallest prime factor of the modulus n.

Since the derivation of the main result is quite lengthy, we attempt to clarify it by breaking the proof into a series of lemmas (Lemma 11 - Lemma 15) leading to the final theorem. One particular result that is generally interesting (not only for this chapter) is the result of Lemma 6. In Lemma 6 we prove what can be viewed as an extension to Bézout's lemma for finite integer rings. It is a well-known fact in algebra and number theory that, for two integers a and n, if $\gcd(a, n) = d$, there exists an integer x such that $x \cdot a \equiv d \pmod{n}$. What we show in Lemma 6 is that for an $a \in \mathbb{Z}_n \backslash \{0\}$ such that $\gcd(a, n) = d$, not only there exists an element $x \in \mathbb{Z}_n$ such that $x \cdot a \equiv d \pmod{n}$ but, further, there exists an *invertible element* $x \in \mathbb{Z}_n^*$ such that $x \cdot a \equiv d \pmod{n}$. This result is essential to generalize our bounds to any finite integer ring and, to the best of our knowledge, has not appeared in the literature of mathematics.

The rest of the chapter is organized as follows. In Section 7.2, we formally state and prove our extension to Bézout's lemma along with some basic properties of the finite integer ring $\mathbb{Z}_n$. Section 7.3 gives two examples of the use of the studied universal hash-function family for the construction of computationally secure MACs and the construction of codes with secrecy. Section 7.4 is devoted to the security analysis. Section 7.5 provides a summary of the choices of moduli and their security ramifications. In Section 7.6 we summarize the chapter.

7.2 Preliminaries

For any nonzero integers a and n with $\gcd(a, n) = d$, by Bézout's lemma [245], there exist two integers x and y so that $ax + ny = d$. Otherwise stated, for any nonzero integers a and n with $\gcd(a, n) = d$, by Bézout's lemma, there exists an integer x so that

$$ax \equiv d \pmod{n}. \tag{7.1}$$

It is further known that the x satisfying equation (7.1) is not necessarily unique. In particular, for a nonzero $a \in \mathbb{Z}_n$, there are $d = \gcd(a, n)$ distinct elements in $\mathbb{Z}_n$ satisfying equation

(7.1), given by

$$\Big\{x_0,\ x_0+\frac{n}{d},\ x_0+2\frac{n}{d},\ \cdots,\ x_0+(d-1)\frac{n}{d}\Big\}, \tag{7.2}$$

where x_0 is the smallest integer in $\mathbb{Z}_n$ satisfying equation (7.1) [245]. The significance of the following lemma is the statement that at least one of the d elements of the set in equation (7.2) *must be* invertible in $\mathbb{Z}_n$. This result is essential for the proofs of this chapter and, to the best of our knowledge, has not appeared in the literature.

Lemma 6. *In any finite integer ring $\mathbb{Z}_n$, for any $\delta \in \mathbb{Z}_n \backslash \{0\}$, if $\gcd(\delta, n) = d$, then there exists an invertible element $\alpha \in \mathbb{Z}_n^*$ such that $\alpha \times \delta \equiv d \pmod{n}$.*

Proof: Let $\gcd(\delta, n) = d$, then by Bézout's lemma [245], there exists an integer α_0 such that

$$\alpha_0 \times \delta \equiv d \pmod{n}. \tag{7.3}$$

Further, all integers in the infinite set

$$A = \Big\{\alpha_k | \alpha_k = \alpha_0 + k\frac{n}{d}, \quad \forall\ k \in \mathbb{Z}\Big\} \tag{7.4}$$

are valid solutions to equation (7.3) [245]. The lemma states that, not only there exists an integer that satisfies equation (7.3), but there exists an *invertible* element in $\mathbb{Z}_n$ that satisfies equation (7.3). We will prove the lemma by finding an integer k such that $\alpha_k \in A$ is relatively prime to n.

If $\gcd(\delta, n) = 1$ then $\alpha_0 = \delta^{-1} \in \mathbb{Z}_n^*$ does exist and is the invertible solution to equation (7.3). Assume, however, that $\gcd(\delta, n) = d > 1$ and write n in its prime factorization as

$$n = \prod_{i=1}^{\ell_1} p_i^{e_i} \prod_{i=1}^{\ell_2} \gamma_i^{e_{\gamma_i}} \prod_{i=1}^{\ell_3} \zeta_i^{e_{\zeta_i}}. \tag{7.5}$$

Assume further that δ can be written in its prime factorization form as

$$\delta = \prod_{i=1}^{\ell_1} p_i^{e'_i} \prod_{i=1}^{\ell_2} \gamma_i^{e'_{\gamma_i}} \prod_{i=1}^{\ell_4} r_i^{e_{r_i}}, \tag{7.6}$$

where $e'_i \geq e_i,\ \forall\ i = 1, \cdots, \ell_1$, and $e'_{\gamma_i} < e_{\gamma_i},\ \forall\ i = 1, \cdots, \ell_2$, with the ζ_i's and r_i's being distinct primes. Then, $d = \prod_{i=1}^{\ell_1} p_i^{e_i} \prod_{i=1}^{\ell_2} \gamma_i^{e'_{\gamma_i}}$ and, by Bézout's lemma, there exists an α_0 such that

$$\alpha_0 \times \delta \equiv d \pmod{n}. \tag{7.7}$$

Which is equivalent to

$$\alpha_0 \times \prod_{i=1}^{\ell_1} p_i^{e_i'-e_i} \prod_{i=1}^{\ell_4} r_i^{e_{r_i}} \equiv 1 \pmod{\prod_{i=1}^{\ell_2} \gamma_i^{e_{\gamma_i}-e'_{\gamma_i}} \prod_{i=1}^{\ell_3} \zeta_i^{e_{\zeta_i}}}. \tag{7.8}$$

Equation (7.8) implies that α_0 is relatively prime to $\prod_{i=1}^{\ell_2} \gamma_i^{e_{\gamma_i}-e'_{\gamma_i}} \prod_{i=1}^{\ell_3} \zeta_i^{e_{\zeta_i}}$, which implies that none of the γ_i's nor the ζ_i's divides α_0. Furthermore, by equation (7.4), none of the γ_i's nor the ζ_i's will divide α_k for any $k \in \mathbb{Z}$. Therefore, to prove that an $\alpha_k \in A$ is relatively prime to n, since the prime factorization of n consists only of p_i's, γ_i's, and ζ_i's, it suffices to show that none of the p_i's divides α_k.

Define $\prod_{i=1}^{\ell_2+\ell_3} q_i^{e_{q_i}} := \prod_{i=1}^{\ell_2} \gamma_i^{e_{\gamma_i}-e'_{\gamma_i}} \prod_{i=1}^{\ell_3} \zeta_i^{e_{\zeta_i}}$, where $q_i = \gamma_i, e_{q_i} = e_{\gamma_i} - e'_{\gamma_i}$, for $i = 1, \cdots, \ell_2$ and $q_{\ell_2+i} = \zeta_i, e_{q_{\ell_2+i}} = e_{\zeta_i}$, for $i = 1, \cdots, \ell_3$. Then, equation (7.8) can be rewritten as

$$\alpha_0 \times \prod_{i=1}^{\ell_1} p_i^{e_i'-e_i} \prod_{i=1}^{\ell_4} r_i^{e_{r_i}} \equiv 1 \pmod{\prod_{i=1}^{\ell_2+\ell_3} q_i^{e_{q_i}}}, \tag{7.9}$$

where none of the q_i's divides any α_k for any $k \in \mathbb{Z}$.

Now, if none of the p_i's divides α_0 then $\gcd(\alpha_0, n) = 1$ and we are done. Assume, however, that some of the p_i's, for $i = 1, \cdots, \ell_1$ divide α_0, and let p_1 be one such prime dividing α_0. Then α_0 can be written as $\alpha_0 = m_1 p_1$, where m_1 is relatively prime to all q_i's (since, otherwise, some of the q_i's will divide α_0). Then, from equation (7.4), we know that

$$\alpha_1 = \alpha_0 + \frac{n}{d} = m_1 p_1 + \prod_{i=1}^{\ell_2+\ell_3} q_i^{e_{q_i}} \tag{7.10}$$

also satisfies equation (7.3). Therefore, $p_1 \nmid \alpha_1$ since it does not divide $\prod_{i=1}^{\ell_2+\ell_3} q_i^{e_{q_i}}$ (also none of the q_i's divides α_1 since none of them divides $m_1 p_1$).

Assume, however, that some of the other p_i's divide α_1, and let p_2 be such a prime. Then α_1 can be written as $\alpha_1 = m_2 p_2$ for some m_2 relatively prime to p_1 and all the q_i's. Then, by equation (7.4),

$$\alpha_2 \overset{(b)}{=} m_2 p_2 + \prod_{i=1}^{\ell_2+\ell_3} q_i^{e_{q_i}} \overset{(a)}{=} m_1 p_1 + 2 \prod_{i=1}^{\ell_2+\ell_3} q_i^{e_{q_i}} \tag{7.11}$$

also satisfies equation (7.3). Therefore, by equality (b), $p_2 \nmid \alpha_2$ since it does not divide $\prod_{i=1}^{\ell_2+\ell_3} q_i^{e_{q_i}}$ and, by equality (a), $p_1 \mid \alpha_2$ iff $p_1 = 2$. Assume that $p_1 = 2$ and write $\alpha_2 = m_3 p_1$

for an m_3 that is relatively prime to p_2 and the q_i's, then

$$\alpha_3 \overset{\text{(b)}}{=} m_3 p_1 + \prod_{i=1}^{\ell_2+\ell_3} q_i^{e_{q_i}} \overset{\text{(a)}}{=} m_2 p_2 + 2 \prod_{i=1}^{\ell_2+\ell_3} q_i^{e_{q_i}} = m_1 p_1 + 3 \prod_{i=1}^{\ell_2+\ell_3} q_i^{e_{q_i}}. \tag{7.12}$$

Thus, since $p_2 \neq 2$, $p_1 \nmid \alpha_3$ and $p_2 \nmid \alpha_3$ by equalities (b) and (a) respectively, and $q_i \nmid \alpha_3 \; \forall i$ by construction.

Assume now that there exists an α_k such that $p_i \nmid \alpha_k \forall \; i = 1, \cdots, \ell_1 - 1$ and $q_i \nmid \alpha_k \forall i$, but $p_{\ell_1} \mid \alpha_k$. Then write $\alpha_k = m_k p_{\ell_1}$ for some m_k relatively prime to all q_i's and all p_i's except possibly p_{ℓ_1}. Then α_{k+1} can be expressed as

$$\alpha_{k+1} = m_k p_{\ell_1} + \prod_{i=1}^{\ell_2+\ell_3} q_i^{e_{q_i}} = \cdots \overset{\text{(b)}}{=} m_2 p_2 + c_2 \prod_{i=1}^{\ell_2+\ell_3} q_i^{e_{q_i}} \overset{\text{(a)}}{=} m_1 p_1 + c_1 \prod_{i=1}^{\ell_2+\ell_3} q_i^{e_{q_i}}, \tag{7.13}$$

for some constants $c_i \in \mathbb{N}$. Recall that all the m_i's are relatively prime to the q_i's by construction. Therefore, to complete the proof, it suffices to show that there exists an integer $h \geq 1$ such that α_{k+h} is not divisible by any p_i. As a function of the p_i's and the c_i's, we conclude the proof by showing how to iteratively find such an h.

Iteration 1. Assume that p_1 divides the α_{k+1} in equation (7.13). This implies, by equality (a), that $p_1 \mid c_1$. However, if $p_1 \mid c_1$ then $p_1 \nmid (c_1 + 1)$, and α_{k+2} can be written as

$$\begin{aligned}\alpha_{k+2} = m_k p_\ell + 2 \prod_{i=1}^{\ell_2+\ell_3} q_i^{e_{q_i}} &= \cdots \overset{\text{(c)}}{=} m_3 p_3 + (c_3 + 1) \prod_{i=1}^{\ell_2+\ell_3} q_i^{e_{q_i}} \\ &\overset{\text{(b)}}{=} m_2 p_2 + (c_2 + 1) \prod_{i=1}^{\ell_2+\ell_3} q_i^{e_{q_i}} \overset{\text{(a)}}{=} m_1 p_1 + (c_1 + 1) \prod_{i=1}^{\ell_2+\ell_3} q_i^{e_{q_i}}.\end{aligned} \tag{7.14}$$

Therefore, by equality (a) in equation (7.14), we get $p_1 \nmid \alpha_{k+2}$.

Iteration 2. Now, assume that $p_2 \mid \alpha_{k+2}$. By equality (b) in equation (7.14), this implies that $p_2 \mid (c_2 + 1)$. However, if $p_2 \mid (c_2 + 1)$ then $p_2 \nmid (c_2 + 1 + p_1)$, and α_{k+2+p_1} can be written as

$$\alpha_{k+2+p_1} = m_k p_\ell + (2+p_1)\prod_{i=1}^{\ell_2+\ell_3} q_i^{e_{q_i}} = \cdots \overset{\text{(c)}}{=} m_3 p_3 + (c_3+1+p_1)\prod_{i=1}^{\ell_2+\ell_3} q_i^{e_{q_i}}$$
$$\overset{\text{(b)}}{=} m_2 p_2 + (c_2+1+p_1)\prod_{i=1}^{\ell_2+\ell_3} q_i^{e_{q_i}} \overset{\text{(a)}}{=} m_1 p_1 + (c_1+1+p_1)\prod_{i=1}^{\ell_2+\ell_3} q_i^{e_{q_i}}. \qquad (7.15)$$

Then, by equality (b) in equation (7.15), $p_2 \nmid \alpha_{k+2+p_1}$ and, by equality (a) in equation (7.15), $p_1 \nmid \alpha_{k+2+p_1}$.

Iteration 3. Similarly, if p_3 divides α_{k+2+p_1} in equation (7.15), by equality (c), $p_3 \mid (c_3+1+p_1)$. However, if $p_3 \mid (c_3+1+p_1)$ then $p_3 \nmid (c_3+1+p_1+p_1p_2)$ and, by writing $\alpha_{k+2+p_1+p_1p_2}$ as

$$\begin{aligned}\alpha_{k+2+p_1+p_1p_2} &= m_k p_\ell + (2+p_1+p_1p_2)\prod_i q_i^{e_{q_i}}\\ &= \cdots\\ &\overset{\text{(c)}}{=} m_3p_3 + (c_3+1+p_1+p_1p_2)\prod_i q_i^{e_{q_i}}\\ &\overset{\text{(b)}}{=} m_2p_2 + (c_2+1+p_1+p_1p_2)\prod_i q_i^{e_{q_i}}\\ &\overset{\text{(a)}}{=} m_1p_1 + (c_1+1+p_1+p_1p_2)\prod_i q_i^{e_{q_i}}, \qquad (7.16)\end{aligned}$$

one can see that neither p_3 nor p_2 nor p_1 divides $\alpha_{k+2+p_1+p_1p_2}$ by equalities (c), (b), and (a) in equation (7.16), respectively.

Iteration ℓ_1. After the ℓ_1^{th} iteration, for an h given by:

$$h = 1 + \beta_1 + \beta_2 p_1 + \beta_3 p_1 p_2 + \cdots + \beta_{\ell_1}\prod_{i=1}^{\ell-1} p_i, \qquad (7.17)$$

where $\beta_i = 1$ if in the i^{th} iteration $p_i \mid (m_ip_i + c_i \prod_i q_i^{e_{q_i}})$ and zero otherwise, α_{k+h} will not be divisible by any p_i. Hence, we have found, by construction, an α_{k+h} with $\gcd(\alpha_{k+h}, n) = 1$ that satisfies equation (7.3). The residue of α_{k+h} modulo n is an invertible element of $\mathbb{Z}_n$ that satisfies equations (7.3), and the lemma follows. ■

For any finite integer ring $\mathbb{Z}_n$, $\mathbb{Z}_n \backslash \mathbb{Z}_n^*$, the complement of $\mathbb{Z}_n^*$, will be the set of elements that are not relatively prime to n. The following result holds for the set of integers that are not relatively prime to n.

Lemma 7. *In any finite integer ring $\mathbb{Z}_n$, for any $\alpha \in \mathbb{Z}_n \backslash \mathbb{Z}_n^*$ and any $\beta \in \mathbb{Z}_n$, $\alpha \times \beta \in \mathbb{Z}_n \backslash \mathbb{Z}_n^*$.*

The proof of this lemma can be found in [224].

Lemma 8. *Given an integer $k \in \mathbb{Z}_n^*$, for an r uniformly distributed over $\mathbb{Z}_n^*$, the value δ given by:*

$$\delta \equiv r \times k \pmod{n} \tag{7.18}$$

is uniformly distributed over $\mathbb{Z}_n^*$.

Lemma 8 follows directly from the general result of Lemma 1. The following is also a general result from number theory.

Lemma 9. *For any positive integer n with a prime factor p, $\varphi(n) \geq p - 1$, with equality iff $n = p$.*

This Lemma is a standard result for integers and its proof can be found in most books in number theory (see, e.g., [107]).

7.3 Examples of constructions

In this section, we give two examples of authentication codes based on the universal hash-function family under analysis. The first example is a construction of a computationally secure message authentication code (MAC) algorithm, while the second construction is an example of authentication codes with secrecy.

7.3.1 Constructing computationally secure MACs

In computationally secure MACs, the message to be authenticated is first compressed using a universal hash function and then the compressed image is processed with a cryptographic function (such as one-time pad ciphers, stream ciphers, or pseudorandom function).

Assume the message to be authenticated can be divided into b blocks, that is, $m = (m_1, \cdots, m_b)$, where $m_i \in \mathbb{Z}_p^*$ for $i = 1, \cdots, b$. Let the key of the universal hash function be $k = (k_1, \cdots, k_b)$, where the k_i's are drawn uniformly at random from the multiplicative group $\mathbb{Z}_p^*$. Then, the compressed image of m is computed as

$$h(m) = \sum_{i=1}^{b} k_i \times m_i \pmod{p}. \tag{7.19}$$

Note that the key need not to be as long as the message, otherwise, such constructions will be impractical. That is, there are standard techniques so that the same key can be used to hash messages of arbitrary lengths (see, e.g., [261, 104, 42] for the description of such techniques).

The security of universal hash-function families based MACs depends on the probability of message collision. That is, if two distinct messages m and m' hash to the same image $\Big($i.e., $h(m) = h(m')\Big)$, then they will have the same authentication tag. Consequently, for a message-tag pair, if an adversary can come up with a different message that hashes to the same value, successful forgery can be accomplished with high probabilities. Therefore, the most important security property of universal hash functions is their probabilities of message collisions.

Carter and Wegman suggested the hash function of equation (7.19) with the primes $p = 2^{16} + 1$ or $p = 2^{32} - 1$ [104]. Halevi and Krawczyk later suggested the same equation with any prime $2^{32} < p < 2^{32} + 2^{16}$. They designed their MMH family, one of the fastest universal hash-function families, with $p = 2^{32} + 15$, the smallest prime between 2^{32} and $2^{32} + 2^{16}$ [104]. Etzel et al. proposed a variant of the MMH family of [104] that can be faster in some applications [73].

When the hash function is computed modulo a prime integer, equation (7.19) is known to be $(p-1)^{-1}$-AU. In fact, it is shown to be $(p-1)^{-1}$-AΔU in [104] (the notion of ϵ-AΔU is a stronger notion than ϵ-AU; interested readers may refer to [104] for the precise definition of ϵ-AΔU hash families).

The security proofs of all such constructions rely on the fact that computations are performed over integer fields, i.e., the moduli must be prime integers. To the best of our knowledge, no previous work has studied the security of such constructions when the computations are performed over finite integer rings, i.e., not restricting the moduli to prime integers. We aim to provide the first such analysis.

7.3.2 Constructing codes with secrecy

In this section, we describe a construction of codes with secrecy based on the same principle of Section 7.3.1; that is, the security of the construction restricts the computations to be performed over an integer field. What we will describe here is a generalization of the construction appeared in [7, 8], in which we allow operations to be performed over a finite integer ring instead of a field. (Similar constructions have also appeared in [234, 13, 14, 17]). As in the computationally secure constructions discussed in Section 7.3.1, the codes in [234, 7, 14] demand that operations must be performed over the integer field $\mathbb{Z}_p$; no previous work has studied the probability of deception of such codes when computations are performed over arbitrary finite integer rings. Other codes with secrecy include, but are not limited to, [239, 240, 250, 62].

Let the legitimate users agree on an ℓ-bit long positive integer n, where ℓ is a security parameter. The users share a secret key $k = k_1||k_2$, where k_1 and k_2 are drawn *uniformly* and *independently* from $\mathbb{Z}_n$ and $\mathbb{Z}_n^*$, respectively.

For any message $m \in \mathbb{Z}_n^*$, define $\psi_{k_1}(m) : \mathbb{Z}_n^* \rightarrow \mathbb{Z}_n$ and $\psi_{k_2}(m) : \mathbb{Z}_n^* \rightarrow \mathbb{Z}_n^*$ as follows:

$$\psi_{k_1}(m) \equiv k_1 + m \pmod{n}, \tag{7.20}$$

$$\psi_{k_2}(m) \equiv k_2 \times m \pmod{n}. \tag{7.21}$$

Equivalently, the exclusive-or operation can be used instead of the addition operation in equation (7.20) without affecting the cipher's security properties [7]. We will refer to $\psi_{k_1}(m)$ and $\psi_{k_2}(m)$ as the ciphertext and authentication tag, respectively. Then, as a function of the key k, the output of the system, $\psi_k(m)$, is the concatenation of the ciphertext and the authentication tag. That is,

$$\psi_k(m) = \psi_{k_1}(m) \mid\mid \psi_{k_2}(m). \tag{7.22}$$

Upon receiving a ciphertext $\psi'_k(m)$, the legitimate receiver extracts the plaintext m' as follows:

$$m' = \psi'_{k_1}(m) - k_1 \pmod{n}. \tag{7.23}$$

The integrity of the extracted m' is verified by the following check:

$$m' \times k_2 \overset{?}{\equiv} \psi'_{k_2}(m) \pmod{n}. \tag{7.24}$$

The notations $\psi'_k(m)$ and m' are to reflect the possibility that the received ciphertext and the extracted plaintext are different than the transmitted ones. The ciphertext is considered valid if and only if the integrity check of equation (7.24) is passed.

7.4 Security analysis

This section will be dedicated to analyzing the security of the authentication with secrecy detailed in Section 7.3.2, although the bounds on deception probabilities applies to both constructions of Section 7.3.1 and Section 7.3.2.

The scheme described in Section 7.3.2 is designed to achieve two security objectives, confidentiality and integrity. More specifically, by restricting computations to be performed over integer fields, the scheme in Section 7.3.2 achieves Shannon's perfect secrecy in addition to message integrity [7]. Even though the main emphasis of this work is to analyze the effect of working with arbitrary finite integer rings on the integrity of the scheme, we will show in Section 7.4.1, for completeness of presentation, the effect on the confidentiality of the scheme when computations are allowed to be performed over arbitrary integer rings. In Section 7.4.2 we address the main focus of the chapter, namely, the bounds on the probabilities of successful message forgery.

7.4.1 Perfect secrecy

Corollary 1. *If encrypted messages are restricted to belong to $\mathbb{Z}_n^*$, the scheme of Section 7.3.2 achieves perfect secrecy (in Shannon's sense).*

Corollary 1 is a direct consequence of Lemma 1. To see this, observe that the results of equations (7.20) and (7.21) are defined on a group $G = \mathbb{Z}_n \times \mathbb{Z}_n^*$.

Remark 10. *Restricting the message m to be relatively prime to n does not impose a significant limitation on the system since, for example, any non-trivial message will satisfy the condition when n is a prime integer. For an arbitrary positive integer n, the message can be padded to be relatively prime to n. Moreover, the system will still work without this restriction; however, perfect secrecy is not achieved.*

To illustrate how perfect secrecy is violated when messages are not restricted to the multiplicative group, consider an arbitrary message $m \in \mathbb{Z}_n$ to be encrypted. If $m \in \mathbb{Z}_n \backslash \mathbb{Z}_n^*$, by Lemma 7, the resulting ψ_{k_2} will be in $\mathbb{Z}_n \backslash \mathbb{Z}_n^*$. On the other hand, since $k_2 \in \mathbb{Z}_n^*$, if $m \in \mathbb{Z}_n^*$, by Lemma 8, the resulting ψ_{k_2} will be in $\mathbb{Z}_n^*$. Therefore, an adversary observing the authentication tag ψ_{k_2} can determine a subset of the message space that the encrypted message belongs to (if $\psi_{k_2} \in \mathbb{Z}_n \backslash \mathbb{Z}_n^*$ then $m \in \mathbb{Z}_n \backslash \mathbb{Z}_n^*$ and if $\psi_{k_2} \in \mathbb{Z}_n^*$ then $m \in \mathbb{Z}_n^*$); thus, revealing partial information about the encrypted message. Otherwise put,

$$\Pr\left[\boldsymbol{m} = m | \psi_{k_2} \in \mathbb{Z}_n^*\right] = \begin{cases} \frac{1}{|\mathbb{Z}_n^*|} & \text{if } m \in \mathbb{Z}_n^*, \\ 0 & \text{if } m \in \mathbb{Z}_n \backslash \mathbb{Z}_n^*, \end{cases} \tag{7.25}$$

and similarly for the case where $\psi_{k_2} \in \mathbb{Z}_n \backslash \mathbb{Z}_n^*$. Therefore,

$$\Pr\left[\boldsymbol{m} = m | \boldsymbol{\psi_{k_2}} = \psi_{k_2}\right] \neq \Pr\left[\boldsymbol{m} = m\right] \tag{7.26}$$

for all plaintext m and all ciphertext ψ_{k_2}; a clear violation of Definition 2 of perfect secrecy.

7.4.2 Message integrity

In what follows, we address message integrity of authentication codes based on the universal hash family under analysis. Even though the analysis applies to both schemes described in Section 7.3, we will use the notations of Section 7.3.2.

As discussed in Section 7.3.2, the main purpose of ψ_{k_2} is to serve as an authentication tag (MAC) for the encrypted message m. Thus, there are two cases to be considered, modifying ψ_{k_1} alone, and modifying both ψ_{k_1} and ψ_{k_2}. Modifying ψ_{k_2} alone, since it serves as a MAC, does not lead to extracting a false plaintext.

- Case I. Modifying the ciphertext only

Assume that ψ_{k_1} has been modified, by a man in the middle, to ψ'_{k_1}. Since k_1 is known to the receiver, this modification will lead to the extraction of an m' different than the encrypted m; that is, $m' = \psi'_{k_1} - k_1 \pmod{n}$. Let $m' = m + \delta \pmod{n}$, for some $\delta \in \mathbb{Z}_n \backslash \{0\}$. To be accepted by the receiver, m' must satisfy the following integrity check:

$$\begin{aligned} m' \times k_2 &\equiv (m + \delta) \times k_2 \pmod{n} && (7.27) \\ &\equiv (m \times k_2) + (\delta \times k_2) \pmod{n} && (7.28) \\ &\overset{?}{\equiv} \psi_{k_2} \pmod{n} && (7.29) \\ &\equiv m \times k_2 \pmod{n}. && (7.30) \end{aligned}$$

Equivalently, the integrity check in equation (7.29) is satisfied if and only if the following condition holds:

$$\delta \times k_2 \equiv 0 \pmod{n}. \tag{7.31}$$

That is, modification of ψ_{k_1} alone will go undetected if and only if it is modified by a δ that satisfies equation (7.31). Section 7.4.2.1 provides detailed probabilistic analysis of equation (7.31).

- Case II. Modifying both the ciphertext and the MAC

In a different scenario, the adversary may attempt to modify both ψ_{k_1} and ψ_{k_2} so that a false message will be validated. Assume that ψ_{k_1} has been modified so that the extracted message becomes $m' = m + \delta \pmod{n}$, for some $\delta \in \mathbb{Z}_n \backslash \{0\}$. Also, assume that ψ_{k_2} has been modified to $\psi'_{k_2} = \psi_{k_2} + \epsilon \pmod{n}$, for some $\epsilon \in \mathbb{Z}_n \backslash \{0\}$. The integrity of m' is verified using the received ψ'_{k_2} as follows:

$$\begin{aligned} \psi_{k_2} + \epsilon &\equiv \psi'_{k_2} \pmod{n} && (7.32)\\ &\overset{?}{\equiv} m' \times k_2 \pmod{n} && (7.33)\\ &\equiv (m+\delta) \times k_2 \pmod{n} && (7.34)\\ &\equiv (m \times k_2) + (\delta \times k_2) \pmod{n} && (7.35)\\ &\equiv \psi_{k_2} + (\delta \times k_2) \pmod{n}. && (7.36) \end{aligned}$$

Equivalently, the false m' will be accepted if and only if the following condition is satisfied:

$$\epsilon \equiv \delta \times k_2 \pmod{n}. \tag{7.37}$$

That is, modification of ψ_{k_1} by a value δ and ψ_{k_2} by a value ϵ will go undetected if and only if δ and ϵ satisfy equation (7.37). Section 7.4.2.2 provides detailed probabilistic analysis of equation (7.37).

7.4.2.1 Analysis of modifying ciphertext only

As derived above, an adversary modifying the ciphertext ψ_{k_1} in order to make the legitimate receiver authenticate a false message is successful if and only if she can solve the congruence

$$\delta \times k_2 \equiv 0 \pmod{n} \tag{7.38}$$

for an unknown k_2 uniformly distributed over $\mathbb{Z}_n^*$. To analyze the adversary's ability to solve this congruence for an arbitrary finite integer n, we start with the following lemma.

Lemma 10. *Let n be any fixed finite integer. For any nonzero elements α and β in $\mathbb{Z}_n$, if n divides $\alpha \times \beta$, then both α and β must belong to $\mathbb{Z}_n \backslash \mathbb{Z}_n^*$. Formally, the following one-way implication must hold:*

$$\alpha \times \beta \equiv 0 \pmod{n} \;\Rightarrow\; \{\alpha, \beta \in \mathbb{Z}_n \backslash \mathbb{Z}_n^*\}. \tag{7.39}$$

Lemma 10 is a corollary of more general results shown by Schwarz in [224]. Given Lemma 10, the adversary's chances of tampering with the ciphertext ψ_{k_1} in a way undetected by the legitimate receiver is stated in the following theorem.

Theorem 10. *Any modification of the ciphertext ψ_{k_1} alone will be detected by the legitimate receiver with probability one.*

Proof: Recall that the modification of ψ_{k_1} will be verified only if

$$\delta \times k_2 \equiv 0 \pmod{n}. \tag{7.40}$$

Lemma 10, however, states that equation (7.40) can be satisfied only if both δ and k_2 belong to $\mathbb{Z}_n \backslash \mathbb{Z}_n^*$. Since, by design, k_2 is chosen from $\mathbb{Z}_n^*$, equation (7.40) can never be satisfied. Therefore, any modification of the ciphertext ψ_{k_1} will be detected by its MAC with probability one. ■

Next, we analyze the possibility of modifying both the ciphertext and MAC, ψ_{k_1} and ψ_{k_2}, in order to make the legitimate receiver authenticate a false message.

7.4.2.2 Analysis of modifying both the ciphertext and the MAC

This section constitutes the main contribution of this chapter. All previous results stated in this chapter were either already known or follow directly from known results. The result of this section, on the other hand, has not appeared in the literature; it will show the direct relation between the prime factorization of the modulus n and the security of any authentication code based on the use of the universal hash family discussed in Section 7.3.

Recall that the adversary has to find a solution to the congruence

$$\epsilon \equiv \delta \times k_2 \pmod{n}, \tag{7.41}$$

where n is an arbitrary fixed modulus and k_2 is chosen uniformly at random from $\mathbb{Z}_n^*$, in order to make the legitimate receiver authenticate a modified message. To be able to analyze the adversary's ability to solve the congruence in equation (7.41), we start by stating a sequence of lemmas.

The first lemma specifies a necessary and sufficient condition for the existence of a k_2 that satisfies equation (7.41).

Lemma 11. *Let n be any finite positive integer. Then, for any nonzero $\epsilon, \delta \in \mathbb{Z}_n$, there exists $k \in \mathbb{Z}_n^*$ satisfying*

$$\epsilon \equiv k \times \delta \pmod{n} \tag{7.42}$$

if and only if

$$\gcd(\epsilon, n) = \gcd(\delta, n). \tag{7.43}$$

Proof: Let $\gcd(\epsilon, n) = \gcd(\delta, n) = r$. By lemma 6, there exist two invertible elements $\alpha, \beta \in \mathbb{Z}_n$ so that, $\epsilon \equiv r \times \alpha^{-1} \pmod{n}$ and $\delta \equiv r \times \beta^{-1} \pmod{n}$. Then,

$$\begin{aligned} \epsilon &\equiv r \times \alpha^{-1} \pmod{n} && (7.44)\\ &\equiv r \times \alpha^{-1} \times \beta^{-1} \times \beta \pmod{n} && (7.45)\\ &\equiv \alpha^{-1} \times \beta \times \delta \pmod{n}. && (7.46) \end{aligned}$$

Hence, $k \equiv \alpha^{-1} \times \beta \pmod{n}$ satisfies equation (7.42). Further, $k \in \mathbb{Z}_n^*$ by Lemma 8. Therefore, equation (7.43) implies equation (7.42).

Now, suppose that $\epsilon \equiv k \times \delta \pmod{n}$ for some $k \in \mathbb{Z}_n^*$. Let $r = \gcd(\epsilon, n)$ and $s = \gcd(\delta, n)$ and suppose, without loss of generality, that $r > s$. Again, by Lemma 6, there exist $\alpha, \beta \in \mathbb{Z}_n^*$ satisfying $\epsilon \equiv r \times \alpha^{-1} \pmod{n}$ and $\delta \equiv s \times \beta^{-1} \pmod{n}$. Then,

$$\begin{aligned} r \times \alpha^{-1} &\equiv \epsilon \pmod{n} && (7.47)\\ &\equiv k \times \delta \pmod{n} && (7.48)\\ &\equiv k \times s \times \beta^{-1} \pmod{n}, && (7.49) \end{aligned}$$

and multiplying both sides by α yields,

$$r \equiv s \times (\alpha \times \beta^{-1} \times k) \pmod{n}. \tag{7.50}$$

Also, since $r \mid n$, there exists an $\ell \in \mathbb{Z}_n$ such that $\ell \cdot r = n$. Multiplying both sides of equation (7.50) by ℓ yields,

$$0 \equiv (\ell \times s) \times (\alpha \times \beta^{-1} \times k) \pmod{n}. \tag{7.51}$$

Since $s < r$ by hypothesis, the first factor on the right hand side is strictly less than $n = \ell \cdot r$; hence, $(\ell \times s)$ is a nonzero element in $\mathbb{Z}_n$. By Lemma 8, the second factor belongs to $\mathbb{Z}_n^*$; a contradiction to Lemma 10, which states that for the product of two nonzero integers to be congruent to zero modulo n, both integers must be in $\mathbb{Z}_n \backslash \mathbb{Z}_n^*$. Therefore, $r = s$, and the lemma follows. ■

Lemma 11 specifies a necessary condition for the successful forgery by modifying the ciphertext by a $\delta \in \mathbb{Z}_n \backslash \{0\}$ and the MAC by an $\epsilon \in \mathbb{Z}_n \backslash \{0\}$. Namely, $\gcd(\delta, n)$ must be equal to $\gcd(\epsilon, n)$; otherwise, there does not exist a shared key $k_2 \in \mathbb{Z}_n^*$ that could possibly satisfy equation (7.41) for the chosen δ and ϵ.

Assume now that an adversary has chosen nonzero δ and ϵ that satisfy the necessary condition of Lemma 11. Given the value of δ, what is the probability that the chosen ϵ will satisfy equation (7.41). To be able to answer this question, we introduce the following set.

Definition 3 (The set of common gcd's)**.**
For any fixed integer δ, define $T(\delta)$ to be the set of ϵ's that satisfy equation (7.41) for at least one $k \in \mathbb{Z}_n^*$. That is,

$$T(\delta) := \Big\{ \epsilon \in \mathbb{Z}_n : \exists\ k \in \mathbb{Z}_n^* \textit{ such that } \epsilon \equiv \delta \times k \pmod{n} \Big\}. \tag{7.52}$$

By Lemma 11, this set is equal to the set of ϵ's in $\mathbb{Z}_n$ such that $\gcd(\epsilon, n) = \gcd(\delta, n)$. *Therefore, it can be written as,*

$$T(\delta) = \Big\{ \epsilon \in \mathbb{Z}_n : \gcd(\epsilon, n) = \gcd(\delta, n) \Big\}. \tag{7.53}$$

For the rest of the chapter, the representations in equations (7.52) and (7.53) of the set of common gcd's will be used interchangeably to define the set $T(\delta)$.

To be able to quantify the adversary's probability of successful forgery, we need to answer the following question: For an $\epsilon \in T(\delta)$, how many possible secret keys k_2's can satisfy equation (7.41) for the given (δ, ϵ) pair? More importantly, for two distinct ϵ's in $T(\delta)$, say ϵ and ϵ', what is the relation between the number of k_2's in $\mathbb{Z}_n^*$ that satisfy equation (7.41) for each of them? This question is important since, for a given δ, an intelligent adversary will choose the ϵ that maximizes her probability of successful forgery. The following lemma addresses this question.

Lemma 12. *Fix any $\delta \in \mathbb{Z}_n$ and let $\epsilon, \epsilon' \in T(\delta)$. Define the set K_ϵ to be the set of all k's in $\mathbb{Z}_n^*$ that satisfy equation (7.41) for the given δ and ϵ. Similarly, define the set $K_{\epsilon'}$ to be the set of all k's in $\mathbb{Z}_n^*$ that satisfy equation (7.41) for δ and ϵ'. That is, $K_\epsilon := \{k \in \mathbb{Z}_n^* : \delta \times k \equiv \epsilon \pmod{n}\}$ and $K_{\epsilon'} := \{k \in \mathbb{Z}_n^* : \delta \times k \equiv \epsilon' \pmod{n}\}$. Then $|K_\epsilon| = |K_{\epsilon'}|$, i.e., the sets K_ϵ and $K_{\epsilon'}$ have the same cardinality.*

Proof: Without loss of generality, assume $|K_\epsilon| < |K_{\epsilon'}| = \ell$, and let $K_{\epsilon'} = \{k_1, \ldots, k_\ell\}$, for distinct k_i's. Since $\epsilon \in T(\delta)$, there exists an r satisfying $r \times \delta \equiv \epsilon \pmod{n}$. Also, since $k_1 \in K_{\epsilon'}$, $\delta \equiv k_1^{-1} \times \epsilon' \pmod{n}$. Now, for $i = 1, \ldots, \ell$, define r_i as,

$$r_i = r \cdot k_1^{-1} \cdot k_i. \tag{7.54}$$

Then, every r_i satisfies,

$$\begin{aligned} r_i \times \delta &\equiv r \times k_1^{-1} \times k_i \times \delta \pmod{n} && (7.55)\\ &\equiv r \times k_1^{-1} \times \epsilon' \pmod{n} && (7.56)\\ &\equiv r \times \delta \pmod{n} && (7.57)\\ &\equiv \epsilon \pmod{n}. && (7.58) \end{aligned}$$

Furthermore, the r_i's are distinct: if $r_i = r_j$, then

$$r \times k_1^{-1} \times k_i \equiv r \times k_1^{-1} \times k_j \pmod{n}. \tag{7.59}$$

Since k_1^{-1} and r are invertible, by cancellation we have $k_i = k_j$, implying that $i = j$. Therefore, the set K_ϵ contains at least ℓ distinct elements, a contradiction to the hypothesis that $|K_\epsilon| < |K_{\epsilon'}|$. Therefore, $|K_\epsilon| = |K_{\epsilon'}|$. ■

Lemma 12 implies that any ϵ which has the same greatest common divisor with n as δ will have the same number of keys as possible candidates for successful forgery. That is, from the adversary's standpoint, there is no advantage of picking one particular $\epsilon \in T(\delta)$ over the others. The following lemma formalizes this argument.

Lemma 13. *Suppose that k is an unknown integer, randomly drawn from $\mathbb{Z}_n^*$. Then for any fixed $\delta \in \mathbb{Z}_n \backslash \{0\}$, the probability of selecting ϵ satisfying $\epsilon \equiv k \times \delta \pmod{n}$ is at most $1/|T(\delta)|$.*

Proof: By the definition of $T(\delta)$ and Lemma 11, all valid ϵ's are in $T(\delta)$, and any ϵ in $T(\delta)$ is a valid choice. Also, by Lemma 12, the number of possible values of k that map δ to any ϵ is the same, so there is no advantage in picking one ϵ over another, i.e., the ϵ's are uniformly distributed in $T(\delta)$. Hence, for a given $\delta \in \mathbb{Z}_n \backslash \{0\}$, the probability of selecting an $\epsilon \in T(\delta)$ that satisfies equation (7.41) is $1/|T(\delta)|$. ■

Lemma 13 implies that the adversary's best strategy for successful forgery is to choose the δ that minimizes $|T(\delta)|$. (Observe that the cardinality of $T(\delta)$ is at least one since $\delta \in T(\delta)$ for any $\delta \in \mathbb{Z}_n$.) The next two lemmas address the problem of minimizing $|T(\delta)|$.

We start with a lemma that relates the cardinality of the set T with the Euler totient function φ.

Lemma 14. *For any integer α that divides n, $|T(n/\alpha)| = \varphi(\alpha)$. More explicitly, the set $T(n/\alpha)$ can be expressed as,*

$$T(n/\alpha) = \frac{n}{\alpha}\,\Big\{\beta \in \mathbb{Z}_\alpha : \gcd(\beta, \alpha) = 1\Big\}. \tag{7.60}$$

Proof: The fact that $\alpha | n$ implies that $\gcd(n/\alpha, n) = n/\alpha$. Therefore, by the definition of T in equation (7.53),

$$T(n/\alpha) = \Big\{\epsilon \in \mathbb{Z}_n : \gcd(\epsilon, n) = \gcd(n/\alpha, n) = n/\alpha\Big\}. \tag{7.61}$$

Now, for any $\beta \in \mathbb{Z}_\alpha$ such that $\gcd(\beta, \alpha) = 1$, using the fact that $\gcd(ka, kb) = k\gcd(a, b)$ [51], we get:

$$\begin{aligned} \gcd(\beta,\alpha) = 1 \quad &\Longleftrightarrow \quad \gcd(\frac{n}{\alpha}\beta, \frac{n}{\alpha}\alpha) = \frac{n}{\alpha} && (7.62)\\ &\Longleftrightarrow \quad \gcd(\frac{n}{\alpha}\beta, n) = \frac{n}{\alpha} && (7.63)\\ &\overset{(7.61)}{\Longleftrightarrow} \quad \frac{n}{\alpha}\beta \in T(n/\alpha). && (7.64) \end{aligned}$$

Furthermore, for distinct $\beta_1, \beta_2 \in \mathbb{Z}_\alpha$, $\frac{n}{\alpha}\beta_1$ and $\frac{n}{\alpha}\beta_2$ are distinct elements of $\mathbb{Z}_n$. This is because, since $\alpha > \max\Big\{\beta_1, \beta_2\Big\}$, $n > \max\Big\{\frac{n}{\alpha}\beta_1, \frac{n}{\alpha}\beta_2\Big\}$. Therefore, there is a one-to-one correspondence between the set $\Big\{\beta \in \mathbb{Z}_\alpha : \gcd(\beta, \alpha) = 1\Big\}$ and the set $\Big\{\gamma \in \mathbb{Z}_n : \gamma \in T(n/\alpha)\Big\}$. ■

We can now state the relation between the cardinality of $T(\delta)$, for any δ, and the choice of the underlying integer ring. More specifically, the following lemma emphasizes the effect of the prime factorization of n on the cardinality of the smallest $T(\delta)$.

