Noname manuscript No.
(will be inserted by the editor)

$\mathcal{E}$-MACs: Towards More Secure and More Efficient Constructions of Secure Channels

Basel Alomair · Radha Poovendran

Received: date / Accepted: date

Abstract In cryptography, secure channels enable the confidential and authenticated message exchange between authorized users. A generic approach of constructing such channels is by combining an encryption primitive with an authentication primitive (MAC). In this work, 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 is the observation that, since the message must be both encrypted and authenticated, there might be some redundancy in the computations performed by the two primitives. Therefore, removing such redundancy can improve the efficiency of the overall composition. Moreover, computations performed by the encryption algorithm can be further utilized to improve the security of the authentication algorithm. In particular, we will show how $\mathcal{E}$-MACs can be designed to reduce the amount of computation required by standard MACs based on universal hash functions, and show how $\mathcal{E}$-MACs can be secured against key-recovery attacks.

Keywords Confidentiality, authenticity, message authentication code (MAC), authenticated encryption, universal hash families

1 Introduction

There are two main approaches for the construction of secure channels in cryptography: a dedicated approach and a generic approach. In the dedicated approach, a cryptographic primitive is designed to achieve authenticated encryption as a standalone system (see, e.g., [11, 27, 32, 44, 66]). In the generic approach, an authentication primitive is combined with an encryption primitive to provide message integrity and confidentiality (see, e.g., [22, 23, 30, 49, 77]).

Generic compositions can be constructed in three different ways: Encrypt-and-Authenticate (E&A), Encrypt-then-Authenticate (EtA), or Authenticate-then-Encrypt (AtE). In E&A 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 EtA compositions, the plaintext is passed to the encryption algorithm to get a ciphertext, the resulting ciphertext is passed to the MAC algorithm to get a tag,

A preliminary version of this paper has appeared in the 13th International Conference on Information Security and Cryptology–ICISC'10 [3].

B. Alomair
Department of Electrical Engineering
University of Washington
Seattle, WA 98195, USA
Tel.: +1 206-616-6130
Fax: +1 206-616-6130
E-mail: alomair@uw.edu

R. Poovendran
Department of Electrical Engineering
University of Washington
Seattle, WA, 98195, USA
E-mail: rp3@uw.edu

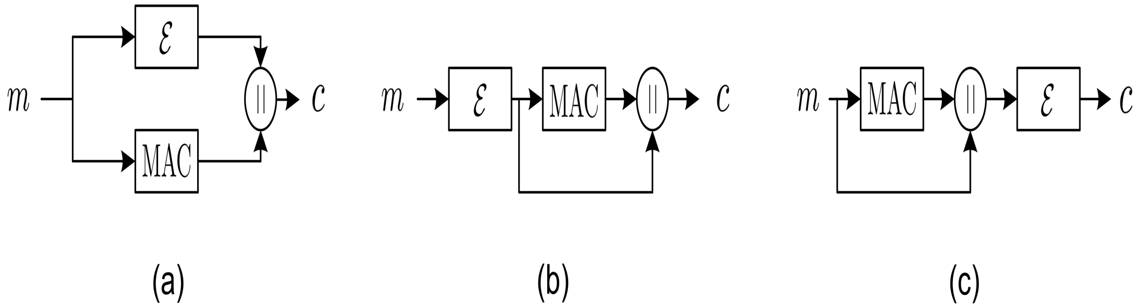

Fig. 1 A schematic of the three generic approaches; (a) E&A, (b) EtA, and (c) AtE.

and the resulting $\big(\mathcal{E}(M), \mathrm{MAC}(\mathcal{E}(M))\big)$ is transmitted to the intended receiver. In AtE 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 $\big(\mathcal{E}\big(M, \mathrm{MAC}(M)\big)\big)$ is transmitted to the intended receiver. The transport layer of SSH uses a variant of the E&A composition [77], IPSEC uses a variant of the EtA composition [23], while SSL and TLS use variants of the AtE composition [22, 30].