Lemma 15. *If p is the smallest prime factor of n, then $|T(\delta)| \geq |T(n/p)|$ for any $\delta \in \mathbb{Z}_n$.*

Proof: Let $\delta \in \mathbb{Z}_n$ and let p be the smallest prime factor of n. By Lemma 14, $|T(n/p)| = \varphi(p) = p - 1$. Now, recall that:

$$T(\delta) = \Big\{\epsilon \in \mathbb{Z}_n : \gcd(\epsilon, n) = \gcd(\delta, n)\Big\}. \tag{7.65}$$

Then, if $\gcd(\delta, n) = 1$, by equation (7.65),

$$|T(\delta)| = |\mathbb{Z}_n^*| = \varphi(n); \tag{7.66}$$

and we know, by Lemma 9, that

$$\varphi(n) \geq p - 1; \tag{7.67}$$

and, by Lemma 14, that

$$p - 1 = |T(n/p)|. \tag{7.68}$$

Thus, $|T(\delta)| \geq |T(n/p)|$ for all δ's that are relatively prime to n.

It remains to show that the same is true for δ's that are not relatively prime to n. Let $\gcd(\delta, n) = d > 1$, then, by equation (7.65),

$$T(\delta) = T\Big(\gcd(\delta, n)\Big) = T(d). \tag{7.69}$$

Therefore, we can assume, without loss of generality, that $\delta | n$ (since for any δ such that $\gcd(\delta, n) = d$, $T(\delta) = T(d)$ and $d|n$). Now, write $\delta = n/\alpha$ and let α be written in its prime factorization form as $\alpha = \prod_i p_i^{e_i}$, where the p_i's are distinct primes. Then, $\varphi(\alpha) = \prod_i (p_i - 1)p_i^{e_i - 1}$. Since $\alpha | n$, and p is the smallest prime factor of n, $p \leq p_i$ for any i. Hence, by Lemma 14,

$$|T(\delta)| = |T(n/\alpha)| = \varphi(\alpha) \geq p - 1 = |T(n/p)|. \tag{7.70}$$

Therefore, for any $\delta \in \mathbb{Z}_n$, $|T(\delta)| \geq |T(n/p)|$. ■

We can now state the main theorem analyzing the adversary's probability of successful forgery by modifying both ciphertext ψ_{k_1} and the MAC ψ_{k_2}.

Theorem 11. *Let p be the smallest prime factor of n. Then, an adversary modifying both the ciphertext ψ_{k_1} and the MAC ψ_{k_2} will be successful with probability at most $1/(p-1)$.*

Proof: Recall that an adversary modifying ψ_{k_1} and ψ_{k_2} will be successful only if she can choose δ, ϵ such that:

$$\epsilon \equiv \delta \times k_2 \pmod{n}. \tag{7.71}$$

By Lemma 13, the probability of choosing δ, ϵ that satisfy equation (7.71) is given by $1/|T(\delta)|$. To maximize the probability of successful forgery, the adversary can choose δ that minimizes the size of $T(\delta)$. By Lemma 15, the best choice of δ that minimizes $T(\delta)$ is $\delta = n/p$, where p is the smallest prime factor of n. Finally, by Lemma 14, $|T(n/p)| = p-1$, and the theorem follows. ∎

7.5 Choice of integer rings

The described authenticated encryption schemes is designed to achieve two main objectives, message confidentiality and integrity. In this section we summarize the effect of the underlying integer ring on the security properties of the scheme.

It has been shown, in Section 7.4.1, that reducing message space to the multiplicative group of integers modulo n is a necessary condition for the scheme to achieve perfect secrecy. Consequently, the choice of the underlying integer ring will be a factor for the number of possible messages that can be encrypted with perfect secrecy.

In Section 7.4.2.1, it was shown that an adversary modifying ψ_{k_1} only will be successful only if ψ_{k_1} is perturbed by an integer δ that satisfies

$$\delta \times k_2 \equiv 0 \pmod{n}. \tag{7.72}$$

Moreover, it was shown that choosing k_2 from the multiplicative group $\mathbb{Z}_n^*$ is a sufficient condition to guarantee that no nonzero $\delta \in \mathbb{Z}_n$ will satisfy equation (7.72). Therefore, the choice of the underlying integer ring does not play an important role in the protection against modifying ψ_{k_1} only, other than restricting k_2 to be chosen from the multiplicative group $\mathbb{Z}_n^*$.

The choice of the underlying integer ring has its most impact when an adversary modifies both ψ_{k_1} and ψ_{k_2}. As discussed in Section 7.4.2.2, the adversary is successful in tampering with the message, in a way undetected by the legitimate receiver, only if she can select ϵ, δ satisfying:

$$\epsilon \equiv \delta \times k_2 \pmod{n}. \tag{7.73}$$

The proof of Theorem 11 describes the following attack on the scheme. Suppose that the scheme designer chooses a modulus with prime factorization given by $n = p_1^{e_1} \cdots p_k^{e_k}$, where the p_i's are ordered increasingly. Assuming the adversary is able to factor n, then she can choose $\delta = n/p_1$ to maximize her probability of successful forgery. The resulting $\delta \times k_2 \pmod{n}$ will be, from the adversary's perspective, a random element in the set of multiples of n/p_1 (excluding 0 because k_2 is known to be relatively prime to n). Consequently, by randomly choosing an integer ϵ from the set $\left\{m\dfrac{n}{p_1} \pmod{n}, \text{ for } m = 1, \ldots, p_1 - 1\right\}$, the adversary can tamper with the message without detection with probability $1/(p_1 - 1)$. The following numerical example illustrates the attack.

Example 2. *Let $n = 45 = 3^2 \times 5$. According to Theorem 11, the adversary can maximize her probability of successful forgery by choosing $\delta = n/3 = 15$. Moreover, the secret key k_2 is restricted to belong to the multiplicative group $\mathbb{Z}_{45}^*$, that is, $k_2 \in \{1, 2, 4, 7, 8, 11, 13, 14, 16, 17, 19, 22, 23, 26, 28, 29, 31, 32, 34, 37, 38, 41, 43, 44\}$. Thus, the resulting $\delta \times k_2 \pmod{45} := \epsilon$ is equal to*

$$15 \times 1 \equiv 15 \pmod{45}, \tag{7.74}$$

$$15 \times 2 \equiv 30 \pmod{45}, \tag{7.75}$$

$$15 \times 4 \equiv 15 \pmod{45}, \tag{7.76}$$

$$15 \times 7 \equiv 15 \pmod{45}, \tag{7.77}$$

$$15 \times 8 \equiv 30 \pmod{45}, \tag{7.78}$$

$$15 \times 11 \equiv 30 \pmod{45}, \tag{7.79}$$

$$15 \times 13 \equiv 15 \pmod{45}, \tag{7.80}$$

$$15 \times 14 \equiv 30 \pmod{45}, \tag{7.81}$$

$$15 \times 16 \equiv 15 \pmod{45}, \tag{7.82}$$

$$15 \times 17 \equiv 30 \pmod{45}, \tag{7.83}$$

$$15 \times 19 \equiv 15 \pmod{45}, \tag{7.84}$$

$$15 \times 22 \equiv 15 \pmod{45}, \tag{7.85}$$

$$15 \times 23 \equiv 30 \pmod{45}, \tag{7.86}$$

$$15 \times 26 \equiv 30 \pmod{45}, \tag{7.87}$$

$$15 \times 28 \equiv 15 \pmod{45}, \tag{7.88}$$

$$15 \times 29 \equiv 30 \pmod{45}, \tag{7.89}$$

$$15 \times 31 \equiv 15 \pmod{45}, \tag{7.90}$$

$$15 \times 32 \equiv 30 \pmod{45}, \tag{7.91}$$

$$15 \times 34 \equiv 15 \pmod{45}, \tag{7.92}$$

$$15 \times 37 \equiv 15 \pmod{45}, \tag{7.93}$$

$$15 \times 38 \equiv 30 \pmod{45}, \tag{7.94}$$

$$15 \times 41 \equiv 30 \pmod{45}, \tag{7.95}$$

$$15 \times 43 \equiv 15 \pmod{45}, \tag{7.96}$$

$$15 \times 44 \equiv 30 \pmod{45}. \tag{7.97}$$

That is, the resulting ϵ will be 15 or 30 with equal probability. (Similarly, one can show that, by choosing $\delta = n/5 = 9$, the resulting ϵ is uniformly distributed over $\{9, 18, 27, 36\}$). In any case, the resulting ϵ will be uniformly distributed over the multiples of n/p, where p is a prime factor of n, and the p that minimizes the cardinality of the set of possible ϵ's is the smallest prime factor of n.

As an illustration of the importance of the underlying integer ring, in what follows, we show the best and the worst choices of integer rings in terms of security against man in the middle attacks.

7.5.1 Prime moduli

When the used modulus is a prime integer p, the underlying integer ring $\mathbb{Z}_p$ becomes a field. Not surprisingly, the use of a prime modulus gives the best security performances against message corruption attacks. Since the smallest prime factor of p is p itself, by Theorem 11, the adversary's probability of successful forgery is $1/(p-1)$. That is, there is no advantage of choosing a δ over another. In other words, no matter what the value of δ an adversary chooses, the resulting ϵ will be uniformly distributed over the entire set of nonzero element $\{1, 2, \cdots, p-1\}$.

7.5.2 Even moduli

Even moduli give the worst security against message modification. An active adversary can take advantage of the even modulus to make the intended receiver authenticate a false message with probability one. This is due to the fact that the smallest prime factor of n is 2. Therefore, by Theorem 11, the adversary's probability of successful forgery is $1/(2-1)$. To illustrate the attack, let the adversary choose $\delta = n/2$. Since $k_2 \in \mathbb{Z}_n^*$, and n is an even integer, k_2 must be an odd integer, which can be written in the form $2r+1$ for some positive integer r. Then,

$$\begin{aligned} \epsilon &\equiv \delta \times k_2 \pmod{n} && (7.98)\\ &\equiv (\frac{n}{2}) \times (2r+1) \pmod{n} && (7.99)\\ &\equiv \frac{n}{2} \pmod{n}. && (7.100) \end{aligned}$$

Therefore, choosing $\delta = \epsilon = n/2$ guarantees that the modification will go undetected with probability one. Consequently, even moduli cannot be used to implement the described scheme since an active adversary can always perturb both ψ_{k_1} and ψ_{k_2} in a way undetected by the legitimate receiver.

7.6 Summary

In this chapter, we investigated authentication based on a class of universal hash-function families that have been appeared in the literature. Although the studied universal hash-function family has appeared in many places, computations have always been performed modulo prime integers. In this work, we analyzed the security of message authentication when computations are performed over arbitrary finite integer rings. We derived a direct relation between the security of authentication and the underlying integer ring $\mathbb{Z}_n$. Specifically, we showed that the bound on successful forgery is proportional to the reciprocal of the smallest prime factor of the used modulus n.

Part II

SECURING RADIO FREQUENCY IDENTIFICATION

The second part of this dissertation investigates the problem of security and privacy in radio frequency identification (RFID) systems. There are three technical contributions in this part, detailed in three different chapters. Before we describe the technical contributions of this part, we give a brief background and preliminaries that will be used for the rest of Part II in Chapter 8. The first technical contribution entitled "*Securing RFID Systems: An Information-Theoretically Secure Approach*" is given in Chapter 9. The second technical contribution entitled "*Securing RFID Systems: A Computationally Secure Approach*" is given in Chapter 10. The third technical contribution entitled "*Reducing Identification Complexity*" is given in Chapter 11.

Chapter 8

RADIO FREQUENCY IDENTIFICATION SYSTEMS

8.1 System Components and Functionality

Typically, radio frequency identification (RFID) systems are composed of three main components: tags, readers, and a database. An RFID tag is a small device that can be attached to products and allow for unique item identification and product description. RFID tags can be battery powered (active) or powerless (passive). Passive RFID tags (which are the main emphasis of this dissertation) are low-cost devices with limited memory and limited computational capabilities. An RFID reader, on the other hand, is a computationally powerful device with ability to interrogate tags and access the database, where information about individual tags and their corresponding items is stored.

When an RFID tag is within communication range of an RFID reader, the reader interrogates that tag (and charges it if it is passive). Upon interrogation, the tag responds with a quantity that allows legitimate readers to access the database and carry out the identification process. If things work as planned, the reader should be able to uniquely identify the interrogated tag.

The specific details of the identification process can vary dramatically from one protocol to another, depending on the targeted applications and the security assumptions on the underlying environment. In its simplest form, identification can be as straightforward as sending unique identifiers in clear text. Figure 8.1 depicts an instance of a simple identification run. The RFID reader interrogates the tag by sending a "Hello" message. The tag responds with its unique identifier in clear text. The reader can then access the database to obtain information about the tag and the item carrying it.

Compared to traditional means of identification, RFID tags possess a unique property that is making the deployment of RFID systems in everyday life a highly controversial issue. RFID tags respond to readers' queries via the wireless medium, no line-of-sight is required

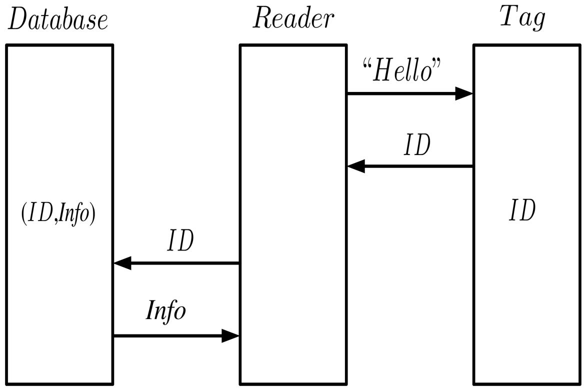


Figure 8.1: A simple identification protocol for reading the ID of an RFID tag within a communication range of an RFID reader. The RFID reader broadcasts a "$Hello$" message to announce. The RFID tag responds to the reader's request by transmitting its unique ID. The tag's unique ID is then used by the reader to lookup the database for information related to the item carrying the RFID tag.

as in traditional identification processes (e.g., ID cards or barcodes). Consequently, the identity of RFID tags, and ultimately their owners, can be revealed to unauthorized parties without the owners' approval nor even their awareness.

Privacy activists have been concerned about the invasion of users' privacy by RFID tags, calling for the delay or even the abandonment of their deployment (naming RFID tags "the spy chips" [6]). In some cases, companies have been forced to repudiate their plans for RFID deployment in response to the threat of being boycotted [87]. Consequently, providing private identification for RFID systems has been an attractive problem for both academic and industrial researchers.

8.2 Security Challenges in RFID Systems

In addition to the primary objective of RFID systems, identification, there are two secondary goals that most RFID systems aim to satisfy: privacy and security. The distinction between the primary and secondary goals could explain the market failure of privacy friendly solutions for RFID systems.

The simple scheme depicted in Figure 8.1 is clearly a violation of users' privacy. A person carrying an item equipped with an RFID tag, a watch or a jacket for example, can be tracked down by the tag he/she is carrying. A rogue reader interrogating a tag multiple times, and receiving the same identifier as in the basic scheme of Figure 8.1, will be able to correlate the tag's responses and ultimately identifying the person carrying the tag, without his/her awareness. Furthermore, the scheme of Figure 8.1 is also a security violation. Users listening to the same radio channel can record the identity of the tag and illegally impersonate that tag later. For instance, a tag used for access control can be easily cloned, granting access to unauthorized, possibly malicious, persons. Therefore, in a typical RFID system, tag authentication is also a basic requirement.

There are two complementing ends in any radio frequency identification system:

1. the interactive protocol between authorized RFID reader-tag pairs,

2. the interaction between RFID readers and the database for data retrieval.

The reader-tag interactive protocol usually involves, in addition to identification, tag authentication or mutual authentication, depending on the specification of the protocol. The data retrieval mechanism is nontrivial since, to satisfy the privacy requirement, tags' information cannot be transmitted in clear text. That is, in one hand, tags' responses must not reveal private information to unauthorized observers and, on the other hand, it must enable authorized readers to access the database and retrieve tags' information.

In most papers in the literature of private RFID systems, those two complementing ends have been treated independently. In fact, the majority of RFID papers study only the interaction between RFID reader-tag pairs, either by proposing new protocols or analyzing existing ones. This is not surprising because it deals with the most challenging issue in RFID designs. The fact that RFID tags are, in most applications, low-cost devices with stringent computational capabilities makes the use of sophisticated cryptographic primitives proven to achieve private identification and secure authentication impractical. In fact, several attempts have been made to come up with cryptographic primitives designed specifically for RFID tags (see, e.g., [45, 77, 75, 226, 112]).

However, the fact that both readers and the database are computationally powerful devices able to establish secure channels does not make the reader-database interaction an easy one. The computational limitation of RFID tags, in addition to its direct implications on the reader-tag interaction, have a significant impact on the reader-database information retrieval process. In this dissertation, the two complementing ends of RFID systems are addressed. In particular, Chapters 9 and 10 address the reader-tag interaction protocol, while Chapter 11 is dedicated to the reader-database data retrieval issue.

8.3 Related Work

There exist multiple good survey papers that study different security aspects of RFID systems (see, e.g., [221, 262, 222, 232, 154, 128, 199, 273, 18]). To formalize security analysis in RFID systems, multiple models have been proposed (see, e.g., [22, 276, 251, 102, 185, 86, 170, 278, 152]).

To mitigate the vulnerabilities of the simple identification protocol of Figure 8.1, the problem of identity authentication in RFID systems has been studied under different constraints. The use of public-key cryptographic solution has been addressed in multiple studies (see, e.g., [129, 21, 97, 177, 50, 161]).

Since symmetric-key operations typically require orders of magnitude less circuitry and consume orders of magnitude less energy than public-key ones [208], symmetric-key solutions have appeared more frequently in the literature. The design of special cryptographic primitives to meet the computational capabilities of RFID tags is of particular interest. Feldhofer et al. proposed a 128-bit version of the Advanced Encryption Standard (AES) designed specifically for RFID tags in [75, 77]. Their implementation requires 3628 gates in expense of more clock cycles. The design of low-power processors to implement AES computations in passive RFID tags has also been proposed in [121, 162, 214, 112]. In a related direction, the design of hardware-efficient one-way cryptographic hash functions targeting RFID tags has appeared in [191, 25, 59, 235]. Privacy and security issues of RFID systems based on one-way hash functions have also been studied extensively (see, e.g., [262, 76, 46, 164]).

Researchers, however, were still not satisfied with the advances in hardware technologies and attempted to come up with special cryptographic primitives for RFID systems. In [132], Juels and Weis proposed HB$^+$, a lightweight authentication protocol based on the human-to-computer authentication protocol pioneered by Hopper and Blum in [117]. Due to its lightweight computations, HB$^+$ attracted significant attention. Gilbert et al. showed a successful linear time active attack on the HB$^+$ protocol in [91], and proposed an improvement to the security and efficiency of HB in [90]. The study and analysis of HB$^+$-like protocols is still an active research area [89, 155, 48, 193, 272, 139, 105].

Furthermore, there is yet another class of protocols targeting RFID system with stringent computational capabilities based on a combination of randomness and simple bitwise operations. Such protocols include, but are not limited to, [247, 200, 201, 202, 59]. Protocols of this class, however, have been shown to be severely vulnerable to attacks [63, 158, 157, 12, 206, 23]. In fact, as shown in [15], simple bitwise operation cannot lead to secure authentication. An up-to-date report listing known broken RFID protocols with detailed descriptions of the suggested attacks is maintained by Van Deursen and Radomirovic in [248].

8.4 Model Assumptions

In this section, we state the system, adversarial, and security models that will be used for the remainder of Part II.

8.4.1 System Model

Tags are assumed to have limited computing power; in particular, public-key operations are assumed beyond their computational capabilities. Tags are equipped with a nonvolatile memory so they can retain their keying information and carry out necessary updates. The reader is a computationally powerful device with the ability to perform sophisticated cryptographic operations. The database is a storage resource at which information about tags in the system is stored. Readers-database communications are assumed to be secure.

8.4.2 Adversarial Model

We assume adversaries with complete control over the communication channel. Adversaries can observe all exchanged messages, modify exchanged messages, block exchanged messages and replay them later, and generate messages of their own. We do not consider an adversary whose only goal is to jam the communication channel. Distinguishing tags by the physical fingerprints of their transmissions requires sophisticated devices and cannot be solved using cryptographic solutions; it is out of the scope of this dissertation.

The adversary $\mathcal{A}$ is modeled as a polynomial-time algorithm. Given a tag, T, and a reader, R, we assume $\mathcal{A}$ has access to the following oracles:

- *Query* (T, m_1, x_2, m_3): $\mathcal{A}$ sends m_1 as the first message to T; receives a response, x_2; and then sends the message $m_3 = f(m_1, x_2)$. This oracle models the adversary's ability to interrogate tags in the system.

- *Send* (R, x_1, m_2, x_3): $\mathcal{A}$ receives x_1 from the reader R; replies with $m_2 = f(x_1)$; and receives the reader's response x_3. This oracle models the adversary's ability to act as a tag in the system.

- *Execute* (T, R): The tag, T, and the reader, R, execute an instance of the protocol. $\mathcal{A}$ eavesdrops on the channel, and can also tamper with the messages exchanged between T and R. This oracle models the adversary's ability to actively monitor the channel between tag and reader.

- *Block* $(\cdot)$: $\mathcal{A}$ blocks any part of the protocol.

- *Reveal* (T): This query models the exposure of the tags' secret parameters to $\mathcal{A}$. The oracle simulates the adversary's ability to physically capture the tag and obtain its secret information.

$\mathcal{A}$ can call the oracles *Query*, *Send*, *Execute*, and *Block* any polynomial number of times. The *Reveal* oracle can be called only once (on the same tag), at which the tag is considered compromised and, thus, there is no point of calling the *Reveal* oracle on the same tag multiple times. To model tag compromise attacks, however, the adversary is allowed to call other oracles after the *Reveal* oracle on the same tag; detailed discussion about this is provided in Section 11.6.

8.4.3 Security Model

The two main security goals of RFID systems are tags' privacy and identity authentication. There are different notions of privacy in the RFID literature (see, e.g., [22, 133, 170]). In this dissertation, privacy is measured by the adversary's ability to trace tags by means of their responses in different interactions. We define three notions of untraceability, *universal*, *forward*, and *existential*.

Definition 4 (Universal Untraceability)**.** *In an RFID system, tags are said to be universally untraceable if an adversary cannot track a tag based on information gained before the tag's last authentication with a valid reader. In other words, there is no correlation between a tag's responses before and after completing a protocol run with a valid reader.*

Universal untraceability is modeled by the following game between the challenger $\mathcal{C}$ (an RFID system) and a polynomial time adversary $\mathcal{A}$.

1. $\mathcal{C}$ selects two tags, T_0 and T_1, and a valid reader, R.

2. $\mathcal{A}$ makes queries on T_0, T_1, and R using the *Query*, *Send*, *Execute*, and *Block* oracles for a number of times of its choice.

3. $\mathcal{A}$ stops calling the oracles and notifies $\mathcal{C}$.

4. $\mathcal{C}$ carries out an instance of the protocol with T_0 and T_1, during which mutual authentication of both tags with R is achieved.

5. $\mathcal{C}$ selects a random bit, b, and sets $T = T_b$.

6. $\mathcal{A}$ makes queries of T and R using the *Query*, *Send*, *Execute*, and *Block* oracles.

7. $\mathcal{A}$ outputs a bit, b', and wins the game if $b' = b$.

The second notion of privacy, forward untraceability, is defined as follows.

Definition 5 (Forward Untraceability). *In an RFID system with forward untraceability, an adversary capturing the tag's secret information cannot correlate the tag with its responses before the last complete protocol run with a valid reader.*

Forward untraceability is modeled by the following game between $\mathcal{C}$ and $\mathcal{A}$.

1. $\mathcal{C}$ selects two tags, T_0 and T_1, and a valid reader, R.
2. $\mathcal{A}$ makes queries of T_0, T_1, and R using the *Query*, *Send*, *Execute*, and *Block* oracles for a number of times of its choice.
3. $\mathcal{A}$ stops calling the oracles and notifies $\mathcal{C}$.
4. $\mathcal{C}$ carries out an instance of the protocol with T_0 and T_1, during which mutual authentication of both tags with R is achieved.
5. $\mathcal{C}$ selects a random bit, b, and sets $T = T_b$.
6. $\mathcal{A}$ calls the oracle *Reveal* (T).
7. $\mathcal{A}$ outputs a bit, b', and wins the game if $b' = b$.

Finally, the third notion of privacy, existential untraceability, is defined as follows.

Definition 6 (Existential Untraceability). *Tags in an RFID system are said to be existentially untraceable if an* active *adversary cannot track a tag based on its responses to multiple interrogation, even if the tag has not been able to accomplish mutual authentication with an authorized reader.*

Existential untraceability is modeled by the following game between $\mathcal{C}$ and $\mathcal{A}$.

1. $\mathcal{C}$ selects two tags, T_0 and T_1.

2. $\mathcal{A}$ makes queries of T_0 and T_1 using the *Query* oracle for at most $C-1$ number of times for each tag, where C is a pre-specified system security parameter.

3. $\mathcal{A}$ stops calling the oracles and notifies $\mathcal{C}$.

4. $\mathcal{C}$ selects a random bit, b, and sets $T = T_b$.

5. $\mathcal{A}$ makes a query of T using the *Query* oracle.

6. $\mathcal{A}$ outputs a bit, b', and wins the game if $b' = b$.

To quantify the adversary's ability to trace RFID tags, we define the adversary's advantage of successfully identifying the tag in the previous games as

$$\mathsf{Adv}_{\mathcal{A}} = \left|\Pr[b' = b] - \frac{1}{2}\right|. \tag{8.1}$$

Tags are said to be untraceable if the adversary's advantage of equation (8.1) is a negligible function in the security paramter.

The other security goal of RFID systems in this dissertation is mutual reader-tag authentication. An *honest protocol run* is defined as follows [14]: A mutual authentication protocol run in the symmetric-key setup is said to be honest if the parties involved in the protocol run use their shared key to exchange messages, and the messages exchanged in the protocol run have been relayed faithfully (without modification).

We now give the formal definition of secure mutual authentication for RFID systems as appeared in [14].

Definition 7 (Secure Mutual Authentication). *A mutual authentication protocol for RFID systems is said to be secure if and only if it satisfies all the following conditions:*

1. No information about the secret parameters of an RFID tag is revealed by messages exchanged in protocol runs.

*2. **Authentication** ⇒ **Honest protocol:** the probability of authentication when the protocol run is not honest is negligible in the security parameter.*

*3. **Honest protocol** ⇒ **Authentication:** if the protocol run is honest, the tag-reader pair must authenticate each other with probability one.*

To model the adversary's attempt to authenticate herself to a reader (tag), we propose the following game between the challenger $\mathcal{C}$ and adversary $\mathcal{A}$.

1. $\mathcal{C}$ chooses a tag, T, at random, and a reader, R.

2. $\mathcal{A}$ calls the oracles *Query*, *Send*, *Execute*, and *Block* using T and R for a number of times of its choice.

3. $\mathcal{A}$ decides to stop and notifies $\mathcal{C}$.

4. $\mathcal{A}$ calls the oracle *Send* (*Query*) to impersonate a tag (reader) in the system.

5. If $\mathcal{A}$ is authenticated as a valid tag (reader), $\mathcal{A}$ wins the game.

Definition 7 implies that the protocol achieves secure mutual authentication only if the adversary's probability of winning the previous game is negligible.

Chapter 9

SECURING RFID SYSTEMS: AN INFORMATION-THEORETICALLY SECURE APPROACH

This chapter is the first in a series of three chapters in which we address the challenges in securing RFID systems. The main idea of this chapter is to come up with a reasonable relaxation on the adversarial capabilities that can be practical for certain applications. Then, given the relaxation, we introduce an identification protocol that meets the stringent computational power of low-cost RFID systems.

9.1 Moore's Law and RFID Systems

While mutual authentication is a well-studied problem in the cryptographic literature, it becomes more challenging with the use of low-cost devices. Low-cost RFID tags, in particular, have limited computational capabilities that render them unable to perform sophisticated cryptographic operations. Hoping Moore's law will eventually render RFID tags computationally powerful, it might be tempting to consider the computational limitations of low-cost tags a temporary problem. The cost of tags, however, will remain a determining factor in the deployment of RFID systems in real life applications. When RFID technology is to replace barcodes to identify individual items, RFID tags will substantially contribute to the cost of these products. Even when the price of tags that can implement provably secure cryptography can be driven to 10 cents or less, it would still be impractical to attach them to low-cost items, e.g., 50-cent or cheaper products. When retailers are to choose between tags that can perform sophisticated cryptographic operations and cheaper tags that cannot, it seems inevitable that the cheaper tags will prevail.

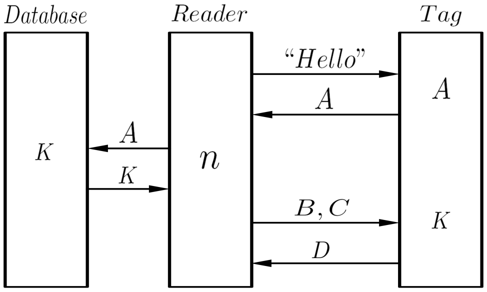
As discussed in Section 8.3, the problem of authenticating RFID systems has been studied under different assumptions, ranging from public-key solutions to simple bitwise operations. As can be inferred from the examples in Section 8.3, previous secure RFID protocols are all *computationally secure*. That is, security can only be proven assuming the hardness of breaking the security of a certain standard cryptographic primitive that is used in the interaction protocol. Hoping to meet the limited computational capabilities of low-cost tags, we start the search for *unconditionally secure* protocols for RFIDs. Unconditional security relies on the freshness of the keys rather than the hardness of solving mathematical problems. Thus, an appropriately designed unconditionally secure protocol will normally require less computational effort.

In this chapter, we introduce the first UnConditionally Secure mutual authentication protocol for RFID systems (UCS-RFID). To minimize the computational effort on tags, we develop an unconditionally secure method for delivering random numbers from RFID readers to tags. Thus, allowing tags to benefit from the functionalities of random numbers without the hardware to generate them. Then, we take advantage of the secrecy of exchanged messages to develop a novel unconditionally secure technique for message authentication using only a single multiplication operation.

The rest of the chapter is organized as follows. In Section 9.2, we describe our UCS-RFID protocol. In Section 9.3, we give detailed security analysis of mutual authentication in the proposed UCS-RFID. In Section 9.4, we address desynchronization attacks on the proposed system. In Section 9.5, we modify the original scheme of Section 9.2 to overcome desynchronization attacks. In Section 9.6 we discuss tags' privacy in the proposed system. We summarize the chapter in Section 9.7.

9.2 The proposed UCS-RFID System

Based on pre-defined security requirements, a security parameter, N, is specified and a $2N$-bit prime integer, p, is chosen. Initially, each tag is loaded with an N-bit long identifier, $A^{(0)}$, and a secret key composed of five subkeys, i.e., $K^{(0)} = \left(k_a^{(0)}, k_b^{(0)}, k_c^{(0)}, k_d^{(0)}, k_u^{(0)}\right)$. The length of k_a and k_d is N bits, while k_b, k_c, and k_u are $2N$-bit long. The subkeys, $k_a^{(0)}$ and $k_d^{(0)}$, and the identifier, $A^{(0)}$, are drawn independently and uniformly from $\mathbb{Z}_{2^N}$; $k_b^{(0)}$ is drawn uniformly from $\mathbb{Z}_p$; while $k_c^{(0)}$ and $k_u^{(0)}$ are drawn independently and uniformly from $\mathbb{Z}_p^*$. The subkeys k_a, k_b, k_c, and k_d will be used to generate messages exchanged in protocol runs, while the sole purpose of k_u is for updating the secret keys to maintain certain properties



$$A \equiv n_\ell + k_a \mod 2^N$$

$$B \equiv n + k_b \mod p$$

$$C \equiv n \times k_c \mod p$$

$$D = n_\ell \oplus k_d$$

Figure 9.1: A schematic of one instance of the proposed UCS-RFID protocol. The reader interrogates the tag which responds with its current identifier, A. Using the received A, the reader looks up the database for the corresponding key K. The reader generates a random nonce, n, and use it with K to compute B and C. The tag responds with D to authenticate itself.

(details are discussed later).

The security of the protocol relies on the reader's ability to convey a random nonce to the tag in an *authenticated* and *secret* manner (inspired by the information-theoretically secure authenticated encryption proposed in [7]). When an RFID reader interrogates a tag within its communication range, the tag responds with its identifier, A. Once the tag has been identified, the reader generates a $2N$-bit long random nonce, n, and delivers it to the tag. If the reader is authenticated successfully, the received n will be used by the tag to authenticate itself to the valid reader.

For the rest of the chapter, quantities involved in the generation of exchanged messages in different protocol runs will be differentiated by superscripts. When differentiation between protocol runs is unnecessary, superscripts will be dropped for ease of notation.The proposed UCS-RFID enables the mutual authentication between an RFID reader and a tag by executing four phases: a tag identification phase, a reader authentication phase, a tag authentication phase, and a key updating phase. Figure 9.1 depicts a single protocol run of the proposed UCS-RFID.

9.2.1 Tag Identification Phase

In order to carry out the authentication process, the reader must identify the tag it is communicating with to access its key information.

Step 1. The reader announces its presence by broadcasting a *"Hello"* message.

Step 2. The tag responds to the *"Hello"* message by sending its current identifier, A.

Step 3. The reader looks up the database for the key $K = (k_a, k_b, k_c, k_d, k_u)$ corresponding to the tag's current identifier, A.[1] If A is not recognized as a valid identifier, the tag is rejected.

9.2.2 Reader Authentication Phase

This is one of the most important phases in the proposed protocol. In this phase, the RFID reader authenticates itself to the tag by proving its knowledge of the tag's subkeys k_b and k_c. More importantly, the reader delivers a nonce, n, to the tag in an authenticated and perfectly secret manner.

Step 4. The reader generates a $2N$-bit random nonce, n, drawn uniformly from the *multiplicative group* $\mathbb{Z}_p^*$. We emphasize that n must be an unpredictable nonce; predictable nonces such as time stamps do not induce the required randomness.

Step 5. With k_b, k_c, and n, the reader broadcasts two messages, B and C, generated

[1] Database management is out of the scope of this work.

according to the following formulas:

$$B \equiv n + k_b \mod p, \tag{9.1}$$

$$C \equiv n \times k_c \mod p. \tag{9.2}$$

Step 6. Upon receiving B and C, the tag extracts n from message B and verifies its integrity using message C. The reader is authenticated if and only if the following integrity check is satisfied,

$$(B - k_b) \times k_c \equiv C \mod p. \tag{9.3}$$

If the integrity check of equation (9.3) does not pass, the reader will not be authenticated and the tag will abort the protocol.

9.2.3 Tag Authentication Phase

In the tag authentication phase, the tag is authenticated by its ability to extract the correct nonce, n, and its knowledge of the secret key k_d.

Step 7. If the reader failed the authentication process, the tag aborts the protocol. Otherwise, the tag broadcasts message D, given by

$$D = n_\ell \oplus k_d, \tag{9.4}$$

where n_ℓ denotes the N most significant bits of n.

Step 8. Upon receiving D, the reader authenticates the tag by verifying that the received D is equal to $n_\ell \oplus k_d$. Otherwise, the tag is rejected.

9.2.4 Key Update Phase

After a mutual authentication between the RFID reader and the tag is achieved, the parameters are updated at the database and the tag for the next mutual authentication run.

Step 9. The reader and the tag update the key, K, and the tag identifier, A. Let $A^{(m)}$, $k_i^{(m)}$, and $n^{(m)}$ denote the identifier A, k_i, and n used to execute the m^{th} protocol run; let n_r denotes the N least significant bits of n. Then, the parameters are updated as follows,

$$
\begin{aligned}
k_a^{(m+1)} &= n_r^{(m)} \oplus k_a^{(m)}, && (9.5)\\
k_b^{(m+1)} &\equiv k_u^{(m)} + (n^{(m)} \oplus k_b^{(m)}) \mod p, && (9.6)\\
k_c^{(m+1)} &\equiv k_u^{(m)} \times (n^{(m)} \oplus k_c^{(m)}) \mod p, && (9.7)\\
k_d^{(m+1)} &= n_r^{(m)} \oplus k_d^{(m)}, && (9.8)\\
k_u^{(m+1)} &\equiv k_u^{(m)} \times n^{(m)} \mod p, && (9.9)\\
A^{(m+1)} &\equiv n_\ell^{(m)} + k_a^{(m+1)} \mod 2^N. && (9.10)
\end{aligned}
$$

It is vital for the security of the protocol that the updated $k_b^{(m+1)}$ and $k_c^{(m+1)}$ remain uniformly distributed over $\mathbb{Z}_p$ and $\mathbb{Z}_p \backslash \{0\}$, respectively. Here is where the updating key, k_u, comes into play. In addition to inducing a desired independence between message $B^{(m)}$ and the updated $k_b^{(m+1)}$, and between message $C^{(m)}$ and the updated $k_c^{(m+1)}$, observe that, since $\mathbb{Z}_p$ is a field, $k_u^{(m)}$ will always be uniformly distributed over $\mathbb{Z}_p^*$ (since the initial $k_u^{(0)}$ is drawn uniformly from $\mathbb{Z}_p^*$ and every generated nonce is a random element of $\mathbb{Z}_p^*$). Therefore, $k_b^{(m+1)}$ is uniformly distributed over $\mathbb{Z}_p$. However, there is a possibility that $k_c^{(m+1)}$ will be equal to zero; which will occur, *with negligible probability*, when $n^{(m)} \oplus k_c^{(m)}$ is congruent to zero modulo p. In this case, $n^{(m)} \oplus k_c^{(m)}$ in equation (9.7) is replaced with $n^{(m)} \times k_c^{(m)}$. Now, $n^{(m)} \times k_c^{(m)}$ is guaranteed not to be congruent to zero (since any field is an integral domain), which guarantees that $k_c^{(m+1)}$ is not zero. The reason for not starting with $n^{(m)} \times k_c^{(m)}$ in the update equation of $k_c^{(m+1)}$ is that this is equal to $C^{(m)}$, which will lead to revealing information about the nonce with the observation of multiple consecutive protocol runs. With the update procedure described above, $n^{(m)} \times k_c^{(m)}$ will be used for updating $k_c^{(m+1)}$ with negligible probability, and even when it is used, the adversary can never know that it is being used. Therefore, without loss of generality, we will assume for the rest of the chapter that equation (9.7) always results in a $k_c^{(m+1)}$ that is uniformly distributed over $\mathbb{Z}_p \backslash \{0\}$.

9.3 Security Analysis

Before we show the security of our UCS-RFID, we will first prove our claims that, under our adversarial model, the integrity of the delivered nonce, n, can be verified using a single modular multiplication, and show that the random nonce is delivered to tags in an unconditionally secure manner.

9.3.1 Integrity of the Delivered Nonce

In this section, we will show how the integrity of the nonce, n, is preserved without resorting to computationally secure cryptographic primitives. The integrity of the delivered nonce in our UCS-RFID is accomplished in a novel way, by taking advantage of the properties of the integer field $\mathbb{Z}_p$, with only a single multiplication operation.

There are two cases to consider: modifying message B alone and modifying both B and C in order to make the tag authenticate a false nonce. Modifying message C alone, since its main purpose is to authenticate the received nonce, does not lead to the extraction of a modified nonce.

Lemma 16. *Given that k_c is uniformly distributed over $\mathbb{Z}_p \backslash \{0\}$, the probability of accepting a modified nonce by a valid tag is at most $1/(p-1)$.*

Proof: Assume that message B has been modified to B'. This modification will lead to the extraction of a nonce, n', different than the authentic n generated by the reader; that is, $n' \equiv B' - k_b \mod p$. Message C, however, is used to verify the integrity of the extracted n'. Let $n' \equiv n + \epsilon \mod p$; for some $\epsilon \in \mathbb{Z}_p \backslash \{0\}$. To be accepted by the tag, n' must satisfy the integrity check of equation (9.3). That is,

$$n' \times k_c = (n + \epsilon) \times k_c = (n \times k_c) + (\epsilon \times k_c) \tag{9.11}$$

$$\overset{?}{\equiv} C \equiv n \times k_c \mod p. \tag{9.12}$$

Clearly, the congruence in equation (9.12) will be satisfied only if $\epsilon \times k_c \equiv 0 \mod p$. However, since k_c is a nonzero element by design, and $\epsilon \not\equiv 0$ (since $\epsilon \equiv 0$ implies that $n' \equiv n \mod p$), the fact that p is prime guarantees that $\epsilon \times k_c \not\equiv 0 \mod p$. Therefore, the congruence of equation (9.12) can never be satisfied, and any modification of message B *alone* will be detected with probability *one*.

The second case to consider here is when both messages B and C are corrupted simultaneously. Assume that message B has been modified so that the extracted nonce becomes $n' \equiv n + \epsilon \mod p$; for some $\epsilon \in \mathbb{Z}_p \backslash \{0\}$. Also, assume that message C has been modified to $C' \equiv C + \delta \mod p$, for some $\delta \in \mathbb{Z}_p \backslash \{0\}$. The integrity of the extracted n' is verified using the received C' as follows:

$$C + \delta \equiv C' \tag{9.13}$$

$$\overset{?}{\equiv} n' \times k_c \equiv (n + \epsilon) \times k_c \equiv (n \times k_c) + (\epsilon \times k_c) \equiv C + (\epsilon \times k_c) \mod p. \tag{9.14}$$

Equivalently, the false n' is accepted only if $\delta \equiv \epsilon \times k_c$. Since k_c is unknown to the adversary, for any fixed δ, since $\mathbb{Z}_p$ is a field, there exists a unique $\epsilon \in \mathbb{Z}_p \backslash \{0\}$ that satisfies (9.14). Therefore, the probability of modifying both B and C in a way undetected by the tag is at most $1/(p-1)$ (equivalently, guessing the value of k_c). ■

9.3.2 Privacy of the Delivered Nonce

Before we show that the nonce is delivered to the tag in an unconditionally secure manner, we need the following lemma.

Lemma 17. *Given that the preloaded subkeys $k_a^{(0)}$, $k_b^{(0)}$, $k_c^{(0)}$, and $k_d^{(0)}$ are mutually independent, the subkeys $k_a^{(m)}$, $k_b^{(m)}$, $k_c^{(m)}$, and $k_d^{(m)}$ at the m^{th} protocol run are mutually independent, for any $m \in \mathbb{N}$.*

Proof: Let $\boldsymbol{k_a^{(m)}}, \boldsymbol{k_b^{(m)}}, \boldsymbol{k_c^{(m)}}$, and $\boldsymbol{k_d^{(m)}}$ be the random variables representing the subkeys involved in the generation of the messages exchanged between an authorized RFID pair during the m^{th} protocol run of our UCS-RFID. Then, for any $k_a^{(1)}, k_b^{(1)}, k_c^{(1)}$, and $k_d^{(1)}$,

$$\begin{aligned}
&\Pr\left(\boldsymbol{k_a^{(1)}} = k_a^{(1)}, \boldsymbol{k_b^{(1)}} = k_b^{(1)}, \boldsymbol{k_c^{(1)}} = k_c^{(1)}, \boldsymbol{k_d^{(1)}} = k_d^{(1)}\right) \\
&= \sum_{n,k_u} \Pr\left(\boldsymbol{k_a^{(1)}} = k_a^{(1)}, \boldsymbol{k_b^{(1)}} = k_b^{(1)}, \boldsymbol{k_c^{(1)}} = k_c^{(1)}, \boldsymbol{k_d^{(1)}} = k_d^{(1)} | \boldsymbol{n} = n, \boldsymbol{k_u} = k_u\right) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \cdot \Pr\left(\boldsymbol{n} = n, \boldsymbol{k_u} = k_u\right) \quad (9.15) \\
&= \sum_{n,k_u} \Pr\left(\boldsymbol{k_a^{(0)}} = k_a^{(1)} \oplus n_r, \boldsymbol{k_b^{(0)}} = (k_b^{(1)} - k_u^{(0)}) \oplus n, \boldsymbol{k_c^{(0)}} = (k_c^{(1)} \times k_u^{(0)^{-1}}) \oplus n\right. \\
&\qquad\qquad\qquad\qquad\qquad\qquad \left., \boldsymbol{k_d^{(0)}} = k_d^{(1)} \oplus n_r\right) \cdot \Pr\left(\boldsymbol{n} = n, \boldsymbol{k_u} = k_u\right) \quad (9.16) \\
&= \sum_{n,k_u} \Pr\left(\boldsymbol{k_a^{(0)}} = k_a^{(1)} \oplus n_r\right) \cdot \Pr\left(\boldsymbol{k_b^{(0)}} = (k_b^{(1)} - k_u^{(0)}) \oplus n\right) \\
&\quad \cdot \Pr\left(\boldsymbol{k_c^{(0)}} = (k_c^{(1)} \times k_u^{(0)^{-1}}) \oplus n\right) \cdot \Pr\left(\boldsymbol{k_d^{(0)}} = k_d^{(1)} \oplus n_r\right) \cdot \Pr\left(\boldsymbol{n} = n, \boldsymbol{k_u} = k_u\right) \quad (9.17) \\
&= \sum_{n,k_u} \frac{1}{2^N} \cdot \frac{1}{p} \cdot \frac{1}{p-1} \cdot \frac{1}{2^N} \cdot \Pr\left(\boldsymbol{n} = n, \boldsymbol{k_u} = k_u\right) \quad (9.18) \\
&= \Pr\left(\boldsymbol{k_a^{(1)}} = k_a^{(1)}\right) \cdot \Pr\left(\boldsymbol{k_b^{(1)}} = k_b^{(1)}\right) \cdot \Pr\left(\boldsymbol{k_c^{(1)}} = k_c^{(1)}\right) \cdot \Pr\left(\boldsymbol{k_d^{(1)}} = k_d^{(1)}\right). \quad (9.19)
\end{aligned}$$

Equations (9.17) and (9.18) hold due to the independence and the uniform distribution of the initial subkeys ($\boldsymbol{k_a^{(0)}}, \boldsymbol{k_b^{(0)}}, \boldsymbol{k_c^{(0)}}, \boldsymbol{k_d^{(0)}}$), respectively; while equation (9.19) holds due to the uniform distribution of the updated subkeys ($\boldsymbol{k_a^{(1)}}, \boldsymbol{k_b^{(1)}}, \boldsymbol{k_c^{(1)}}, \boldsymbol{k_d^{(1)}}$). The existence of $k_u^{(0)^{-1}}$, the multiplicative inverse of $k_u^{(0)}$ in $\mathbb{Z}_p$, is a direct consequence of the fact that $k_u^{(0)} \in \mathbb{Z}_p^*$. The proof of the lemma follows by induction. ■

Lemma 18. *At each instance of the protocol, the random nonce generated by the authorized reader in an instance of our UCS-RFID protocol is delivered to the tag in a perfectly secret manner.*

Proof: Fix k_b, k_c, and let $\boldsymbol{n}$ be uniformly distributed over $\mathbb{Z}_p \backslash \{0\}$. (Recall that, by Lemma 17, $\boldsymbol{k_b}$ and $\boldsymbol{k_c}$ are statistically independent in every protocol run; so, the superscript will be dropped for ease of notation.) Then the resulting $\boldsymbol{B}$ and $\boldsymbol{C}$ will be uniformly distributed over $\mathbb{Z}_p$ and $\mathbb{Z}_p \backslash \{0\}$, respectively. Consequently, for any arbitrary $b \in \mathbb{Z}_p$ and $c \in \mathbb{Z}_p \backslash \{0\}$, the probability of $\boldsymbol{B}$ and $\boldsymbol{C}$ taking these specific values are $\Pr(\boldsymbol{B} = b) = 1/p$ and $\Pr(\boldsymbol{C} = c) = 1/(p-1)$.

Now, given a specific value of the random nonce $\boldsymbol{n} = n$, the probability that $\boldsymbol{B}$ takes a value b is

$$\begin{aligned}\Pr(\boldsymbol{B} = b|\boldsymbol{n} = n) &= \Pr(\boldsymbol{k_b} = b - n) \qquad (9.20)\\ &= 1/p. \qquad (9.21)\end{aligned}$$

Similarly, given a specific value of the random nonce $\boldsymbol{n} = n$, the probability that $\boldsymbol{C}$ takes a value c is

$$\begin{aligned}\Pr(\boldsymbol{C} = c|\boldsymbol{n} = n) &= \Pr(\boldsymbol{k_c} = c \times n^{-1}) \qquad (9.22)\\ &= 1/(p-1). \qquad (9.23)\end{aligned}$$

Equations (9.21) and (9.23) hold since, by design, $\boldsymbol{k_b}$ and $\boldsymbol{k_c}$ are uniformly distributed over $\mathbb{Z}_p$ and $\mathbb{Z}_p\backslash\{0\}$, respectively. The existence of n^{-1} is a direct consequence of the fact that $n \in \mathbb{Z}_p^*$.

Therefore, for any nonce n and any values of b and c, Bayes' theorem [101] can be used to show that $\Pr(\boldsymbol{n} = n|\boldsymbol{B} = b) = \Pr(\boldsymbol{n} = n) = \Pr(\boldsymbol{n} = n|\boldsymbol{C} = c)$. That is, the a priori probabilities that the random nonce is n are the same as the a posteriori probabilities that the random nonce is n given the corresponding B and C. Hence, both B and C "individually" provided perfect secrecy. However, since they are both functions of the same variable, there might be information leakage about n revealed by the *combination* of B and C. One way of measuring how much information is learned by the observation of two quantities is the notion of mutual information. Consider an arbitrary $b \in \mathbb{Z}_p$ and arbitrary $c, n \in \mathbb{Z}_p^*$. Then, for independent $\boldsymbol{k_b}$ and $\boldsymbol{k_c}$ uniformly distributed over $\mathbb{Z}_p$ and $\mathbb{Z}_p^*$, respectively, we get:

$$\begin{aligned}
\Pr(\boldsymbol{B}=b,\boldsymbol{C}=c) &= \sum_n \Pr(\boldsymbol{B}=b,\boldsymbol{C}=c|\boldsymbol{n}=n)\Pr(\boldsymbol{n}=n) && (9.24)\\
&= \sum_n \Pr(\boldsymbol{k_b}=b-n,\boldsymbol{k_c}=c\times n^{-1})\Pr(\boldsymbol{n}=n) && (9.25)\\
&= \sum_n \Pr(\boldsymbol{k_b}=b-n)\Pr(\boldsymbol{k_c}=c\times n^{-1})\Pr(\boldsymbol{n}=n) && (9.26)\\
&= \sum_n \frac{1}{p}\cdot\frac{1}{p-1}\cdot\Pr(\boldsymbol{n}=n) && (9.27)\\
&= \Pr(\boldsymbol{B}=b)\cdot\Pr(\boldsymbol{C}=c). && (9.28)
\end{aligned}$$

Equation (9.26) holds by the independence of $\boldsymbol{k_b}$ and $\boldsymbol{k_c}$, while equations (9.27) and (9.28) hold by the uniform distribution of $\boldsymbol{k_b}$, $\boldsymbol{k_c}$, $\boldsymbol{B}$, and $\boldsymbol{C}$. Consequently, $\boldsymbol{B}$ and $\boldsymbol{C}$ are independent and, thus, their mutual information is *zero* [61]. In other words, observing both messages B and C gives no extra information about n than what they give individually. ■

Remark 11. Lemma 18 does not hold for an adversary who has observed multiple *consecutive* protocol runs between authorized reader-tag pairs. Consider observing three consecutive B messages, say $B^{(0)}, B^{(1)}, B^{(2)}$. The fundamental problem is that only $k_b^{(0)} \in \mathbb{Z}_p$ and $k_u^{(0)} \in \mathbb{Z}_p^*$ are involved in the update equation of the subkey k_b. Therefore, out of the total $(p-1)^3$ possible sequences of $\{n^{(0)}, n^{(1)}, n^{(2)}\}$, to an adversary who has observed $\{B^{(0)}, B^{(1)}, B^{(2)}\}$, there are only $p(p-1)$ possible $\{n^{(0)}, n^{(1)}, n^{(2)}\}$ sequences that could have generated the observed B's; a violation to the definition of perfect secrecy. Obviously, one can include more variables in the update equation of k_b but that will only increase the number of consecutive protocol runs an adversary is allowed to observe to a certain number.

However, breaking perfect secrecy does not imply breaking the system. In what follows, we provide an example to further illustrate the remark; then, we give detailed probabilistic analysis to show that the system can still provide unconditional security given some practical assumptions about the RFID system.

Example 3. This example illustrates the effect of observing consecutive protocol runs between authorized reader-tag pairs. For simplicity, assume that the used prime number is $p = 7$. Assume further that the initial keys $k_b^{(0)} = 2$ and $k_u^{(0)} = 5$ are preloaded into the tag. Consider the first three protocol runs as follows.

First run: let the generated nonce be $n_0 = 1$. Then by equation (9.1) the adversary can observe $B_0 = 3$ broadcasted by the reader. The tag and the reader will then update the keys according to equations (9.6) and (9.9) to $k_b^{(1)} = 1$ and $k_u^{(1)} = 5$.

Second run: let the generated nonce be $n_1 = 6$. Then by equation (9.1) the adversary can observe $B_1 = 0$. The tag and the reader will then update the keys according to equations (9.6) and (9.9) to $k_b^{(2)} = 5$ and $k_u^{(2)} = 2$.

Third run: let the generated nonce be $n_2 = 2$. Then the adversary can observe $B_2 = 0$. Now, with some algebra the adversary can construct the following system of equations:

$$B_0 = k_b^{(0)} + n_0, \tag{9.29}$$

$$B_1 = k_u^{(0)} + (n_0 \oplus k_b^{(0)}) + n_1, \tag{9.30}$$

$$B_2 = (k_u^{(0)} \times n_0) + \Big(n_1 \oplus \big(k_u^{(0)} + (n_0 \oplus k_b^{(0)})\big)\Big) + n_2. \tag{9.31}$$

Consider now the sequence $\{n_0 = 1, n_1 = 1, n_2 = 1\}$. Given the observed B's, by checking equations (9.29), (9.30), and (9.31), one can see that the sequence $\{n_0 = 1, n_1 = 1, n_2 = 1\}$ cannot satisfy the three equations simultaneously. Moreover, by checking all possible $6 \times 6 \times 6$ sequences, one can find that only 7×6 of them can satisfy all three equations simultaneously. The fundamental problem is that only $k_b^{(0)} \in \{0, 1, 2, 3, 4, 5, 6\}$ and $k_u^{(0)} \in \{1, 2, 3, 4, 5, 6\}$ are involved in the three equations.

Observe, however, that this does not imply anything more than that the sequence $\{n_0 = 1, n_1 = 1, n_2 = 1\}$ cannot generate the observed B's. That is, it does not imply that $n_0 \neq 1$, nor that $n_1 \neq 1$, nor that $n_2 \neq 1$. In other words, individually, any one of the n's can be equal to one (indeed, $n_0 = 1$ in the above example).

Therefore, for the adversary to obtain meaningful information, she must know the exact value of at least one of the nonces (so that possible values of other nonces can be eliminated). This can only occur if for at least one nonce n_i, only one value in $\mathbb{Z}_p^*$ is possible. That is, all the possible values n_i is allowed to take can be eliminated, except for exactly one value.

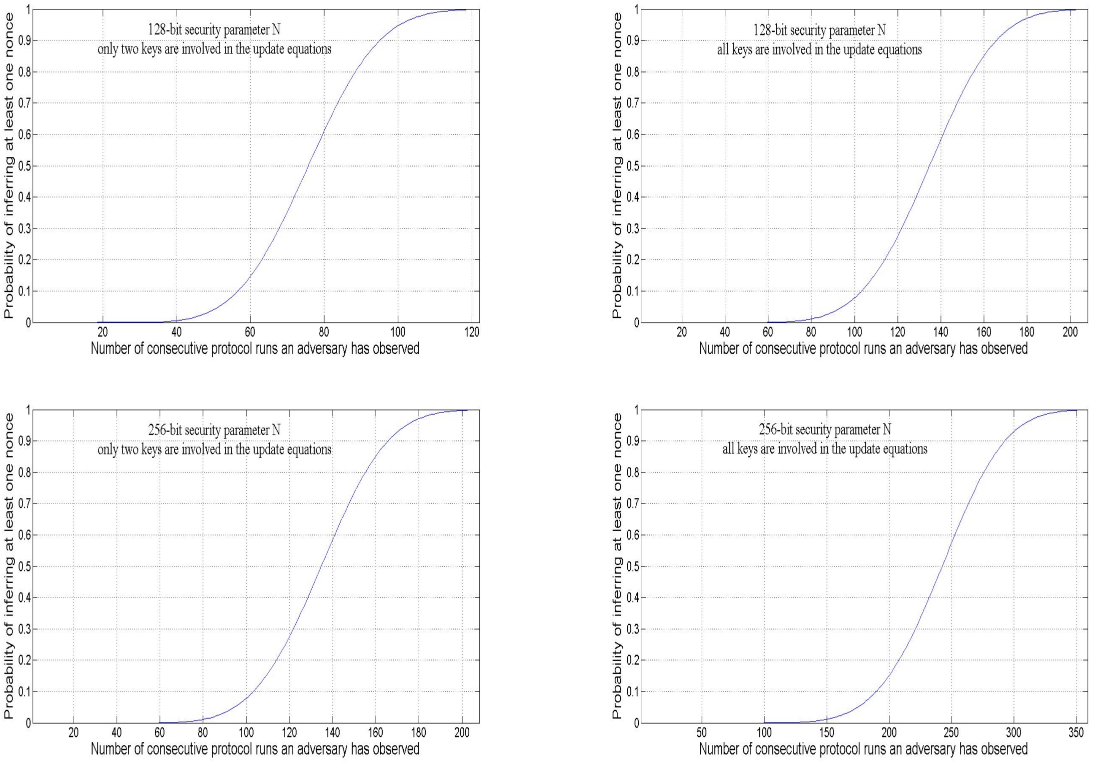


Figure 9.2: The probability of exposing at least one nonce as a function of the number of consecutive protocol runs observed by the adversary, for different size of security parameter and different number of parameters involved in the update equation.

We will now provide probabilistic analysis of the number of consecutive protocol runs an adversary must observe in order to learn the value of at least one of the transmitted nonces.

By the randomness nature of the generated nonces, the total number of possible sequences is uniformly distributed over the nonces. That is, given there are $p(p-1)$ possible sequences, if the adversary has observed m consecutive protocol runs, each of the m nonces is expected to have $\sqrt[m]{p(p-1)}$ possible values. Therefore, for m consecutive protocol runs, the total number of possible values distributed over the m nonces is $m\sqrt[m]{p(p-1)}$.

To give a lower bound on the number of consecutive protocol runs an adversary must observe in order to infer at least one nonce with a certain probability, we use the well-known "balls in bins without capacity" problem in probability theory. Given r balls thrown uniformly at random at m bins, the probability that at least one bin remains empty is given by [78]:

$$\Pr[\text{at least one bin remains empty}] = \frac{\binom{r-1}{m-1}}{\binom{m+r-1}{m-1}}. \tag{9.32}$$

Given that each nonce will take at least one value, the problem reduces to distributing $(m\sqrt[m]{p(p-1)}-m)$ values uniformly at random at m nonces and finding the probability that at least one nonce does not receive another possible value. Substituting $r = m\sqrt[m]{p(p-1)} - m$ in equation (9.32), we plot the results in Figure 9.2. Each plot shows the number of consecutive protocol runs an adversary must observe in order to infer at least one nonce with a certain probability. In the top left plot, the security parameter N is 128-bit long, with only k_b and k_u are involved in the update equation of k_b. The plot in the top right shows the result when all secret keys are involved in the update equation of k_b. The two bottom plots shows the result when the used security parameter is 256-bit long.

As can be seen in Figure 9.2, the number of consecutive protocol runs an adversary has to observe to learn the value of at least one nonce is much higher than the number of protocol runs needed to break perfect secrecy. Depending on how many secret keys are used in the update equations and the length of the security parameter, for an adversary to have a 50% chance of exposing a secret nonce value, the number of *consecutive* protocol runs needed to be observed can be as high as 240 complete runs.

Remark 12. *Observe that, unlike general computer communications, that many consecutive protocol runs can be sufficiently high for RFID systems. Consider, for example, an RFID tag used for a pay-at-the-pump application. If the user goes to the same gas station every single time, this implies that for an adversary to extract secret tag information, she must be in a close proximity to the user for about 240 consecutive gas pumping. In the case in which the user goes to different gas stations, this implies that the adversary is following the user everywhere. Both scenarios are highly unlikely to occur in real life applications. In a different example, consider low-cost tags replacing barcodes for identifying grocery items. In such applications, that many authorized protocol runs are unlikely to occur during the entire life time of a low-cost tag. The following corollary is a direct consequence of this remark.*

Corollary 2. *In order to expose secret tag information, the adversary must observe a* *sufficiently high number of honest protocol runs between authorized reader-tag pairs.*

This implies that adversaries, regardless of their computational power, must rely on authorized reader-tag interactions to have a chance of inferring secret information.

Assumption 1. *For the rest of the chapter, we will adopt the assumption that observing enough protocol runs to expose the value of a nonce is impractical in low-cost RFID systems.*

9.3.3 Security of Mutual Authentication

Before we can state our main theorem regarding the security of mutual authentication in our protocol, we need two more lemmas.

Lemma 19. *Under Assumptions 1, given that the reader generates random nonces, no information about the secret key, K, is revealed by observing protocol runs of the proposed protocol.*

Proof: We start with the basic assumption that the key is loaded to the tag secretly; that is $k_a^{(0)}$, $k_b^{(0)}$, $k_c^{(0)}$, and $k_d^{(0)}$ are secret. By Lemma 18, messages $B^{(0)}$ and $C^{(0)}$ provide perfect secrecy. That is, no information about the nonce, $n^{(0)}$, nor the keys, $k_b^{(0)}$ and $k_c^{(0)}$, will be leaked by $B^{(0)}$ and $C^{(0)}$. Now, $n^{(0)}$ will be used to generate $D^{(0)} = n_\ell^{(0)} \oplus k_d^{(0)}$ and $A^{(1)} = n_\ell^{(0)} + (n_r^{(0)} \oplus k_a^{(0)})$. Since $n^{(0)}$ is delivered in a perfectly secret manner, and $k_d^{(0)}$ and $k_a^{(0)}$ are secret, no information will be revealed by the observation of $D^{(0)}$ and $A^{(1)}$. (The proof is very similar to the proof of Lemma 18; it is based on the fact that $k_a^{(0)}$ and $k_d^{(0)}$ are random and independent.)

So far, no secret information about the initial key $K^{(0)}$ nor the nonce $n^{(0)}$ has been revealed. Therefore, there is no information leakage about the updated subkeys $k_a^{(1)} = n_r^{(0)} \oplus k_a^{(0)}$, $k_b^{(1)} = k_u^{(0)} + (n^{(0)} \oplus k_b^{(0)})$, $k_c^{(1)} = k_u^{(0)} \times (n^{(0)} \oplus k_c^{(0)})$, and $k_d^{(1)} = n_r^{(0)} \oplus k_d^{(0)}$. Given that the keys are updated to remain independent and to have the same distribution as the outdated keys, and that $n^{(1)}$ is random and independent from the previous nonce and from the secret keys, the proof follows by induction (given Assumption 1). ■

Lemma 20. *Under Assumption 1, an adversary making q_q Query oracles, q_s Send oracles will succeed with probability at most*

$$\max\left\{\frac{q_q}{p-1}, \frac{q_s}{2^N}\right\}. \tag{9.33}$$

Proof: By Lemma 19 and Corollary 2, calling the *Execute* oracle a practical number of consecutive times is of no help to the adversary, since no information is leaked by observing messages exchanged between authorized RFID pairs.

On the other hand, an adversary calling the *Query* oracle will receive A as the tag's response. Depending on the adversary's response, the tag will respond with message D with probability $1/(p-1)$ (the probability of successful forgery by Lemma 16), or abort the protocol with probability $(p-2)/(p-1)$. If the tag does respond, the protocol is considered broken. However, upon unsuccessful forgery, the tag will abort, and responds to the next *Query* with the same identifier, A. Therefore, no information about the tag's secret key is revealed by multiple *Query* calls.

Finally, an adversary calling the *Send* oracle to impersonate a valid tag will be successful with probability at most $1/2^N$. This is due to the fact that A might or might not be a valid tag identifier. If it is not, the reader will abort the protocol. Assume, however, that A is a valid identifier (the adversary can obtain a valid one by interrogating a tag in the system). An authorized reader, responding with B and C, will accept D if and only if $D = n_\ell \oplus k_d$. To extract the correct n_ℓ, however, the adversary must know k_b or k_c, which are kept secret by Lemma 19. Moreover, k_d is unknown to the adversary (also by Lemma 19). Hence, the adversary's probability of success is $1/2^N$, and the lemma follows. ■

Given that p is a $2N$-bit prime integer, the adversary's probability of falsely authenticating herself to a valid tag is at most $1/2^{2N-1}$, and the adversary's probability of authenticating herself to a valid reader is $1/2^N$. That is, the probability of mutual authentication when the protocol is not honest is negligible in the security parameter N. We can now state our main theorem.

Theorem 12. *Under Assumption 1, the proposed UCS-RFID is a secure mutual authentication protocol for RFID systems.*

Proof: Lemma 19 implies that the first condition of Definition 7 is satisfied. The second condition of Definition 7 can be easily verified; it merely means that if the messages exchanged between legitimate RFID pairs are relayed faithfully to one another, mutual authentication is achieved. The third condition of Definition 7 is shown to be satisfied in Lemma 20. Thus, all three conditions of Definition 7 are satisfied. ■

9.4 Desynchronization Analysis

Observe that the analysis of Section 9.3 does not include the adversarial ability to block messages exchanged in parts of the protocol. In this section, we extend the analysis to include the *Block* oracle introduced in Section 8.4.2 and discuss the possible attacks that can be launched by blocking exchanged messages along with their corresponding mitigations.

9.4.1 Desynchronization Attacks

As in stateful protocols, the update procedure is critical for the security and the correctness of the protocol. Without a valid identifier and a valid set of keys, tags cannot be identified successfully in the proposed protocol. Consequently, not only the secrecy and authenticity of tags parameters must be preserved, but also tags' states must be synchronized with the database. That is, if the parameters of a certain tag in the database are different than the parameters stored at the tag itself, the tag cannot be identified successfully by authorized readers. Therefore, it is important to show that tags in the proposed system are secured against possible desynchronization attacks.

From Theorem 12, an adversary cannot cause a desynchronization between tags and the database with a non-negligible probability by authenticating herself to the tag or the reader. The protocol as described in Section 9.2, however, allows an adversary to mount a desynchronization attack by blocking some messages exchanged between authorized reader-tag pairs. In order to formalize such attacks, we need the *Block* oracle defined in Section 8.4.2

In what follows, we investigate the effect of blocking different messages in a protocol run and then give an extension to the protocol description of Section 9.2.

1. *Block*(“*Hello*”): An adversary blocking the first protocol message, the “Hello” message from the reader to the tag, will cause the tag not to respond. That is, neither the tag nor the database will update the tag’s state. Therefore, no desynchronization will occur by blocking the first protocol message.

2. *Block*(A): Consider an adversary blocking the second message containing the tag’s identifier A. Again, neither the tag nor the database will update the tag’s parameters and no desynchronization will occur. An adversary replaying message A to the reader will cause no harm either. To see this, recall that the adversary does not know k_b, k_c nor k_d, hence, she cannot extract the correct n and generate a valid D with a non-negligible probability.

3. *Block*(B, C): Consider an adversary blocking messages B and C, and replaying them to the tag. Of course, the adversary will be authenticated. However, this is considered as faithfully relaying messages, which does not affect the honesty of the protocol. This makes sense, because the tag will respond with a message D which does not reveal extra information about the tag that has not been revealed by A.

4. *Block*(D): Consider an adversary blocking message D sent to the reader. The reader will assume that the tag has not updated its parameters while, in fact, it has. Consequently, the secret keys at the tag’s side will be different than the secret keys stored at the database, causing a possible desynchronization between the tag and the reader. A solution to this problem is that the reader updates the parameters even if it does not receive message D from the tag. The reader, however, must store both the updated and the outdated parameter values at the database to count for the possible scenario that the tag has not updated its parameters (more details in Section 9.5).

9.4.2 Key Exposure

A more dangerous attack can be launched by blocking messages B and C sent to the tag, or message D sent to the reader, if the *same* keys, with *different* nonces are used in the next protocol run. To see this, assume that $B^{(1)} \equiv n^{(1)} + k_b^{(1)} \mod p$ and $C^{(1)} \equiv n^{(1)} \times k_c^{(1)} \mod p$ have been blocked by an active adversary. If the same keys $k_b^{(1)}$ and $k_c^{(1)}$ are used to generate $B^{(2)} \equiv n^{(2)} + k_b^{(1)} \mod p$ and $C^{(2)} \equiv n^{(2)} \times k_c^{(1)} \mod p$, the difference between the two nonces, $n^{(1)}$ and $n^{(2)}$, is simply the difference between $B^{(1)}$ and $B^{(2)}$. It can be easily seen that:

$$\begin{aligned} C^{(2)} &\equiv n^{(2)} \times k_c^{(1)} \mod p && (9.34)\\ &\equiv (n^{(1)} + \delta) \times k_c^{(1)} \mod p && (9.35)\\ &\equiv (n^{(1)} \times k_c^{(1)}) + (\delta \times k_c^{(1)}) \mod p && (9.36)\\ &\equiv C^{(1)} + (\delta \times k_c^{(1)}) \mod p. && (9.37) \end{aligned}$$

Hence, with the knowledge that $n^{(2)} \equiv n^{(1)} + \delta \mod p$, where $\delta \equiv B^{(2)} - B^{(1)} \mod p$, the value of $k_c^{(1)}$ can be easily computed as $k_c^{(1)} \equiv (C^{(2)} - C^{(1)}) \times \delta^{-1} \mod p$. Thus, we emphasize that whenever the reader receives an outdated identifier, the reader retransmits the same messages $B^{(1)}$ and $C^{(1)}$, as opposed to generating a new nonce and transmitting $B^{(2)}$ and $C^{(2)}$ as above.

The requirement that the reader responds with the same B and C when receiving an outdated A, however, introduces a vulnerability to a man-in-the-middle (MITM) attack. Consider an adversary observing messages $A^{(1)}, B^{(1)}, C^{(1)}$, and then intercepting message $D^{(1)}$. The reader will assume that the tag has not updated its parameter. Hence, in the next protocol run, the adversary can impersonate the tag by sending its $A^{(1)}$ and, upon receiving the same $B^{(1)}$ and $C^{(1)}$, she can replay the intercepted $D^{(1)}$, which will be accepted by the reader.

Fortunately, there is an easy fix for this vulnerability. Whenever a valid reader receives an outdated identifier $A^{(1)}$, it responds with the same $B^{(1)}$ and $C^{(1)}$ to avoid key exposure (as discussed above). But the tag does *not* get authenticated upon the reception of $D^{(1)}$ (to avoid the man-in-the-middle attack described above). The reader continues by carrying out another protocol run with the tag (with updated keys this time), and *only* if the second authentication run is passed, with updated parameters to generate $A^{(2)}, B^{(2)}, C^{(2)}$, and $D^{(2)}$, the tag is authenticated. Below, we extend the basic description of our protocol to take into account the desynchronization and key exposure attacks described above.

9.5 The Complete UCS-RFID Description

As discussed in the previous section, the basic protocol description of Section 9.2 is vulnerable to desynchronization and key recovery attacks. We she below an extension the basic protocol description that is secure against the desynchronization and key recovery attacks illustrated in the previous section.

Step 1. The reader announces its presence by broadcasting a *"Hello"* message.

Step 2. The tag responds to the *"Hello"* message by sending its current identifier, A.

Step 3. The reader looks up the database for the key $K = (k_a, k_b, k_c, k_d, k_u)$ corresponding to the tag's current identifier, A. If A is not recognized as a valid identifier, the tag is rejected.

Step 4. There are two possible scenarios if A is identified: 1. A can be updated (corresponding to the case in which the tag has updated its parameters successfully during its last protocol run) and, 2. A can be outdated (corresponding to the case in which the tag has not updated its parameters during its last protocol run, while the database has).

4.1. If A is updated, the reader generates a $2N$-bit random nonce, n, drawn uniformly at random from the multiplicative group $\mathbb{Z}_p^*$. The reader also deletes the outdated information of the tag from the database, if existed.

4.2. If A is outdated, the reader uses the same nonce n used for the last protocol run with the tag.

Step 5. With k_b, k_c, and n, the reader broadcasts two messages, B and C, generated according to the following formulas:

$$B \equiv n + k_b \mod p, \tag{9.38}$$

$$C \equiv n \times k_c \mod p. \tag{9.39}$$

Again, there are two possible scenarios.

5.1. If A was updated, the reader updates the tags parameters. Let $A^{(m)}$, $k_i^{(m)}$, and $n^{(m)}$ denote the identifier A, k_i, and n used to execute the m^{th} protocol run; let n_r denotes the N least significant bits of n. Then, the parameters are updated as follows,

$$k_a^{(m+1)} = n_r^{(m)} \oplus k_a^{(m)}, \tag{9.40}$$

$$k_b^{(m+1)} \equiv k_u^{(m)} + (n^{(m)} \oplus k_b^{(m)}) \mod p, \tag{9.41}$$

$$k_c^{(m+1)} \equiv k_u^{(m)} \times (n^{(m)} \oplus k_c^{(m)}) \mod p, \tag{9.42}$$

$$k_d^{(m+1)} = n_r^{(m)} \oplus k_d^{(m)}, \tag{9.43}$$

$$k_u^{(m+1)} \equiv k_u^{(m)} \times n^{(m)} \mod p, \tag{9.44}$$

$$A^{(m+1)} \equiv n_\ell^{(m)} + k_a^{(m+1)} \mod 2^N. \tag{9.45}$$

5.2. If A was outdated, the reader does nothing (since the parameters have already been updated during the last protocol run).

Step 6. Upon receiving B and C, the tag extracts n from message B and verifies its integrity using message C. The reader is authenticated if and only if the following integrity check is satisfied,

$$(B - k_b) \times k_c \equiv C \mod p. \tag{9.46}$$

If the integrity check of equation (9.46) does not pass, the reader will not be authenticated and the tag will abort the protocol. (This is an example where the tag's identifier used in the next protocol run will be outdated.)

Step 7. If the reader is authenticated, the tag broadcasts message D, given by

$$D = n_\ell \oplus k_d, \tag{9.47}$$

where n_ℓ denotes the N most significant bits of n (the subscript ℓ refers to the ℓeft half of n). The tag then updates its parameters according to equations (9.40)-(9.45).

Step 8. Upon receiving D, the reader authenticates the tag by verifying that the received D is equal to $n_\ell \oplus k_d$. If D does not pass the validity check, the tag is rejected. If D passes the validity check, there are two possible scenario.

8.1. If A was updated, the tag is authenticated.

8.2. If A was outdated, the reader goes back to Step 1 to start another protocol run with the tag.

9.6 Privacy of RFID Tags

In this section we analyze the level of privacy achieved by UCS-RFID according to the privacy definitions of Section 8.4.3. Consider a tag T_a that has been interrogated by the adversary to get its identifier $A_a^{(i)}$.

Lemma 21. *Under the adversarial and security models of Section 8.4, tags executing the UCS-RFID protocol are universally untraceable.*

Proof: Assume the tag has accomplished mutual authentication with an authorized reader. Its identifier $A_a^{(i)}$ is updated according to equation (9.10) to $A_a^{(i+1)} = n_\ell^{(i)} + k_a^{(i)}$, where both $n_\ell^{(i)}$ and $k_a^{(i)}$ are unknown, random strings. That is, according to Theorem 12, the adversary cannot correctly predict the value of $A_a^{(i+1)}$ with a non-negligible probability. Therefore, given the tag, T_a, with identifier $A_a^{(i+1)}$, and another tag, T_b, with identifier A_b, the adversary can do no better than a random guess. That is, $\Pr(T_g = T_a) = 0.5$, and the adversary's advantage of tracking the tag is $\mathsf{Adv}_{\mathcal{A}} = 2\Big(\Pr(T_g = T_a) - 0.5\Big) = 0$. Consequently, tags using our protocol are passively private. ■

Lemma 22. *Under the adversarial and security models of Section 8.4, tags executing the UCS-RFID protocol are not existentially untraceable.*

Proof: With the absence of authorized readers, tags are unable to update their parameters. Hence, an adversary interrogating the same tag multiple times will receive the same response, and the adversary's advantage of recognizing the tag is *one*. Consequently, tags using our UCS-RFID for mutual authentication are not actively private. ■

The case of tracking tags by their other responses can be handled similarly. Section 9.6.1 addresses the problem of active privacy and discuss possible solutions.

9.6.1 Tag Probing Attack

In the tag probing attack, a rogue reader probes the tag by sending a "*Hello*" message. The tag challenges the reader by revealing its identifier, A, and the reader replies with false authentication information. Given that the authentication fails, the tag will not update its keying information and its identifier. Hence, the tag will respond with the same A on every "*Hello*" message after every false authentication attempt.

This is a common issue shared by all stateful RFID protocols. The fundamental problem here is that tags are identified via their states, the identifiers in the proposed protocol. To prevent illegal tag tracking, the state must be updated. Since the state in both the tag and the database must be the same, to enable identification, the tag cannot update its identifier in a way unrecognized by valid readers. Consequently, active adversaries interrogating the same tag multiple times, without successfully completing the protocol run, will be able to correlate the tag's responses. Thus, leading to the ability to illegally track RFID tags.

Using the tag probing attack, the adversary can track the tag and the user that carries it, until a successful authentication is performed with an authorized reader. The problem of active privacy, however, is very challenging in RFID protocols based on symmetric-key cryptography (see, e.g., [181, 49, 265, 58, 9] for references addressing this issue). Most existing protocols that provide active privacy require readers to perform linear search of all tags in the system in order to identify every single tag response [25]. (Although out of the scope of this chapter, the identification process can be performed more efficiently in this protocol since identifiers are broadcasted in clear text.) Given the stringent computational power of tags in this protocol, we discuss the following non-cryptographic techniques to mitigate the tag probing vulnerability.

In [130], Juels et al. introduced the idea of a blocker tag. The blocker tag can simulate many ordinary RFID tags simultaneously to enhance users' privacy. In [82], Floerkemeier et al. proposed the use of a watchdog tag. The watchdog tag enables users to be aware of their tags being interrogated. In [215], Rieback et al. proposed the idea of using an RFID guardian. The RFID guardian is a battery powered device that protects tags from being illegally scanned. In [131], Juels et al. proposed the idea of using an RFID Enhancer Proxy (REP) to enhance the privacy of RFID tags.

9.7 Summary

In this chapter, a new direction into the problem of authenticating low-cost RFID systems is proposed. The aim of this chapter was to investigate the possibilities of unconditional security in the design of RFID protocols. An instance of such protocols was proposed. Under a restriction on the number of consecutive protocol runs an adversary is assumed to observe, the proposed protocol is shown to achieve unconditional secrecy and unconditional integrity. The main goal of this new approach is to design secure RFID protocols with minimum hardware requirements to meet the demand of secure low-cost RFID systems.

Chapter 10

SECURING RFID SYSTEMS: A COMPUTATIONALLY SECURE APPROACH

In the previous chapter, we proposed an unconditionally secure approach to address the privacy and authenticated identification in RFID systems. The security of the UCS-RFID system of the previous chapter, however, relies on the assumption that the adversary cannot observe a large number of consecutive protocol runs. Since this assumption might be impractical in some application, we propose here two approaches for authenticated message exchange suitable for low-cost devices that are provably secure without the above assumption.

10.1 Message Authentication Codes and Pervasive Computing

There are two important observations to make about standard MAC algorithms. First, they are designed independently of any other operations required to be performed on the message to be authenticated. For instance, if the authenticated message must also be encrypted, existing MACs are not designed to utilize the functionalities that can be provided by the underlying encryption algorithm. Second, most existing MACs are designed for the general computer communication systems, independently of the properties that messages can possess. For example, one can find that most existing MACs are inefficient when the messages to be authenticated are short. (For instance, UMAC, the fastest reported message authentication code in the cryptographic literature [249], has undergone large algorithmic changes to increase its speed on short messages [151].)

Nowadays, however, there is an increasing demand for the deployment of networks consisting of a collection of small devices. In many practical applications, the main purpose of such devices is to communicate short messages. A sensor network, for example, can be deployed to monitor certain events and report some collected data. In many sensor network applications, reported data consist of short confidential measurements. Consider, for instance, a sensor network deployed in a battlefield with the purpose of reporting the existence of moving targets or other temporal activities. In such applications, the confidentiality and integrity of reported events are of critical importance [5, 204, 203].

In another application, consider the increasingly spreading deployment of radio frequency identification (RFID) systems. In such systems, RFID tags need to identify themselves to authorized RFID readers in an authenticated way that also preserves their privacy. In such scenarios, RFID tags usually encrypt their identity, which is typically a short string, to protect their privacy. Since the RFID reader must also authenticate the identity of the RFID tag, RFID tags must be equipped with a message authentication mechanism [221, 128, 199].

Another application that is becoming increasingly important is the deployment of body sensor networks. In such applications, small sensors can be embedded in the patient's body to report some vital signs. Again, in some applications the confidentiality and integrity of such reported messages can be important [277, 252, 244].

There have been significant efforts devoted to the design of hardware efficient implementations that suite such small devices. For instance, hardware efficient implementations of block ciphers have been proposed in, e.g., [75, 163, 116, 153, 45, 171]. Implementations of hardware efficient cryptographic hash functions have also been proposed in, e.g., [192, 227, 46, 141]. However, there has been little or no effort in the design of special algorithms that can be used for the design of message authentication codes that can utilize other operations and the special properties of such networks. In this chapter, we provide the first such work.

In this chapter, we propose two new techniques for authenticating short encrypted messages that are more efficient than existing approaches. In the first technique, we utilize the fact that the message to be authenticated is also encrypted, with any secure encryption algorithm, to append a short random string to be used in the authentication process. Since the random strings used for different operations are independent, the authentication algorithm can benefit from the simplicity of unconditional secure authentication to allow for faster and more efficient authentication, without the difficulty to manage one-time keys. In the second technique, we make the extra assumption that the used encryption algorithm is block cipher based to further improve the computational efficiency of the first technique.

The rest of the chapter is organized as follows. In Section 10.2 we describe the first authentication technique assuming messages do not exceed a maximum length, discuss its performance advantages over existing techniques, and prove its security. In Section 10.3, we propose a modification to the scheme of Section 10.2 that provides a stronger notion of integrity. In Section 10.4, we describe the second technique assuming the encryption is block cipher based, discuss its performance, and prove its security. In Section 10.5, we summarize the chapter.

10.2 Authenticating Short Encrypted Messages

In this section, we describe our first authentication scheme that can be used with any IND-CPA secure encryption algorithm. An important assumption we make is that messages to be authenticated are no longer than a predefined length. This includes applications in which messages are of fixed length that is known a priori, such as RFID systems in which tags need to authenticate their identifiers, sensor nodes reporting events that belong to certain domain or measurements within a certain range, etc. The approach of this section can be viewed as a special case of the $\mathcal{E}$-MAC scheme proposed in Chapter 4 in which authenticated message are of a pre-defined length.

10.2.1 The Proposed System

The proposed scheme is based on the encode-then-encrypt principle of Bellare and Rogaway [36]. Let $N-1$ be an upper bound on the length, in bits, of exchanged messages. That is, messages to be authenticated can be no longer than $(N-1)$-bit long. Choose p to be an N-bit long prime integer. (If N is too small to provide the desired security level, p can be chosen large enough to satisfy the required security level.) Choose an integer k_s uniformly at random from the multiplicative group $\mathbb{Z}_p^*$; k_s is the secret key of the scheme. The prime integer, p, and the secret key, k_s, are distributed to legitimate users and will be used for message authentication. Note that the value of p need not be secret, only k_s is secret.

Let $\mathcal{E}$ be any IND-CPA secure encryption algorithm. Let m be a short messages ($N-1$ bit or shorter) that is to be transmitted to the intended receiver in a confidential manner (by encrypting it with $\mathcal{E}$). Instead of authenticating the message using a traditional MAC algorithm, consider the following procedure. On input a message m, a random nonce $r \in \mathbb{Z}_p$ is chosen. (We overload m to denote both the binary string representing the message, and the integer representation of the message as an element of $\mathbb{Z}_p$. The same applies to k_s and r. The distinction between the two representations will be omitted when it is clear from the context.) We assume that integers representing distinct messages are also distinct, which can be achieved by appropriately encoding messages [42].

Now, r is appended to the message and the resulting $m \parallel r$, where "$\|$" denotes the concatenation operation, goes to the encryption algorithm as an input. Then, the authentication tag of message m can be calculated as follows:

$$\tau \equiv mk_s + r \pmod{p}. \tag{10.1}$$

Remark 13. *We emphasize that the nonce, r, is generated internally and is not part of the chosen message attack. In fact, r can be thought of as a replacement to the coin tosses that can be essential in many MAC algorithms. In such a case, the generation of r imposes no extra overhead on the authentication process. We also point out that, as opposed to one-time keys, r needs no special key management; it is delivered to the receiver as part of the encrypted ciphertext.*

Since the generation of pseudorandom numbers can be considered expensive for computationally limited devices, there have been several attempts to design true random number generators that are suitable for RFID tags (see, e.g., [165, 114, 115]) and for low-cost sensor nodes (see, e.g., [205, 52, 83]). Thus, we assume the availability of such random number generators.

Now, the ciphertext $c = \mathcal{E}(m||r)$ and the authentication tag τ, computed according to equation (10.1), are transmitted to the intended receiver.

Upon receiving the ciphertext, the intended receiver decrypts it to extract m and r. Given τ, the receiver can check the validity of the message by performing the following integrity test:

$$\tau \overset{?}{\equiv} mk_s + r \pmod{p}. \tag{10.2}$$

If the integrity check of equation (10.2) is satisfied, the message is considered authentic. Otherwise, the integrity of the message is denied.

Note, however, that the authentication tag is a function of the confidential message. Therefore, the authentication tag must not reveal information about the plaintext since, otherwise, the confidentiality of the encryption algorithm is compromised. Before we give formal security analysis of the proposed technique, we first discuss its performance compared to existing techniques.

10.2.2 Performance Discussion

As discussed in Section 3.1, there are three classes of standard message authentication codes (MACs) that can be used to preserve message integrity in mobile and pervasive computing. One can use a MAC based on block ciphers, a MAC based on cryptographic hash functions, or a MAC based on universal hash-function families. Since MACs based on universal hashing are known to be more computationally-efficient than MACs based on block ciphers and cryptographic hash function [249], we focus on comparing the proposed MAC to universal hash functions based MACs.

In MACs based on universal hashing, two phases of computations are required: 1. a message compression phase using a universal hash function and, 2. a cryptographic phase in which the compressed image is processed with a cryptographic primitive (a block cipher or a cryptographic hash function). The compression phase is similar to the computation of equation (10.1) of the proposed MAC (in fact, the proposed MAC of equation (10.1) is an instance of strongly universal hash functions). As opposed to standard universal hash functions based MACs, however, there is no need to process the the result of equation (10.1) with a cryptographic function in the proposed technique.

When the messages to be authenticated are short, the modulus prime, p, can also be small. For a small modulus, the modular multiplication of equation (10.1) is not a time consuming operation. That is, for short messages, the cryptographic phase is the most time consuming phase. Since we target applications in which messages are short, eliminating the need to perform such a cryptographic operation will have a significant impact on the performance of the MAC operation. For instance, while the cryptographic hash function SHA-256 hashes at around 21 cycles/byte [184], the modular multiplication of equation (10.1) runs in about 1.5 cycles/byte [42], which illustrates the significance of removing the cryptographic phase from our MAC.

Another advantage of the proposed method is hardware efficiency. The hardware required to perform modular multiplication is less than the hardware required to perform sophisticated cryptographic operations. This advantage is particularly important for low-cost devices.

Compared to single-pass authenticated encryption algorithms, when combined with a stream cipher, the technique of Section 10.2.1 will be much faster (recall that all secure single-pass authenticated encryption methods are block cipher based). Furthermore, our construction is an instance of the encrypt-and-authenticate (E&M) generic composition. That is, the encryption and authentication operations can be performed in parallel. If the underlying encryption algorithm is a block cipher based, the time to complete the entire operation will be the time it takes for encryption only. Even with the added time to encrypt the nonce, which depending on the length of r and the size of the block cipher might not require any additional block cipher calls, single-pass authenticated encryption methods typically require at least two additional block cipher calls.