Over dedicated primitives, generic compositions possess several design and analysis advantages due to their modularity and the fact that encryption and authentication schemes can be designed, analyzed, and replaced independently from each other [52]. Further, and most important, 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 [52].

However, generic compositions are more involved than just combining an encryption algorithm and a MAC algorithm. In [9, 52] 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 EtA will guarantee the construction of secure channels [9, 52]. Therefore, special attention must be paid to the design of secure channels if the E&A or the AtE compositions are used.

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 paper, we provide the first such work. Specifically, we introduce the design of special purpose MACs to be used in the construction of E&A and AtE compositions. 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.

The E&A and AtE compositions, however, impose an extra requirement on the MAC algorithm. As opposed to the EtA compositions, the tag in the E&A and AtE compositions is a function of the plaintext message (not the ciphertext as in EtA). Therefore, the tag must be at least as confidential as the ciphertext since, otherwise, the secrecy 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. In universal hash-function families based MACs, the message to be authenticated is first compressed using a universal hash function in the Carter-Wegman style [19, 20, 72, 73] 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 block ciphers. 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 [14] and hash127 [12]; both of which are based on universal hash functions [53, 70].)

Recently, however, Handschuh and Preneel [36] discovered a vulnerability in universal hash functions based MACs. They demonstrated that once a collision in the universal hash function is achieved, subsequent forgeries can succeed with higher probabilities. Their attack is not directed to a specific universal hash family and can be applied to all such MACs. The recommendation of the work in [36] is not to reuse the universal hash function keys, thus going back to the earliest use of universal hash families for unconditionally secure authentication, or proceeding with the less efficient, yet more secure, block cipher based MACs.

Contributions. In this paper, 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&A and AtE compositions. We introduce the design of $\mathcal{E}$-MACs: *Authentication Codes for $\mathcal{E}$ncrypted Messages*. By proposing the first instance of $\mathcal{E}$-MACs, we show how the structure of the E&A and AtE 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&A and AtE 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&A and AtE structures.

Organization. The rest of the paper is organized as follows. In Section 2, we discuss related work. In Section 3, we list the used notations, security model, and preliminaries. An instance of $\mathcal{E}$-MACs is proposed in Section 4. The performance discussion and advantage of $\mathcal{E}$-MACs are addressed in Section 5. The security analysis of the proposed $\mathcal{E}$-MAC and the security of the generic compositions constructed using the proposed $\mathcal{E}$-MAC is detailed in Section 6. Section 7 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&A and AtE systems to overcome this vulnerability. The paper is concluded in Section 8.

2 Related Work

Many standard MACs that can be used in the construction of authenticated encryption schemes have appeared in the literature. Standard MACs can be block ciphers based, cryptographic hash functions based, or universal hash functions based.

CBC-MAC is one of the most known block cipher based MACs specified in FIPS publication 113 [28] and the International Organization for Standardization ISO/IEC 9797-1 [39]. CMAC, a modified version of CBC-MAC, is presented in the NIST special publication 800-38B [24], which was based on OMAC of Iwata and Kurosawa [41]. Other block cipher based MACs include, but are not limited to, XOR-MAC [6] and PMAC [15]. In [6], Bellare et al. proposed the XOR-MACs to achieve provably secure message authentication via the use of any finite pseudorandom function (PRF). Rogaway and Black proposed a parallelizable block cipher based message authentication code (PMAC) in [15]. The security of different MACs has been exhaustively studied (see, e.g., [7, 10, 42, 43, 64]).

The use of one-way cryptographic hash functions for message authentication was introduced by Tsudik in [71]. HMAC is a popular example of the use of iterated cryptographic hash functions in the design of MACs [5], which was adopted as a standard [29]. Another cryptographic hash function based MAC is the MDx-MAC of Preneel and Oorschot [63]. HMAC and two variants of MDx-MAC are specified in the International Organization for Standardization ISO/IEC 9797-2 [40]. 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 [17]. Similar to the case of block cipher based MACs, the security of cryptographic hash function based MACs has been extensively studied (see, e.g., [4, 21, 47, 62, 76]).