10.2.3 Security Analysis

In this section, we prove the confidentiality of the system, give a formal security analysis of the proposed message authentication mechanism, and then discuss the security of the composed authenticated encryption system. The security model that will be used for security analysis is the same security model of Chapter 4.

10.2.3.1 Data Privacy

We show in this section that the privacy of the proposed compositions is provably secure assuming the underlying encryption algorithm provides indistinguishability under chosen plaintext attacks (IND-CPA). Let Σ denote the proposed authenticated encryption composition described in Section 10.2.1. Let $\mathcal{A}$ be an adversary against the privacy of Σ and let $\mathsf{Adv}_{\Sigma}^{\text{priv}}(\mathcal{A})$ denote adversary's $\mathcal{A}$ advantage in breaking the privacy of the system, where the privacy of the system is modeled as its indistinguishability under chosen plaintext attacks. One gets the following theorem.

Theorem 13. *Let Σ be the authenticated encryption composition described in Section 10.2.1 using $\mathcal{E}$ as the underlying encryption algorithm. Then given an adversary, $\mathcal{A}$, against the privacy of Σ, one can construct an adversary, $\mathcal{B}$, against $\mathcal{E}$ such that*

$$\mathsf{Adv}_{\Sigma}^{\text{priv}}(\mathcal{A}) \leq \mathsf{Adv}_{\mathcal{E}}^{\text{ind-cpa}}(\mathcal{B}).$$

Theorem 13 states that an adversary breaking the privacy of the proposed system will also be able to break the IND-CPA of the underlying encryption algorithm. Therefore, if $\mathcal{E}$ provides IND-CPA, the adversary's advantage of exposing private information about the system is negligible. Note that private information here refers not only to the encrypted messages, but also the secret key, k_s, as well as the secret key of the encryption algorithm. Observe that, unlike Chapter 9, no specific protocol is proposed in this chapter. That is, the proposed idea can be used to construct different protocols. Therefore, while Theorem 13 implies universal untraceability, forward and existential untraceabilities will depend on the specifics of the identification protocol.

Proof: [Proof of Theorem 13] Recall that each authentication tag, τ, computed according to equation (10.1) requires the generation of a random nonce, r. Recall further that r is generated internally and is not part of the chosen message attack. Now, if r is delivered to the receiver using a secure channel (e.g., out of band), then equation (10.1) is an instance of a perfectly secret (in Shannon's information theoretic sense) one-time pad cipher (encrypted with the one-time key r) and, hence, no information will be exposed. However, the r corresponding to each tag is delivered via the ciphertext. Therefore, the only way to expose private information is from the ciphertext.

Assume now that $\mathcal{A}$ is an adversary against the privacy of the system proposed in Section 10.2.1. Let $\mathcal{B}$ be an adversary with access oracle to the encryption algorithm $\mathcal{E}$ and let adversary $\mathcal{A}$ use adversary $\mathcal{B}$ to attack the privacy of observed ciphertexts. Then,

$$\mathsf{Adv}_{\Sigma}^{\text{priv}}(\mathcal{A}) \leq \mathsf{Adv}_{\mathcal{E}}^{\text{ind-cpa}}(\mathcal{B})$$

and the theorem follows. ■

10.2.3.2 Data Authenticity

We can now proceed with the main theorem formalizing the adversary's advantage of successful forgery against the proposed scheme. As before, let Σ denotes the proposed authenticated encryption composition of Section 10.2.1 and let $\mathsf{Adv}_{\Sigma}^{\text{auth}}(\mathcal{A})$ denotes adversary's $\mathcal{A}$ advantage of successful forgery against Σ.

Theorem 14. *Let Σ denotes the proposed authenticated encryption composition of Section 10.2.1 in which the authentication tag is computed over the the finite integer field $\mathbb{Z}_p$. Let $\mathcal{A}$ be an adversary making a q signing queries before attempting its forgery. Then, one can come up with an adversary, $\mathcal{B}$, against the IND-CPA security of the underlying encryption algorithm, $\mathcal{E}$, such that*

$$\mathsf{Adv}_{\Sigma}^{\text{auth}}(\mathcal{A}) \leq \mathsf{Adv}_{\mathcal{E}}^{\text{ind-cpa}}(\mathcal{B}) + \frac{1}{p-1}.$$

Theorem 14 states that if the adversary's advantage in breaking the IND-CPA security of the underlying encryption algorithm as negligible, then so is her advantage in breaking the integrity of the scheme. That is, the integrity of the scheme of Section 10.2.1 is provably secure provided the underlying encryption algorithm is IND-CPA secure.

Proof: [Proof of Theorem 14] Assume an adversary calling the signing oracle for q times and recording the sequence

$$\mathsf{Seq} = \Big\{(m_1, \tau_1), \cdots, (m_q, \tau_q)\Big\} \tag{10.3}$$

of message-tag pairs. We aim to bound the probability that an (m, τ) pair of the adversary's choice will be accepted as valid, where $(m, \tau) \neq (m_i, \tau_i)$ for any $i \in \{1, \cdots, q\}$, since otherwise the adversary does not win by definition.

Let $m \equiv m_i + \epsilon \pmod{p}$ for any $i \in \{1, \cdots, q\}$, where ϵ can be any function of the recorded values. Similarly, let $r \equiv r_i + \delta \pmod{p}$, where δ is any function of the recorded values (r here represents the value of the coin tosses extracted by the legitimate receiver after decrypting the ciphertext). Assume further that the adversary knows the values of ϵ and δ. Then,

$$\begin{aligned} \tau &\equiv mk_s + r \pmod{p} && (10.4) \\ &\equiv (m_i + \epsilon)k_s + (r_i + \delta) \pmod{p} && (10.5) \\ &\equiv \tau_i + \epsilon k_s + \delta \pmod{p}. && (10.6) \end{aligned}$$

Therefore, for (m, τ) to be validated, τ must be congruent to $\tau_i + \epsilon k_s + \delta$ modulo p. Now, by Theorem 13, k_s will remain secret as long as the adversary does not break the IND-CPA security of the encryption algorithm. Hence, by Lemma 1, the value of ϵk_s is an unknown value uniformly distributed over the multiplicative group $\mathbb{Z}_p^*$ (observe that ϵ cannot be the zero element since, otherwise, m will be equal to m_i). Therefore, unless the adversary can break the IND-CPA security of the underlying encryption algorithm, her advantage of successful forgery is $1/(p-1)$ for each verify query, and the theorem follows. ■

Remark 14. *Observe that, if both k_s and r are used only once (i.e., one-time keys), the authentication tag of equation (10.1) is a well-studied example of a strongly universal hash family (see [243] for a definition of strongly universal hash families and detailed discussion showing that equation (10.1) is indeed strongly universal hash family). The only difference is that we restrict k_s to belong to the multiplicative group modulo p, whereas it can be equal to zero in unconditionally secure authentication. This is because, in unconditionally secure authentication, the keys can only be used once. In our technique, since k_s can be used to authenticate an arbitrary number of messages, it cannot be chosen to be zero. Otherwise, mk_s will always be zero and the system will not work. The novelty of our approach is to utilize the encryption primitive to reach the simplicity of unconditionally secure authentication, without the need for impractically long keys.*

Note also that, unless further assumptions about the encryption algorithm is assumed (such as the pseudorandom permutation property as in Section 10.4), it is critical for the security of authentication to perform the multiplication modulo a prime integer. That is, as shown in Chapter 7, the security of authentication based on universal hash families similar to the one in equation (10.1) is proportional to the reciprocal of the smallest prime factor of the used modulus.

10.3 From Weak to Strong Unforgeability

As per [34], there are two notions of unforgeability in authentication codes. Namely, a MAC algorithm can be weakly unforgeable under chosen message attacks (WUF-CMA), or strongly unforgeable under chosen message attacks (SUF-CMA). A MAC algorithm is said to be SUF-CMA if, after launching chosen message attacks, it is infeasible to forge a message-tag pair that will be accepted as valid regardless of whether the message is "new" or not, as long as the tag has not been previously attached to the message by an authorized user. If it is only hard to forge valid tags for "new" messages, the MAC algorithm is said to be WUF-CMA.

The authentication code, as described in Section 10.2, is only WUF-CMA. To see this, let $\mathcal{E}$ works as follows. On input a message m, generate a random string s, compute $PRF_x(s)$, where PRF_x is a pseudorandom function determined by a secret key x, and transmit $c = (s, PRF_x(s) \oplus m)$ as the ciphertext. Then, $\mathcal{E}$ is an IND-CPA secure encryption. Applied to our construction, on input a message m, the ciphertext will be $c = \big(s, PRF_x(s) \oplus (m||r)\big)$ and the corresponding tag will be $\tau \equiv mk_s + r \pmod{p}$. Now, let s' be a string of length equal to the concatenation of m and r. Then, $c' = \big(s, PRF_x(s) \oplus (m||k) \oplus s'\big) = \big(s, PRF_x(s) \oplus (m||k \oplus s')\big)$. Let s' be a string of all zeros except for the least significant bit, which is set to one. Then, either $\tau_1 \equiv mk_s + r + 1 \pmod{p}$ or $\tau_2 \equiv mk_s + r - 1 \pmod{p}$ will be a valid tag for m, when c' is transmitted as the ciphertext. That is, the same message can be authenticated using different tags with high probabilities.

While WUF-CMA can be suitable for some applications, it can also be inadequate for other applications. Consider RFID systems, for instance. If the message to be authenticated is the tag's fixed identity, then WUF-CMA allows the authentication of the same identity by malicious users. In this section, we will modify the original scheme described in Section 10.2 to make it SUF-CMA, without incurring any extra computational overhead.

As can be observed from the above example, the forgery is successful if the adversary can modify the value of r and predict its effect on the authentication tag τ. To rectify this problem, not only the message but also the coin tosses, r, must be authenticated. Obviously, this can be done with the use of another secret key k'_s and computing the tag as

$$\tau \equiv mk_s + rk'_s \pmod{p}. \tag{10.7}$$

This, however, requires twice the amount of shared key material and an extra multiplication operation. A more efficient way of achieving the same goal can be done by computing the modular multiplication

$$\sigma = mk_s \pmod{p} \tag{10.8}$$

and transmitting an encrypted version of the result of equation (10.8) as the authentication tag. That is, since r is the main reason for the successful forgery illustrated above, instead of authenticating r as in equation (10.7), it is removed from the equation. However, since r was necessary for the privacy of the scheme of Section 10.2.1, it is required to encrypt the result of equation (10.8) before transmission to provide data privacy. This implies that the scheme described here is an instance of the MAC-then-Encrypt (MtE) composition as apposed to the Encrypt-and-MAC (E&M) composition of Section 10.2.1.

The description of the modified system is as follows. Assume the users have agreed on a security parameter N, exchanged an N-bit prime integer p, and a secret key $k_s \in \mathbb{Z}_p^*$. On input a message $m \in \mathbb{Z}_p$, compute the modular multiplication $\sigma = mk_s \pmod{p}$. The transmitter encrypts m and σ and transmits the ciphertext $c = \mathcal{E}(m, \sigma)$ to the intended receiver. The ciphertext can be the encryption of the plaintext message concatenated with σ, i.e. $\mathcal{E}(m||\sigma)$, or it can be the concatenation of the encryption of the message and the encryption of σ, i.e. $\mathcal{E}(m)||\mathcal{E}(\sigma)$. For ease of presentation, we will assume the latter scenario and call the ciphertext $c = \mathcal{E}(m)$ and the tag $\tau = \mathcal{E}(\sigma)$. Decryption and authentication are performed accordingly.

The proof that this modified scheme provides data privacy can be found in [34]. In particular, since the modified scheme of this section is an instance of MtE compositions, Bellare and Namprempre showed that if the underlying encryption algorithm is IND-CPA secure, then so is the generic MtE composition [34]. The proof that the modified scheme achieves weak unforgeability under chosen message attacks is similar to the proof of Theorem 14 and, thus, is omitted. Below we show that the modified system described in this section is indeed strongly unforgeable under chosen message attacks.

Theorem 15. *The proposed scheme is strongly unforgeable under chosen message attacks (SUF-CMA), provided the adversary's inability to break the IND-CPA security of the underlying encryption algorithm.*

Proof: Let (m, τ) be a valid message-tag pair recorded by the adversary. By equation (10.8), for the same m, the resulting σ will always be the same. Assume the adversary is attempting to authenticate the same message, m, with a different tag τ'. Since σ in both cases is the same, the difference between τ and τ' is due to the probabilistic behavior of the encryption algorithm. Therefore, the adversaries advantage of breaking the SUF-CMA security of the scheme is negligible provided the IND-CPA security of the encryption algorithm. That is,

$$\mathsf{Adv}_{\Sigma}^{\text{suf-cma}}(\mathcal{A}) \leq \mathsf{Adv}_{\mathcal{E}}^{\text{ind-cpa}}(\mathcal{B}) + \mathsf{negl}(N),$$

where $\mathsf{negl}(N)$ is a negligible function in the security parameter N, and the theorem follows.

■

10.4 Encrypting with Pseudorandom Permutations (Block Ciphers)

In this section, we describe a message authentication approach that is faster than the one described in previous sections. The main idea of this approach is that the input-output relation of the used encryption operation can be realized as a pseudorandom permutation. The scheme of this section can be viewed as a special case of the KMAC scheme introduced in Chapter 5 in which authenticated message are of a pre-determined length.

10.4.1 Background

Informally speaking, a strong pseudorandom permutation is an invertible function in which the input-output relation cannot be distinguished from a true random function by compu-

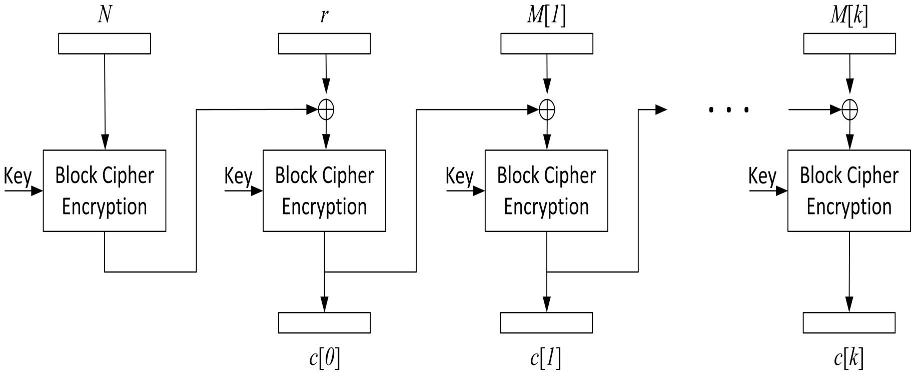


Figure 10.1: The Cipher Block Chaining (CBC) mode of encryption used for message encryption. The random number, r, is treated as the first block of the plaintext.

tationally bounded users. In practice, most modern block ciphers can be modeled as strong pseudorandom permutations. In fact, in the call for proposals for the Advanced Encryption Standard (AES) one can find the following statement [138]:

"The security provided by an algorithm is the most important factor ... . Algorithms will be judged on the following factors ...

- *The extent to which the algorithm output is indistinguishable from a random permutation on the input block."*

Since the design of hardware efficient block ciphers is an active research area nowadays, with solution already proposed (see, e.g., [75, 163, 116, 153, 45, 171]), we propose here another authentication code assuming the used encryption algorithm is a strong pseudorandom permutation.

10.4.2 The Proposed System

Let $\mathcal{F} : \{0,1\}^N \rightarrow \{0,1\}^N$ be the function representing the bock cipher. We assume that $\mathcal{F}$ acts as a strong pseudorandom permutation, a typical assumption satisfied by secure block ciphers. Assume further that exchanged messages are N-bit long.

10.4.2.1 Message Encryption

Let m be a short message that is to be transmitted to the intended receiver in a confidential manner. For every message to be transmitted, a random nonce $r \in \mathbb{Z}_{2^N}$ is chosen. (We overload m to denote both the binary string representing the message, and the integer representation of the message as an element of $\mathbb{Z}_{2^N}$; the same applies to r. The distinction between the two representations will be omitted when it is clear from the context.)

Now, the concatenation of r and m goes to the encryption algorithm, call it $\mathcal{E}$, as an input. Ideally, we may desire $\mathcal{E}$ to be a strong pseudorandom permutation; however, since N can be sufficiently long (e.g., 128 or larger), constructing a block cipher that maps $2N$-bit strings to $2N$-bit strings can be expensive. Therefore, we resort to the well-studied cipher block chaining (CBC) mode of operation to construct $\mathcal{E}$ from $\mathcal{F}$, as illustrated in Figure 10.1.[1]

Consider the CBC mode of operation depicted in Figure 10.1. The nonce r is treated as the first plaintext block and is XORed with the initialization vector (IV) to insure IND-CPA security. The first ciphertext block,

$$c_1 = \mathcal{F}_{k_{\mathcal{E}}}(IV \oplus r), \tag{10.9}$$

is then XORed with the second plaintext block, m in our construction, to produce the second ciphertext block,

$$c_2 = \mathcal{F}_{k_{\mathcal{E}}}(c_1 \oplus m), \tag{10.10}$$

where $k_{\mathcal{E}}$ is the key corresponding to the block cipher. The resulting

$$c = \mathcal{E}(r, m) = IV||c_1||c_2 \tag{10.11}$$

is then transmitted to the intended receiver as the ciphertext.

[1] Although other modes of operations, such as, counter (CTR), output feedback (OFB), etc., can be used, we restrict our attention to the CBC mode for two reasons. First, the CBC mode of operation is sufficient to illustrate the main ideas of our construction. Second, it is a reasonable mode of operation for low-cost devices that are unable to perform parallel computing.

10.4.2.2 Message Authentication

With the encryption described above, authentication becomes simpler than the ones in previous sections; the authentication tag of message m is calculated as follows:

$$\tau \equiv m + r \pmod{2^N}. \tag{10.12}$$

Upon receiving the ciphertext, the intended receiver decrypts it to extract r and m. Given τ, the receiver can check the validity of the message by performing the following integrity test:

$$\tau \overset{?}{\equiv} m + r \pmod{2^N}. \tag{10.13}$$

If the integrity check of equation (10.13) is satisfied, the message is considered authentic. Otherwise, the integrity of the message is denied.

10.4.3 Performance Discussion

Assuming devices are already equipped with a secure block cipher to encrypt messages, the authentication technique of Section 10.4.2 requires only one modular addition. While addition is performed in $O(n)$ time, the fastest integer multiplication algorithms typically require $O(n \ \log n \ \log\log n)$ time [85]. Therefore, as efficient as the scheme proposed in Section 10.2.1, the authentication technique of Section 10.4.2 is at least $O(\log n \ \log\log n)$ faster. (We note, however, that the scheme of Section 10.2.1 can result in a faster authenticated encryption if combined with a stream cipher since the one of Section 10.4.2 requires the use of block cipher based encryption.)

10.4.4 Security Analysis

In this section, we prove the privacy of the system, give a formal security analysis of the proposed message authentication mechanism, and then discuss the security of the composed authenticated encryption system. The security model that will be used here is the same as the security model used to analyze the security in Chapter 5.

10.4.4.1 Data Privacy

Recall that two pieces of information are transmitted to the intended receiver (the ciphertext and the authentication tag), both of which are functions of the private plaintext message. Now, when it comes to the authentication tag, observe that the nonce r serves as a one-time key (similar to the role r plays in the construction of Section 10.2). The formal analysis that the authentication tag does not compromise message privacy is the same as the one provided in Section 10.2.3.1 and, thus, is omitted.

The ciphertext of equation (10.9), on the other hand, is a standard CBC encryption and its security is well-studied; thus, we give the theorem statement below without a formal proof (interested readers may refer to textbooks in cryptography, e.g., [179, 243, 138] for more details). Let the privacy of the system be modeled as its indistinguishability from a pseudorandom permutation, one gets the following.

Theorem 16. *Let $\mathcal{E}$ be the encryption algorithm of Figure 10.1 and let $\mathcal{F}$ be the block cipher used to construct $\mathcal{E}$. Then given an adversary $\mathcal{A}$ against the privacy of $\mathcal{E}$, one can construct an adversary $\mathcal{B}$ against the pseudorandomness of $\mathcal{F}$ such that*

$$\mathsf{Adv}^{\mathrm{priv}}_{\mathcal{E}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathrm{prp}}_{\mathcal{F}}(\mathcal{B}).$$

Furthermore, the experiment for $\mathcal{B}$ takes the same time as the experiment for $\mathcal{A}$ and, if $\mathcal{A}$ makes at most q_e oracle queries, then $\mathcal{B}$ makes at most $2q_e$ oracle queries.

Theorem 16 states that an adversary breaking the privacy of the encryption algorithm of Figure 10.1 is also able to break the pseudorandomness of the underlying block cipher. Therefore, the adversary's advantage of breaking the privacy of the encryption algorithm is negligible, provided the use of a secure block cipher.

10.4.4.2 Data Authenticity

Before we provide a bound on the probability of successful forgery, we give an informal discussion on how the structure of the authenticated encryption composition will be utilized.[2]

[2] The discussion is similar to the one at the beginning of Section 5.3.2.

Recall that, in standard MACs, the security is modeled by the adversary's probability of predicting a valid authentication tag for a certain message. That is, given the adversary's knowledge of a polynomial number of valid message-tag pairs, the goal of the adversary is to forge a new message-tag pair that will be accepted as valid.

MACs in an our authenticated encryption composition, on the other hand, are fundamentally different than standard MACs. The intended receiver in an authenticated encryption system receives a ciphertext-tag pair as opposed to message-tag pair. This implies that, for an attempted forgery to be successful, the adversary must come up with a *ciphertext-tag* pair that will be accepted as valid, not a *message-tag* pair. (Note, however, that we do not hide messages from the adversary. In fact, we assume the adversary's ability to launch chosen message attacks, as can be seen in the security model and the formal proof below.)

Remark 15. *We emphasize that this security model can also be used for the analyses of previous sections (since it is also the case that the intended user receives a ciphertext-tag pair). The reason for not using this security model in previous sections is that the security can be proven using the standard model. For the technique proposed in this section, however, security cannot be proven without the modified model (as can be seen in the proof below).*

Following the standard convention in cryptography, we give below information-theoretic bound on the adversary's probability of successful forgery assuming the block cipher is a true random permutation (the complexity-theoretic analogy is given after the theorem). Let Σ denote the proposed composition of Section 10.4.2 and let $\mathsf{Adv}_{\Sigma}^{\mathrm{auth}}(\mathcal{A})$ denote adversary's $\mathcal{A}$ advantage of successful forgery against Σ.

Theorem 17. *Let $\mathcal{F} : \{0,1\}^N \to \{0,1\}^N$ be a true random permutation used to construct an encryption algorithm, $\mathcal{E}$, in the cipher block chaining mode, as depicted in Figure 10.1. Let Σ denote the proposed composition of Section 10.4.2 and let $\mathcal{A}$ be an adversary calling the signing oracle q times before making a forgery attempt. Then,*

$$\mathsf{Adv}_{\Sigma}^{\mathrm{auth}}(\mathcal{A}) \leq 2^{1-N}.$$

To pass a complexity-theoretic analog of Theorem 17, one will need access to an $\mathcal{F}^{-1}$ oracle in order to verify a forgery attempt, which translates into needing the strong pseudorandom permutation assumption. One gets the following. Fix a block cipher $\mathcal{F} : \mathcal{K} \times \{0,1\}^N \to \{0,1\}^N$ that is used to construct the mode of encryption of Figure 10.1. Let $\mathcal{A}$ be an adversary that asks q signing queries then attempting its forgery. Then, there is an adversary $\mathcal{B}$ attacking the block cipher in which

$$\mathsf{Adv}_{\Sigma}^{\mathrm{auth}}(\mathcal{A}) \leq \mathsf{Adv}_{\mathcal{F}}^{\mathrm{sprp}}(\mathcal{B}) + 2^{1-N},$$

where $\mathsf{Adv}_{\mathcal{F}}^{\mathrm{sprp}}(\mathcal{B})$ is as defined in equation (2.3). Furthermore, adversary $\mathcal{B}$ takes the same time adversary $\mathcal{A}$ takes, minus the time of generating the coin tosses and the generation and authentication of tags, and makes at most $2(q+1)$ oracle queries.

Proof: [Proof of Theorem 17] When $q = 0$ it is rather straightforward. It follows directly from the fact that each value of the authentication tag is equally probable (by Lemma 1).

Now, assume $\mathcal{A}$ has made q signing queries and recorded the sequence

$$\mathsf{Seq} = \Big\{(m_1, c_1, \tau_1), \cdots, (m_q, c_q, \tau_q)\Big\}. \tag{10.14}$$

$\mathcal{A}$ then calls the verify oracle with (c, τ), where $(c, \tau) \neq (c_i, \tau_i)$ for any $i = 1, \cdots, q$ since otherwise $\mathcal{A}$ does not win by definition. We aim to bound the probability that (c, τ) will be validated. Let r and m be the nonce and the message corresponding to the decryption of c, respectively. There are two possible strategies for forgery:

1. attempt to forge a valid ciphertext-tag pair corresponding to a specific plaintext of $\mathcal{A}$'s choice,

2. attempt to authenticate a ciphertext-tag pair regardless of their corresponding plaintext (i.e., modify a recorded ciphertext-tag pair in a way undetected by the legitimate receiver).

Call the former $\mathsf{forgery}_1$ and the latter $\mathsf{forgery}_2$.

To bound the probability of $\mathsf{forgery}_1$, assume $\mathcal{A}$ attempts to falsely authenticate a plaintext $r||m \neq r_i||m_i$ for any $i = 1, \cdots, q$. If $r \neq r_i$, the adversary must predict the two ciphertext blocks and the probability of successful forgery is 2^{-2N} (since $\mathcal{F}$ is a true random permutation). If $r = r_i$, the adversary must predict the ciphertext block corresponding to m, which is equal to 2^{-N}. Therefore, the probability of $\mathsf{forgery}_1$ is at most 2^{-N}.

To bound the probability of $\mathsf{forgery}_2$, denote by $\mathsf{Collision}$ the event that $m + r \equiv m_i + r_i \pmod{2^N}$ for some $i \in \{1, \cdots, q\}$. That is, the tag corresponding to the modified ciphertext, τ, collides with τ_i, one of the recorded tags in the sequence of equation (10.14). Also, we use $\overline{\mathsf{Collision}}$ as the typical notation for the complement of $\mathsf{Collision}$.

Obviously, when the event $\mathsf{Collision}$ occurs, $(c, \tau_i) \neq (c_i, \tau_i)$ will pass the integrity check, leading to successful forgery. Recall, however, that $\mathcal{F}$ is a true random permutation; hence, m and r, the message and the nonce corresponding to c, cannot be correlated to m_i and r_i, the plaintext and the nonce corresponding to c_i. That is, from the adversary's standpoint, m and r are random elements of $\mathbb{Z}_{2^N}$. Therefore, the probability that the plaintext-nonce pair corresponding to c (the modified version of c_i), will result in a τ that collides with τ_i is

$$\Pr\left[\mathsf{Collision}\right] = \Pr\left[m + r \equiv m_i + r_i \pmod{2^N}\right] \leq 2^{-N}. \tag{10.15}$$

Assume now that the event $\overline{\mathsf{Collision}}$ is true. If no collision has occurred, then the adversary's probability of successful forgery is bounded by the probability of predicting the plaintext message corresponding to c. That is, similar to the probability of $\mathsf{forgery}_1$,

$$\Pr\left[\mathsf{forgery}_2 | \overline{\mathsf{Collision}}\right] \leq 2^{-N}. \tag{10.16}$$

By equations (10.15) and (10.16), it follows that the probability of $\mathsf{forgery}_2$ can be bounded by:

$$\begin{aligned}
\Pr\left[\mathsf{forgery}_2\right] &= \Pr\left[\mathsf{forgery}_2 | \mathsf{Collision}\right] \cdot \Pr\left[\mathsf{Collision}\right] \\
&\qquad + \Pr\left[\mathsf{forgery}_2 | \overline{\mathsf{Collision}}\right] \cdot \Pr\left[\overline{\mathsf{Collision}}\right] && (10.17) \\
&\leq \Pr\left[\mathsf{Collision}\right] + \Pr\left[\mathsf{forgery}_2 | \overline{\mathsf{Collision}}\right] && (10.18) \\
&\leq 2^{-N} + 2^{-N}. && (10.19)
\end{aligned}$$

Hence, $\max\left\{\Pr\left[\mathsf{forgery}_1\right], \Pr\left[\mathsf{forgery}_2\right]\right\} = 2^{1-N}$ is $\mathcal{A}$'s maximum advantage of successful forgery, and the theorem follows. ■

10.5 Summary

In this chapter, a new technique for authenticating short encrypted messages is proposed. The fact that the message to be authenticated must also be encrypted is used to deliver a random nonce to the intended receiver via the ciphertext. This allowed the design of an authentication code that benefits from the simplicity of unconditionally secure authentication without the need to manage one-time keys. In particular, it has been demonstrated in this chapter that authentication tags can be computed with one addition and a one modular multiplication. Given that messages are relatively short, addition and modular multiplication can be performed faster than existing computationally secure MACs in the literature of cryptography. When devices are equipped with block ciphers to encrypt messages, a second technique that utilizes the fact that block ciphers can be modeled as strong pseudorandom permutations is proposed to authenticate messages using a single modular addition.

Chapter 11

REDUCING IDENTIFICATION COMPLEXITY

Recall that there are two complementing ends in any radio frequency identification system: the interactive protocol between authorized RFID reader-tag pairs and the interaction between RFID readers and the database for data retrieval. In the previous two chapters, we addressed the reader-tag interactive protocol side of RFID system. In this chapter, we turn our attention to the reader-database data retrieval mechanism. In particular, we present the first privacy-preserving protocol with constant-time identification.

11.1 The Identification Paradox

Privacy-preserving symmetric-key protocols are faced with the following paradox. On one side, a tag must encrypt its identity with its secret key so that only authorized readers can extract the identity. On the opposite side, authorized readers must first determine the identity of the tag in order to know which key is to be used for decryption. Therefore, given that tags' responses are randomized (to protect users' privacy), and that the length of tags' responses is sufficiently long (so that easy to implement attacks such as random guessing and exhaustive search will have small probability of success), searching the database for those responses is a nontrivial task.

Most privacy-preserving RFID protocols trade-off identification efficiency for the sake of privacy. That is, private identification is accomplished, but the reader is required to perform an exhaustive search among all tags in the system in order to identify the tag being interrogated (see, e.g., [191, 25, 59, 235]). In a typical protocol of this class, the reader interrogates a tag by sending a random nonce, r_1. The tag generates another nonce, r_2, computes $h(ID, r_1, r_2)$, where h is a cryptographic hash function, and responds with $s = \big(r_2, h(ID, r_1, r_2)\big)$. (Different protocols implement variants of this approach; but this is the main idea of this class of protocols.) Upon receiving the tag's response, the reader performs a linear search of all tags in the system, computing the hash of their identifiers with the transmitted nonces, until it finds a match. Obviously, unauthorized observers cannot correlate different responses of the same tag, as long as the nonce is never repeated.

Although protocols of this class have been shown to provide private identification, their practical implementation has a scalability issue. In a large-scale RFID system, performing a linear search for every identification can be time consuming. Moreover, denial of service attacks can be launched by giving authorized readers false identifiers causing them to perform exhaustive search amongst all tags in the system before realizing that the received response is invalid. Hence, for an RFID system to be practical, one must aim for a scheme that can break the barrier of *linear-time* identification complexity.

A big step towards solving the scalability issue in privacy-preserving RFID systems was proposed in [181]. This new approach traded-off computational and communication overhead on tags to speed up the identification process. The authors utilized a tree data structure, where each edge in the tree corresponds to a unique secret key, each leaf of the tree corresponds to a unique tag, and each tag carries the set of keys on the corresponding path from the root of the tree to its leaf. When a reader interrogates a tag, the tag responds with a message encrypted with its first key. By decrypting the tag's response with the keys corresponding to all edges of the first level of the tree, the reader can determine to which edge the tag belongs. By traversing the tree from top to bottom, the tag can be identified in $O(\log N_T)$ time using $O(\log N_T)$ reader-tag interactions, where N_T is the number of tags in the system.

Arranging tags in a tree, based on secret keys they possess, however, introduced a new security threat to the RFID system known as the "*tag compromise vulnerability*": every compromised tag will reveal the secret keys from the root of the tree to its leaf. Since these keys are shared by other tags in the system, compromising one tag will reveal secret information about all tags sharing a subset of those keys. In [25], the tree structure is analyzed showing that in a tree with a branching factor of two, compromising 20 tags in a system of 2^{20} tags leads to the identification of uncompromised tags with an average probability close to one.

Another major drawback of the tree-based class of protocols is the increase in communication and computation overhead on tags. In a typical RFID system, the reader interrogates multiple tags simultaneously. Consequently, even in the linear-time identification protocols, where communication overhead is $O(1)$, collision avoidance and medium access control are among the most challenging problems in the design of efficient RFID systems [183, 145, 143]. Increasing the communication overhead to $O(\log N_T)$ can only complicate collision avoidance even further. Moreover, requiring the tag to perform $O(\log N_T)$ cryptographic operations can also be problematic for passive tags as it leads to more energy consumption.

Researchers believing that reducing identification complexity from $O(N_T)$ to $O(\log N_T)$ cannot be overlooked as a result of the vulnerability it introduced have been making significant efforts to mitigate the tag compromise problem [167, 255, 168]. The idea shared by all such attempts is to employ key updating mechanisms to mitigate the effect of tag compromise. Other researchers, however, believe that the new threat overweighs the reduction in identification complexity, thus, proceeding with the linear-time class of protocols and trying to improve on its performance (see, e.g., [25, 59, 235]).

In this chapter, we address the private identification problem in large-scale RFID systems. We propose a protocol that, in addition to being *resilient to tag compromise attacks*, allows *constant-time identification*, without imposing extra *communication or computation overhead* on the resource limited tags. The main drive behind devising our protocol is the intuition that, in order to overcome the problems in both linear and logarithmic identification classes, one must aim for a solution that is fundamentally different than both of them. We do not resort to tree structure, nor do we incur more communication overhead. Instead, we utilize resources that are already available in RFID systems to improve identification efficiency. That is, since in any RFID system there is a database, to store information about tags in the system, and since storage is relatively cheap in today's technology, we trade-off storage for the sake of better identification efficiency. The novelty of the proposed protocol is the architecture of the database and the utilization of off-line computations to allow for constant-time tag identification. In addition to its obvious advantage in identification efficiency, our protocol has also a security advantage. In [24], Avoine et al. introduced a new attack on the privacy of RFID systems based on time measurements. After analyzing various protocols, the authors concluded that the protocol proposed in this chapter is, to

Table 11.1: Performance comparison as a function of the number of tags in the system, N_T, and a security parameter C introduced in our protocol. Class 1 represents protocols with linear-time identification, while Class 2 represents protocols with log-time identification. The overhead in the last column refers to computation and communication overhead on the tags' side.

	Search time	Key size	Database size	Overhead
Class 1	$O(N_T)$	$O(1)$	$O(N_T)$	$O(1)$
Class 2	$O(\lg N_T)$	$O(\lg N_T)$	$O(N_T)$	$O(\lg N_T)$
Proposed	$O(1)$	$O(1)$	$O(CN_T)$	$O(1)$

date, the best protocol in terms of efficiency and security [24]. Table 11.1 compares our protocol to the classes of linear-time and log-time identification protocols.

The rest of the chapter is organized as follows. The proposed system is detailed in Section 11.2. In Section 11.3, we prove our claim of constant-time identification, and provide a case study in Section 11.4. Section 11.5 is dedicated to the security proofs of the proposed system. The robustness against tag compromise attacks is detailed in Section 11.6. In Section 11.7, we discuss desynchronization attacks against the proposed system and extend our system to prevent such attacks. We conclude our chapter in Section 11.8.

11.2 System Description

11.2.1 Protocol Overview

Each tag has an internal counter, c, and is preloaded with a unique *secret* pseudonym, ψ, and a secret key, k. The secret key and the secret pseudonym are updated whenever mutual authentication with a valid reader is accomplished, while the counter is incremented every time authentication fails.

When an RFID reader is to identify and authenticate a tag within its range, it generates a random nonce, $r \overset{\$}{\leftarrow} \{0,1\}^L$, and transmits it to the tag. Upon receiving r, the tag computes $h(\psi, c)$ and $\tilde{r} := h(0, \psi, c, k, r)$, where ψ is the tag's current pseudonym, k is the

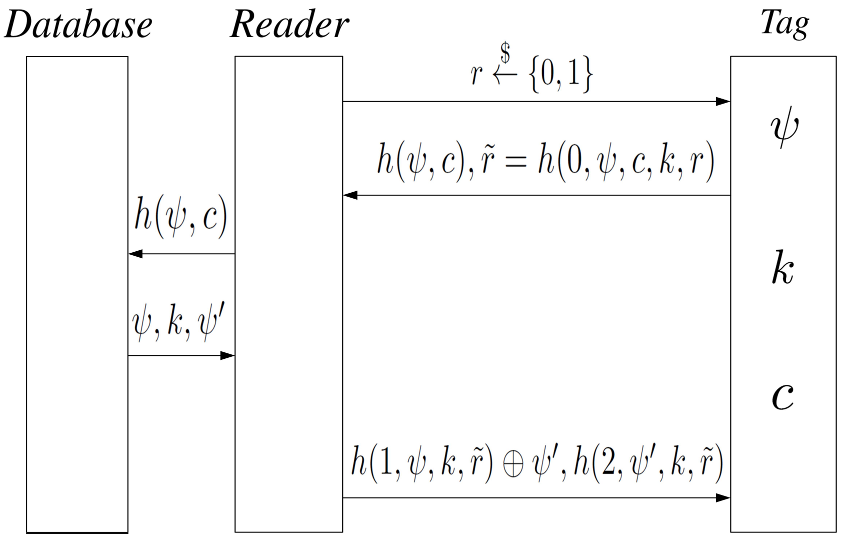

Database
Reader
Tag
$r \overset{\$}{\leftarrow} \{0,1\}$
$h(\psi, c), \tilde{r} = h(0, \psi, c, k, r)$
$h(\psi, c)$
ψ, k, ψ'
$h(1, \psi, k, \tilde{r}) \oplus \psi', h(2, \psi', k, \tilde{r})$
ψ
k
c

Figure 11.1: A schematic of one instance of the protocol.

tag's current secret key, c is the tag's internal counter, and r is the received nonce. The tag then increments its counter, $c \leftarrow c + 1$ and responds to the reader. With $h(\psi, c)$, the reader accesses the database to identify the tag and obtain its information, including its pseudonym, ψ, its secret key, k, and a new pseudonym, ψ', to update the tag. With $\tilde{r}$, the reader authenticates the tag by confirming its knowledge of the secret key, k, obtained from the database.

Once the tag has been identified and authenticated, the reader responds with $h(1, \psi, k, \tilde{r}) \oplus \psi'$ and $h(2, \psi', k, \tilde{r})$. With $h(1, \psi, k, \tilde{r}) \oplus \psi'$ the tag extracts its new pseudonym, ψ'. With $h(2, \psi', k, \tilde{r})$, the tag authenticates the reader and verifies the integrity of the received ψ'. If the reader is authenticated, the tag resets its counter to zero and updates its secret key and pseudonym to $k' = h(k)$ and ψ', respectively. Otherwise, the protocol is terminated. Figure 11.1 depicts a single protocol run between an RFID reader-tag pair.

11.2.2 Database Overview

As mentioned above, the tag is identified by its randomized response, $h(\psi, c)$, which is an L-bit long string. Since security requires that L is sufficiently long, it is infeasible to construct a physical storage that can accommodate all possible 2^L responses, for direct addressing. (This is the reason why previous schemes resorted to linear search amongst all tags in the system to identify a response.) For ease of presentation, the structure of the database is divided into three logical parts, M-I, M-II, and M-III.

To allow for constant-time identification, with feasible storage, we truncate the L-bit identifiers to their s most significant bits, where s is small enough so that a storage of size 2^s is feasible. Of course, many identifiers will share the same s most significant bits (to be exact, 2^{L-s} possible identifiers will share the same truncated value). M-I is a table of size $O(2^s)$, with addresses ranging from 0 to $2^s - 1$, and each table entry contains a pointer to an entry in M-II (similar to a hashtable data structure, with truncation instead of hashing). All identifiers with the same s most significant bits will be stored in a smaller table in M-II, and the pointer at address s in M-I will point to the head of this smaller table. Finally, actual information about tags in the system is stored in M-III. Detailed construction of the database and description of the identification process will be the focus of the remainder of this section.

The proposed protocol can be broken into four main phases: parameters selection phase, system initialization phase, tag identification phase, and identity randomization and system update phase. Each phase is detailed below.

11.2.3 Parameters Selection

During this phase, the database is initialized and each tag is loaded with secret information. The secret information includes the tag's secret key, which the tag and reader use to authenticate one another, and the tag's pseudonym, which is used for tag identification.

Given the total number of tags the RFID system is suppose to handle, N_T, and predefined security and performance requirements (more about this later), the system designer chooses the following parameters to start the initialization phase:

- The total number of pseudonyms, N. Since pseudonyms will be used as unique tag identifiers, there must be at least one pseudonym for every tag in the system. Furthermore, since tags are assigned new identifiers following every successful mutual

Table 11.2: A list of parameters and used notations.

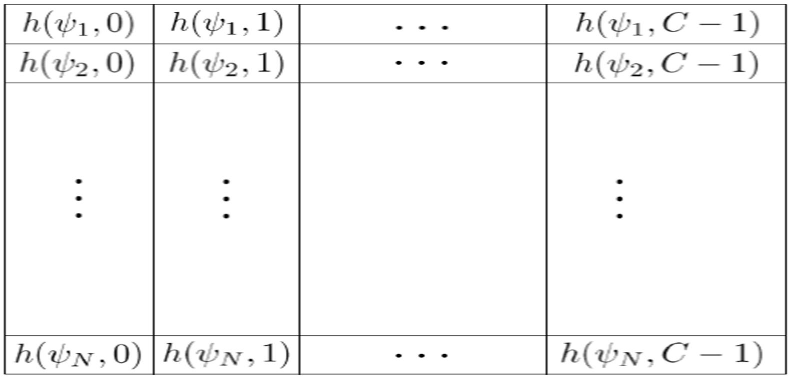
Symbol	Definition
N_T	The total number of tags in the system
N	The total number of pseudonyms in the system
ψ_i	The pseudonym corresponding to the i^{th} tag
C	The maximum counter value
ℓ	The length of the secret parameter in bits
h	Cryptographic hash function
L	The output length of the used hash function
n	The length of the truncated hash values
$\Psi_{i,c}$	A tag identifier, $\Psi_{i,c} := h(\psi_i, c)$
$\Psi_{i,c}^n$	The n most significant bits of $\Psi_{i,c}$

authentication process with an authorized reader, the total number of pseudonyms must be greater than the total number of tags in the system, i.e., $N > N_T$.

- The maximum counter value, C. The counter is used by RFID tags to mitigate traceability by active adversaries; the larger the counter is, the more difficult it will be for active adversaries to track the tag; on the downside, the size of the database will grow linearly with the counter (the database size is $O(NC)$). Therefore, the size of the counter is a trade-off between tags' privacy and system complexity.

- The length, ℓ, in bits, of the tags' secret parameters (pseudonyms and keys). As in any symmetric key cryptosystem, ℓ should be chosen properly to prevent easy-to-implement attacks, such as exhaustive search and random guessing. Obviously, ℓ must be long enough to generate N distinct pseudonyms, i.e., $\ell \geq \lceil \log_2 N \rceil$. In practice, however, ℓ will be much longer.

- The hash function, h. In particular, the output length of the hash values, L, is of



$h(\psi_1, 0)$	$h(\psi_1, 1)$	$\cdots$	$h(\psi_1, C-1)$
$h(\psi_2, 0)$	$h(\psi_2, 1)$	$\cdots$	$h(\psi_2, C-1)$
$\vdots$	$\vdots$		$\vdots$
$h(\psi_N, 0)$	$h(\psi_N, 1)$	$\cdots$	$h(\psi_N, C-1)$

Figure 11.2: During database initialization, all values of $h(\psi, c)$ are computed.

special importance. The length must be chosen large enough so that there are no collisions during database initialization, which is described below.

- The length, n, of the truncated hashes. The size of n is the key for constant-time identification and practicality of the system. It will be determined in Section 11.3.