The use of universal hash families was first introduced by Carter and Wegman in the context of designing unconditionally secure authentication [19, 20, 72, 73]. The use of universal hash functions for the design of computationally secure MACs appeared in [1, 2, 12–14, 26, 35, 45, 56]. The basic concept behind the design of computationally secure universal hash functions based MACs is to compress the message using universal hash functions and then process the compressed output using a cryptographic function. The key idea is that processing messages using universal hash functions is faster than processing them block by block using block ciphers. Then, since the hashed image is typically much shorter than the message itself, processing the hashed image with a cryptographic function is faster then processing the entire message.

The design of authenticated encryption schemes has attracted a lot of attention historically. Variety of earlier schemes based on adding some redundancy to messages before cipher block chaining (CBC) encryption were found vulnerable to attacks [9]. Establishing secure channels by means of generic constructions of authenticated encryption schemes was of particular interest. The network layer of SSH [77], the IPsec [23], and the SSL [30] (followed by the TLS [22]) use variants of E&A, EtA, and AtE compositions, respectively. Kohno et al. [49] proposed the high-performance Carter-Wegman Counter (CWC) mode of encrypt-then-authenticate, which the NIST standard Galois/Counter Mode (GCM) was based on [25]. The security relations among different notions of security in authenticated encryption schemes was studied in detail by Bellare and Namprempre in [9]. Canetti and Krawczyk showed that EtA schemes build secure channels [18]. Krawczyk analyzed the security of the three generic constructions methods in [52]. Bellare et al. showed that SSH is provably secure in [8]. Maurer and Tackmann showed that the AtE can result in secure channels in [55].

In a different direction, block ciphers that combine encryption and message authentication have been proposed in the literature. Proposals that use simple checksum or manipulation detection code (MDC) have appeared in [31, 48, 57]. Such simple schemes, however, are known to be vulnerable to attacks [44]. Other dedicated schemes that combine encryption and message authenticity include [11, 27, 32, 44, 66].

In [44], Jutla proposed the integrity aware parallelizable mode (IAPM), an encryption scheme with authentication. The authenticated encryption requires a total of $m + 2$ block cipher evaluation for a message of m blocks. Gligor and Donescu proposed the XECB-MAC [32]. The XECB-MAC possesses all the operational properties of the XOR-MAC [6] with about only half the block cipher calls of the XOR-MAC. Rogaway et al. [66] proposed OCB: a block-cipher mode of operation for efficient authenticated encryption. For a message of length M-bits and an n-bit cipher block size, their method requires $\lceil \frac{M}{n} \rceil + 2$ block cipher runs.

3 Notations and Model Assumptions

In this section we state our assumptions and describe the notations and definitions that will be used for the rest of the paper.

3.1 Notations

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

- For the set $\mathbb{Z}_n \overset{\text{def}}{=} \{0, 1, ..., n-1\}$, define $\mathbb{Z}_n^*$ to be the set of integers relatively prime (co-prime) to n. When n is a prime integer $\mathbb{Z}_n^* = \mathbb{Z}_n \backslash \{0\} \overset{\text{def}}{=} \{1, 2, ..., n-1\}$. For the rest of the paper, 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.
- For any non-empty set I, the cardinality of the set is denoted as $|I|$.
- 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 the rest of the paper, addition and multiplication are performed over elements in the ring $\mathbb{Z}_p$, even if the $\pmod p$ part is dropped for convenience.
- $\{0, 1\}^\alpha$ denotes a binary string of length α, whereas $\{0, 1\}^*$ denotes a binary string of arbitrary length.
- Throughout the rest of the paper, 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.

3.2 Definitions and Adversarial Model

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 ℓ and N describing the length of the shared key and the resulting authentication tag, respectively. On input an ℓ-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 ℓ-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. We ask for a basic validity condition, namely 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.

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 ℓ 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 [9], 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.

As in standard fast MACs, the proposed $\mathcal{E}$-MAC is based on a universal hash-function family. 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 A to B, it can be viewed as a single function $\mathcal{H} : \mathcal{K} \times A \to B$, whose first argument is usually written as a subscript. A random element $h \in \mathcal{H}$ is determined by selecting a $k \in \mathcal{K}$ uniformly at random and setting $h = \mathcal{H}_k$.

There has been a number of different definitions of universal hash families (see, e.g., [19,35,50,51,65, 68,72]). We give below a formal definition of one class of universal hash families called ϵ-almost universal hash families.

Definition 1 [14,68] [Universal Hash Families] Let $\mathcal{H} = \{h : A \to B\}$ be a family of hash functions and let $\epsilon \geq 0$ be a real number. We say that $\mathcal{H}$ is ϵ-almost universal, denoted ϵ-AU, if for all distinct $M, M' \in A$, we have that $\Pr_{h \leftarrow \mathcal{H}}[h(M) = h(M')] \leq \epsilon$. We say that $\mathcal{H}$ is ϵ-almost universal on equal-length strings if for all distinct, equal-length strings $M, M' \in A$, we have that $\Pr_{h \leftarrow \mathcal{H}}[h(M) = h(M')] \leq \epsilon$.

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

As mentioned earlier, the main idea of this work is to introduce the design of special purpose MACs that can be used to authenticate $\mathcal{E}$ncrypted messages (thus the name "$\mathcal{E}$-MACs"). 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-Authenticate (E&A) composition and the other is based on the Authenticate-then-Encrypt (AtE) composition.

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

The basic goal of any encryption scheme is secrecy; 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 work is to introduce the general idea of utilizing the encryption operation for better designs of

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

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) [46].

Just like fast MACs, the proposed $\mathcal{E}$-MAC utilizes universal hash families in the Carter-Wegman style [72, 73]. 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 [14, 36].) 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 proposed 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 paper 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 [35] or the NH family of Black et al. [14], will satisfy the security requirements detailed in Section 6, 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 [35] 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. Whether the assumption that the encryption algorithm is also a pseudorandom permutation or a strong pseudorandom permutation can be utilized for further improvements in $\mathcal{E}$-MACs performance is left for a continuing research in this direction.

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,[2] 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.

$$\{\alpha\beta \equiv 0 \mod n\} \Rightarrow \{\alpha \equiv 0 \ \vee \ \beta \equiv 0 \mod n\} \tag{1}$$

Property 1 is satisfied by any $\mathbb{Z}_n$ that is also an integral domain [38].

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

$$\delta \equiv rk \mod n \tag{2}$$

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.[3] Thus, the operations used for the rest of the paper are performed over the integer field $\mathbb{Z}_p$.

4.2 Encrypt-and-Authenticate Composition

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

[2] In Section 6, a tighter bound will be derived in the proof of Theorem 3.

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

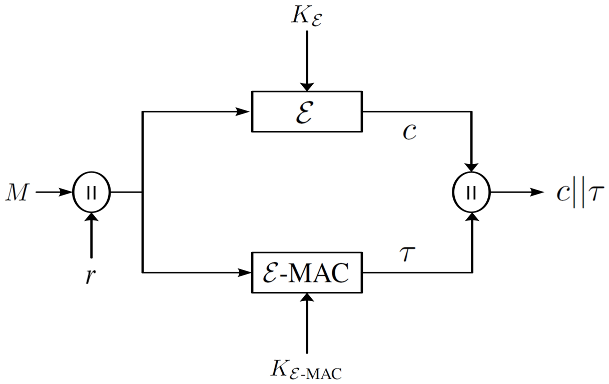

Fig. 2 A block diagram illustrating the use of $\mathcal{E}$-MAC to construct an Encrypt-and-Authenticate (E&A) generic composition. A random number, r, is appended to the plaintext message. The resulting $M||r$ is compressed according to equation (3) and the same $M||r$ goes to the encryption algorithm. The output of the encryption algorithm and $\mathcal{E}$-MAC are concatenated and transmitted to the intended receiver.

4.2.1 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., [16, 37]. 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 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.

4.2.2 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, $M = m_1||m_2||\ldots||m_L$, 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 N-bit long tag of message M is computed as:

$$\tau = \sum_{i=1}^{L} k_i m_i + k_B r \mod p, \tag{3}$$

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-Authenticate generic composition is shown in Figure 2.