Table 11.2 summarizes the list of system parameters and used notations.

11.2.4 System Initialization

Once the system parameters have been chosen, the initialization phase can start. The initialization phase can be summarized in the following steps.

1) Given the number of pseudonyms, N, and the length of each pseudonym, ℓ, the system designer draws, *without replacement*, N pseudonyms randomly from the set of all possible ℓ-bit strings. That is, N *distinct* pseudonyms, $\psi_1, \psi_2, \ldots, \psi_N$, are chosen at random from $\{0,1\}^\ell$. Each tag is given a unique pseudonym and a secret key, and each tag's counter is initially set to zero. *We emphasize that the drawn pseudonyms are not publicly known; otherwise, tags' privacy can be breached.*

2) For each pseudonym, ψ_i, the hash value $h(\psi_i, c)$ is computed for all $i = 1, \ldots, N$ and all

$c = 0, \ldots, C-1$. That is, a total of NC hash operations must be performed, as depicted in Figure 11.2. Each row of the table in Figure 11.2 corresponds to the same pseudonym. Therefore, all entries in the i^{th} row must point to the same memory address carrying information about the tag identified by the pseudonym ψ_i.

In order for tags to be identified uniquely, the hash values in the table of Figure 11.2 *must be distinct.* This can be achieved by choosing the hash function, h, to be an expansion function, as opposed to the usual use of hash functions as compression functions, so that collision will occur with small probability.[1] We will assume that the output of the hash function has length L bits, which must be at least equal to $\lceil \log_2 NC \rceil$ so that the table in Figure 11.2, which is of size NC, can be constructed without collisions (L will be much larger in practice). If a pseudonym that causes a collision in Figure 11.2 is found, the pseudonym is replaced by another one that does not cause a collision. (Observe that the pool of possible pseudonyms is of size 2^{ℓ}, which is much larger than the required number of pseudonyms N, giving the system designer a sufficient degree of freedom in constructing the system.) With the appropriate choice of the hash function, a table of hash values with no collisions can be constructed. *Note that this operation is performed only once during the initialization phase, thus, it does not undermine the performance of the system.*

Since the length of $h(\psi_i, c)$ (the tags' identifiers), L, is large to avoid collision, it would be infeasible to have a physical storage that can accommodate all possible L-bit strings (for direct addressing). For example, if $L = 128$, a database of size in the order of 4×10^{28} *Gigabyte* will be required. Previously proposed privacy-preserving schemes solve this problem in one of two approaches. The first approach requires $O(N_T)$ memory space to store information about each tag in the system, and requires the reader to perform a linear search among tags in the system to identify tags' responses; thus requiring $O(N_T)$ space and $O(N_T)$ time for identification. The other method identifies tags based on their key information and requires the reader to perform logarithmic search to identify tags' responses; thus requiring $O(N_T)$ space and $O(\log N_T)$ time for identification.

[1] For example, this can be accomplished by concatenating multiple hash functions, i.e., $h(x) = h_1(x)|| \cdots ||h_m(x)$, so that $h(x)$ has the required length.

3) For ease of presentation, we will divide the database into three logical parts, M-I, M-II, and M-III. The first part, M-I, consists of a single table of size $O(2^n)$.The second part, M-II, consists of multiple smaller tables; the total size of all the tables in M-II is $O(NC)$. Finally, the last part, M-III, is of size $O(N)$.

M-I is a table of pointers. The addresses of M-I range from 0^n to 1^n; each entry in the table points to the head of one of the mini tables in M-II (according to a specific relation explained below).

Each entry of M-II contains two fields. In the first field, the hash values obtained in the table of Figure 11.2 are stored (i.e., $h(\psi_i, c)$ for all $i = 1, \ldots, N$ and all $c = 0, \ldots, C-1$). M-II is organized based on the hash values stored in the first field. We say that two hash values $h(\psi_1, c_1)$ and $h(\psi_2, c_2)$ are in the same *position*, b, if their n most significant bits are the same (recall that the output length of the hash function is $L > n$). All hash values that have the same position, i.e., share the n most significant bits, are stored in the same *mini* table in M-II (e.g., the hash values with $b = s$ in Figure 11.3). Hash values with distinct positions are stored in different tables (e.g., hash values with $b = 1, s, 1^n$ in Figure 11.3). (Recall that Figure 11.2 contains the computed hash values; hence, table M-II can be viewed as a reorganized version of the two-dimensional table in Figure 11.2 into a one-dimensional table of size $O(NC)$.) The second field of each entry of M-II stores a pointer to an entry in M-III containing information about a tag in the system (depending on the value of the first field). For example, if the value stored in the first field is $h(\psi_i, c)$, then the value in the second field will be a pointer to the data entry in M-III where information about the tag with pseudonym ψ_i can be found.

After M-II has been constructed, the pointers at M-I are chosen to satisfy the following: the pointer stored at address a in M-I must point to the mini table in M-II that stores identifiers with position a. In other words, each pointer in M-I must point to the identifiers with position equal to the address of the pointer.

Finally, M-III is the actual storage where tags' information is stored. Figure 11.3 depicts the architecture of the database with the three logical partitions. The identification phase below will further illustrate the structure of the database.

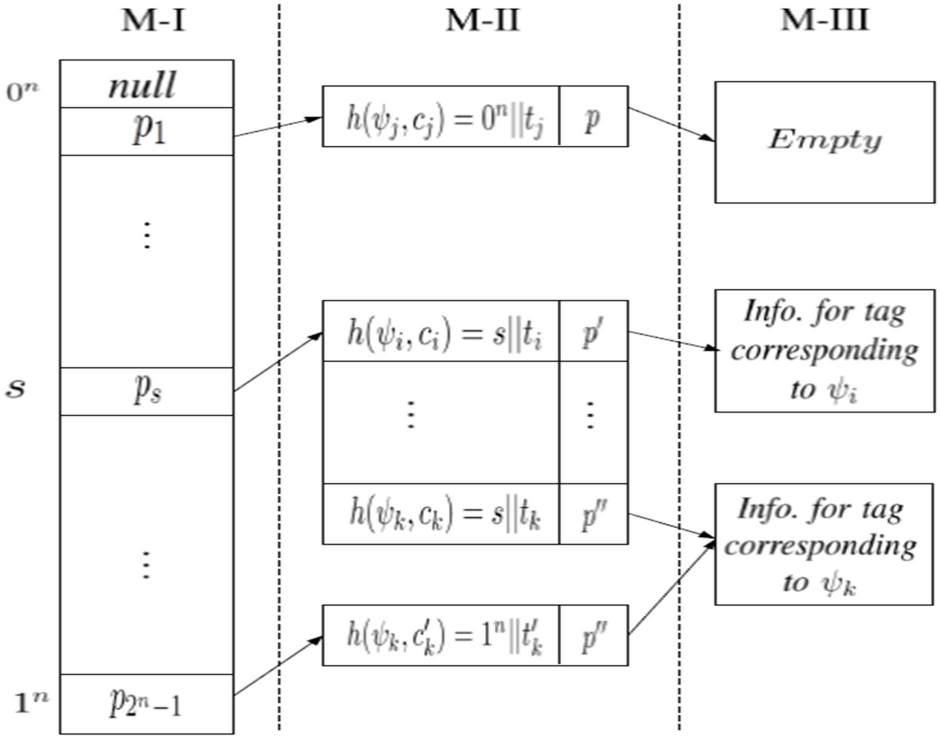

M-I
M-II
M-III
0^n
null
p_1
s
p_s
1^n
p_{2^n-1}
$h(\psi_j, c_j) = 0^n \| t_j$
p
$h(\psi_i, c_i) = s \| t_i$
p'
$h(\psi_k, c_k) = s \| t_k$
p''
$h(\psi_k, c'_k) = 1^n \| t'_k$
p''
Empty
Info. for tag corresponding to ψ_i
Info. for tag corresponding to ψ_k

Figure 11.3: The architecture of the database. Each entry in M-I points to another, smaller table in M-II. The entries of the smaller tables in M-II point to tags' information.

11.2.5 Tag Identification

Tags in a protocol run of the system are identified by the hash of their pseudonyms concatenated with their internal counters. Denote by $\Psi_{i,c}$ the hash value of the i^{th} pseudonym concatenated with a counter c; that is, $\Psi_{i,c} := h(\psi_i, c)$. Furthermore, we will denote by $\Psi^n_{i,c}$ the truncated value of $\Psi_{i,c}$; more precisely, $\Psi^n_{i,c}$ represents the n most significant bits of $\Psi_{i,c}$ (i.e., the position of $\Psi_{i,c}$).

Once $\Psi_{i,c}$ has been received, the reader accesses the data entry at address $\Psi^n_{i,c}$ in M-I. This table entry is actually a pointer, p, to one of the tables in M-II. There are three possible scenarios here:

1) The value at address $\Psi^n_{i,c}$ in M-I is a *null*. This implies that, during the construction of the table in Figure 11.2, no identifier with position $\Psi^n_{i,c}$ is constructed. Therefore, either the tag is not a valid one or the tag's response has been modified. In the example of Figure 11.3, if the n most significant bits of the received $\Psi_{i,c}$ are *zeros*, then no valid tag matches this response.

2) The pointer, p, at address $\Psi^n_{i,c}$ points to a table in M-II with exactly one entry. In this scenario, the first field of the entry pointed at by p must be the entire (untruncated) $\Psi_{i,c}$; the value at the second field will be a pointer to the entry in M-III that contains information about the interrogated tag. In the example of Figure 11.3, if the n most significant bits of the received $\Psi_{i,c}$ are *ones*, then the pointer at address 1^n in M-I will point to the entry at M-II at which $\Psi_{k,c'_k} = 1^n||t'_k$ and the pointer, p'', are stored. In turn, p'' will point to the entry at M-III where information about the tag with pseudonym ψ_k is stored.

3) The pointer at address $\Psi^n_{i,c}$ of M-I points to a table in M-II with more than one entry. In this scenario, the reader searches the first fields of the mini table in M-II until it reaches the entry that matches the complete (untruncated) received identifier, $\Psi_{i,c}$; and then follows the pointer (in the corresponding second field) to get the tag's information. In the example of Figure 11.3, if the received identifier is $\Psi_{k,c_k} = s||t_k$, the reader will follow the pointer at address s of M-I. The pointer, however, points to a table in M-II with more than one entry. Therefore, the reader must search until it reaches the last entry of the table to find a match for the received $\Psi_{k,c_k} = s||t_k$. Once the match is found, the reader can follow the pointer, p'', to the entry in M-III containing information about the tag with pseudonym ψ_k.

The identification process allows for unique identification of tags in the system. This is due to the requirement that, in the initialization phase, the values in the table of Figure 11.2 are distinct. Consequently, the entries in M-II are distinct, allowing for the unique identification of tags.

Remark 16. *Recall that the pseudonyms drawn in the initialization are not publicly known.*

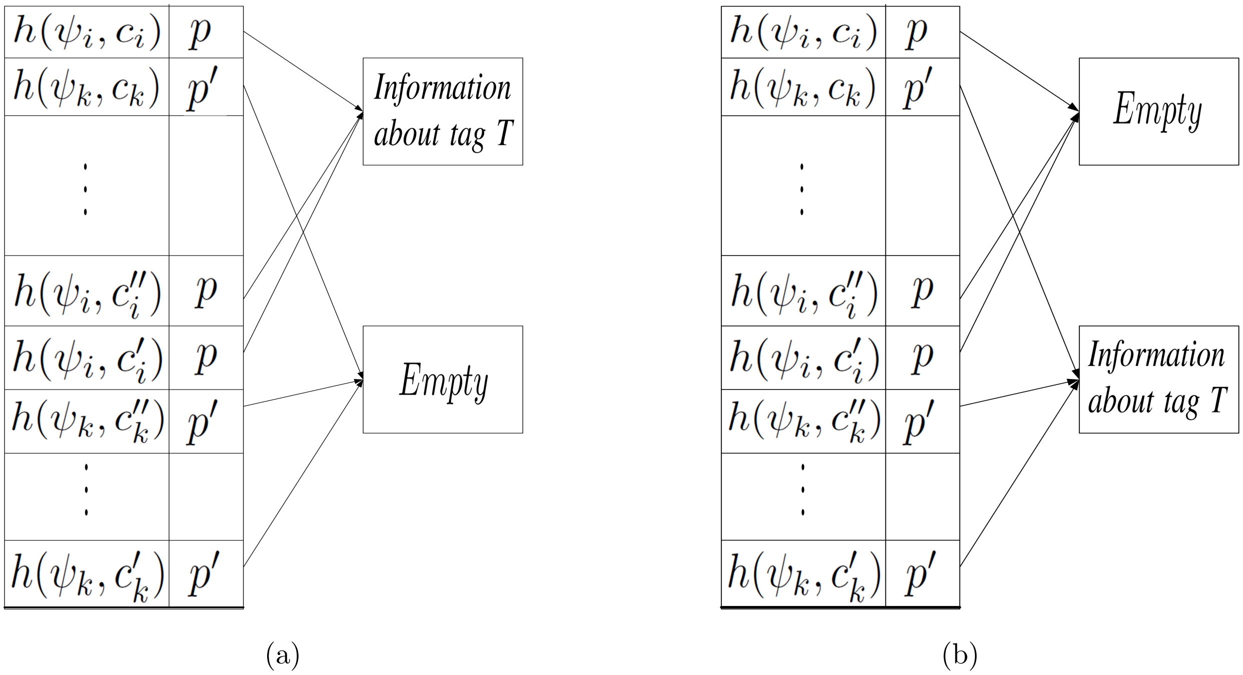


Figure 11.4: (a) Before (b) After; an illustration of database update. Note that only the tag information is updated, rather than the pointer values. This way, we only have to update two entries instead of $O(C)$ entries.

If the pseudonyms were published, an adversary can, in principle, construct her own system and identify tags in constant-time. Further discussion about the adversary's ability to expose secret pseudonyms is provided in Section 11.6.

11.2.6 Identity Randomization and System Update

Once a tag has been authenticated, the reader draws one of the unoccupied pseudonyms generated in the initialization phase. (Recall that the number of pseudonyms is greater than the number of tags in the system; consequently, there will always be unused pseudonyms available for identity randomization.) Once an unoccupied pseudonym has been chosen, it is to be transmitted to the tag in a secret and authenticated way.

To allow for correct identification of a tag after its pseudonym has been updated, the database must be updated accordingly. A straightforward way of updating the database is by updating the pointers corresponding to the outdated and updated pseudonyms. For example, if the tag's outdated pseudonym is ψ_i and its updated pseudonym is ψ_k, then all pointers in M-II corresponding to entries $\Psi_{i,0}, \Psi_{i,1}, \ldots, \Psi_{i,C-1}$ must point to a null; and all pointers in M-II corresponding to entries $\Psi_{k,0}, \Psi_{k,1}, \ldots, \Psi_{k,C-1}$ must point to the entry in M-III containing information about the tag. This method, however, requires $O(C)$ updating time. Since C will typically be a large constant (say hundreds of thousands), this will be a "practical" violation of our constant-identification claim (similar to the protocol of [265]).

An alternative method that allows a more practical update is depicted in Figure 11.4. Instead of updating the pointers as in the previous method, the tag's information is moved to the entry in M-III pointed at by the pointers corresponding to the updated pseudonym in M-II. The only price to pay for this method over the previous one is that the size of M-III will increase from $O(N_T)$ to $O(N)$ (asymptotically, N and N_T are of the same size). In the example of Figure 11.4, instead of changing all entries in M-II with pointer p' to p, and changing entries with pointer p to *null*, the tag's information is moved to the entry in M-III pointed at by p' and the entry pointed at by p is emptied.

11.3 Performance Analysis

For the proposed scheme to be practical, we must show that a set of parameters can be chosen such that our claim of constant-time identification can be achieved with feasible resources (namely, feasible database size). This section is devoted to showing that, with a set of appropriately chosen parameters, the proposed technique can achieve constant-time identification with a database of size $O(N_T)$.

Assuming that the $\Psi_{i,c}$s are uniformly distributed, the probability that the truncated version $\Psi^n_{i,c}$ takes a specific value, s, is $\alpha = \Pr[\Psi^n_{i,c} = s] = 2^{-n}$, for any $s \in \{0,1\}^n$. Let $M := NC$ and define $m := \log_2 M$, where N is the total number of pseudonyms and C is the maximum counter value. Then, out of the M values of $\Psi_{i,c}$s, the probability that exactly k of them share the same truncation value (i.e., exactly k of them have the same n

most significant bits) is

$$\Pr[\boldsymbol{k} = k] = \binom{M}{k} \alpha^k (1-\alpha)^{M-k}, \tag{11.1}$$

where $\boldsymbol{k}$ is the random variable representing the number of $\Psi^n_{i,c}$ sharing the same value, s, for any $s \in \{0,1\}^n$. Then, for $k \ll M$,

$$\binom{M}{k} = \frac{M!}{k!(M-k)!} \approx \frac{M^k}{k!}. \tag{11.2}$$

Using the facts that $\lim_{n\to\infty}(1 - \frac{1}{n})^n = e^{-1}$, $M = 2^m$, and $\alpha = 2^{-n}$ we get:

$$\begin{aligned} (1-\alpha)^{M-k} &\approx (1-\alpha)^M && (11.3)\\ &= (1-2^{-n})^{2^m} && (11.4)\\ &= (1-2^{-n})^{2^n \cdot 2^{m-n}} && (11.5)\\ &\approx e^{-2^{m-n}}. && (11.6) \end{aligned}$$

Substituting equations (11.2) and (11.6) into (11.1) yields,

$$\begin{aligned} \Pr[\boldsymbol{k} = k] &\approx \frac{M^k}{k!} \cdot \alpha^k \cdot e^{-2^{m-n}} && (11.7)\\ &= \frac{2^{mk}}{k!} \cdot \frac{1}{2^{nk}} \cdot e^{-2^{m-n}} && (11.8)\\ &= \frac{1}{k!} \cdot \beta^k \cdot e^{-\beta}, && (11.9) \end{aligned}$$

where $\beta = 2^{m-n}$. Choosing $m = n$ yields $\beta = 1$ and equation (11.9) can be reduced to

$$\Pr[\boldsymbol{k} = k] \approx \frac{1}{k!} \cdot e^{-1} \text{ for } k = 0, 1, \ldots\ . \tag{11.10}$$

(It can be easily verified that $\Pr[\boldsymbol{k} = k]$ in equation (11.10) is a valid probability mass function by verifying that $\sum_{k=0}^{\infty} \Pr[\boldsymbol{k} = k] = 1$.)

Using the fact that $e = \sum_{k=0}^{\infty} \frac{1}{k!}$, the expected number of truncated $\Psi_{i,c}$'s with the same value is

$$\begin{aligned} \mathbb{E}[\boldsymbol{k}] &= \sum_{k=0}^{\infty} k \cdot \Pr[\boldsymbol{k} = k] && (11.11)\\ &= \sum_{k=1}^{\infty} k \cdot \frac{1}{k!} \cdot e^{-1} && (11.12)\\ &= 1. && (11.13) \end{aligned}$$

Recall that identifiers $\Psi_{i,c}$ with the same truncated value $\Psi^n_{i,c}$ will be in the same table in M-II; and when the reader receives one of these identifiers it will have to search the table to be able to identify the tag. Equation (11.13), however, implies that the expected size of the tables in M-II is *one*. Therefore, upon receiving a tag identifier $\Psi_{i,c}$, the reader goes to the table entry in M-I at address $\Psi^n_{i,c}$, follows the pointer p_1 stored at that address, searches the table in M-II pointed at by p_1 for the received $\Psi_{i,c}$ (by equation (11.13), there will be only one entry on average), and then follows a pointer p_2 to information about the tag. Indeed, the search time is independent of the number of tags in the system (on average).

Since the database consists of three parts, M-I, M-II, and M-III; and since the size of M-I is $O(2^n)$, the size of M-II is $O(NC)$, and the size of M-III is $O(N)$, the only concern is the size of M-I. The above analysis shows that, by choosing $n = \lceil \log_2 NC \rceil$, the system achieves the constant-time identification claim. Therefore, the size of M-I is $O(NC)$ and, consequently, the total size of the database is $O(NC)$.

11.4 Case Study

Since asymptotic analysis can be misleading by absorbing big constants, we give here a numerical example of the practicality of our system.

11.4.1 Database Size

Assume an enterprise with one million items to be tagged, i.e., $N_T = 10^6$. Assume further that the total number of pseudonyms is two millions, i.e., $N = 2N_T$, and $C = 10^6$. Then, the truncated identifiers are $n = \lceil \log_2 NC \rceil = 41$-bit long. Therefore, M-I can be constructed with a storage smaller than 12 terabyte; a practical storage even for personal usage.[2]

Then, an active adversary must interrogate a tag more than a million consecutive times, not separated by a protocol run with a valid reader, in order to correlate its responses. We emphasize that the adversary's interrogations must be consecutive. That is, once the tag completes a protocol run with an authorized reader, its pseudonym will be updated and the adversary will have to start all over again. Observe that, unlike security models in

[2] Western Digital has already released 8-TB hard drives for personal use [1].

general computer communications, that much consecutive interrogations is a highly unlikely scenario for RFID systems. A web server, for instance, is always online and available for interactions from distances. In a typical RFID systems, however, adversaries must be in close proximity to tags in order to interrogate them. Observe, moreover, that an adversary who is always in the vicinity of a tag can track it down visually without interrogation. So, in typical designs, the goal is to protect tags privacy against adversaries that are not always in close proximity to the RFID tags. Therefore, limiting the number of consecutive tag interrogations is a typical relaxation in RFID models [128].

In another example, assume an enterprise with one billion items to be tagged, i.e., $N_T = 10^9$. Assume further that the total number of pseudonyms is two billions and $C = 1000$. Then, M-I can be constructed with the same storage of the above example and the adversary must interrogate the tag more than a thousand consecutive times to correlate its responses.

11.4.2 Tags Privacy and the Value of C

As discussed earlier, an active adversary interrogating the same tag C consecutive number of times will be able to correlate the tag's responses. A typical response time of EPC tags, for instance, is around 4ms [274]. Therefore, when $C = 10^6$, the adversary must interrogate the tag for about 66.7 consecutive minutes in order to correlate its responses. In the other example where $C = 10^3$, the adversary only needs few seconds. While the first case might provide reasonable privacy, the second one definitely does not.

To mitigate such attacks, tag response time, call it τ, can be designed so that the product $\tau \cdot C$ is sufficiently long. Setting $\tau = 1$ second, for instance, will require the adversary to interrogate the tag for 16.67×10^3 and 16.67 consecutive minutes for the first and second example, respectively. Alternatively, let the tag delay its response as a function of the counter (up to a certain threshold to avoid denial of service attacks). For example, let tags delay their responses by 10 seconds after the tenth consecutive false protocol run. Then, even when C is as small as 10^3, the adversary will need 165 minutes of consecutive interrogations.

11.5 Security Analysis

In this section, we prove that our protocol preserves the integrity of the tag and reader while maintaining user privacy. Before we proceed with the proofs of privacy and integrity, we state some important assumptions about the used hash function that are necessary for our security proofs.

11.5.1 Cryptographic Hash Functions

We assume the use of a secure cryptographic one-way hash function (there exist hash functions designed solely for RFID tags, such as, [227, 192, 141]). Under practical assumptions about the adversary's computational power, the used hash function satisfies the following properties.

1. Given the output of the hash function, it is computationally difficult to infer the input. That is, given the value of $h(x)$, the probability to predict the correct value of x by computationally bounded adversaries is negligible.

2. Given x and $h(x)$, the probability to predict $h(x+i)$, for any i, without actually evaluating $h(x+i)$ is negligible.

Given the above properties of the used hash function, the following lemma states an important result that will be used for the privacy and integrity proofs.

Lemma 23. *The secret parameters of RFID tags in the proposed protocol cannot be exposed without calling the Reveal oracle.*

Proof: In any interrogation, the tag responds with its current identifier $\Psi_{i,c} = h(\psi_i, c)$, where ψ_i is the tag current pseudonym and c is its internal counter. Given the above properties of the used hash function, the pseudonym cannot be exposed by the observation of $h(\psi_i, c)$ with a non-negligible probability. Furthermore, the new pseudonym is delivered to the tag by transmitting $(h(1, \psi_i, k_i, \tilde{r}) \oplus \psi_{i+1})$, which can be viewed as an encryption of ψ_{i+1} with the key $h(1, \psi_i, k_i, \tilde{r})$. Since ψ_i and k_i are unknown to adversaries, $h(1, \psi_i, k_i, \tilde{r})$ will act as a random key and the new pseudonym ψ_{i+1} will be delivered secretly. Moreover, since the outdated and the updated pseudonyms, ψ_i and ψ_{i+1}, are unknown to adversaries, the two identifiers, $h(\psi_i, c)$ and $h(\psi_{i+1}, c)$, cannot be correlated with a non-negligible probability and, similarly, the identifiers $h(\psi_i, c)$ and $h(\psi_i, c+1)$, cannot be correlated with a non-negligible probability.

Therefore, unless $\mathcal{A}$ calls the *Reveal* oracle, no secret information about RFID tags in the proposed protocol can be revealed. ■

Before we proceed with the formal proofs, we discuss the effect of the *Block* oracle and desynchronization attacks.

11.5.2 Desynchronization Attacks

Jamming the communication channel, i.e., blocking all messages, is not of an interest to this work, since it does not lead to breaching of tags' privacy nor does it lead to authenticating unauthorized users.

Blocking the first message (from the reader to the tag) will just cause the tag not to respond. Similar to jamming, no information will be leaked by blocking the first message.

Blocking the second message (from the tag to the reader) can be modeled by the *Query* oracle. In fact, intercepting the tag's response is equivalent to a *Query* oracle in which the adversary does not control the value of r transmitted in the first message.

Blocking the last message (from the reader to the tag) has two effects. First, it will cause the tag to increase its internal counter (since the protocol run is incomplete), but this can also be modeled using the *Query* oracle. Second, and more important, it will lead the reader to update the tag's pseudonym while the tag has not,[3] i.e., a desynchronization attack. Fortunately, however, this can be solved by storing both the updated and the outdated pseudonyms in the database (the database must be designed accordingly, as detailed in Section 11.7).

In what follows, we formally prove the privacy and integrity of the proposed protocol.

[3] This is an inherited problem shared by all interactive protocols. The fundamental problem here is that the sender of the last message has no means of confirming that the message has been successfully delivered.

11.5.3 Privacy

In this section, we show that the proposed protocol satisfies the three notions of tag privacy defined in Section 8.4.3.

Theorem 18. *In the proposed protocol, tags are universally untraceable.*

Proof: Assume the challenger $\mathcal{C}$ has chosen two tags, T_0 and T_1, and a reader R for the game. $\mathcal{A}$ starts the game by calling the *Query, Send, Execute* and *Block* oracles on T_0, T_1, and R for a number of times of its choice before deciding to stop. $\mathcal{A}$ records all the outputs of the oracle calls and notifies $\mathcal{C}$.

Now, R carries out protocol runs with T_0 and T_1 causing their pseudonyms and keys to update. $\mathcal{C}$ chooses a bit b uniformly at random and sets $T = T_b$. By Lemma 23, $\mathcal{A}$ cannot infer the outdated nor the updated values of the tags' pseudonyms and keys. $\mathcal{A}$ now calls the oracles *Query, Send, Execute* and *Block* and outputs a bit b'. Since $\mathcal{A}$ does not know the outdated or the updated pseudonyms, by the assumptions on the used hash function, the probability $\Pr[b = b']$ will be greater than 1/2 with a non-negligible probability.

Therefore, the adversary's advantage, as defined in equation 8.1, will be greater than zero with only a negligible probability. ■

The following theorem concerns forward untraceability in our protocol.

Theorem 19. *In the proposed protocol, tags are forward untraceable.*

Proof: Similar to the proof of universal untraceability, assume the challenger $\mathcal{C}$ has chosen two tags, T_0 and T_1, and a reader R for the game. $\mathcal{A}$ starts the game by calling the *Query, Send, Execute* and *Block* oracles on T_0, T_1, and R for a number of times of its choice before deciding to stop. $\mathcal{A}$ records all the outputs of the oracle calls and notifies $\mathcal{C}$.

Now, R carries out protocol runs with T_0 and T_1 causing their pseudonyms and keys to update. $\mathcal{C}$ chooses a bit b uniformly at random and sets $T = T_b$ and gives it to $\mathcal{A}$. By Lemma 23, $\mathcal{A}$ cannot infer the outdated nor the updated values of the tags' pseudonyms and keys. $\mathcal{A}$ now calls the *Reveal*(T) oracle, thus getting T's secret parameters, and then outputs a bit b'. Since $\mathcal{A}$ cannot infer the outdated pseudonyms and keys of T_0 and T_1 from the recorded oracle outputs, and since the updated pseudonyms are chosen independently of the outdated ones, $\mathcal{A}$ cannot correlate T's updated pseudonym with its previous responses. Furthermore, since the updated key is a hashed function of the outdated key, by the assumptions on the used hash function, $\mathcal{A}$ cannot infer the value of the outdated key with a non-negligible probability. Hence, the probability $\Pr[b = b']$ will be greater than $1/2$ with only a non-negligible probability.

Therefore, the adversary's advantage, as defined in equation 8.1, will be greater than zero with only a negligible probability. ■

Finally, the following theorem concerns existential untraceability in our protocol.

Theorem 20. *Without being able to achieve mutual authentication with an authorized reader, a tag interrogated fewer than C number of times by an active adversary is untraceable.*

Proof: Assume that $\mathcal{C}$ has given T_0 and T_1 to $\mathcal{A}$. Let ψ_0 and ψ_1 denote the pseudonyms of T_0 and T_1, respectively. Without loss of generality, assume that tags T_0 and T_1 have their internal counters at zero. $\mathcal{A}$ calling the *Query* oracle on T_0 and T_1 for m and n times, respectively, where $m, n < C$ will observe the following sequences

$$\{h(\psi_0, 0), \ldots, h(\psi_0, m-1)\}, \tag{11.14}$$

$$\{h(\psi_1, 0), \ldots, h(\psi_1, n-1)\}. \tag{11.15}$$

The challenger $\mathcal{C}$ now chooses a bit b at random, sets $T = T_b$, and gives T to $\mathcal{A}$. By interrogating the tag, $\mathcal{A}$ gets an identifier $h(\psi_b, \ell)$, where $b \in \{0, 1\}$ and $\ell \in \{m, n\}$. Again, by Lemma 23, ψ_0 and ψ_1 cannot be recovered by the observation of the sequences in equations (11.14) and (11.15). Furthermore, by the assumptions on the hash function, $h(\psi_0, m)$ and $h(\psi_1, n)$ cannot be correlated to the observed values in equations (11.14) and (11.15) with a non-negligible probability. Therefore, the probability that $\mathcal{A}$'s guess b' is equal to b can be higher than $1/2$ with only a negligible probability and, hence, $Adv_{\mathcal{A}} = 0$ and tags are existentially untraceable, provided that $m, n < C$. ■

11.5.4 Mutual Authentication

We shift our attention now to the other security requirement, authenticity.

Theorem 21. *The proposed protocol performs secure mutual authentication.*

Proof: Assume that $\mathcal{C}$ has given $\mathcal{A}$ a tag T and a reader R. Assume further that $\mathcal{A}$ has called the *Query, Send, Execute* and *Block* oracles for a number of times of its choice and recorded the oracle outputs.

The first condition of Definition 7 of secure mutual authentication is satisfied by Lemma 23.

Assume now that $\mathcal{A}$ attempts to impersonate the tag T. $\mathcal{A}$ must answer the reader's challenge r with a response $s = \Big(h(\psi, c), \tilde{r} = h(0, \psi, c, k, r)\Big)$, where ψ is the tag's current pseudonym and k is its key. Since ψ and k remain secret, by Lemma 23, $\mathcal{A}$ can be successful with only a negligible probability. Observe further that, even if $\mathcal{A}$ attempts to impersonate an arbitrary tag in the system (the one with pseudonym ψ), $\mathcal{A}$ must know the value of k corresponding to the tag with pseudonym ψ in order to be authenticated with a non-negligible probability. Therefore, the probability of impersonating a tag in the system is negligible.

On the other hand, assume that $\mathcal{A}$ attempts to impersonate the reader R. $\mathcal{A}$ sends r to the tag and receives $h(\psi, c)$ and $\tilde{r} = h(0, \psi, c, k, r)$, where ψ is the tag's pseudonym, k is its secret key, and c is its internal counter. Since, by the assumption on the hash function, $\mathcal{A}$ cannot infer the secret parameters, the probability of coming up with a response $h(1, \psi, k, \tilde{r}) \oplus \psi', h(2, \psi', k, \tilde{r})$ that will be validated is negligible. Consequently, the probability of impersonating an authorized reader in the system is negligible.

Therefore, the probability of mutual authentication when the protocol is not honest is negligible and, hence, the second condition of Definition 7 of secure mutual authentication is satisfied.

As shown above, the adversary's probability of causing a desynchronization between the tag and the reader by authenticating herself to either one of them is negligible. Causing a desynchronization by blocking the last message of the protocol can be solved by making the reader store both the updated and the outdated values (as will be discussed in Section 11.7). Therefore, if the protocol run is honest, mutual authentication will be achieved with probability one and, consequently, the third condition of Definition 7 of secure mutual authentication is satisfied.

Hence, all conditions of Definition 7 of secure mutual authentication are satisfied and the proposed protocol is shown to provide secure mutual authentication. ■

11.6 Tag Compromise Analysis

In this section, we show that, unlike log-time identification protocols [181, 167, 255, 168] and the protocol of Song and Mitchell [236], the proposed protocol is secure against tag compromise attacks.

11.6.1 The Compromise attack

Each tag in the proposed protocol has two pieces of secret information, its pseudonym and its key. Since tags' pseudonyms and keys are designed to be statistically independent for different tags, compromising some tags in the system does not affect the security of other, uncompromised tags. An adversary, however, can compromise a tag in the system and attempt to harvest as many pseudonyms as possible by performing multiple protocol runs with a valid reader.

The adversarial model of Section 8.4 can be modified to capture the tag compromise attack. Let an adversary calling the *Reveal* (T) oracle, thus capturing the tag T, have the ability to perform multiple protocol runs with the system. Let q be the number of protocol runs an adversary has performed with the system using compromised tags. The number of interest here is how many distinct pseudonyms the adversary has collected, after q protocol runs. This is known in the literature of probability theory as the "coupon collecting problem" [78]. Given there are N distinct pseudonyms and the adversary has performed q protocol runs, assuming each pseudonym is equally likely to be selected, the expected number of distinct pseudonyms collected by the adversary is [78]:

$$N\left(1-\left(\frac{N-1}{N}\right)^q\right). \tag{11.16}$$

Assume an adversary has built a system, similar to our construction, with the collected pseudonyms. The adversary's advantage of distinguishing between two tags, given by equation (8.1), will be greater than zero if at least one of the two tags' pseudonyms is in the

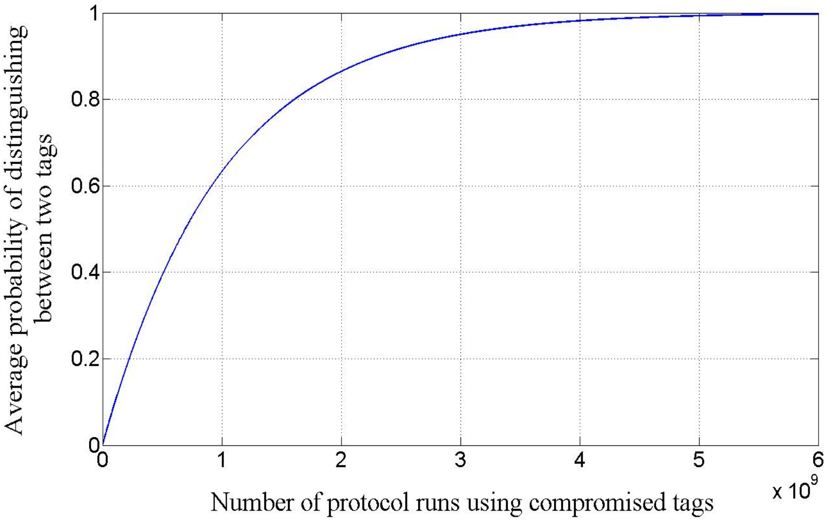


Figure 11.5: The adversary's average probability of distinguishing between two tags vs. the number of protocol runs using a compromised tag, in a system with 2×10^9 pseudonyms.

constructed table. Thus, given the adversary has performed q protocol runs with a system of N pseudonyms, the probability of distinguishing between two tags is:

$$1 - \left(\frac{N-1}{N}\right)^{2q}. \tag{11.17}$$

Consider the numbers given in Section 11.4, i.e., $N = 2 \times 10^9$. To have a 0.001 probability of distinguishing between two tags, an adversary needs to compromise a tag and complete more than a million protocol runs with the system. Figure 11.5 shows the adversary's probability of having an advantage greater than zero as a function of the number of protocol runs performed with the system using compromised tags.

11.6.2 Countermeasures

Remember, however, that the database is a powerful device. Therefore, designing the database to record timing information about the tag's past protocol runs can mitigate this threat. For example, the database can store information about the tag's last five protocol runs (this can be stored as part of the tag's information, i.e., in M-III). If the adversary attempts to harvest different pseudonyms by performing multiple protocol runs with the system, the tag will be detected. Therefore, to harvest enough pseudonyms, the adversary will need to compromise more than one tag, depending on the system's parameters and the required probability of success.

Furthermore, the database can periodically update the system by replacing vacant pseudonyms with new pseudonyms (recall that the number of pseudonyms in the database, N, is only a small fraction of the number of all possible pseudonyms, 2^ℓ). This pseudonym update procedure is performed offline by the database, thus, not affecting identification time. Moreover, as a result of the independence of secret parameters amongst tags, the updating procedure is independent of tags.

With the periodic update described earlier, the space of possible pseudonyms will increase to all possible ℓ-bit long strings, as opposed to the predefined *smaller* number N. Therefore, for a bounded adversary, any polynomial number of collected pseudonyms is negligible in the security parameter ℓ. (Recall that the size of the actual database is still proportional to N; only from the adversary's point of view the size is proportional to 2^ℓ.) Consequently, the adversary's probability of breaking the privacy of the system is negligible in ℓ, provided the periodic update of the database.

11.7 Preventing Desynchronization Attacks

Recall that if the tag does not accept the reader's response, the database will update the tag's pseudonym while the tag has not. Consequently, the reader will not be able to identify the tag in future protocol runs. As mentioned in Section 11.5.2, however, the database can be designed to overcome such attacks by storing both the updated and outdated pseudonyms; details are as follows.

11.7.1 Redesigning the Update Procedure

Consider the update procedure described in Section 11.2.6. Let each entry of M-III consists of a linked list data structure, as opposed to a single entry as in the basic description. For illustration purposes, assume the linked list consists of four fields containing the following data. The first field contains information about a tag T_i with pseudonym ψ_i, where ψ_i is

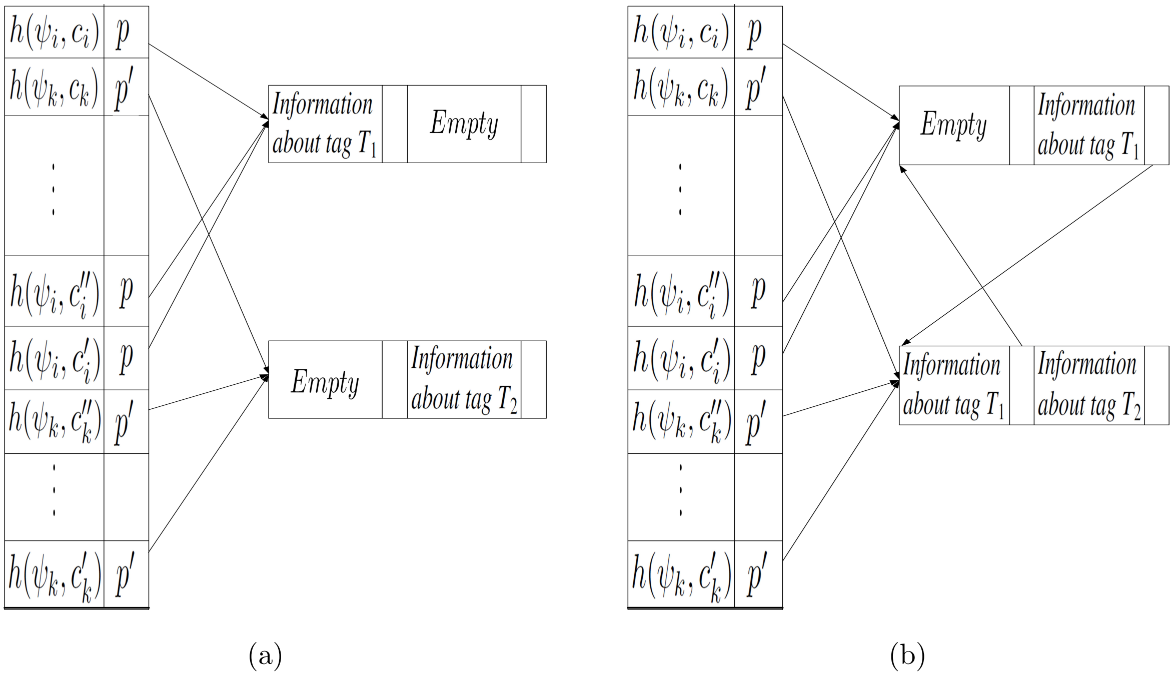


Figure 11.6: (a) Before (b) After; an illustration of database update. Note that only the tag information is updated, rather than the pointer values. This way, we only have to update two entries instead of $O(C)$ entries.

the T_i's "updated" pseudonym. The second field will contain a pointer to the entry of M-III corresponding to T_i's outdated pseudonym, if it existed (i.e., if the tag has been interrogated previously). The third field contains information about a tag T_k with pseudonym ψ_k, where ψ_k is the T_k's "outdated" pseudonym. The fourth field will contain a pointer to the entry of M-III corresponding the T_k's updated pseudonym, if it existed. The construction is best illustrated through the following example.

Consider Figure 11.6 for updating the database. Assume the reader has authenticated the tag T_1 with a current pseudonym ψ_i. Assume further that the database returns a new pseudonym ψ_k as the updated pseudonym for the tag T_1. Just like the update procedure described in Section 11.2.6, the information about tag T_1 in M-III will be copied into the data entry pointed at by the pointers in M-II corresponding to the updated pseudonym ψ_k

(i.e., pointer p' in the example of Figure 11.6). However, instead of deleting the information about tag T_1 in the entry pointed at by the pointer corresponding to the outdated pseudonym ψ_i (i.e., pointer p in the example of Figure 11.6), the information remains there.

Observe, however, that by continuing in this fashion, information about the tag will have multiple copies in the database, one for each identification run. To prevent this problem, we use the pointer field in M-III. That is, the use of the new pointer fields in M-III will allow preventing the desynchronization attack with only two copies of tag information in M-III, one corresponding the updated pseudonym and one corresponding to the outdated pseudonym. Observe, in Figure 11.6-b, that the information about tag T_1 corresponding to the outdated pseudonym ψ_i is followed by a pointer field that stores a pointer to the information about T_1 corresponding to the updated pseudonym ψ_k. Similarly, that the information about tag T_1 corresponding to the updated pseudonym ψ_k is followed by a pointer field that stores a pointer to the information about T_1 corresponding to the outdated pseudonym ψ_i.

Assume now that the tag T_1 has received the updated identifier ψ_k successfully and, hence, no desynchronization attack has been attempted. Upon interrogation, the tag will respond with its identifier $\Psi_{k,c}$, which will enable the reader to identify the tag. Once the entry in M-III with information about the tag has been found (the bottom box of M-III in the example of Figure 11.6-b), the pointer in the field after the tag's information is followed to empty the data entry with information corresponding to the tags outdated pseudonym ψ_i (in the top box of M-III in the example of Figure 11.6-b). The database then draws an unused pseudonym ψ_j to update the tag, mark the information corresponding to ψ_k as outdated, and copies the tag's information to the entry corresponding to ψ_j. Therefore, only two copies of the tag's information need to be stored in M-III.