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).

A pseudocode describing the signing algorithm of the proposed E&A composition is shown in Algorithm 1.

Algorithm 1 $\mathcal{S}(K, M, r)$

```
if $|M| >$ MaxLen then
  Return 0
end if
Write $K$ as a sequence of blocks $k_1 \parallel \cdots \parallel k_B$ to get $k_1$ through $k_B$;
Set $M = M||1$;
Write $M$ as a sequence of blocks $m_1 \parallel \cdots \parallel m_L$ to get $m_1$ through $m_L$;
$\tau = \sum_{i=1}^{L} k_i m_i + k_B r \mod p$;
Return $\tau$
```

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 paper, without loss of generality, we will assume messages with $|M| \leq$ 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 6, 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 [50, 54] (see, e.g., [14] 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 [14,35,50,54] for more details.

4.2.3 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

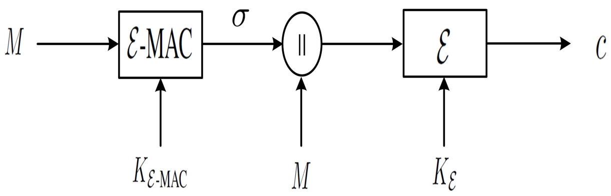

Fig. 3 A block diagram illustrating the use of $\mathcal{E}$-MAC to construct an Authenticate-then-Encrypt generic composition. A compressed image of the message is first computed according to equation (5). The compressed image is then appended to the plaintext message and the result goes to the encryption algorithm as input. The output of the encryption algorithm is transmitted to the intended receiver.

integrity check must be satisfied for the message to be authenticated:

$$\tau \overset{?}{\equiv} \sum_{i=1}^{L} k_i m_i + k_B r \mod p. \tag{4}$$

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.

A pseudocode describing the verifying algorithm of the proposed Encrypt-and-Authenticate composition is shown in Algorithm 2.

Algorithm 2 $\mathcal{V}(K, M, r, \tau)$

```
Write $K$ as a sequence of blocks $k_1 \parallel \cdots \parallel k_B$ to get $k_1$ through $k_B$;
Write $M$ as a sequence of blocks $m_1 \parallel \cdots \parallel m_L$ to get $m_1$ through $m_L$;
$\tau' = \sum_{i=1}^{L} k_i m_i + k_B r \mod p$;
if $\tau' = \tau$ then
    Return 1
else
    Return 0
end if
```

4.3 Authenticate-then-Encrypt Composition

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

4.3.1 Instantiation

As in the E&A of the Section 4.2, 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.

4.3.2 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

$$\sigma = \sum_{i=1}^{L} k_i m_i \mod p, \tag{5}$$

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 Authenticate-then-Encrypt composition is shown in Figure 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 paper, we will assume the latter scenario ($c = \mathcal{E}(M)$ will denote the ciphertext and $\tau = \mathcal{E}(\sigma)$ will denote the authentication tag). A pseudocode describing the signing algorithm of the proposed AtE composition is shown in Algorithm 3.

Algorithm 3 $\mathcal{S}(K, M)$

```
if $|M| >$ MaxLen then
    Return 0
end if
Write $K$ as a sequence of blocks $k_1 \parallel \cdots \parallel k_B$ to get $k_1$ through $k_B$;
Set $M = M||1$;
Write $M$ as a sequence of blocks $m_1 \parallel \cdots \parallel m_L$ to get $m_1$ through $m_L$;
$\tau = \mathcal{E}\Big( \sum_{i=1}^{L} k_i m_i \mod p \Big)$;
Return $\tau$
```

4.3.3 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:

$$\tau \overset{?}{\equiv} \mathcal{E}\Big(\sum_{i=1}^{L} k_i m_i \mod p \Big). \tag{6}$$

A pseudocode describing the verifying algorithm of the proposed Authenticate-then-Encrypt composition is shown in Algorithm 4.

Algorithm 4 $\mathcal{V}(K, M, \tau)$

```
Write $K$ as a sequence of blocks $k_1 \parallel \cdots \parallel k_B$ to get $k_1$ through $k_B$;
Write $M$ as a sequence of blocks $m_1 \parallel \cdots \parallel m_L$ to get $m_1$ through $m_L$;
$\tau' = \mathcal{E}\Big( \sum_{i=1}^{L} k_i m_i \mod p \Big)$;
if $\tau' = \tau$ then
    Return 1
else
    Return 0
end if
```

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

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. As discussed earlier, 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

[14], the cryptographic hash functions SHA-256 and SHA-512 run in about 23.73 cycles/byte and 40.18 cycles/byte, respectively [59]. 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 [46, 69]. 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[4]). This is due to the fact that when combining block ciphers with universal hashing to construct secure channels, the encryption operation is the most time-consuming process [52]. Since stream ciphers are much faster than block ciphers, it is evident that using stream ciphers to build secure channels will be faster [52].

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-Authenticate (EtA) generic composition, on the other hand, authentication tags are functions of ciphertexts. Since ciphertexts must be sent in the clear, EtA compositions cannot take advantage of $\mathcal{E}$-MACs.

6 Security Analysis

This section is devoted to the security analysis of the proposed compositions of Sections 4.2 and 4.3. We start be stating general lemmas.

6.1 General Lemmas

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

Lemma 1 *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,

$$k_i m_i' - k_i m_i = k_i(m_i' - m_i) \tag{7}$$

$$= k_i \delta \overset{?}{\equiv} 0 \mod p. \tag{8}$$

Trivially, the value $\delta \equiv 0 \mod p$ satisfies the condition in equation (8). 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 (8) 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 2 *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] Although stream cipher based authenticated encryption primitives have appeared in [27, 74], such proposals have been analyzed and shown to be vulnerable to attacks [58, 60, 61, 75].

6.2 Security of Encryption

In this section, we show that the privacy of the proposed compositions is provably secure assuming the used encryption algorithm provides indistinguishability under chosen plaintext attacks (IND-CPA). Consider an adversary, $\mathcal{B}$, who is given oracle access to the encryption algorithm, $\mathcal{E}$. The adversary calls the encryption oracle on a polynomial number of messages of her choice and records the corresponding ciphertexts. The adversary then chooses tow equal-length messages, m_0 and m_1, and gives them to the encryption oracle. The oracle draws a bit $b \in \{0, 1\}$ uniformly at random, encrypts m_b, and gives the adversary the resulting ciphertext. The adversary is allowed to perform additional call to the encryption oracle and eventually outputs a bit, b'. We define the adversary's advantage of breaking the security of the encryption algorithm, $\mathcal{E}$, as her probability of successfully guessing the correct bit (equivalently knowing to which plaintext the ciphertext corresponds); that is,

$$\mathsf{Adv}_{\mathcal{E}}^{\text{ind-cpa}}(\mathcal{B}) = \Big| \Pr[b' = b] - 1/2 \Big|. \tag{9}$$

If $\mathcal{E}$ provides IND-CPA, the adversary's advantage of equation (9) is negligible.

We now show that the privacy of the proposed $\mathcal{E}$-MACs is provably secure provided the security of the underlying encryption algorithm.

Theorem 1 *Let $\mathcal{E}$-MAC$_{\text{E\&A}}$ be the authenticated encryption composition described in Section 4.2. Then given an adversary, $\mathcal{A}$, against the privacy of $\mathcal{E}$-MAC$_{\text{E\&A}}$, one can construct an adversary $\mathcal{B}$ against $\mathcal{E}$ such that*

$$\mathsf{Adv}_{\mathcal{E}\text{-MAC}_{\text{E\&A}}}^{\text{priv}}(\mathcal{A}) \le \mathsf{Adv}_{\mathcal{E}}^{\text{ind-cpa}}(\mathcal{B}). \tag{10}$$