On the other hand, assume that there has been a desynchronization attempt during the last protocol run and, thus, the tag has not updated its pseudonym to ψ_k. Therefore, upon the next interrogation, the tag will respond with its identifier $\Psi_{i,c}$. Since both the updated and the outdated pseudonyms are stored, the database can still identify the tag via its outdated pseudonym (in the top box of M-III in the example of Figure 11.6-b). Once the tag's information has been found, the pointer is followed to delete the tag's information corresponding to the undelivered pseudonym ψ_k (in the bottom box of M-III in the example of Figure 11.6-b). Just like the previous case, the database then draws an unused pseudonym ψ_j to update the tag, mark the information corresponding to ψ_i as outdated, and copies the tag's information to the entry corresponding to ψ_j. Therefore, whether a desynchronization attack has been attempted or not, only two copies of the tag's information need to be stored in M-III.

As can be observed in the example of Figure 11.6-a, the tag T_2 has ψ_k as its outdated pseudonym. This does not prevent the database from choosing ψ_k as the new pseudonym to update tag T_1. If the existence of a tag with an outdated pseudonym prevents the database from using this pseudonym to update other tags, then each tag in the system will occupy two pseudonyms. As this might not cause a problem when the number of tags in the system is not too large, it can be problematic if the number of tags in the system is very large (a billion tags, for instance). Therefore, we allow the database to update tags with any pseudonym as long as there is no other tag in the system with this pseudonym as its "updated" pseudonym, even if other tags have this pseudonym as their "outdated" pseudonym. In the example of Figure 11.6, ψ_k is chosen to update the tag T_1 even though the tag T_2 has the same pseudonym as its outdated pseudonym.

Assume now that the tag T_1 in the example of Figure 11.6 has received ψ_k successfully. Since in the next interrogation, T_1 will respond with $\Psi_{k,c}$, the pseudonym ψ_k will be marked now as the tag's outdated pseudonym. Therefore, there might be more than one tag with the same outdated pseudonym and, hence, upon receiving an identifier corresponding to such pseudonym, the database will search linearly (amongst tag stored in the same entry of M-III) until it finds the match. We show next that this does not violate the constant-time identification claim by showing that the expected number of tags in the same entry of M-III is independent of the total number of tags in the system.

11.7.2 Identification Complexity

We seek to find the number of tags with the same outdated pseudonym, thus, falling in the same entry of M-III, causing the database to search linearly amongst them. Recall that pseudonyms are drawn uniformly at random to update tags. That is, the tag's information can fall into any entry of M-III with equal probability. This problem is equivalent to a well-studied problem in probability theory called the "balls in bins" problem [78]. In a classic variant of the balls in bins problem, m balls are thrown at n bins and the probability of any ball falling in a certain bin is the same for all balls and all bins.

Instead of m balls and n bins, we are interested in throwing N_T RFID tags into N possible pseudonyms (recall that each entry in M-III corresponds to one pseudonym). Therefore, the probability that a certain tag will fall into a particular entry in M-III is $1/N$. Consequently, the expected number of tags that will fall in a particular entry of M-III is $\sum_{i=1}^{N_T} \frac{1}{N} = \frac{N_T}{N}$. Since $N > N_T$ by design, the expected number of outdated information in a single entry of M-III is less than one. Therefore, given the redesigned updating procedure described in Section 11.7.1 to prevent desynchronization attacks, the identification complexity of the proposed protocol is constant.

11.8 Summary

In this chapter, we addressed the problem of individual tag identification in large-scale RFID systems. We proposed a protocol that enables the private identification of tags in the system with constant-time complexity. By utilizing the existence of a large storage device in the system, the constant-time identification is achieved by performing the necessary time consuming computations offline (independent of the reader-tag interactions). As opposed to tree-based protocols, the proposed protocol does not further complicate the already challenging problems in RFID systems, namely, collision avoidance and medium access control. Furthermore, tag compromise threats can be mitigated by periodically updating the database which, due to independence of secret parameters amongst tags, can be performed independent of any tag-reader interaction. To the best of our knowledge, this is the first symmetric-key, constant-time identification protocol in the literature of RFID that allows for provably secure mutual authentication and private identification.

Part III

ANONYMOUS WIRELESS SENSOR NETWORKS

Chapter 12

TOWARDS ANONYMOUS WIRELESS SENSOR NETWORKS

12.1 Anonymity in Sensor Networks

Sensor networks are deployed to sense, monitor, and report events of interest in a wide range of applications including, but are not limited to, military, health care, and animal tracking [5, 20, 270]. In many applications, such monitoring networks consist of energy constrained nodes that are expected to operate over an extended period of time, making energy efficient monitoring an important feature for unattended networks. In such scenarios, nodes are designed to transmit information *only* when a relevant event is detected (i.e., *event-triggered* transmission). Consequently, given the location of an event-triggered node, the location of a real event reported by the node can be approximated within the node's sensing range. In the example depicted in Figure 12.1, the locations of the combat vehicle at different time intervals can be revealed to an adversary observing nodes transmissions.

There are three parameters that can be associated with an event detected and reported by a sensor node: the description of the event, the time of the event, and the location of the event. When sensor networks are deployed in untrustworthy environments, protecting the privacy of the three parameters that can be attributed to an event-triggered transmission becomes an important security feature in the design of wireless sensor networks.

While transmitting the "description" of a sensed event in a private manner can be achieved via encryption primitives [72, 204, 116, 45], hiding the timing and spatial information of reported events cannot be achieved via cryptographic means [178, 230]. Encrypting a message before transmission, for instance, can hide the context of the message from unauthorized observers, but the mere existence of the ciphertext is indicative of information transmission.

The source anonymity problem in wireless sensor networks is the problem of studying techniques that provide time and location privacy for events reported by sensor nodes.

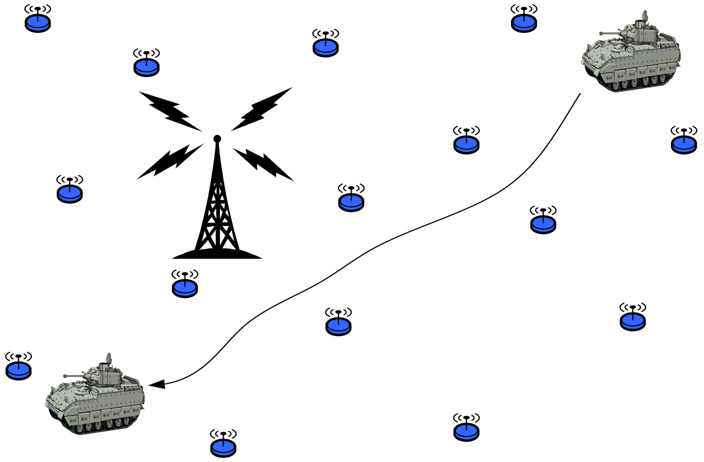


Figure 12.1: A sensor network deployed in a battlefield. Only nodes in close proximity to the combat vehicle are broadcasting information, while other nodes are in sleep mode.

(Time and location privacy will be used interchangeably with source anonymity throughout the chapter.) The source anonymity problem has been drawing increasing research attention recently [135, 196, 194, 256, 160, 266, 113, 178, 230, 268, 156].

In the existing literature, the source anonymity problem has been addressed under two different types of adversaries, namely, local and global adversaries. A local adversary is defined to be an adversary having limited mobility and partial view of the network traffic. Routing-based techniques have been shown to be effective in hiding the locations of reported events against local adversaries [196, 135, 194, 256, 160]. A global adversary is defined to be an adversary with ability to monitor the traffic of the entire network (e.g., coordinating adversaries spatially distributed over the network). Against global adversaries, routing-based techniques are known to be ineffective in concealing location information in event-triggered transmission. This is due to the fact that, since a global adversary has full spatial view of the network, it can immediately detect the origin and time of the event-triggered transmission.

The first step towards achieving source anonymity for sensor networks in the presence of global adversaries is to refrain from event-triggered transmissions [178]. To do that, nodes are required to transmit *fake messages* even if there is no detection of events of interest (*real*

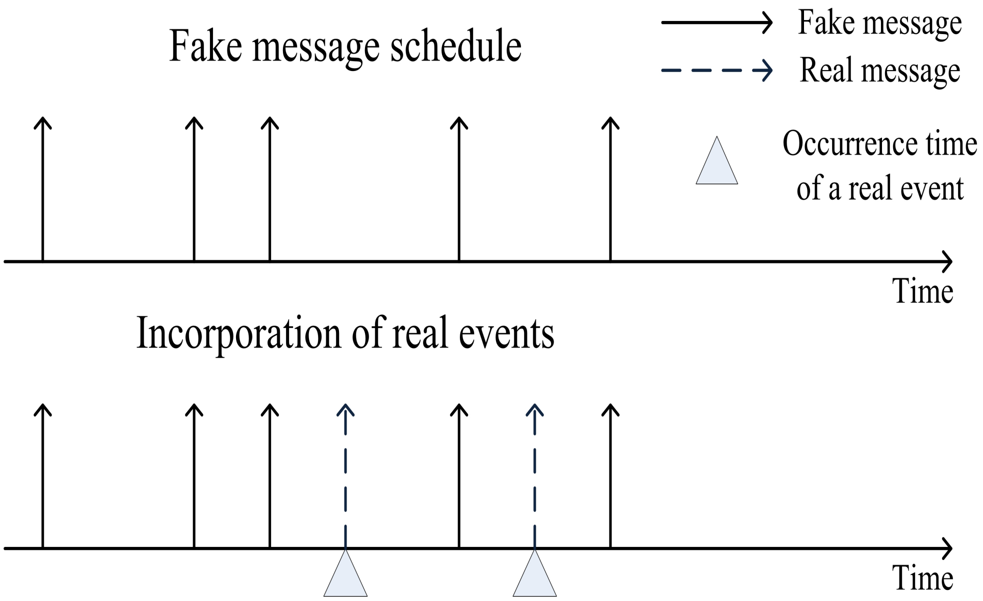


Figure 12.2: Different approaches for embedding the report of real events within a series of fake transmissions; (a) shows the pre-specified distribution of fake transmissions, (b) illustrates how real events are transmitted as soon as they are detected, (c) illustrates how nodes report real events instead of the next scheduled fake message.

events will be used to denote events of interest for the rest of the chapter). When a real event occurs, its report can be embedded within the transmissions of fake messages. Thus, given an individual transmission, an observer cannot determine whether it is fake or real with a probability significantly higher than 1/2, assuming messages are encrypted.

In the above approach, there is an implicit assumption of the use of a probabilistic distribution to schedule the transmission of fake messages. However, the arrival distribution of real events is, in general, time-variant and unknown a priori. If nodes report real events as soon as they are detected (independently of the distribution of fake transmissions), given the knowledge of the fake transmission distribution, statistical analysis can be used to identify outliers (real transmissions) with a probability higher than 1/2, as illustrated in Figure 12.2(b). In other words, transmitting real events as soon as they are detected does not provide source anonymity against statistical adversaries analyzing a series of fake and real transmissions.

One way to mitigate the above statistical analysis is illustrated in Figure 12.2(c). As opposed to transmitting real events as they occur, they can be transmitted instead of the next scheduled fake one. For example, consider programming sensor nodes to deterministically transmit a fake message every minute. If a real event occurs within a minute from the last transmission, its report must be delayed until exactly one minute has elapsed. This approach, however, introduces additional delay before a real event is reported (in the above example, the average delay of transmitting real events is half a minute). When real events have time-sensitive information, such delays might be unacceptable. Reducing the delay of transmitting real events by adopting a more frequent scheduling algorithm is impractical for most sensor network applications since sensor nodes are battery powered and, in many applications, unchargeable. Therefore, a frequent transmission scheduling will drastically reduce the desired lifetime of the sensor network.

The Statistical Source Anonymity (SSA) problem in sensor networks is the study of techniques that prevent global adversaries from exposing source location by performing statistical analysis on nodes transmissions [230, 268, 156, 254, 229, 54, 275]. Practical SSA solutions need to be designed to achieve their objective under two main constraints: minimizing delay and maximizing the lifetime of sensors' batteries.

In this chapter, we investigate the problem of statistical source anonymity in wireless sensor networks. The main contributions of this chapter can be summarized by the following points.

- We introduce the notion of "interval indistinguishability" and illustrate how the problem of statistical source anonymity can be mapped to the problem of interval indistinguishability.

- We propose a quantitative measure to evaluate statistical source anonymity in sensor networks.

- We map the problem of breaching source anonymity to the statistical problem of binary hypothesis testing with nuisance parameters.

- We demonstrate the significance of mapping the problem in hand to a well-studied problem in uncovering hidden vulnerabilities. In particular, realizing that the SSA problem can be mapped to the hypothesis testing with nuisance parameters implies that breaching source anonymity can be converted to finding an appropriate data transformation that removes the nuisance information.

- We analyze existing solutions under the proposed model. By finding a transformation of observed data, we convert the problem from analyzing real-valued samples to binary codes and identify a possible anonymity breach in the current solutions for the SSA problem.

- We pose and answer the important research question of why previous studies were unable to detect the possible anonymity breach identified in this chapter.

- We discuss, by looking at the problem as a coding problem, a new direction to enhance the anonymity of existing SSA solutions.

The rest of the chapter is organized as follows. In Section 12.2, we describe our network and adversarial assumptions. In Section 12.3, we describe the proposed framework. In Section 12.4, we describe the notion of statistical goodness of fit tests and study its use in designing SSA solutions. In Section 12.5, we provide experimental analysis of statistical goodness of fit test based approaches and quantify their anonymity. In Section 12.6 we demonstrate the importance of converting the SSA problem into binary codes for uncovering the hidden vulnerabilities missed by previous studies. In Section 12.7, we suggest a modification of the current solutions and show its effectiveness in improving anonymity. In Section 12.8 we extend the source anonymity problem in sensor networks to illustrate the effect of possible multi-hop analysis. In Section 12.9, we discuss related work and conclude the chapter in Section 12.10.

12.2 Model Assumptions

In this section, we describe the network and adversarial assumption that will be used in this chapter.

12.2.1 Network Model

Communication is assumed to take place in a network of energy constrained sensor nodes. Nodes are deployed to sense events of interest and report them with minimum delay. Consequently, given the location of a certain node, the location of the reported event of interest can be approximated within the node's communication range at the time of transmission. When a node senses an event, it places information about the event in a message and broadcast an encrypted version of the message. To obscure the report of an event of interest, nodes are assumed to broadcast fake messages, even if no event of interest has been detected. Nodes are also assumed to be equipped with a semantically secure encryption algorithm, so that adversaries are unable to distinguish between the reports of events of interest and the fake transmissions by means of cryptographic tests.[1] Furthermore, the network is assumed to be deployed in an unreachable environment and, therefore, the conservation of nodes' energy is a design requirement.

12.2.2 Adversarial Model

The adversarial model used in this chapter is similar to the one considered in [178, 230], in that it is *external*, *passive*, and *global*. An external adversary is an adversary who does not control any of the nodes in the network. As opposed to active adversaries injecting their own traffic or jamming the network, a passive adversary is only capable of observing the network traffic. A global adversary is an adversary who can monitor the traffic of the entire network and can determine the node responsible for the initial transmission reporting an event of interest.

[1] In cryptography, semantic security implies that, given a ciphertext, unauthorized users without the knowledge of the decryption key have no means of distinguishing between two plaintexts in which one of them corresponds to the observed ciphertext [96].

The adversary is assumed to know the locations of all nodes in the networks. The adversary is also assumed to know the distribution of fake message transmissions. Furthermore, the adversary is assumed capable of observing nodes transmissions over extended periods of times and performing sophisticated statistical analysis to compare the observed transmission with the known distribution of fake messages. The adversary, however, is not assumed able to break the security of the encryption algorithm and distinguish the report of event of interests via cryptographic tests.

12.3 Proposed Framework for SSA

In this section, we introduce our source anonymity model for wireless sensor networks. Intuitively, anonymity should be measured by the amount of information about the occurrence time and location of reported events an adversary can extract by monitoring the sensor network. The challenge, however, is to come up with an appropriate model that captures all possible sources of information leakage and a proper way of quantifying anonymity in different systems.

12.3.1 Interval Indistinguishability

Currently, statistical anonymity in sensor networks is modeled by the adversary's ability to distinguish between real and fake transmissions by means of statistical analysis. That is, given a series of transmissions of a certain node, the adversary must be unable to distinguish, with significant confidence, which transmission carries real information and which transmission is fake, regardless of the number of transmissions the adversary may observe.

Consider now an adversary observing a sensor network over multiple *time intervals*. Assume that, during a given time interval, the adversary is able to notice a change in the statistical behavior of transmission times of a certain node in the network. This distinguishable change in the transmission behavior of the node can be indicative of the existence of real activities detected and reported by that node during that interval, even if the adversary was unable to distinguish between individual transmissions.

Consequently, in many applications, modeling source anonymity in sensor networks by

Table 12.1: A list of used terms and notations.

SSA	Statistical Source Anonymity
$\boldsymbol{E_i}$	The random variable representing the type of event reported in the i^{th} transmission (either fake or real)
$\boldsymbol{X_i}$	The random variable representing the inter-transmission time between the i^{th} and the $i+1^{\text{st}}$ transmissions
μ	The desired mean of the $\boldsymbol{X_i}$'s
I_F	A fake interval: an interval consisting of fake events only
I_R	A real interval: an interval containing a mix of real and fake event transmissions
short inter-transmission times	Inter-transmission times that are shorter than the mean of the pre-defined distribution
long inter-transmission times	Inter-transmission times that are longer than the mean of the pre-defined distribution
short-long pattern	A short inter-transmission time followed by a long inter-transmission time

the adversary's ability to distinguish between *individual transmissions* is insufficient to guarantee location privacy. It must be the case that an adversary monitoring the network over multiple *time intervals*, in which some intervals contain real event transmissions and the others do not, is unable to determine, with significant confidence, which of the intervals contain the real traffic. Formally, the notion of interval indistinguishability can be defined as follows [10].

Definition 8 (Interval Indistinguishability)**.** *Let I_F denotes a time interval* without *any real event transmission (called the "fake interval" for the rest of the paper), and I_R denotes a time interval* with *real event transmissions (called the "real interval" for the rest of the paper). The two time intervals are said to be statistically indistinguishable if the distri-*

butions of inter-transmission times during these two intervals cannot be distinguished with significant confidence.

12.3.2 Interval versus Event Indistinguishability

This section illustrates the relation between the traditional anonymity notion (i.e., individual event indistinguishability) and the proposed anonymity notion (i.e., interval indistinguishability). First, observe that as the length of intervals decreases, interval indistinguishability approaches event indistinguishability. If each interval consists of a single transmission, interval indistinguishability is equivalent to event indistinguishability.

However, in the more general scenario, in which intervals contain more than a single transmission, interval indistinguishability implies indistinguishability of individual transmissions. To see this, assume a system satisfying interval indistinguishability but does not satisfy individual event indistinguishability. Since real and fake transmissions are distinguishable, given a fake interval and a real interval, the real interval can be identified as the one with the real transmission; a contradiction to the hypothesis that the system satisfies interval indistinguishability. That is, if intervals are indistinguishable, then individual events within them must also be indistinguishable.

In fact, the notion of interval indistinguishability is strictly stronger than the traditional notion individual event indistinguishability. That is, while interval indistinguishability implies individual indistinguishability, the converse is not true in general. This will be shown in Section 12.5 by demonstrating that there exist schemes that achieve high levels of individual indistinguishability while failing to achieving satisfactory levels of interval indistinguishability.

12.3.3 Mapping Statistical Source Anonymity to Binary Hypothesis Testing

In binary hypothesis testing, given two hypothesis, H_0 and H_1, and a data sample that belongs to one of the two hypotheses (e.g., a bit transmitted through a noisy communication channel), the goal is to decide to which hypothesis the data sample belongs. In the statistical strong anonymity problem under interval indistinguishability, given an interval of inter-transmission times, the goal is to decide whether the interval is fake or real (i.e., consists of fake transmissions only or contains real transmissions).

Given Definition 8 of interval indistinguishability, consider the following game between a challenger, $\mathcal{C}$ (the system designer), and a statistical adversary, $\mathcal{A}$.

Game 3 (Anonymity Game)**.**

1. *$\mathcal{C}$ chooses two intervals I_R and I_F, in which I_R is a real interval and I_F is a fake one.*
2. *$\mathcal{C}$ draws a bit $b \in \{0,1\}$ uniformly at random and sets $I_R = I_b$ and $I_F = I_{\overline{b}}$, where $\overline{b}$ denotes the binary complement of b.*
3. *$\mathcal{C}$ gives I_b and $I_{\overline{b}}$ to $\mathcal{A}$.*
4. *$\mathcal{A}$ makes any statistical test of her choice on I_b and $I_{\overline{b}}$ and outputs a bit b'.*
5. *If $b' = b$, $\mathcal{A}$ wins the game.*

Game 3 can be viewed as a standard binary hypothesis testing problem. That is, given two hypotheses (a real interval and a fake interval) and an observed data (an interval of inter-transmission times of a sensor node), the goal of the adversary is to determine to which hypothesis the observed data belongs (i.e., whether the observed interval contains real event transmissions).

Remark 17. *Although giving the adversary two intervals might seem too strong of an assumption, it is actually a practical one. To see this, note that the adversary can always observe multiple time intervals, two for instance. Then, all that is needed is to analyze these two observed intervals. If they are distinguishable, then it is likely that one of them is a real interval and the other is fake. Moreover, an adversary can discover the distribution of fake intervals by monitoring a node in the absence of real events. Then, all that is needed is to observe different time intervals. The more distinguishable a time interval from the known fake interval, the more likely it is to contain real events. Therefore, Game 3 is suitable to analyze practical systems.*

12.3.4 Quantifying Statistical Source Anonymity

With Definition 8 and Game 3, we aim to find a security measure that can formally quantify the anonymity of different systems. Let σ denote any adversarial strategy for breaching the anonymity of the system. Let $\Pr[b' = b]_\sigma$ denote the adversary's probability of winning Game 3 using strategy σ. We quantify the anonymity of a sensor network against the strategy σ by

$$\Lambda_\sigma := 1 - 2 \left| \Pr[b' = b]_\sigma - 0.5 \right|. \tag{12.1}$$

In the best case scenario, from the challenger's standpoint, the adversary's strategy is a pure random guess; leading to $\Pr[b' = b]_\sigma = 1/2$ and $\Lambda_\sigma = 1$ (absolute anonymity). In the worst case, the adversary will have a strategy with $\Pr[b' = b]_\sigma = 1$ leading to $\Lambda_\sigma = 0$ (no anonymity). Any intelligent strategy will result in a probability of winning the game belonging to the interval $[0.5, 1]$, leading to an anonymity measure in the interval $[0, 1]$.

Now, let Σ be the set of all possible adversarial strategies to breach the anonymity of the sensor network. Then, we define the anonymity of the system as:

$$\Lambda := \min_{\sigma \in \Sigma} \Lambda_\sigma, \tag{12.2}$$

where Λ_σ is as defined in equation (12.1).

With the above definition of interval indistinguishability, we introduce the notion of Λ-anonymity in sensor networks.

Definition 9 (Λ-anonymity)**.** *A wireless sensor network is said to be Λ-anonymous if it satisfies two conditions*

1. *the anonymity of the system, as defined in equation (12.2), is at least Λ,*

2. *there is no distinguishable transitional behavior between intervals.*

The second condition in Definition 9 ensures that the adversary is unable to infer when an interval starts or when it ends. This is necessary since an adversary with the knowledge that a node is transitioning from one interval to another will infer that either real events have started to arrive or stopped from arriving. In either case, source anonymity can be breached. In Table 12.1, the terms and notations that will be used throughout the chapter are listed.

12.4 Statistical Goodness of Fit Tests and the SSA Problem

In the literature, statistical source anonymity are shown to be achieved via the use of statistical goodness of fit tests [230, 268, 156, 254, 229, 54, 275]. In this section, we describe the current use of statistical goodness of fit tests in designing anonymous sensor networks.

12.4.1 SSA Solutions Based on Statistical Goodness of Fit Tests

The statistical goodness of fit of an observed data describes how well the data fits a given statistical model. Measures of goodness of fit typically summarize the discrepancy between observed values and the values expected under the statistical model in question. Such measures can be used, for example, to test for normality of residuals, to test whether two samples are drawn from identical distributions, or to test whether outcome frequencies follow a specified distribution. Examples of well-studied goodness of fit tests include, but are not limited to, the Anderson-Darling (A-D) test [19], the Kolmogorov-Smirnov (K-S) test [174], the Jarque-Bera (J-B) test [123].

The following is a description of how statistical goodness of fit tests have been used to design anonymous sensor networks. Let sensor nodes be designed to transmit independent identically distributed (iid) fake messages according to a pre-specified probabilistic distribution, $\mathcal{D}$, with a desired mean, μ. Furthermore, let nodes store a sliding window of times between consecutive transmissions (inter-transmission times), say $\boldsymbol{X_i}, \boldsymbol{X_{i+1}}, \cdots, \boldsymbol{X_{k+i-1}}$, where $\boldsymbol{X_j}$ is the random variable representing the time between the j^{th} and the $j+1^{\text{st}}$ transmissions, and k is the length of the sliding window.

Assume that, after the $k+i^{\text{th}}$ transmission, a real event is detected. Ideally, the inter-transmission time for reporting the detected event, represented by $\boldsymbol{X_{k+i}}$, should be a random variable drawn from $\mathcal{D}$ independently of all the $\boldsymbol{X_j}$'s. To minimize delay, however, consider the following use of a statistical goodness of fit test. Let $\boldsymbol{Y}$ be a random variable drawn from $\mathcal{D}$ and let $\boldsymbol{X_{k+i}} = \boldsymbol{Y} - \epsilon$, where ϵ is defined to be the largest positive number such that the sequence of random variables in the sliding window, $\{\boldsymbol{X_i}, \cdots, \boldsymbol{X_{k+i}}\}$, passes the statistical goodness of fit test for a sequence following the distribution $\mathcal{D}$. That is, an adversary recording the sequence of inter-transmission times will observe a sequence that is statistically indistinguishable from an iid sequence of random variables with the pre-specified distribution of fake transmissions.

Observe, however, that by continuing in the same fashion of transmitting real event as soon as possible the mean of the probabilistic distribution will skew away from the mean of the desired distribution of only fake transmissions, μ, since nodes always favor shorter times to transmit real events. To adjust the mean, the inter-transmission time between the report of the real event and next transmission, $\boldsymbol{X_{k+i+1}}$ in this example, will be purposely delayed. That is, let $\boldsymbol{Y}$ be a random variable drawn from $\mathcal{D}$ and set $\boldsymbol{X_{k+i+1}} = \boldsymbol{Y} + \delta$, where δ is defined to be the largest positive number such that the sequence of random variables in the sliding window, $\{\boldsymbol{X_{i+1}}, \cdots, \boldsymbol{X_{k+i+1}}\}$, passes the statistical goodness of fit test for a sequence following the distribution $\mathcal{D}$. Then, as shown in [230], an adversary observing the sensor node cannot differentiate between real and fake transmissions. Figure 12.3 illustrates an instance of this approach.

12.4.2 Statistical Goodness of Fit Under Interval Indistinguishability

As discussed in Section 12.3.1, when an adversary can distinguish between real and fake intervals, source location can be exposed, even if the adversary cannot distinguish between individual transmissions. In this section, we analyze statistical goodness of fit based solutions under the proposed model of interval indistinguishability.

As before, let $\boldsymbol{X_i}$ be the random variable representing the time between the i^{th} and the $i+1^{\text{st}}$ transmissions and let the desired mean of these random variables be μ; i.e., $\mathbb{E}[\boldsymbol{X_i}] = \mu$, for all i (since the $\boldsymbol{X_i}$'s are iid). We now examine two intervals, a fake interval and a real one.

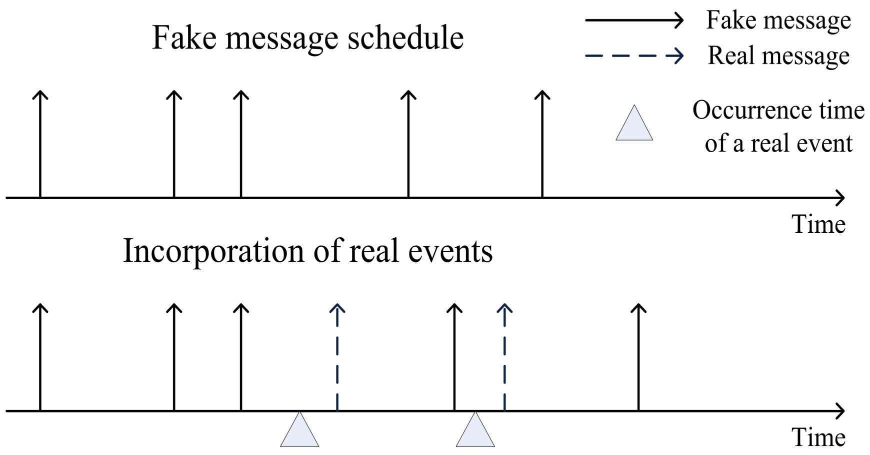


Figure 12.3: An illustration of solutions based on statistical goodness of fit tests. Nodes transmit fake messages according to a pre-specified probabilistic distribution and maintain a sliding window of inter-transmission times. When a real event occurs, it is transmitted as soon as possible under the condition that the samples in the sliding window maintain the designed distribution. The transmission following the real transmission is delayed to maintain the mean of the distribution of inter-transmission times in the sliding window.

12.4.2.1 Fake Interval (I_F)

Recall that, in the absence of real events, nodes are programmed to transmit iid fake messages according to a pre-specified probability distribution. That is, the $\boldsymbol{X_i}$'s in fake intervals are iid random variables with mean μ. Therefore, during any fake interval, I_F, for any $\boldsymbol{X_{i-1}}, \boldsymbol{X_i} \in I_F$, one gets

$$\mathbb{E}\big[\boldsymbol{X_i} \mid \boldsymbol{X_{i-1}} < \mu\big] = \mu, \tag{12.3}$$

by the fact that $\boldsymbol{X_{i-1}}$ and $\boldsymbol{X_i}$ are independent by definition and that $\mathbb{E}[\boldsymbol{X_j}] = \mu$, for all j's.

12.4.2.2 Real Interval (I_R)

By definition, real intervals will have both fake and real transmissions. Let $\boldsymbol{E_i}$ be the random variable representing the type of the event reported in the i^{th} transmission, i.e., fake or real. Then, $\boldsymbol{E_i}$ can take the values R and F, where R denotes a real event and F denotes a fake one. Since, in the most general scenario, the distribution of inter-arrival times of real events can be time-variant and unknown beforehand, we will assume that $\boldsymbol{E_i}$ can take the values R and F with arbitrary probabilities.

Recall that the time between the transmission of a real event and its preceding fake one is usually shorter than the mean, μ, by design (to reduce delay). Recall further that the time between the transmission of a real event and its successive one is usually longer than μ by design (to adjust the ensemble mean). That is, during any real interval, I_R, for any $\boldsymbol{X_{i-1}}, \boldsymbol{X_i} \in I_R$, one gets

$$\mathbb{E}\big[\boldsymbol{X_i} \mid \boldsymbol{X_{i-1}} < \mu, \boldsymbol{E_i} = R\big] > \mu, \tag{12.4}$$

and,

$$\mathbb{E}\big[\boldsymbol{X_i} \mid \boldsymbol{X_{i-1}} < \mu, \boldsymbol{E_i} = F\big] = \mu, \tag{12.5}$$

by design. Combining equations (12.4) and (12.5) one gets

$$\begin{aligned}\mathbb{E}\big[\boldsymbol{X_i} \mid \boldsymbol{X_{i-1}} < \mu\big] &= \mathbb{E}\big[\boldsymbol{X_i} \mid \boldsymbol{X_{i-1}} < \mu, \boldsymbol{E_i} = R\big] \cdot \Pr[\boldsymbol{E_i} = R] \\ &\qquad + \mathbb{E}\big[\boldsymbol{X_i} \mid \boldsymbol{X_{i-1}} < \mu, \boldsymbol{E_i} = F\big] \cdot \Pr[\boldsymbol{E_i} = F] \qquad (12.6) \\ &> \mu \cdot \Pr[\boldsymbol{E_i} = R] + \mu \cdot \Pr[\boldsymbol{E_i} = F] \qquad (12.7) \\ &= \mu. \qquad (12.8)\end{aligned}$$

An inter-transmission time can be either shorter or longer than μ.[2] For the rest of the chapter, we call an inter-transmission time that is shorter than μ "short inter-transmission time" and an inter-transmission time that is longer than μ "long inter-transmission time".

Equation (12.8) implies that short inter-transmission times are most likely to be followed by long inter-transmission times during real intervals. Therefore, by equations (12.3) and (12.8), short inter-transmission times followed by long inter-transmission times occur more frequently in real intervals than fake intervals (for the rest of the chapter, a short-long pattern will be used to denote a short inter-transmission time followed by a long inter-transmission time). Figure 12.4 illustrates the sort-long patterns.

[2] Since inter-transmission times are typically drawn from continuous random variables, the probability of an inter-transmission time to be equal to the mean, μ, is zero.

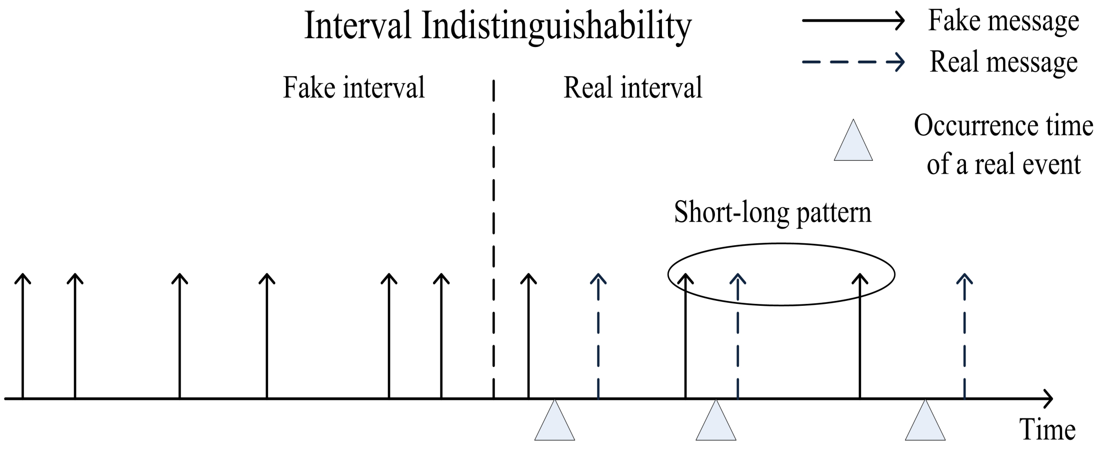


Figure 12.4: An illustration of interval distinguishability in the current state-of-the-art solutions based on statistical goodness of fit tests. Real events are transmitted sooner than what is determined by the probabilistic distribution, while the transmission following the real event is later than what is determined by the probabilistic distribution to fix the mean of the pre-defined distribution.

12.4.3 Questions Arising from our Analysis

Our analysis in the previous section shows that real and fake intervals in approaches based on statistical goodness of fit tests can be theoretically distinguishable. This raises the following question: *can the analysis in Section 12.4.2 be applied in practical scenarios?* If the presented analysis is indeed applicable in practical setups, then the next questions will be: *what is the mathematical explanation for the seemingly contradicting results of Section 12.4.2 and prior studies acknowledging the effectiveness of statistical goodness of fit tests in designing anonymous systems? That is, how can one explain the fact that the use of statistical goodness of fit is known to be secure in the literature while the analysis of Section 12.4.2 state otherwise?* The answers to these questions will be the main focus of Sections 12.5 and 12.6, respectively. First we provide experimental analysis in an attempt to investigate the first question.

12.5 Experimental Analysis of SSA Solutions based on Statistical Goodness of Fit

The use of statistical goodness of fit tests in designing anonymous sensor networks was pioneered by Shao et al. in [230] and followed by schemes that build on it or acknowledge its effectiveness in providing secure SSA for sensor networks, such as [268, 156, 254, 229, 54, 275]. In this section, we analyze schemes based on statistical goodness of fit tests using the ideas implied by the theoretical analysis of Section 12.4.2.

12.5.1 Converting Real-Valued Samples to Binary Codes

Let every inter-transmission time that is shorter than the mean μ be represented by the binary digit '0', and every inter-transmission time that is longer than the mean μ be represented by the binary digit '1'. That is, given a sequence of real-valued inter-transmission times $X = \{x_1, \cdots, x_n\}$, the function g is applied to every inter-transmission time as follows:

$$g(x_i) = \begin{cases} 1, & \text{if } x_i > \mu \\ 0, & \text{if } x_i \leq \mu \end{cases} \tag{12.9}$$

for each $i = 1, \cdots, n$. (We use g to denote the indicator function instead of the commonly used notation, I, since I is already used to denote an interval.) Then, the real-valued sequence, X, is transformed into a binary code as follows:

$$f\big(X\big) = f\big(\{x_1, \cdots, x_n\}\big) = \{g(x_1), \cdots, g(x_n)\}. \tag{12.10}$$

Observe that this is the same transformation used implicitly in Section 12.4.2. That is, short-long patterns will be represented by the ordered sequence '01'. Next, we describe the statistical measure that will be used in our experimental analysis of SSA solutions based on statistical goodness of fit tests.

12.5.2 Correlation Measure for Binary Hypothesis Testing

In this section, we specify the statistical measure that will be used to perform our experimental analysis of SSA approaches based statistical goodness of fit tests. Let $X = \{x_1, \cdots, x_n\}$

and $Y = \{y_1, \cdots, y_n\}$ be two sequences of length n. Define the correlation coefficient of the two sequences by:

$$\rho(X,Y) = \frac{\mid n\sum_{i=1}^{n} x_i y_i - (\sum_{i=1}^{n} x_i)(\sum_{i=1}^{n} y_i) \mid}{\sqrt{\left(n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2\right)\left(n\sum_{i=1}^{n} y_i^2 - (\sum_{i=1}^{n} y_i)^2\right)}}, \tag{12.11}$$

where x_i and y_i denote the i^{th} elements of sequences X and Y, respectively. It can be verified that the value of ρ is always in the interval $[0, 1]$ [98]. When X and Y are uncorrelated, ρ will be equal to zero. The higher value of ρ, the more the two sequences are correlated.

12.5.3 Correlation Analysis of SSA Solutions Based on Statistical Goodness of Fit Tests

The interpretation of the analysis of Section 12.4.2 in terms of the transformation of the previous section is that each bit in a binary code representing a fake interval is independent of the all other bits, while bits in a binary code representing a real interval are correlated. More specifically, a binary code representing a real interval is likely to have more '01' patterns than a binary code representing a fake interval. This suggests to the following approach to distinguish between fake and real intervals. First, generate a "reference" binary code of the form

$$\mathsf{Ref} = \{0, 1, 0, 1, \cdots, 0, 1\}. \tag{12.12}$$

Now, let I_0 and I_1 be two time intervals in which one of them contains real event transmissions and the other does not. Let S_0 and S_1 be the two sequences of real-valued inter-transmission times corresponding to I_0 and I_1, respectively. Let $X_0 = f(S_0)$ and $X_1 = f(S_1)$ be the conversion of S_0 and S_1 into their corresponding binary codes according to the transformation of Section 12.5.1. Correlate X_0 and X_1 with the reference code of equation (12.12); the binary code having a higher correlation coefficient with the reference code is the one corresponding to the real interval.

In the context of Game 3, given two intervals I_0 and I_1 in which one is real and the other is fake, the adversary's decision is given by:

$$D(I_0, I_1) = \begin{cases} 0, & \text{if } \rho(\mathsf{Ref}, X_0) > \rho(\mathsf{Ref}, X_1) \\ \gamma, & \text{if } \rho(\mathsf{Ref}, X_0) = \rho(\mathsf{Ref}, X_1) \\ 1, & \text{if } \rho(\mathsf{Ref}, X_0) < \rho(\mathsf{Ref}, X_1) \end{cases}, \tag{12.13}$$

where γ denotes any decisional strategy to break a tie. That is, the interval corresponding to the binary code that is more correlated to the reference code is decided to be the real one.

12.5.3.1 Experimental Parameters and Setup

In this section, We specify our parameters selection and setup our experimental analysis of approaches based on statistical goodness of fit tests.

Inter-transmission times between fake transmissions are chosen to be iid exponentials with a rate parameter $\lambda = 20$. Real events arrive according to a Poisson Arrival process with mean 1/20. The Anderson-Darling (A-D) goodness of fit test is used to determine the transmission times of real events and the mean recovery algorithm. The two parameters of the A-D test are the significance level of the test and the allowed deviation from the mean which are set to 0.05 and 0.1, respectively.[3]

The experiment was run for 10,000 independent trials. Each trial consists of two intervals, a real one, I_R, and a fake one, I_F. Every trial starts with a "warm-up" period, where 200 iid exponential random variables with rate 20 are drawn to constitute a backlog to be used in the A-D goodness of fit test. Then real events start arriving and they are transmitted according to the procedure described in Section 12.4.1 (interested readers may refer to [230] for more detailed algorithms of the transmission mechanism). Each real interval consists of 50 real events. After the 50th real event has been transmitted, the fake interval starts for the same amount of time the real interval lasted.

For each of the 10,000 independent trials, denote by $S_R^{(i)}$ the sequence of inter-transmission times of the real interval of the i^{th} trial and, similarly, denote by $S_F^{(i)}$ the sequence of inter-transmission times of the fake interval of the i^{th} trial. The numbers in $S_R^{(i)}$ and $S_F^{(i)}$

[3] These are the same parameters appeared in [230].

will be real-valued that are indistinguishable from iid exponential random variables. Let $X_R^{(i)} = f\big(S_R^{(i)}\big)$ and $X_L^{(i)} = f\big(S_L^{(i)}\big)$, where f is the function defined in equation (12.10), be the binary conversion of the real-valued inter-transmission times of the real and fake intervals of the i^{th} trial.

Following the decision rule in equation (12.13), we correlate $X_R^{(i)}$ and $X_F^{(i)}$ with the reference sequence Ref for all $i = 1, \cdots, 10,000$. Intuitively, the test is said to be successful in distinguishing between real and fake intervals in the i^{th} trial if $\rho(\mathsf{Ref}, X_R^{(i)}) > \rho(\mathsf{Ref}, X_F^{(i)})$ and unsuccessful if $\rho(\mathsf{Ref}, X_R^{(i)}) < \rho(\mathsf{Ref}, X_F^{(i)})$. When $\rho(\mathsf{Ref}, X_R^{(i)}) = \rho(\mathsf{Ref}, X_F^{(i)})$ one of the intervals is chosen to be the real one uniformly at random.

12.5.3.2 Experimental Results and Anonymity Interpretation

Out of the $10,000$ independent trials, the following results were obtained:

- $\rho(\mathsf{Ref}, X_R^{(i)}) > \rho(\mathsf{Ref}, X_F^{(i)})$ in $7,301$ trials;

- $\rho(\mathsf{Ref}, X_R^{(i)}) < \rho(\mathsf{Ref}, X_F^{(i)})$ in $2,695$ trials;

- $\rho(\mathsf{Ref}, X_R^{(i)}) = \rho(\mathsf{Ref}, X_F^{(i)})$ in 4 trials.

Now, consider Game 3 for analyzing interval indistinguishability. Given two intervals I_0 and I_1 at which one of them is real and one is fake, let the adversary's strategy for deciding which is which be according to the decision rule in equation (12.13). Then, given the simulation results provided above, the adversary's probability of correctly identifying real intervals is 0.730. In other words, the anonymity of the system is at most $\Lambda = 0.539$, significantly far away from the desired $\Lambda \approx 1$ claimed and acknowledged in prior studies such as [230, 268, 156, 254, 229, 54, 275].

12.6 Explanation for Discrepancies Between Our Results and Prior Studies

The results of Section 12.5 provide an answer to the first question raised in Section 12.4.3. Namely that the analysis of Section 12.4.2 can improve the adversary's chances of distinguishing real from fake intervals and, ultimately, breaching the anonymity of the system in practical setups is possible. Now, it remains to investigate the rest of the questions raised in Section 12.4.3. Namely, is there a contradiction between our results and previous studies and, if not, how can we explain such discrepancies mathematically. The keys to answer such questions are "interval indistinguishability" and "nuisance information". We start by a brief background.