Theorem 1 states that if the adversary can expose private information from the proposed E&A composition of Section 4.2, 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 3 *In the Encrypt-and-Authenticate composition described in Section 4.2, 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 $L \le B-1$. Then for any tag $\tau \in \mathbb{Z}_p$ computed according to equation (3) and any plaintext message M the following holds:

$$\Pr(\boldsymbol{\tau} = \tau | \boldsymbol{M} = M) = \Pr\Big(\boldsymbol{r} = (\tau - \sum_{i=1}^{L} k_i m_i)\ k_B^{-1}\Big) \tag{11}$$

$$= \frac{1}{p}, \tag{12}$$

where m_i denotes the i^{th} block of the message M. Equation (12) 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:

$$\Pr(\boldsymbol{\tau} = \tau) = \frac{1}{p}. \tag{13}$$

Combining Bayes' theorem [34] with equations (12) and (13) yields:

$$\Pr(\boldsymbol{M} = M | \boldsymbol{\tau} = \tau) = \frac{\Pr(\boldsymbol{\tau} = \tau | \boldsymbol{M} = M) \Pr(\boldsymbol{M} = M)}{\Pr(\boldsymbol{\tau} = \tau)} \tag{14}$$

$$= \Pr(\boldsymbol{M} = M). \tag{15}$$

Equation (15) 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:

$$\begin{aligned}
&\Pr(\boldsymbol{\tau}_1 = \tau_1, \cdots, \boldsymbol{\tau}_\ell = \tau_\ell) \\
&\quad = \sum_{M_1, \cdots, M_\ell} \Pr(\boldsymbol{\tau}_1 = \tau_1, \cdots, \boldsymbol{\tau}_\ell = \tau_\ell | \boldsymbol{M_1} = M_1, \cdots, \boldsymbol{M_\ell} = M_\ell) \\
&\qquad\qquad \cdot \Pr(\boldsymbol{M_1} = M_1, \cdots, \boldsymbol{M_\ell} = M_\ell) \qquad (16) \\
&\quad = \sum_{M_1, \cdots, M_\ell} \Pr\left(\boldsymbol{r}_1 = (\tau_1 - \sum_{i=1}^{L_1} k_i m_{1_i})\; k_B^{-1}, \cdots, \boldsymbol{r}_\ell = (\tau_\ell - \sum_{i=1}^{L_\ell} k_i m_{\ell_i})\; k_B^{-1}\right) \\
&\qquad\qquad \cdot \Pr(\boldsymbol{M_1} = M_1, \cdots, \boldsymbol{M_\ell} = M_\ell) \qquad (17) \\
&\quad = \sum_{M_1, \cdots, M_\ell} \Pr\left(\boldsymbol{r}_1 = (\tau_1 - \sum_{i=1}^{L_1} k_i m_{1_i})\; k_B^{-1}\right) \cdots \Pr\left(\boldsymbol{r}_\ell = (\tau_\ell - \sum_{i=1}^{L_\ell} k_i m_{\ell_i})\; k_B^{-1}\right) \\
&\qquad\qquad \cdot \Pr(\boldsymbol{M_1} = M_1, \cdots, \boldsymbol{M_\ell} = M_\ell) \qquad (18) \\
&\quad = \sum_{M_1, \cdots, M_\ell} \left(\frac{1}{p}\right)^\ell \cdot \Pr(\boldsymbol{M_1} = M_1, \cdots, \boldsymbol{M_\ell} = M_\ell) \qquad (19) \\
&\quad = \Pr(\boldsymbol{\tau}_1 = \tau_1) \cdots \Pr(\boldsymbol{\tau}_\ell = \tau_\ell), \qquad (20)
\end{aligned}$$

where m_{j_i} denotes the i^{th} block of the j^{th} message M_j. Equation (18) holds due to the independence of the $\boldsymbol{r}_i$'s; equation (19) holds due to the uniform distribution of the $\boldsymbol{r}_i$'s; and equation (20) 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 3, 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 3, 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&A composition and let $\mathcal{B}$ be an adversary with oracle access to the encryption algorithm $\mathcal{E}$. Adversary $\mathcal{A}$ runs adversary $\mathcal{B}$ to attack the privacy of observed ciphertexts. Then,