12.6.1 Nuisance Parameters

In statistical decision theory, the term "nuisance parameters" refers to information that is not needed for hypothesis testing and, further, can preclude a more accurate decision making [223]. When performing hypothesis testing of data with nuisance parameters, it is desired (even necessary in some scenarios) to find an appropriate transformation of the data that removes or minimizes the effect of the nuisance information before performing the hypothesis testing [223]. That is, given a data sample $X = (x_1, \cdots, x_n)$ that belongs to one of two possible hypotheses H_0 or H_1, the test is performed on a transformation of the data sample, $f(X)$, rather than the original data itself, X. The transformation function, f, is an application dependent and choosing the right function is a critical step in hypothesis testing with nuisance parameters [223].

12.6.2 Significance of Interval Indistinguishability and Nuisance Removal

The answer to the seeming contradiction between the results of Section 12.5 and previous studies can be divided into two parts. First, previous studies model statistical source anonymity by the adversary's ability to distinguish between *individual* transmissions. That is, given a sequence of inter-transmission times, the adversary is shown to be unable to determine which transmission is fake and which one is real. The interval indistinguishability notion introduced in this chapter,[4] on the other hand, assumes that source anonymity can be breached when adversaries can successfully distinguish between real and fake *intervals*.

Observe that no tool in our analysis is introduced to allow the adversary to infer which transmission is real and which one is fake within the real interval itself. That is, if the

[4] Recall that, by Corollary 12.3.2, interval indistinguishability implies individual transmission indistinguishability.

analysis of Section 12.5 is repeated with the assumption that anonymity is breached only if the adversary can distinguish between individual fake and real transmissions, the anonymity of the system will be different the the obtained 0.539 (it might very well be close to the desired $\Lambda \approx 1$ since we do not present any mechanism to distinguish between individual transmissions). Therefore, the notion of interval indistinguishability is essential in explaining the discrepancies between our results and prior studies that model SSA by the adversary's ability to distinguish between individual real and fake transmissions.

Interval indistinguishability alone, however, does not explain why our results are different than what is believed in prior work. That is, even though different statistical tools are used to measure anonymity (statistical goodness of fit tests are used to analyze anonymity in previous studies while we use the correlation measure specified in Section 12.5.2), the difference in the used statistical measure does not explain the discrepancies between our results and prior work.

The conversion of real-valued inter-transmission times into binary codes is the main reason for the differences between our anonymity results of Section 12.5 and prior studies. The conversion to binary codes is a key-enabling tool for the removal of nuisance information precluding successful hypothesis testing. The following experimental analysis demonstrate the significance of the binary code conversion.

12.6.2.1 Experimental Parameters and Setup

In order to examine the effect of binary code conversion for nuisance removal, the experimental analysis of Section 12.5 is repeated with the real-valued inter-transmission times as opposed to their binary transformation. $10,000$ independent trials are performed. In very trial, a real interval, $S_R^{(i)}$, and a fake interval, $S_F^{(i)}$, are generated with the same parameters of Section 12.5.3.1. In each trial, the two intervals are correlated with a reference sequence using the formula in equation (12.11). However, as opposed to the binary reference code of equation (12.12), one real interval, $\mathsf{Ref}_{\mathrm{rv}}$, that serves as a reference sequence of real-valued inter-transmission times is generated as a reference sequence.

12.6.2.2 Experimental Results and Anonymity Interpretation

Out of the $10,000$ independent trials, the following results were obtained:

- $\rho(\mathsf{Ref}_{\mathrm{rv}}, S_R^{(i)}) > \rho(\mathsf{Ref}_{\mathrm{rv}}, S_F^{(i)})$ in $5,076$ trials;
- $\rho(\mathsf{Ref}_{\mathrm{rv}}, S_R^{(i)}) < \rho(\mathsf{Ref}_{\mathrm{rv}}, S_F^{(i)})$ in $4,924$ trials;
- $\rho(\mathsf{Ref}_{\mathrm{rv}}, S_R^{(i)}) = \rho(\mathsf{Ref}_{\mathrm{rv}}, S_F^{(i)})$ in 0 trials.[5]

Under the same adversarial strategy of deciding which interval is real and which is fake given in equation (12.13), the system is 0.984-anonymous using real-valued inter-transmission times. This result agrees with previous studies in that the sequences corresponding to any trial, whether real or fake, are statistically indistinguishable from iid exponential random variables. On the other hand, when the same system is analyzed using the binary code conversion of inter-transmission times it was only 0.539-anonymous. The importance of this result is that it shows how the actual lengths of inter-transmission times can act as nuisance and prevent accurate hypothesis testing.

The results of this section conclude our explanations to the questions posed in Section 12.4.3. In particular, the results show that there is no contradiction between the results obtained and acknowledged in prior studies and our result of Section 12.4.2, and that the combination of the interval indistinguishability model and the existence of nuisance information is the mathematical explanation for such seemingly contradicting results.

12.7 Improving SSA via Induced Correlation in Fake Intervals

Our analysis of SSA solutions based on statistical goodness of fit tests shows that the use of such statistical tools is insufficient to guarantee source anonymity. In particular, not only the real-valued inter-transmission times must be indistinguishable from the desired distribution of fake transmissions, but also the binary codes representing the inter-transmission times of fake and real intervals must have indistinguishable statistical properties. In what follows, we describe a modification to approaches based on statistical goodness of fit tests to improve

[5] this is expected since we are dealing with real valued inter-transmission times in this case

their anonymity. The main idea behind the proposed approach is the attempt to induce the same correlation pattern of inter-transmission times during real intervals into inter-transmission times during fake intervals.

12.7.1 The Proposed Approach

As can be seen from the analysis in Section 12.4.2, inter-transmission times during fake intervals are iid's, while inter-transmission times during real intervals are neither independent nor identically distributed. In theory, the only way to guarantee that a sequence of random variables is statistically indistinguishable from a given iid sequence is to generate it as an iid sequence with the same distribution.

The notion of interval indistinguishability, suggests a different approach for the design of anonymous sensor networks. Observe that Definition 8 of interval indistinguishability does not impose any requirements, such as iid, on the distribution of inter-transmission times during fake intervals. Therefore, designing fake intervals with the distribution that is easiest to emulate during real intervals is the most logical solution. This idea opens the door for more solutions as it gives more flexibility for system designers.

To improve anonymity, we suggest introducing the same correlation of inter-transmission times during real intervals to inter-transmission times during fake intervals. That is, let the transmission procedure consists of two different algorithms: A_R and A_F. In the presence of real events (i.e., in real intervals), algorithm A_R is implemented. In the absence of real events (i.e., in fake intervals), algorithm A_F is implemented. Algorithm A_R is the same as the algorithm described in Section 12.4.1. In algorithm A_F, the nodes generates two sets of events independently of each other: “dummy events” and fake events. Fake events serve the same purpose they serve in algorithm A_R, that is, they are used to hide the existence of real transmissions. Since there are no real events in fake intervals, however, *dummy events* are generated to be handled as if they are real events. That is, dummy events are generated independently of fake messages and, upon their generation, their transmission times are determined according to the used statistical goodness of fit test.

The purpose of this procedure is to introduce the same correlation of real intervals into fake intervals. That is, not only the two sequences of inter-transmission times will be statistically indistinguishable by means of statistical goodness of fit tests, but also the binary codes representing fake and real intervals will have the same statistical behavior. (There is more to be done to decide how nodes switch from algorithm A_R to A_F and vice versa, but since this is not the main focus of this chapter, we defer detailed discussion to future investigation that converts the solution to coding problem.)

12.7.2 Experimental Parameters and Setup

The same experimental analysis of Section 12.5 is performed with one major difference. To make fake intervals possess the same correlation of real intervals, we implemented the A_F algorithm described above. Dummy events were generated according to iid Gaussian inter-arrival times with mean 0.05 seconds and a variance of 0.02. (We reemphasize the distinction between fake messages and dummy events: fake messages are the ones transmitted to hide the existence of real transmissions, while dummy events are the ones generated, during fake intervals only, to resemble the existence of real events.) Note that the inter-arrival distribution of dummy events is purposely different than the inter-arrival distribution of real events to count for the general case of unknown distribution of real events inter-arrivals. The A-D test is used in both algorithms, A_R and A_F, to determine the transmission times of real events and dummy events, respectively.

12.7.3 Experimental Results and Anonymity Interpretations

By running the experiment for $10,000$ independent trials, the following observations were recorded.

- $\rho(\mathsf{Ref}, X_R^{(i)}) > \rho(\mathsf{Ref}, X_F^{(i)})$ in $5,161$ trials;
- $\rho(\mathsf{Ref}, X_R^{(i)}) < \rho(\mathsf{Ref}, X_F^{(i)})$ in $4,832$ trials;
- $\rho(\mathsf{Ref}, X_R^{(i)}) = \rho(\mathsf{Ref}, X_F^{(i)})$ in 7 trials.

In terms of the anonymity measure of equation (12.1), the system is 0.967-anonymous under the adversarial strategy of equation (12.13). Observe the improvement in anonymity

Table 12.2: A quantitative comparison of the statistical goodness of fit test based approach of Section 12.4.1 after the transformation of Section 12.5.1 (i.e., without nuisance), the statistical goodness of fit test based approach of Section 12.4.1 without the transformation of Section 12.5.1 (i.e., with nuisance), and our improved SSA solution of Section 12.7 after the transformation of Section 12.5.1 (i.e., without nuisance). $\rho_R > \rho_F$ denotes larger correlation coefficient in real intervals, $\rho_R < \rho_F$ denotes larger correlation coefficient in fake intervals, while $\rho_R = \rho_F$ denotes equal correlation coefficient in real and fake intervals. The simulation results are obtained from $10,000$ independent trials.

	$\rho_R > \rho_F$	$\rho_R < \rho_F$	$\rho_R = \rho_F$	Anonymity bound
Statistical goodness of fit based approach (without nuisance)	$7,301$	$2,695$	4	0.539
Statistical goodness of fit based approach (with nuisance)	$5,076$	$4,924$	0	0.984
Our modified approach (without nuisance)	$5,161$	$4,832$	7	0.967

against correlation attacks in our modified version (from 0.539 without the use of dummy events to 0.967 when dummy events are used). Table 12.2 summarizes our experimental results.

12.7.4 Performance of the Solution

Compared to the original SSA scheme described in Section 12.4.1, the solution presented in this section induces more computational overhead. That is, while the original scheme described in Section 12.4.1 requires nodes to perform statistical goodness of fit tests during real intervals only, the solution of this section involves the use of statistical goodness of fit test in both real and fake intervals. Note, however, that the solution of this section does not involve extra communication overhead, only rescheduling of fake transmissions that must be sent anyway. This is an important observation since communication consumes orders of magnitude more energy than computations (depending on hardware, transmitting one bit may consume up to 2,900 times the energy consumed by performing one instruction) [137].

We emphasize, however, that this solution is merely presented to illustrate how to improve the anonymity of approaches based on statistical goodness of fit tests. The main focus of this work is to come up with a framework that can be used to design and analyze anonymous sensor networks. Using the proposed framework, including the mapping of the problem of statistical source anonymity to coding theory, in order to design more efficient schemes that satisfy the notion of interval indistinguishability is an open research problem.

12.8 Effect of Network Topology on Source Anonymity

So far, anonymity discussions were restricted to single-hop analysis. However, since the adversary, by assumption, has a global view of the network, the adversary can utilize his/her knowledge of the network's topology to increase the advantage of exposing secret location information. In this section, we bring the network's topology into the picture to illustrate the importance of increasing the anonymity of each node.

Assume the network is deployed to monitor a moving target. Assume further that a global adversary will have a 55% chance of distinguishing between real and fake intervals. In some scenarios, a 0.45 probability of false alarm (the probability that the adversary has concluded a certain interval is real while it is fake) can be considered high enough to prevent the adversary from taking the risk. Since the adversary has a global view of the network, however, he/she can correlate the analysis to the next hop by monitoring adjacent sensor nodes.

Consider the example of Figure 12.5(a) and assume the adversary's chance of distinguishing between real and fake intervals of each node's transmissions is 55%. In such a scenario, according to equation (12.1), the anonymity of each node is $\Lambda = 0.9$. The monitored target, however, is moving and node b will start reporting its existence. On average, the adversary will also have a 55% chance of breaking the anonymity of node b. Combining the observations from node a and b, the anonymity is reduced to be $\Lambda^2 = 0.81$. Consequently, by the time the target reaches node e, the anonymity is already reduced to 0.59. That is, the anonymity of a moving target is an exponentially decreasing function of the

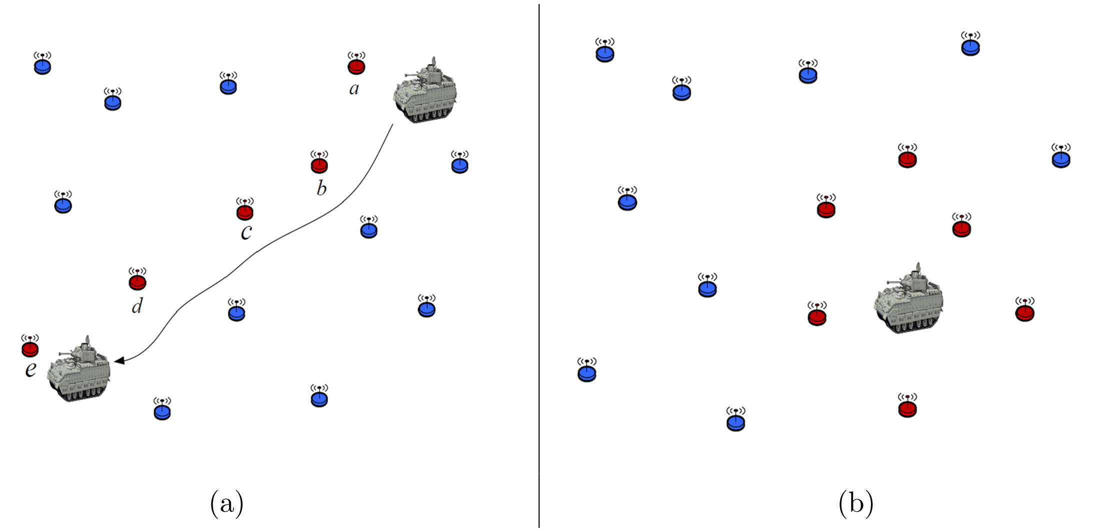


Figure 12.5: (a) An example of a sensor networks monitoring a moving target. As the tank moves along its path, nodes a, b,c, d, and e report that the tank is within their sensing range. (b) An example of multiple sensor nodes reporting a stationary event. Six nodes are simultaneously reporting that the tank is within their sensing range.

number of hops reporting its proximity.

In a different direction, consider the case in which multiple nodes are reporting the same event simultaneously, as depicted in Figure 12.5(b). Then, even if the target is stationary, the anonymity is reduced to $\Lambda^6 = 0.53$ (assuming the anonymity of each node is 0.9).

Therefore, unless the anonymity of each node is $\Lambda = 1$, or if there is a multi-hop anonymous design, global adversaries can substantially increase their advantages of breaking the anonymity of the sensor network by utilizing their knowledge of the network topology and performing multi-hop analysis.

12.9 Related Work

The privacy problem in wireless sensor networks comes in different flavors. Proposals dealing with providing sink anonymity in wireless sensor networks have appeared in, e.g., [100, 225, 190, 189, 166]. Network coding based approaches that protect against traffic analysis have appeared in, e.g., [26, 74, 125]. The privacy problem most relevant to this work is the source location privacy in wireless sensor networks. Li et al. presented a state-of-the-art survey on privacy preservation in wireless sensor networks [156].

The source location privacy in sensor networks is part of a broader area, the design of anonymous communication systems. The foundation for this field was laid by Chaum in [57], and since then has become a very active area of research. In particular, topics related to location anonymity have been discussed by Reed et al. in [213], who introduced the idea of preserving anonymity through onion routing, and by Gruteser and Grunwald in [99], who discussed ways to provide anonymity in location-based services, such as Global Positioning Systems.

In wireless sensor networks, much of the work in source location privacy assumes a passive, local eavesdropper operating close to the base station. Privacy is maintained in such models through anonymous routing. The location privacy problem was first introduced in [135, 196]. The local eavesdropper model was introduced and the authors demonstrated that existing routing methods were insufficient to provide location privacy in this environment. They also proposed a phantom flooding scheme to solve the problem. In [266], Xi et al. proposed a new random walk routing method that reduces energy consumption at the cost of increased delivery time. Path confusion has also been proposed as an anonymity-preserving routing scheme by Hoh and Gruteser in [113]. In [194], Ouyang et al. developed a scheme in which cycles are introduced at various points in the route, potentially trapping the adversary in a loop and forcing the adversary to waste extra resources. In [254], Wang et al. proposed a technique to maximize source location privacy by designing routing protocols that distribute message flows to different routes.

However, in the global adversarial model, in which the adversary has access to all transmissions in the network, routing-based schemes are insufficient to provide location privacy [178, 230]. The global adversarial model was first introduced by Mehta et al. in [178]. The authors motivated the problem, analyzed the security of existing routing-based schemes under the new model, and proposed two new schemes. In the first scheme, some sensor nodes act as fake sources by mimicking the behavior of real events. For example, if the network is deployed to track an animal, the fake sources could send fake messages with a distribution resembling that of the animal's movements. This, however, assumes some knowledge of the time distribution of real events. In the second scheme, packets (real and fake) are sent either at constant intervals or according to a predetermined probabilistic schedule. Although this scheme provides perfect location privacy, it also introduces undesirable performance characteristics, in the form of either relatively high delay or relatively high communication and computational overhead. The scheme of [230] was proposed to address this delay/overhead tradeoff.

In [230], Shao et al. introduced the notion of statistically strong source anonymity in which a global adversary with ability to monitor the traffic in the entire network is unable to infer source locations by performing statistical analysis on the observed traffic. In order to realize their notion of statistical anonymity, nodes are programmed to transmit fake events according to pre-specified distribution. More specifically, after the transmission of every fake event, the node draws an exponentially distributed random variable $t \sim \text{Exp}(\lambda)$, where λ is the pre-specified rate of the exponential distribution. The node then waits for t time units and then transmits another fake event. That is, in the absence of real event transmissions, an adversary monitoring the sensor node will observe inter-transmission times that are iid exponentials with mean $\mu = 1/\lambda$.

Upon the occurrence of real events, the goal of a sensor node is to transmit them while maintaining the exponential distribution of the inter-transmission times. Obviously, if nodes delay their transmission of real events to the next scheduled fake transmission, no statistical test can be used to distinguish between real and fake events (since inter-transmission times are kept exponential iid's with the same rate). The goal in [230], however, is to minimize the latency of reporting real events while maintaining statistical indistinguishability between real and fake transmissions.

To reduce the latency, the authors of [230] proposed the following procedure: let imd_i represent the inter-transmission time between the i^{th} and the $i+1^{\text{st}}$ transmissions. Assume a real event has occurred after the transmission of the i^{th} event. Given $\{imd_1, imd_2, \ldots, imd_i\}$, imd_{i+1}, the time after the transmission of the i^{th} event the node must wait before it can transmit the real event, is determined as follows: imd_{i+1} is the smallest positive value such that the sequence $\{imd_1, imd_2, \ldots, imd_i, imd_{i+1}\}$ passes the Anderson-Darling (A-D) goodness of fit test [238] for a sequence of iid exponentials with mean μ.

Observe, however, that on average $imd_{i+1} < \mu$ since imd_{i+1} is, by definition, the minimum value that passes the test. Therefore, continuing in this fashion will cause the mean of the entire sequence to skew away from desired mean.

To solve the problem of mean deviation described above, the scheme in [230] includes a mean recovery algorithm. The mean recovery algorithm outputs a delay δ and the time between the transmission of a real event and the following event (fake or real) is set to $imd_{i+2} = t + \delta$, where $t \sim \text{Exponential}(\lambda)$. The scheme in [230] is designed so that the sequence $\{imd_1, \ldots, imd_n\}$, where n is the last transmitted message, always passes the A-D goodness of fit test.

To reduce the amount of traffic in the network that is due to the transmission of fake events, techniques based on node proxies and data aggregation have been proposed [268, 267]. In such techniques, the overall communication overhead is reduced by making intermediate nodes act as proxies that filter out fake messages or by aggregating multiple messages in a single transmission. Such approaches make schemes based on generating fake messages more attractive by mitigating the high communication overhead issue.

Shao et al. also consider the problem of an active adversary in [231]. Their adversary also has the ability to perform node compromise attacks, and they develop tools to prevent the adversary from gaining access to event data stored in a node even if the adversary possesses that node's secret keys.

In recent works, Li and Ren [159] proposed a scheme to provide both content confidentiality and source-location privacy through routing to a randomly selected intermediate node (RRIN) and a network mixing ring (NMR), where the RRIN provides local source location privacy and NMR yields network-level (global) source location privacy. Ouyang et al. [195] proposed four schemes: naive, global, greedy, and probabilistic to protect the source location against global adversaries in. Abbasi et al. [2] proposed a distributed algorithm to mix real event traffic with carefully chosen dummy traffic to hide the real event traffic pattern.

12.10 Summary

In this chapter, we provided a statistical framework based on binary hypothesis testing for modeling, analyzing, and evaluating statistical source anonymity in wireless sensor networks. We introduced the notion of interval indistinguishability to model source location privacy. We showed that the current approaches for designing statistically anonymous systems introduce correlation in real intervals while fake intervals are uncorrelated. By mapping the problem of detecting source information to the statistical problem of binary hypothesis testing with nuisance parameters, we showed why previous studies were unable to detect the source of information leakage that was demonstrated in this chapter. Finally, we proposed a modification to existing solutions to improve their anonymity against correlation tests.

Future extensions to this work include mapping the problem of statistical source anonymity to coding theory in order to design an efficient system that satisfies the notion of interval indistinguishability.

Chapter 13

CONCLUSION

In this chapter, we conclude the dissertation and discuss future work plans. The conclusion is divided into three sections representing the three main parts of the dissertation.

13.1 Message Authentication

In Chapter 4, we studied the generic composition of authenticated encryption systems. We introduced $\mathcal{E}$-MACs, a new symmetric-key cryptographic primitive that can be used in the construction of E&M and MtE compositions. By taking advantage of the E&M and MtE structures, the use of $\mathcal{E}$-MACs is shown to improve the efficiency and security of the authentication operation. More precisely, since the message to be authenticated is encrypted, universal hash functions based $\mathcal{E}$-MACs can designed without the need to apply cryptographic operations on the compressed image, since this can be replaced by operations performed by the encryption algorithm. Further, by appending a random string at the end of the plaintext message, $\mathcal{E}$-MAC can be secured against key-recovery attacks.

In Chapter 5, a different approach has been taken to design message authentication codes that can be used in the construction of the E&M and MtE generic composition. The fact that the ciphertext, as opposed to the plaintext, is transmitted to the intended receiver along with the authentication tag is used to propose a new security model to analyze MACs in the E&M or MtE compositions. Furthermore, the security of the underlying encryption algorithm is utilized to reduce the computations performed by the MAC algorithm. In particular, another instance of $\mathcal{E}$-MACs (called KMAC) is proposed. KMAC is the first secure keyless MAC in the cryptographic literature and is $O(\log n \ \log\log n)$ faster than the fastest MAC (where n is the block size). Furthermore, KMAC is shown to construct an authenticated encryption system that is faster than existing authenticated encryption primitives.

In Chapter 6, we proposed yet another method that utilizes the security of encryption algorithms to design more efficient authentication. We introduced the p-var authenticated encryption scheme. The authentication tag in p-var is computed using modular multiplication. However, as opposed to relying solely on the secret key to grant message integrity, we utilize the existence of the encryption algorithm to change the field under which the computation is performed. By doing so, we show that regardless of the length of the message to be authenticated, only the residue of the message (in its integer representation) modulo the used prime needs to be authenticated, without affecting the integrity of the entire message. To the best of our knowledge, this is the first MAC in which message integrity is preserved without the need to process the entire message by the authentication algorithm.

In Chapter 7, we investigated authentication based on a class of universal hash-function families that have been appeared in the literature. Although the studied universal hash-function family has appeared in many places, computations have always been performed modulo prime integers. In this work, we analyzed the security of message authentication when computations are performed over arbitrary finite integer rings. We derived a direct relation between the security of authentication and the underlying integer ring $\mathbb{Z}_n$. Specifically, we showed that the bound on successful forgery is proportional to the reciprocal of the smallest prime factor of the used modulus n.

13.2 Radio Frequency Identification

In Chapter 9, we explored a new direction towards solving the identity authentication problem in RFID systems. We break the RFID authentication process into two main problems: message authentication and random number generation. For parties equipped with a good source of randomness and a secure cryptographic primitive to authenticate messages, the literature of cryptography is rich with well-studied solutions for secure identity authentication. However, the two operations, random number generation and message authentication, can be expensive for low-cost RFID tags. In Chapter 9, we laid down the foundations of a new direction towards solving these problems in RFID systems. We proposed an unconditionally secure direction for authenticating RFID systems. We used the fact that RFID readers are computationally powerful devices to design a protocol that allows RFID readers to deliver random numbers to RFID tags in an unconditionally secure manner. Then, by taking advantage of the information-theoretic security of the transmitted messages, we developed a novel unconditionally secure message authentication code that is computed with a single multiplication operation. The main goal of this idea was to bring more research to the design of such unconditionally secure protocols, as opposed to the computationally secure protocols that have been proposed extensively, for the purpose of suiting the stringent computational capabilities of low-cost devices.

In Chapter 10, a new technique for authenticating short encrypted messages is proposed. The fact that the message to be authenticated must also be encrypted is used to deliver a random nonce to the intended receiver via the ciphertext. This allowed the design of an authentication code that benefits from the simplicity of unconditionally secure authentication without the need to manage one-time keys. In particular, it has been demonstrated in this chapter that authentication tags can be computed with one addition and a one modular multiplication. Given that messages are relatively short, addition and modular multiplication can be performed faster than existing computationally secure MACs in the literature of cryptography. When devices are equipped with block ciphers to encrypt messages, a second technique that utilizes the fact that block ciphers can be modeled as strong pseudorandom permutations is proposed to authenticate messages using a single modular addition.

In RFID literature, most *privacy-preserving* protocols require the reader to search, linearly, all tags in the system in order to identify a single tag. In another class of protocols, extra communication is traded-off to reduce the search complexity to be logarithmic in the number of tags, but it has two major drawbacks: it requires a large communication overhead over the fragile wireless channel, and the compromise of a tag in the system reveals secret information about other, uncompromised, tags in the same system. In Chapter 11, we took a different approach to address time-complexity of private identification in large-scale RFID systems. We utilized the special architecture of RFID systems to propose the first symmetric-key privacy-preserving authentication protocol for RFID systems with *constant-time* identification complexity. Instead of increasing communication overhead, the existence of a large storage device in RFID systems, the database, is utilized for improving the time efficiency of tag identification. That is, we used a larger, yet practical, database that is designed to perform all time consuming computations offline (independent of reader-tag interactions) so that, upon receiving tags' responses, they can be identified in constant-time.

13.3 Wireless Sensor Networks

In Chapter 12, we provided a statistical framework based on binary hypothesis testing for modeling, analyzing, and evaluating statistical source anonymity in wireless sensor networks. We introduced the notion of interval indistinguishability to model source location privacy. We showed that the current approaches for designing statistically anonymous systems introduce correlation in real intervals while fake intervals are uncorrelated. By mapping the problem of detecting source information to the statistical problem of binary hypothesis testing with nuisance parameters, we showed why previous studies were unable to detect the source of information leakage that was demonstrated in this chapter. Finally, we proposed a modification to existing solutions to improve their anonymity against correlation tests.

13.4 Future Research Directions

As hardware technology continues to evolve, the use of small devices to form communication networks is bound to expand. Based on the studies in this dissertation, the search for appropriate techniques to secure communications in such networks is fertile and emerging. In addition, as we move towards nanotechnology and nanonetworks, the need for new primitives can only increase. Not only continuing some of the current wireless security and applied cryptography research is of research interest in the future, but also the expansion to emerging areas of (a) cognitive networks, (b) personal health care networks, (c) distributed storage and computing systems, and (d) the emerging area of cyber-physical systems.

This dissertation's research in the security of energy constrained networks will be transitioned into a detailed investigation of the security of cyber-physical systems (CPSs). CPSs are unique in their ability to directly couple and control objects in the physical world with cyber components and will become a new paradigm in critical infrastructure protection, smart grids, personal/home-based health care, automated manufacturing, energy distribution, avionics, and national security applications. The unique feature of physical influence and control in CPSs emphasizes the importance of understanding robustness to errors and attacks that have direct consequences in the physical world.

Cognitive radio networking is another emerging fertile area where the security has to be guaranteed not only for spectrum sensing but also operation of multiple protocols and new incentive mechanisms. At present, security research in this area is focused on the sending of white spaces and identification of primary user with no investigation on other aspects. The security of this emerging field is of special interest in future research.

Also of interest is investigating the security and privacy in the emerging area of Cloud Computing. Cloud computing brings challenges and concerns for security and privacy in the short and long terms as data and computations are outsourced to potentially untrusted service providers.

BIBLIOGRAPHY

[1] http://www.westerndigital.com/en/.

[2] A. Abbasi, A. Khonsari, and M. Talebi. Source location anonymity for sensor networks. In *Proceedings of the 6th IEEE Conference on Consumer Communications and Networking Conference – CCNC'09*, pages 588–592. IEEE Communications Society, 2009.

[3] V. Afanassiev, C. Gehrmann, and B. Smeets. Fast message authentication using efficient polynomial evaluation. In *Proceedings of the 4th International Workshop on Fast Software Encryption – FSE'97*, volume 1267 of *Lecture Notes in Computer Science*, pages 190–204. Springer, 1997.

[4] M. Agrawal, N. Kayal, and N. Saxena. PRIMES is in P. *Annals of Mathematics*, 160(2):781–793, 2004.

[5] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.

[6] K. Albrecht and L. McIntyre. Consumers Against Supermarket Privacy Invasion and Numbering – CASPIAN. `http://www.spychips.com/`.

[7] B. Aloamir and R. Poovendran. Information Theoretically Secure Encryption with Almost Free Authentication. *Journal of Universal Computer Science*, 15(15):2937–2956, 2009.

[8] B. Aloamir and R. Poovendran. Unconditionally Secure Authenticated Encryption with Shorter Keys. In *Proceedings of the 7th International Workshop on Security in Information Systems – WOSIS'09*, pages 3–15. INSTICC Press, 2009.

[9] B. Alomair, A. Clark, J. Cuellar, and R. Poovendran. Scalable RFID Systems: a Privacy-Preserving Protocol with Constant-Time Identification. In *Proceedings of the 40th Annual IEEE/IFIP International Conference on Dependable Systems and Networks – DSN'10*, pages 1–10. IEEE Computer Society, 2010.

[10] B. Alomair, A. Clark, J. Cuellar, and R. Poovendran. Statistical Framework for Source Anonymity in Sensor Networks. In *Proceedings of the 53rd IEEE Global Communications Conference – GLOBECOM'10*, pages 1–6. IEEE Communications Society, 2010.

[11] B. Alomair, A. Clark, and R. Poovendran. The power of primes: security of authentication based on a universal hash-function family. *Journal of Mathematical Cryptology*, 4(2):121–147, 2010.

[12] B. Alomair, L. Lazos, and R. Poovendran. Passive attacks on a class of authentication protocols for RFID. In *Proceedings of the 10th International Conference on Information Security and Cryptology – ICISC'07*, volume 4817 of *Lecture Notes in Computer Science*, pages 102–115. Springer, 2007.

[13] B. Alomair, L. Lazos, and R. Poovendran. Securing Low-cost RFID Systems: an Unconditionally Secure Approach. In *The Asia Workshop Proceedings on Radio Frequency Identification System Security – RFIDsec'10*, volume 4 of *Cryptology and Information Security Series*, pages 1–17. IOS Press, 2010.

[14] B. Alomair, L. Lazos, and R. Poovendran. Securing Low-cost RFID Systems: an Unconditionally Secure Approach. *Journal of Computer Security*, 19(2):229–256, 2011.

[15] B. Alomair and R. Poovendran. On the Authentication of RFID Systems with Bitwise Operations. In *Proceedings of the 2nd IFIP International Conference on New Technologies, Mobility and Security – NTMS'08*, pages 1–6. IEEE Xplore, 2008.

[16] B. Alomair and R. Poovendran. Efficient Authentication for Mobile and Pervasive Computing. In *the 12th International Conference on Information and Communications Security – ICICS'10*, volume 6476 of *Lecture Notes in Computer Science*, pages 186–202. Springer, 2010.

[17] B. Alomair and R. Poovendran. $\mathcal{E}$-MACs: Towards More Secure and More Efficient Constructions of Secure Channels. In *Proceedings of the 13th International Conference on Information Security and Cryptology – ICISC'10*, Lecture Notes in Computer Science. Springer, 2010.

[18] B. Alomair and R. Poovendran. Privacy versus Scalability in Radio Frequency Identification Systems. *Computer Communications*, 33(18):2155–2163, 2010.

[19] T. Anderson and D. Darling. Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes. *The Annals of Mathematical Statistics*, 23(2):193–212, 1952.

[20] T. Arampatzis, J. Lygeros, and S. Manesis. A survey of applications of wireless sensors and wireless sensor networks. In *Proceedings of the 13th IEEE Mediterranean Conference on Control and Automation – MED'05*, pages 719–724. IEEE Control System Society, 2006.

[21] G. Avoine. Privacy issues in RFID banknote protection schemes. In *Proceedings of the 6th Smart Card Research and Advanced Application IFIP Conference – CARDIS'04*, pages 33–48. Kluwer Academic Publishers, 2004.

[22] G. Avoine. Adversarial model for radio frequency identification. Technical Report LASEC-REPORT-2005-001, Swiss Federal Institute of Technology (EPFL), Security and Cryptography Laboratory (LASEC), 2005.

[23] G. Avoine, X. Carpent, and B. Martin. Strong Authentication and Strong Integrity (SASI) is not that Strong. In *Proceeding of the 6th Workshop on RFID Security – RFIDSec'10*, volume 6370 of *Lecture Notes in Computer Science*, pages 50–64. Springer, 2010.

[24] G. Avoine, I. Coisel, and T. Martin. Time Measurement Threatens Privacy-Friendly RFID Authentication Protocols. In *Proceedings of the 6th Workshop on RFID Security and Privacy – RFIDsec'10*, volume 6370 of *Lecture Notes in Computer Science*, pages 138–157. Springer, 2010.

[25] G. Avoine, E. Dysli, and P. Oechslin. Reducing time complexity in RFID systems. In *Proceedings of the 12th International Workshop on Selected Areas in Cryptography – SAC'05*, volume 3897 of *Lecture Notes in Computer Science*, pages 291–306. Springer, 2005.

[26] E. Ayday, F. Delgosha, and F. Fekri. Location-aware security services for wireless sensor networks using network coding. In *Proceedings of the 26th IEEE International Conference on Computer Communications – INFOCOM'07*, pages 1226–1234. IEEE Communications Society, 2007.

[27] M. Bellare. New proofs for NMAC and HMAC: Security without collision-resistance. In *Proceedings of the 16th Annual International Cryptology Conference – CRYPTO'06*, volume 4117 of *Lecture Notes in Computer Science*, pages 602–619. Springer, 2006.

[28] M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. In *Proceedings of the 16th Annual International Cryptology Conference – CRYPTO'96*, volume 1109 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 1996.

[29] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption. In *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science – FOCS'97*, pages 394–403. IEEE Computer Society, 1997.

[30] M. Bellare, R. Guerin, and P. Rogaway. XOR MACs: New methods for message authentication using finite pseudorandom functions. In *Proceedings of the 15th Annual*

International Cryptology Conference – CRYPTO'95, volume 963 of *Lecture Notes in Computer Science*, pages 15–28. Springer, 1995.

[31] M. Bellare, J. Kilian, and P. Rogaway. The Security of the Cipher Block Chaining Message Authentication Code. *Journal of Computer and System Sciences*, 61(3):362–399, 2000.

[32] M. Bellare, T. Kohno, and C. Namprempre. Authenticated encryption in SSH: provably fixing the SSH binary packet protocol. In *Proceedings of the 9th ACM Conference on Computer and Communications Security – CCS'02*, pages 1–11. ACM SIGSAC, 2002.

[33] M. Bellare, T. Kohno, and C. Namprempre. Breaking and provably repairing the SSH authenticated encryption scheme: A case study of the Encode-then-Encrypt-and-MAC paradigm. *ACM Transactions on Information and System Security*, 7(2):241, 2004.

[34] M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. *Journal of Cryptology*, 21(4):469–491, 2008.

[35] M. Bellare, K. Pietrzak, and P. Rogaway. Improved security analyses for CBC MACs. In *Proceedings of the 25th Annual International Cryptology Conference – CRYPTO'05*, volume 3621 of *Lecture Notes in Computer Science*, pages 527–545. Springer, 2005.

[36] M. Bellare and P. Rogaway. Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient cryptography. In *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security – ASIACRYPT'00*, volume 1976, pages 317–330. Springer, 2000.

[37] M. Bellare, P. Rogaway, and D. Wagner. The EAX mode of operation. In *Proceedings of the 11th International Workshop on Fast Software Encryption – FSE'04*, volume 3017 of *Lecture Notes in Computer Science*, pages 389–407. Springer, 2004.

[38] S. Bellovin. Problem areas for the IP security protocols. In *Proceedings of the 6th Usenix Security Symposium – USENIX'96*, pages 205–214. Usenix Association, 1996.

[39] D. Bernstein. Floating-point arithmetic and message authentication. Unpublished manuscript, 2004. Available at `http://cr.yp.to/hash127.html`.

[40] D. Bernstein. The Poly1305-AES message-authentication code. In *Proceedings of the 12th International Workshop on Fast Software Encryption – FSE'05*, volume 3557 of *Lecture Notes in Computer Science*, pages 32–49. Springer, 2005.

[41] T. Berson. Differential cryptanalysis mod 2 32 with applications to MD5. In *Proceedings of the 7th Workshop on the Theory and Application of Cryptographic Techniques – EUROCRYPT'92*, volume 658, pages 71–80. Springer, 1993.

[42] J. Black, S. Halevi, H. Krawczyk, T. Krovetz, and P. Rogaway. UMAC: Fast and Secure Message Authentication. In *Proceedings of the 19th Annual International Cryptology Conference – CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 482–499. Springer, 1999.

[43] J. Black and P. Rogaway. A block-cipher mode of operation for parallelizable message authentication. In *Proceedings of the 21st International Conference on the Theory and Applications of Cryptographic Techniques – EUROCRYPT'02*, volume 2332 of *Lecture Notes in Computer Science*, pages 384–397. Springer, 2002.

[44] L. Blum, M. Blum, and M. Shub. A Simple Unpredictable Pseudo-random Number Generator. *SIAM Journal on Computing*, 15(2):364, 1986.

[45] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. Robshaw, Y. Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In *Proceedings of the 9th International Workshop on Cryptographic Hardware and Embedded Systems – CHES'07*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, 2007.

[46] A. Bogdanov, G. Leander, C. Paar, A. Poschmann, M. Robshaw, and Y. Seurin. Hash Functions and RFID Tags : Mind The Gap. In *Proceedings of the 10th International Workshop on Cryptographic Hardware and Embedded Systems – CHES'08*, volume 5154 of *Lecture Notes in Computer Science*, pages 283–299. Springer, 2008.

[47] A. Bosselaers, R. Govaerts, and J. Vandewalle. Fast hashing on the Pentium. In *Proceedings of the 16th Annual International Cryptology Conference – CRYPTO'96*, volume 1109 of *Lecture Notes in Computer Science*, pages 298–312. Springer, 1996.

[48] J. Bringer and H. Chabanne. Trusted-HB: A Low-Cost Version of HB^+ Secure Against Man-in-the-Middle Attacks. *IEEE Transactions on Information Theory*, 54(9):4339–4342, 2008.

[49] M. Burmester, B. de Medeiros, and R. Motta. Anonymous RFID authentication supporting constant-cost key-lookup against active adversaries. *Journal of Applied Cryptography*, 1(2):79–90, 2008.

[50] M. Burmester, B. De Medeiros, and R. Motta. Robust, anonymous RFID authentication with constant key-lookup. In *Proceedings of the 3rd ACM Symposium on Information, Computer and Communications Security – ASIACCS'08*, pages 283–291. ACM SIGSAC, 2008.

[51] D. Burton. *Elementary Number Theory.* McGraw Hill, 2002.

[52] S. Callegari, R. Rovatti, and G. Setti. Embeddable ADC-based true random number generator for cryptographic applications exploiting nonlinear signal processing and chaos. *IEEE Transactions on Signal Processing*, 53(2):793–805, 2005.

[53] H. Canetti and H. Krawczyk. Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels. In *Proceedings of the 20th International Conference on the Theory and Application of Cryptographic Techniques – EUROCRYPT'01*, volume 2045 of *Lecture Notes in Computer Science*, pages 451–472. Springer, 2001.

[54] B. Carbunar, Y. Yu, W. Shi, M. Pearce, and V. Vasudevan. Query privacy in wireless sensor networks. *ACM Transactions on Sensor Networks*, 6(2):1–34, 2010.

[55] J. Carter and M. Wegman. Universal classes of hash functions. In *Proceedings of the 9th ACM Symposium on Theory of Computing – STOC'77*, pages 106–112. ACM SIGACT, 1977.

[56] L. Carter and M. Wegman. Universal hash functions. *Journal of Computer and System Sciences*, 18(2):143–154, 1979.

[57] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[58] J. H. Cheon, J. Hong, and G. Tsudik. Reducing RFID Reader Load with the Meet-in-the-Middle Strategy. Cryptology ePrint Archive, Report 2009/092, 2009.

[59] H.-Y. Chien. SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity. *IEEE Transactions on Dependable and Secure Computing*, 4(4):337–340, 2007.

[60] S. Contini and Y. Yin. Forgery and partial key-recovery attacks on HMAC and NMAC using hash collisions. In *Proceedings of the 12th International Conference on the Theory and Application of Cryptology and Information Security – ASIACRYPT'06*, volume 4284 of *Lecture Notes in Computer Science*, pages 37–53. Springer, 2006.

[61] T. Cover and J. Thomas. *Elements of Information Theory.* Wiley-Interscience New York, 2006.

[62] M. De Soete. Some constructions for authentication-secrecy codes. In *Proceedings of the 7th Workshop on the Theory and Application of Cryptographic Techniques – EUROCRYPT'88*, volume 330 of *Lecture Notes in Computer Science*, pages 57–75, 1988.

[63] B. Defend, K. Fu, and A. Juels. Cryptanalysis of two lightweight RFID authentication schemes. In *Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications – PerCom'07*, pages 211–216. IEEE Computer Society Press, 2007.

[64] T. Dierks and E. Rescorla. RFC 5246: The transport layer security (TLS) protocol version 1.2. Technical report, Internet Engineering Task Force, 2008.

[65] D. Dolev, C. Dwork, and M. Naor. Nonmalleable cryptography. *SIAM journal on computing*, 30(2):391–437, 2001.

[66] N. Doraswamy and D. Harkins. *IPSec: the new security standard for the Internet, intranets, and virtual private networks.* Prentice Hall, 2003.

[67] D. Dummit and R. Foote. *Abstract algebra.* Wiley, 1999.

[68] M. Dworkin. Recommendation for block cipher modes of operation: The CMAC mode for authentication. National Institute of Standards and Technology (NIST) Special Publication 800-38B, 2005.

[69] M. Dworkin. Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. National Institute for Standards and Technology (NIST) Special Publication 800-38D, 2007.

[70] D. Eastlake and P. Jones. US secure hash algorithm 1 (SHA1), 2001.

[71] H. Edwards. *Riemann's zeta function.* Dover Pubns, 2001.

[72] L. Eschenauer and V. Gligor. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM Conference on Computer and Communications Security – CCS'02*, pages 41–47. ACM SIGSAC, 2002.

[73] M. Etzel, S. Patel, and Z. Ramzan. Square hash: Fast message authentication via optimized universal hash functions. In *Proceedings of the 19th Annual International Cryptology Conference – CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 234–251. Springer, 1999.

[74] Y. Fan, Y. Jiang, H. Zhu, and X. Shen. An efficient privacy-preserving scheme against traffic analysis attacks in network coding. In *Proceedings of the 28th IEEE International Conference on Computer Communications – INFOCOM'09*, pages 19–25. IEEE Communications Society, 2009.

[75] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer. Strong Authentication for RFID Systems using the AES Algorithm. In *Proceedings of the 6th International Workshop on Cryptographic Hardware and Embedded Systems – CHES'04*, volume 3156 of *Lecture Notes in Computer Science*, pages 357–370. Springer, 2004.

[76] M. Feldhofer and C. Rechberger. A Case Against Currently Used Hash Functions in RFID Protocols. The 2nd Workshop on RFID Security – RFIDsec'06, 2006.

[77] M. Feldhofer, J. Wolkerstorfer, and V. Rijmen. AES Implementation on a Grain of Sand. *IEE Proceedings – Information Security*, 152(1):13–20, 2005.

[78] W. Feller. *An Introduction to Probability Theory and its Applications.* Wiley India Pvt. Ltd., 2008.

[79] N. Ferguson, D. Whiting, B. Schneier, J. Kelsey, and T. Kohno. Helix: Fast encryption and authentication in a single cryptographic primitive. In *Proceedings of the 10th International Workshop on Fast Software Encryption – FSE'03*, volume 2887 of *Lecture Notes in Computer Science*, pages 330–346. Springer, 2003.

[80] FIPS 113. Computer Data Authentication. Federal Information Processing Standards Publication, 1985.

[81] FIPS 198. The Keyed-Hash Message Authentication Code (HMAC). Federal Information Processing Standards Publication, 2002.

[82] C. Floerkemeier, R. Schneider, and M. Langheinrich. Scanning with a Purpose-Supporting the Fair Information Principles in RFID Protocols. In *Proceedings of the 2nd International Symposium on Ubiquitous Computing Systems – UCS'04*, volume 3598 of *Lecture Notes in Computer Science*, pages 214–231. Springer, 2004.

[83] A. Francillon, C. Castelluccia, and P. Inria. TinyRNG: A cryptographic random number generator for wireless sensors network nodes. In *Proceedings of the 5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops – WiOpt'07*, pages 1–7, 2007.

[84] A. Freier, P. Karlton, and P. Kocher. The SSL Protocol Version 3.0. Internet Engineering Task Force (IETF), 2011.

[85] M. Fürer. Faster integer multiplication. In *Proceedings of the 39th ACM Symposium on Theory of Computing – STOC'07*, pages 57–66. ACM SIGACT, 2007.

[86] F. D. Garcia and P. van Rossum. Modeling Privacy for Off-line RFID Systems. The 5th Workshop on RFID Security – RFIDSec'09, 2009.

[87] S. Garfinkel, A. Juels, and R. Pappu. RFID Privacy: An Overview of Problems and Proposed Solutions. *IEEE Security & Privacy Magazine*, 3(3):34–43, 2005.

[88] E. Gilbert, F. MacWilliams, and N. Sloane. Codes which detect deception. *Bell System Technical Journal*, 53(3):405–424, 1974.

[89] H. Gilbert, M. Robshaw, and Y. Seurin. Good Variants of HB+ are Hard to Find. In *Proceedings of the the 12th International Conference on Financial Cryptography and Data Security – FC'08*, volume 5143 of *Lecture Notes in Computer Science*, pages 156–170. Springer, 2008.

[90] H. Gilbert, M. Robshaw, and Y. Seurin. HB#: Increasing the Security and Efficiency of HB. In *Proceedings of the 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques – EUROCRYPT'08*, volume 4965 of *Lecture Notes in Computer Science*, pages 361–378. Springer, 2008.

[91] H. Gilbert, M. Robshaw, and H. Sibert. Active attack against HB^+: a provably secure lightweight authentication protocol. *IET Electronics Letters*, 41(21):1169–1170, 2005.

[92] V. Gligor and P. Donescu. Integrity-Aware PCBC Encryption Schemes. In *Proceedings of the 7th International Workshop on Security Protocols – SP'99*, volume 1796 of *Lecture Notes in Computer Science*, pages 153–168. Springer, 2000.

[93] V. Gligor and P. Donescu. Fast Encryption and Authentication: XCBC Encryption and XECB Authentication Modes. In *Proceedings of the 9th International Workshop on Fast Software Encryption – FSE'01*, volume 2355 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2002.

[94] O. Goldreich. *Foundations of Cryptography.* Cambridge University Press, 2001.

[95] S. Goldwasser and J. Kilian. Almost all primes can be quickly certified. In *Proceedings of the 18th ACM Symposium on Theory of Computing – STOC'86*, pages 316–329. ACM SIGACT, 1986.

[96] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[97] P. Golle, M. Jakobsson, A. Juels, and P. Syverson. Universal re-encryption for mixnets. In *Proceedings of the Cryptographers' Track at the RSA Conference – CT-RSA'04*, volume 2964 of *Lecture Notes in Computer Science*, pages 163–178. Springer, 2004.

[98] S. Golomb and G. Gong. *Signal design for good correlation.* Cambridge University Press, 2005.

[99] M. Gruteser and D. Grunwald. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *Proceedings of the 1st international conference on Mobile systems, applications and services – MobiSys'03*, pages 31–42. ACM SIGMOBILE, 2003.

[100] Q. Gu, X. Chen, Z. Jiang, and J. Wu. Sink-Anonymity Mobility Control in Wireless Sensor Networks. In *Proceedings of the 5th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications – WiMob'09*, pages 36–41. IEEE Computer Society, 2009.

[101] J. Gubner. *Probability and random processes for electrical and computer engineers.* Cambridge University Press, 2006.

[102] J. Ha, S. Moon, J. Zhou, and J. Ha. A new formal proof model for RFID location privacy. In *Proceedings of the 13th European Symposium on Research in Computer Security – ESORICS'08*, volume 5283 of *Lecture Notes in Computer Science*, pages 267–281. Springer, 2008.

[103] J. Hadamard. Sur la distribution des zéros de la fonction ζ (s) et ses conséquences arithmétiques. *Bull. Soc. Math. France*, 24:199–220, 1896.

[104] S. Halevi and H. Krawczyk. MMH: Software message authentication in the Gbit/second rates. In *Proceedings of the 4th International Workshop on Fast Software Encryption – FSE'97*, volume 1267 of *Lecture Notes in Computer Science*, pages 172–189. Springer, 1997.

[105] T. Halevi, N. Saxena, and S. Halevi. Tree-based HB Protocols for Privacy-Preserving Authentication of RFID Tags. *Journal of Computer Security*, 19(2):343–363, 2011.

[106] H. Handschuh and B. Preneel. Key-Recovery Attacks on Universal Hash Function Based MAC Algorithms. In *Proceedings of the 28th Annual International Cryptology Conference – CRYPTO'08*, Lecture Notes in Computer Science, pages 144–161. Springer, 2008.

[107] G. Hardy and E. Wright. *An Introduction to the Theory of Numbers.* Clarendon Press, 1979.

[108] J. Hastad, R. Impagliazzo, L. Levin, and M. Luby. A Pseudorandom Generator from Any One-Way Function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[109] J. Havil. *Gamma: exploring Euler's constant.* Princeton University Press, 2003.

[110] T. Helleseth and T. Johansson. Universal hash functions from exponential sums over finite fields and Galois rings. In *Proceedings of the 16th Annual International Cryptology Conference – CRYPTO'96*, volume 1109 of *Lecture Notes in Computer Science*, pages 31–44. Springer, 1996.

[111] I. Herstein. *Abstract algebra*. Macmillan New York, 1986.

[112] C. Hocquet, D. Kamel, F. Regazzoni, J. Legat, D. Flandre, D. Bol, and F. Standaert. Harvesting the potential of nano-CMOS for lightweight cryptography: an ultra-low-voltage 65 nm AES coprocessor for passive RFID tags. *Journal of Cryptographic Engineering*, 1(1):79–86, 2011.

[113] B. Hoh and M. Gruteser. Protecting Location Privacy Through Path Confusion. In *Proceedings of the 1st International Conference on Security and Privacy for Emerging Areas in Communications Networks – SecureComm'05*, pages 194–205. IEEE Computer Society Press, 2005.

[114] D. Holcomb, W. Burleson, and K. Fu. Initial SRAM state as a Fingerprint and Source of True Random Numbers for RFID Tags. The 3rd Conference on RFID Security – RFIDsec'07, 2007.

[115] D. Holcomb, W. Burleson, and K. Fu. Power-up SRAM State as an Identifying Fingerprint and Source of True Random Numbers. *IEEE Transactions on Computers*, 58(9):1198–1210, 2009.

[116] D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B. Koo, C. Lee, D. Chang, J. Lee, K. Jeong, H. Kim, J. Kim, and S. Chee. HIGHT: A New Block Cipher Suitable for Low-Resource Device. In *Proceedings of the 8th International Workshop on Cryptographic Hardware and Embedded Systems – CHES'06*, volume 4249 of *Lecture Notes in Computer Science*, pages 46–59. Springer, 2006.

[117] N. Hopper and M. Blum. Secure human identification protocols. In *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security – ASIACRYPT'01*, volume 2248 of *Lecture Notes in Computer Science*, pages 52–66. Springer, 2001.

[118] IEEE Standard for Information technology. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. IEEE Standards Association, 2011.

[119] ISO/IEC 9797-1. Information technology – Security techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher, 1999.

[120] ISO/IEC 9797-2. Information technology – Security techniques – Message Authentication Codes (MACs) – Part 2: Mechanisms using a dedicated hash-function, 2002.

[121] P. Israsena. Design and implementation of low power hardware encryption for low cost secure RFID using TEA. In *Proceedings of the 5th International Conference on Information, Communications and Signal Processing – ICICS'05*, pages 1402–1406. IEEE Communications Society, 2005.

[122] T. Iwata and K. Kurosawa. omac: One-key cbc mac. In *Proceedings of the 10th International Workshop on Fast Software Encryption – FSE'03*, volume 2887 of *Lecture Notes in Computer Science*, pages 129–153. Springer, 2003.

[123] C. Jarque and A. Bera. A test for normality of observations and regression residuals. *International Statistical Review/Revue Internationale de Statistique*, 55(2):163–172, 1987.

[124] E. Jaulmes, A. Joux, and F. Valette. On the Security of Randomized CBC-MAC Beyond the Birthday Paradox Limit A New Construction. In *Proceedings of the 9th International Workshop on Fast Software Encryption – FSE'02*, volume 2365 of *Lecture Notes in Computer Science*, pages 613–616. Springer, 2002.

[125] Y. Jiang, Y. Fan, X. Shen, and C. Lin. A self-adaptive probabilistic packet filtering scheme against entropy attacks in network coding. *Computer Networks*, 53(18):3089–3101, 2009.

[126] T. Johansson. Bucket hashing with a small key size. In *Proceedings of the 16th International Conference on the Theory and Application of Cryptographic Techniques – EUROCRYPT'97*, volume 1233 of *Lecture Notes in Computer Science*, pages 149–162. Springer, 1997.

[127] J. Jonsson. On the security of CTR+ CBC-MAC. In *Proceedings of the 9th Annual International Workshop on Selected Areas in Cryptography – SAC'02*, volume 2595 of *Lecture Notes in Computer Science*, pages 76–93. Springer, 2002.

[128] A. Juels. RFID security and privacy: A research survey. *IEEE Journal on Selected Areas in Communications*, 24(2):381–394, 2006.

[129] A. Juels and R. Pappu. Squealing euros: Privacy protection in RFID-enabled banknotes. In *Proceedings of the 7th International Conference on Financial Cryptography – FC'03*, volume 2742 of *Lecture Notes in Computer Science*, pages 103–121. Springer, 2003.

[130] A. Juels, R. Rivest, and M. Szydlo. The blocker tag: Selective blocking of RFID tags for consumer privacy. In *Proceedings of the 10th ACM Conference on Computer and Communications Security – CCS'03*, pages 103–111. ACM SIGSAC, 2003.

[131] A. Juels, P. Syverson, and D. Bailey. High-power proxies for enhancing RFID privacy and utility. In *Proceedings of the 5th Workshop on Privacy Enhancing Technologies – PET'05*, volume 3856 of *Lecture Notes in Computer Science*, pages 210–226. Springer, 2005.

[132] A. Juels and S. Weis. Authenticating pervasive devices with human protocols. In *Proceedings of the 25th Annual International Cryptology Conference – CRYPTO'05*, volume 3126 of *Lecture Notes in Computer Science*, pages 293–308. Springer, 2005.

[133] A. Juels and S. Weis. Defining Strong Privacy for RFID. In *Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications – PerCom'07*, pages 342–347. IEEE Computer Society, 2007.

[134] C. Jutla. Encryption modes with almost free message integrity. *Journal of Cryptology*, 21(4):547–578, 2008.

[135] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk. Enhancing Source-Location Privacy in Sensor Network Routing. In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems – ICDCS'05*, pages 599–608. IEEE Computer Society, 2005.

[136] J. Kaps, K. Yuksel, and B. Sunar. Energy scalable universal hashing. *IEEE Transactions on Computers*, 54(12):1484–1495, 2005.

[137] H. Karl and A. Willig. *Protocols and architectures for wireless sensor networks.* Wiley, 2005.

[138] J. Katz and Y. Lindell. *Introduction to modern cryptography.* Chapman & Hall/CRC, 2008.

[139] J. Katz, J. S. Shin, and A. Smith. Parallel and Concurrent Security of the HB and HB+ Protocols. *Journal of Cryptology*, 23(3):402–421, 2010.

[140] J. Katz and M. Yung. Unforgeable Encryption and Chosen Ciphertext Secure Modes of Operation. In *Proceedings of the 7th International Workshop on Fast Software Encryption – FSE'00*, volume 1978 of *Lecture Notes in Computer Science*, pages 25–36. Springer, 2001.

[141] E. B. Kavun and T. Yalcin. A Lightweight Implementation of Keccak Hash Function for Radio-Frequency Identification Applications. In *Proceedings of the 6th International Workshop on RFID Security – RFIDsec'10*, volume 6370 of *Lecture Notes in Computer Science*, pages 258–269. Springer, 2010.

[142] A. Kerckhoffs. La cryptographie militaire. *Journal des sciences militaires*, 9(1):5–38, 1883.

[143] G. Khandelwal, K. Lee, A. Yener, and S. Serbetli. ASAP: a MAC protocol for dense and time-constrained RFID systems. *EURASIP Journal on Wireless Communications and Networking*, 2007(2):1–13, 2007.

[144] J. Kim, A. Biryukov, B. Preneel, and S. Hong. On the security of HMAC and NMAC based on HAVAL, MD4, MD5, SHA-0 and SHA-1 (Extended abstract). In *Proceedings of the 5th International Conference on Security and Cryptography for Networks – SCN'06*, volume 4116 of *Lecture Notes in Computer Science*, pages 242–256. Springer, 2006.

[145] M. Kodialam and T. Nandagopal. Fast and reliable estimation schemes in RFID systems. In *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking – MobiCom'06*, pages 322–333. ACM SIGMOBILE, 2006.

[146] J. Kohl and C. Neuman. The Kerberos Network Authentication Service (V5). Technical report, RFC 1510, 1993.

[147] T. Kohno, J. Viega, and D. Whiting. CWC: A high-performance conventional authenticated encryption mode. In *Proceedings of the 11th International Workshop on Fast Software Encryption – FSE'04*, volume 3017 of *Lecture Notes in Computer Science*, pages 408–426. Springer, 2004.

[148] H. Krawczyk. LFSR-based hashing and authentication. In *Proceedings of the 14th Annual International Cryptology Conference – CRYPTO'94*, volume 839 of *Lecture Notes in Computer Science*, pages 129–139. Springer, 1994.

[149] H. Krawczyk. New hash functions for message authentication. In *Proceedings of the 15th Annual International Cryptology Conference – CRYPTO'95*, volume 921 of *Lecture Notes in Computer Science*, pages 301–310. Springer, 1995.

[150] H. Krawczyk. The order of encryption and authentication for protecting communications(or: How secure is SSL?). In *Proceedings of the 21st Annual International Cryptology Conference – CRYPTO'01*, volume 2139 of *Lecture Notes in Computer Science*, pages 310–331. Springer, 2001.

[151] T. Krovetz. `http://fastcrypto.org/umac/`.

[152] J. Lai, R. H. Deng, and Y. Li. Revisiting Unpredictability-Based RFID Privacy Models. In J. Zhou and M. Yung, editors, *Proceedings of the 8th International Conference on Applied Cryptography and Network Security – ACNS'10*, volume 6123 of *Lecture Notes in Computer Science*, pages 475–492. Springer, 2010.

[153] G. Leander, C. Paar, A. Poschmann, and K. Schramm. New Lightweight DES Variants. In *Proceedings of the 14th International Workshop on Fast Software Encryption –*

FSE'07, volume 4593 of *Lecture Notes in Computer Science*, pages 196–210. Springer, 2007.

[154] M. Lehtonen, T. Staake, F. Michahelles, and E. Fleisch. From identification to authentication - a review of RFID product authentication techniques. The 2nd Workshop on RFID Security – RFIDsec'06, 2006.

[155] X. Leng, K. Mayes, and K. Markantonakis. HB-MP+ Protocol: An Improvement on the HB-MP Protocol. In *Proceedings of the 2nd IEEE International Conference on RFID – IEEE RFID'08*, pages 118–124. IEEE Computer Society, 2008.

[156] N. Li, N. Zhang, S. Das, and B. Thuraisingham. Privacy preservation in wireless sensor networks: A state-of-the-art survey. *Elsevier Journal on Ad Hoc Networks*, 7(8):1501–1514, 2009.

[157] T. Li and R. H. Deng. Vulnerability analysis of EMAP - an efficient RFID mutual authentication protocol. In *Proceedings of the 2nd International Conference on Availability, Reliability and Security – ARES'07*, pages 238–245. IEEE Computer Society, 2007.

[158] T. Li and G. Wang. Security analysis of two ultra-lightweight RFID authentication protocols. In *Proceedings of the IFIP TC11 22nd International Information Security Conference New Approaches for Security, Privacy and Trust in Complex Environments – SEC'07*, volume 232 of *IFIP Advances in Information and Communication Technology*, pages 109–120. Springer, 2007.

[159] Y. Li and J. Ren. Preserving source-location privacy in wireless sensor networks. In *Proceedings of the 6th Annual IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks – SECON'09*, pages 493–501. IEEE Communications Society, 2009.

[160] Y. Li and J. Ren. Source-location privacy through dynamic routing in wireless sensor networks. In *Proceedings of the 29th IEEE International Conference on Computer Communications – INFOCOM'10*, pages 1–9. IEEE Communications Society, 2010.

[161] B. Liang, Y. Li, C. Ma, T. Li, and R. Deng. On the Untraceability of Anonymous RFID Authentication Protocol with Constant Key-Lookup. In *Proceedings of the 5th International Conference on Information Systems Security – ICISS'09*, volume 5905 of *Lecture Notes in Computer Science*, pages 71–85. Springer, 2009.

[162] C. Lim and T. Korkishko. mCrypton–A lightweight block cipher for security of low-cost RFID tags and sensors. In *Proceedings of the 6th International Workshop on Information Security Applications – WISA'06*, volume 3786 of *Lecture Notes in Computer Science*, pages 243–258. Springer, 2006.

[163] C. H. Lim and T. Korkishko. mCrypton - A Lightweight Block Cipher For Security of Low-Cost RFID Tags and Sensors. In *Proceedings of the 6th International Workshop on Information Security Applications – WISA'05*, volume 3786 of *Lecture Notes in Computer Science*, pages 243–258. Springer, 2005.

[164] T. Lim, T. Li, and Y. Li. A Security and Performance Evaluation of Hash-Based RFID Protocols. In *Proceedings of the 4th International Conferences on Information Security and Cryptology – Inscrypt'08*, volume 5487 of *Lecture Notes in Computer Science*, pages 406–424. Springer, 2008.

[165] Z. Liu and D. Peng. True Random Number Generator in RFID Systems Against Traceability. In *Proceedings of the 2nd IEEE Consumer Communications and Networking Conference – CCNS'06*, pages 620–624. IEEE Communications Society, 2006.

[166] Z. Liu and W. Xu. Zeroing-in on network metric minima for sink location determination. In *Proceedings of the 3rd ACM Conference on Wireless Network Security – WiSec'10*, pages 99–104. ACM SIGSAC, 2010.

[167] L. Lu, J. Han, L. Hu, Y. Liu, and L. Ni. Dynamic Key-Updating: Privacy-Preserving Authentication for RFID Systems. In *Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications – PerCom'07*, pages 13–22. IEEE Computer Society, 2007.

[168] L. Lu, J. Han, R. Xiao, and Y. Liu. ACTION: Breaking the Privacy Barrier for RFID Systems. *Proceedings of the 28th IEEE International Conference on Computer Communications – INFOCOM'09*, pages 1953–1961, 2009.

[169] M. Luby and C. Rackoff. How to Construct Pseudorandom Permutations from Pseudorandom Functions. *SIAM Journal on Computing*, 17(2):373–386, 1988.

[170] C. Ma, Y. Li, R. Deng, and T. Li. RFID privacy: Relation Between Two Notions, Minimal Condition, and Efficient Construction. In *Proceedings of the 16th ACM Conference on Computer and Communications Security – CCS'09*, pages 54–65. ACM SIGSAC, 2009.

[171] F. Macé, F.-X. Standaert, and J.-J. Quisquater. ASIC Implementations of the Block Cipher SEA for Constrained Applications. The 3rd International Workshop on RFID Security – RFIDsec'07, 2007.

[172] Y. Mansour, N. Nisan, and P. Tiwari. The computational complexity of universal hashing. In *Proceedings of the 22nd ACM Symposium on Theory of Computing – STOC'90*, pages 235–243. ACM SIGACT, 1990.

[173] J. Massey. Shift-register synthesis and BCH decoding. *Information Theory, IEEE Transactions on*, 15(1):122–127, 1969.

[174] F. Massey Jr. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253):68–78, 1951.

[175] U. Maurer and B. Tackmann. On the soundness of authenticate-then-encrypt: formalizing the malleability of symmetric encryption. In *Proceedings of the 17th ACM Conference on Computer and Communications Security – CCS'10*, pages 505–515. ACM SIGSAC, 2010.

[176] D. McGrew and J. Viega. The security and performance of the Galois/Counter Mode (GCM) of operation. In *Proceedings of the 5th International Conference on Cryptology in India – INDOCRYPT'04*, volume 3348 of *Lecture Notes in Computer Science*, pages 343–355. Springer, 2004.

[177] M. McLoone and M. Robshaw. Public key cryptography and RFID tags. In *Proceedings of the 7th Cryptographers' Track at the RSA Conference – CT-RSA'07*, volume 4377 of *Lecture Notes in Computer Science*, pages 372–384. Springer, 2006.

[178] K. Mehta, D. Liu, and M. Wright. Location Privacy in Sensor Networks Against a Global Eavesdropper. In *Proceedings of the 15th IEEE International Conference on Network Protocols – ICNP'07*, pages 314–323. IEEE Computer Society, 2007.

[179] A. Menezes, P. Van Oorschot, and S. Vanstone. *Handbook of applied cryptography.* CRC, 1997.

[180] C. Meyer and S. Matyas. *Cryptography: A New Dimension in Computer Data Security.* John Wiley & Sons, 1982.

[181] D. Molnar and D. Wagner. Privacy and Security in Library RFID: Issues, Practices, and Architectures. In *Proceedings of the 11th ACM Conference on Computer and Communications Security – CCS'04*, pages 210–219. ACM SIGSAC, 2004.

[182] F. Muller. Differential attacks against the Helix stream cipher. In *Proceedings of the 11th International Workshop on Fast Software Encryption – FSE'04*, volume 3017 of *Lecture Notes in Computer Science*, pages 94–108. Springer, 2004.

[183] J. Myung, W. Lee, and J. Srivastava. Adaptive binary splitting for efficient RFID tag anti-collision. *IEEE Communications Letters*, 10(3):144–146, 2006.

[184] J. Nakajima and M. Matsui. Performance analysis and parallel implementation of dedicated hash functions. In *Proceedings of the 21st International Conference on the Theory and Applications of Cryptographic Techniques – EUROCRYPT'02*, volume 2332 of *Lecture Notes in Computer Science*, pages 165–180. Springer, 2002.

[185] M. a. Naser, M. Al Majaly, M. Rafie, and R. Budiarto. A Framework for RFID Systems Security for Human Identification Based on Three-Tier Categorization Model. In *Proceedings of the 1st International Conference on Signal Acquisition and Processing – ICSAP'09*, pages 103–107. IEEE Computer Society, April 2009.

[186] U. D. o. C. National Bureau of Standards. Data Encryption Standard. *Federal Information Processing Standards, FIPS-46*, 1977.

[187] U. D. o. C. National Bureau of Standards. Advanced Encryption Standard. *Federal Information Processing Standard, FIPS-197*, 2001.

[188] W. Nevelsteen and B. Preneel. Software performance of universal hash functions. In *Proceedings of the 18th International Conference on the Theory and Application of Cryptographic Techniques – EUROCRYPT'99*, volume 1592 of *Lecture Notes in Computer Science*, pages 24–41. Springer, 1999.

[189] E. Ngai. On providing sink anonymity for sensor networks. In *Proceedings of the 5th International Conference on Wireless Communications and Mobile Computing – IWCMC'09*, pages 269–273. IEEE Communications Society, 2009.

[190] E. Ngai and I. Rodhe. On providing location privacy for mobile sinks in wireless sensor networks. In *Proceedings of the 12th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems – MSWiM'09*, pages 116–123. ACM SIGSIM, 2009.

[191] M. Ohkubo, K. Suzuki, and S. Kinoshita. Cryptographic approach to privacy-friendly tags. RFID Privacy Workshop, 2003.

[192] M. O'Neill. Low-Cost SHA-1 Hash Function Architecture for RFID Tags. The 4th Workshop on RFID Security – RFIDsec'08, 2008.

[193] K. Ouafi, R. Overbeck, and S. Vaudenay. On the Security of HB# against a Man-in-the-Middle Attack. In *Proceeding of the 14th International Conference on the Theory and Application of Cryptology and Information Security – ASIACRYPT'08*, volume 5350 of *Lecture Notes in Computer Science*, pages 108–124. Springer, 2008.

[194] Y. Ouyang, Z. Le, G. Chen, J. Ford, F. Makedon, and U. Lowell. Entrapping Adversaries for Source Protection in Sensor Networks. In *Proceedings of the 7th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks – WOWMOM'06*, pages 32–41. IEEE Computer Society, 2006.

[195] Y. Ouyang, Z. Le, D. Liu, J. Ford, and F. Makedon. Source location privacy against laptop-class attacks in sensor networks. In *Proceedings of the 4th International Conference on Security and privacy in Communication Networks – SecureComm'08*, pages 1–10. ACM, 2008.

[196] C. Ozturk, Y. Zhang, and W. Trappe. Source-location privacy in energy-constrained sensor network routing. In *Proceedings of the 2nd ACM workshop on Security of Ad hoc and Sensor Networks – SASN'04*, pages 88–93. ACM SIGSAC, 2004.

[197] S. Paul and B. Preneel. Near Optimal Algorithms for Solving Differential Equations of Addition with Batch Queries. In *Proceedings of the 6th International Conference on Cryptology in India – INDOCRYPT'05*, volume 3797 of *Lecture Notes in Computer Science*, pages 90–103. Springer, 2005.

[198] S. Paul and B. Preneel. Solving systems of differential equations of addition. In *Proceedings of the 10th Australasian Conference on Information Security and Privacy – ICISP'05*, volume 3574 of *Lecture Notes in Computer Science*, pages 75–88. Springer, 2005.

[199] P. Peris-Lopez, J. Hernandez-Castro, J. Estevez-Tapiador, and A. Ribagorda. RFID systems: A survey on security threats and proposed solutions. In *Proceedings of the IFIP TC6 11th International Conference on Personal Wireless Communications – PWC'06*, volume 4217 of *Lecture Notes in Computer Science*, pages 159–170. Springer, 2006.

[200] P. Peris-Lopez, J. C. Hernandez-Castro, J. Estevez-Tapiador, and A. Ribagorda. LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags. The 2nd Workshop on RFID Security – RFIDsec'06, 2006.

[201] P. Peris-Lopez, J. C. Hernandez-Castro, J. Estevez-Tapiador, and A. Ribagorda. M2AP: A minimalist mutual-authentication protocol for low-cost RFID tags. In *Proceedings of the 3rd International Conference on Ubiquitous Intelligence and Computing – UIC'06*, volume 4159 of *Lecture Notes in Computer Science*, pages 912–923. Springer, 2006.

[202] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. EMAP: An efficient mutual authentication protocol for low-cost RFID tags. In *Proceedings of the 1st International Workshop on Information Security – IS'06*, volume 4277 of *Lecture Notes in Computer Science*, pages 352–361. Springer, 2006.

[203] A. Perrig, J. Stankovic, and D. Wagner. Security in wireless sensor networks. *Communications of the ACM*, 47(6):53–57, 2004.

[204] A. Perrig, R. Szewczyk, J. Tygar, V. Wen, and D. Culler. SPINS: Security protocols for sensor networks. *Wireless networks*, 8(5):521–534, 2002.

[205] C. Petrie and J. Connelly. A noise-based IC random number generator for applications incryptography. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 47(5):615–621, 2000.

[206] R. C.-W. Phan. Cryptanalysis of a New Ultralightweight RFID Authentication Protocol - SASI. *IEEE Transactions on Dependable and Secure Computing*, 99(1), 2008.

[207] C. Poussin. *Recherches analytiques sur la théorie des nombres premiers.* Hayez, 1897.

[208] B. Preneel. Using Cryptography Well. Printed handout available at `http://secappdev.org/handouts/2010/Bart%20Preneel/using_crypto_well.pdf`, 2010.

[209] B. Preneel and P. Van Oorschot. MDx-MAC and building fast MACs from hash functions. In *Proceedings of the 15th Annual International Cryptology Conference – CRYPTO'95*, volume 963 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 1995.

[210] B. Preneel and P. van Oorschot. On the security of two MAC algorithms. In *Proceedings of the 15th International Conference on the Theory and Application of Cryptographic Techniques – EUROCRYPT'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 19–32. Springer, 1996.

[211] B. Preneel and P. Van Oorschot. On the security of iterated message authentication codes. *IEEE Transactions on Information theory*, 45(1):188–199, 1999.

[212] M. Rabin. Probabilistic algorithm for testing primality. *Journal of Number Theory*, 12(1):128–138, 1980.

[213] M. Reed, P. Syverson, and D. Goldschlag. Anonymous connections and onion routing. *IEEE Journal on Selected areas in Communications*, 16(4):482–494, 1998.

[214] A. Ricci, M. Grisanti, I. De Munari, and P. Ciampolini. Design of a 2 μW RFID baseband processor featuring an AES cryptography primitive. In *Proceedings of the 15th IEEE International Conference on Electronics, Circuits and Systems – ICECS'08*, pages 376–379. IEEE Circuits and Systems Society, 2008.

[215] M. Rieback, B. Crispo, and A. Tanenbaum. RFID Guardian: A battery-powered mobile device for RFID privacy management. In *Proceedings of the 10th Australasian Conference on Information Security and Privacy – ACISP'05*, volume 3574 of *Lecture Notes in Computer Science*, pages 184–194. Springer, 2005.

[216] R. Rivest. RFC1321: The MD5 message-digest algorithm, 1992.

[217] P. Rogaway. Bucket hashing and its application to fast message authentication. *Journal of Cryptology*, 12(2):91–115, 1999.

[218] P. Rogaway, M. Bellare, J. Black, and T. Krovetz. OCB: A block-cipher mode of operation for efficient authenticated encryption. In *Proceedings of the 8th ACM Conference on Computer and Communications Security – CCS'01*, pages 196–205. ACM SIGSAC, 2001.

[219] P. Rogaway and D. Wagner. A critique of CCM. Available at `http://eprint.iacr.org/2003/070/`, 2003.

[220] J. Rosser and L. Schoenfeld. Approximate formulas for some functions of prime numbers. *Illinois Journal of Mathematics*, 6(1):64–94, 1962.

[221] S. Sarma, S. Weis, and D. Engels. RFID systems and security and privacy implications. In *Proceedings of the 4th International Workshop on Cryptographic Hardware and Embedded Systems – CHES'02*, volume 2523 of *Lecture Notes in Computer Science*, pages 454–469. Springer, 2002.

[222] S. Sarma, S. Weis, and D. Engels. Radio-Frequency Identification: Security Risks and Challenges. *Cryptobytes, RSA Laboratories*, 6(1):2–9, 2003.

[223] L. Scharf. *Statistical signal processing: detection, estimation, and time series analysis.* Addison-Wesley, 1991.

[224] S. Schwarz. The role of semigroups in the elementary theory of numbers. *Math. Slovaca*, 31(4):369–395, 1981.

[225] E. Shakshuki, T. Sheltami, N. Kang, and X. Xing. Tracking Anonymous Sinks in Wireless Sensor Networks. In *Proceedings of the 23rd IEEE International Conference on Advanced Information Networking and Applications – AINA'09*, pages 510–516. IEEE Computer Society, 2009.

[226] A. Shamir. SQUASH–a new MAC with provable security properties for highly constrained devices such as RFID tags. In *Fast Software Encryption – FSE'08*, volume 5086 of *Lecture Notes in Computer Science*, pages 144–157. Springer, 2008.

[227] A. Shamir. SQUASH–A New MAC with Provable Security Properties for Highly Constrained Devices Such as RFID Tags. In *Proceedings of the 15th International Workshop on Fast Software Encryption – FSE'08*, volume 5086 of *Lecture Notes in Computer Science*, pages 144–157. Springer, 2008.

[228] C. Shannon. *Communication Theory and Secrecy Systems.* Bell Telephone Laboratories, 1949.

[229] M. Shao, W. Hu, S. Zhu, G. Cao, S. Krishnamurthy, and T. La Porta. Cross-layer Enhanced Source Location Privacy in Sensor Networks. In *Proceedings of the 6th Annual IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks – SECON'09*, pages 324–332. IEEE Communications Society, 2009.

[230] M. Shao, Y. Yang, S. Zhu, and G. Cao. Towards Statistically Strong Source Anonymity for Sensor Networks. In *Proceedings of the 27th IEEE International Conference on Computer Communications – INFOCOM'08*, pages 466–474. IEEE Communications Society, 2008.

[231] M. Shao, S. Zhu, W. Zhang, G. Cao, and Y. Yang. pDCS: Security and privacy support for data-centric sensor networks. *IEEE Transactions on Mobile Computing*, 8(8):1023–1038, 2008.

[232] D. Shih, C. Lin, and B. Lin. RFID tags: privacy and security aspects. *International Journal of Mobile Communications*, 3(3):214–230, 2005.

[233] V. Shoup. On fast and provably secure message authentication based on universal hashing. In *Proceedings of the 16th Annual International Cryptology Conference – CRYPTO'96*, volume 1109 of *Lecture Notes in Computer Science*, pages 313–328. Springer, 1996.

[234] B. Smeets, P. Vanroose, and Z. Wan. On the construction of authentication codes with secrecy and codes withstanding spoofing attacks of order $L \geq 2$. In *Proceedings of the 9th Workshop on the Theory and Application of Cryptographic Techniques – EUROCRYPT'90*, volume 473 of *Lecture Notes in Computer Science*, pages 307–312. Springer, 1990.

[235] B. Song and C. J. Mitchell. RFID Authentication Protocol for Low-cost Tags. In *Proceedings of the 1st ACM Conference on Wireless Network Security – WiSec'08*, pages 140–147. ACM SIGSAC, 2008.

[236] B. Song and C. J. Mitchell. Scalable RFID Pseudonym Protocol. In *Proceedings of the 3rd International Conference on Network ans System Security – NSS'09*, pages 216–224. IEEE Computer Society, 2009.

[237] W. Stallings. *Cryptography and network security: principles and practice*. Prentice Hall, 2010.

[238] M. Stephens. EDF statistics for goodness of fit and some comparisons. *Journal of the American Statistical Association*, 69(347):730–737, 1974.

[239] D. Stinson. A construction for authentication/secrecy codes from certain combinatorial designs. *Journal of Cryptology*, 1(2):119–127, 1988.

[240] D. Stinson. The combinatorics of authentication and secrecy codes. *Journal of Cryptology*, 2(1):23–49, 1990.

[241] D. Stinson. Universal hashing and authentication codes. *Designs, Codes and Cryptography*, 4(3):369–380, 1994.

[242] D. Stinson. On the connections between universal hashing, combinatorial designs and error-correcting codes. *Congressus Numerantium*, 114:7–28, 1996.

[243] D. Stinson. *Cryptography: Theory and Practice*. CRC Press, 2006.

[244] C. Tan, H. Wang, S. Zhong, and Q. Li. Body sensor network security: an identity-based cryptography approach. In *Proceedings of the 1st ACM Conference on Wireless Network Security – WiSec'08*, pages 148–153. ACM SIGSAC, 2008.

[245] J. Tignol. *Galois' Theory of Algebraic Equations*. World Scientific, 2001.

[246] G. Tsudik. Message authentication with one-way hash functions. *ACM Computer Communication Review*, 22(5):38, 1992.

[247] I. Vajda and L. Buttyán. Lightweight authentication protocols for low-cost RFID tags. In *Proceedings of the 2nd Workshop on Security in Ubiquitous Computing – Ubicomp'03*, pages 1–10, 2003.

[248] T. van Deursen and S. Radomirović. Attacks on RFID Protocols. Cryptology ePrint Archive, Report 2008/310, 2008.

[249] H. van Tilborg. *Encyclopedia of cryptography and security*. Springer, 2005.

[250] T. Van Tran. On the construction of authentication and secrecy codes. *Designs, codes and cryptography*, 5(3):269–280, 1995.

[251] S. Vaudenay. On Privacy Models for RFID. In *Proceeding of the 13th International Conference on the Theory and Application of Cryptology and Information Security – ASIACRYPT'07*, volume 4833 of *Lecture Notes in Computer Science*, pages 68–87. Springer, 2007.

[252] K. Venkatasubramanian, A. Banerjee, and S. Gupta. Ekg-based key agreement in body sensor networks. In *Proceedings of the 2nd Workshop on Mission Critical Nets*, pages 1–6. Citeseer, 2008.

[253] G. Vernam. Cipher printing telegraph systems for secret wire and radio telegraphic communications. *Journal of the American Institute of Electrical Engineers*, 45:109–115, 1926.

[254] H. Wang, B. Sheng, and Q. Li. Privacy-aware routing in sensor networks. *Elsevier Journal on Computer Networks*, 53(9):1512–1529, 2009.

[255] W. Wang, Y. Li, L. Hu, and L. Lu. Storage-Awareness: RFID Private Authentication based on Sparse Tree. In *Proceedings of the 3rd International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing – SECPerU'07*, pages 61–66. IEEE Computer Society, 2007.

[256] X. Wang, X. Li, Z. Wan, and M. Gu. CLEAR: A confidential and Lifetime-Aware Routing Protocol for wireless sensor network. In *Proceedings of the 20th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications – PIMRC'09*, pages 2265–2269. IEEE Communications Society, 2009.

[257] X. Wang, Y. Yin, and H. Yu. Finding collisions in the full SHA-1. In *Proceedings of the 25th Annual International Cryptology Conference – CRYPTO'05*, volume 3126 of *Lecture Notes in Computer Science*, pages 17–36. Springer, 2005.

[258] X. Wang and H. Yu. How to break MD5 and other hash functions. In *Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques – EUROCRYPT'05*, volume 3494, pages 19–35. Springer, 2005.

[259] X. Wang, H. Yu, and Y. Yin. Efficient collision search attacks on SHA-0. In *Proceedings of the 25th Annual International Cryptology Conference – CRYPTO'05*, volume 3126 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2005.

[260] M. Wegman and J. Carter. New classes and applications of hash functions. In *Proceedings of the 20th Annual IEEE Symposium on Foundations of Computer Science – FOCS'79*, pages 175–182. IEEE Computer Society, 1979.

[261] M. Wegman and L. Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences*, 22(3):265–279, 1981.

[262] S. Weis, S. Sarma, R. Rivest, and D. Engels. Security and privacy aspects of low-cost radio frequency identification systems. In *Proceedings of the 1st International Conference on Security in Pervasive Computing – PERVASIVE'03*, volume 2802 of *Lecture Notes in Computer Science*, pages 201–212. Springer, 2004.

[263] D. Whiting, B. Schneier, S. Lucks, and F. Muller. Phelix – fast encryption and authentication in a single cryptographic primitive. ECRYPT Stream Cipher Project, Report 2005/020, `www.ecrypt.eu.org/stream`, 2005.

[264] H. Wu and B. Preneel. Differential-linear attacks against the stream cipher Phelix. In *Proceedings of the 14th International Workshop on Fast Software Encryption –*

FSE'07, volume 4593 of *Lecture Notes in Computer Science*, pages 87–100. Springer, 2007.

[265] J. Wu and D. Stinson. A Highly Scalable RFID Authentication Protocol. In *Proceedings of the 14th Australasian Conference on Information Security and Privacy – ACISP'09*, volume 5594 of *Lecture Notes in Computer Science*, pages 360–376. Springer, 2009.

[266] Y. Xi, L. Schwiebert, and W. Shi. Preserving source location privacy in monitoring-based wireless sensor networks. In *Proceedings of the 20th IEEE International Parallel & Distributed Processing Symposium – IPDPS'06*, pages 1–8. IEEE Computer Society, 2006.

[267] W. Yang and W. Zhu. Protecting source location privacy in wireless sensor networks with data aggregation. In *Proceedings of the 7th International Conference on Ubiquitous Intelligence and Computing – UIC'10*, volume 6406 of *Lecture Notes in Computer Science*, pages 252–266. Springer, 2010.

[268] Y. Yang, M. Shao, S. Zhu, B. Urgaonkar, and G. Cao. Towards event source unobservability with minimum network traffic in sensor networks. In *Proceedings of the 1st ACM Conference on Wireless Network Security – WiSec'08*, pages 77–88. ACM SIGSAC, 2008.

[269] K. Yasuda. Multilane HMAC– Security beyond the birthday limit. In *Proceedings of the 8th International Conference on Cryptology in India – INDOCRYPT'07*, volume 4859 of *Lecture Notes in Computer Science*, pages 18–32. Springer, 2007.

[270] J. Yick, B. Mukherjee, and D. Ghosal. Wireless sensor network survey. *Computer Networks*, 52(12):2292–2330, 2008.

[271] T. Ylonen and C. Lonvick. The Secure Shell (SSH) Transport Layer Protocol. Technical report, RFC 4253, 2006.

[272] B. Yoon. HB-MP++ Protocol: An Ultra Light-weight Authentication Protocol for RFID System. In *Proceedings of the 3rd IEEE International Conference on RFID – IEEE RFID'09*, pages 186–191. IEEE Computer Society, 2009.

[273] Y. Yousuf and V. Potdar. A Survey of RFID Authentication Protocols. In *Proceedings of the 22nd IEEE International Conference on Advanced Information Networking and Applications – AINA'08*, pages 1346–1350. IEEE Computer Society, 2008.

[274] D. Zanetti, B. Danev, and S. Čapkun. Physical-layer identification of uhf rfid tags. In *Proceedings of the 16th ACM Conference on Mobile Computing and Networking – MobiCom'10*, pages 353–364. ACM SIGMOBILE, 2010.

[275] Y. Zeng, J. Cao, S. Zhang, S. Guo, and L. Xie. Random-walk based approach to detect clone attacks in wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, 28(5):677–691, 2010.

[276] X. Zhang and B. King. Modeling RFID Security. In *Proceedings of the 8th International Conference on Information Security and Cryptology – ICISC'05*, volume 3822 of *Lecture Notes in Computer Science*, pages 75–90. Springer, 2005.

[277] Y. Zhang. A design proposal of security architecture for medical body sensor networks. In *Proceedings of the 3rd International Workshop on Wearable and Implantable Body Sensor Networks – BSN'06*, pages 4–9. IEEE Computer Society, 2006.

[278] Y. Zuo. RFID Survivability Quantification and Attack Modeling. In *Proceedings of the 3rd ACM Conference on Wireless Network Security – WiSec'10*, pages 13–18. ACM, ACM Press, 2010.