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

as desired. □

Theorem 2 *Let $\mathcal{E}$-MAC$_{\text{AtE}}$ be the authenticated encryption composition described in Section 4.3. Then given an adversary, $\mathcal{A}$, against the privacy of $\mathcal{E}$-MAC$_{\text{AtE}}$, one can construct an adversary $\mathcal{B}$ against $\mathcal{E}$ such that*

$$\mathsf{Adv}^{\text{priv}}_{\mathcal{E}\text{-MAC}_{\text{AtE}}}(\mathcal{A}) \leq \mathsf{Adv}^{\text{ind-cpa}}_{\mathcal{E}}(\mathcal{B}). \tag{21}$$

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.

6.3 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 4.2 and 4.3. We give here information-theoretic bounds on the adversary's probability of successful forgery assuming the use of an information-theoretically secure encryption (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*

$$\mathsf{Adv}^{\text{auth}}_{\mathcal{E}\text{-MAC}}(\mathcal{A}) \leq \begin{cases} \dfrac{1}{p} & \text{if } \ q = 0 \\ \\ \dfrac{1}{p-1} & \text{if } \ q > 0. \end{cases} \tag{22}$$

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 4.2 and 4.3. 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 3, 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&A composition of Section 4.2, 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:

$$\sum_{\ell=1}^{B} k_\ell \; m'_\ell \overset{?}{\equiv} \tau' \equiv \tau_i \equiv \sum_{\ell=1}^{B} k_\ell \; m_\ell \mod p, \tag{23}$$

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 (23) 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 (23) is equivalent to

$$k_1 m'_1 \equiv k_1 m_1 \mod p. \tag{24}$$

   Therefore, by Lemma 1, 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 (23) is equivalent to

$$k_1\delta_1 + k_2\delta_2 \equiv 0 \mod p. \tag{25}$$

   Therefore, by Lemma 2, 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 (23) is equivalent to

$$k_j \delta_j + \sum_{\substack{\ell \in J \\ \ell \neq j}} k_\ell \delta_\ell \equiv 0 \mod p, \tag{26}$$

for some $j \in J$. Therefore, using Lemma 2 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:

$$\sum_{\ell=1}^{B} k_\ell\ m'_\ell \overset{?}{\equiv} \tau' \equiv \tau_i + \gamma \equiv \sum_{\ell=1}^{B} k_\ell\ m_\ell + \gamma \mod p, \tag{27}$$

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 (27) 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 (27) is equivalent to

$$k_1 \delta \equiv \gamma \mod p. \tag{28}$$

Therefore, by Lemma 2, 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 (27) is equivalent to

$$\sum_{j \in J} k_j \delta_j \equiv \gamma \mod p. \tag{29}$$

By Lemma 2 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 $(\frac{1}{p-1})$-AΔU. The proof also illustrates why any ϵ-AΔU hash family can be used to construct the proposed $\mathcal{E}$-MAC. That is, any ϵ-AΔU 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 [9].

Theorem 4 *The* E&A *generic composition using the $\mathcal{E}$-MAC described in Section 4.2 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 AtE composition of Section 4.3 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 [33, 46]). Although most practical secure encryption algorithms that can be used to construct the AtE of Section 4.3 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 AtE system of Section 4.3. 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.

6.4 Security of the Generic Compositions

In [9], 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 [9] analyzes the security of the three generic compositions.

In [9], Bellare and Namprempre showed that the E&A and AtE generic compositions are generally insecure, the results do not apply to all variants of E&A and AtE constructions. For instance, as per [9], E&A compositions do not generally provide IND-CPA because there exist secure MACs that reveal information about the plaintext (the authors of [9] provide a detailed example). Obviously, if such a MAC is used in the construction of an E&A 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&A composition does not reveal any information about the plaintext that is not revealed by the ciphertext.

Another result of [9] is that generic E&A and AtE 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 [9].) The reason why E&A and AtE 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 [9]. Obviously, when such an encryption algorithm is combined with the proposed $\mathcal{E}$-MACs to construct an E&A or AtE 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&A and AtE 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&A and AtE systems have been constructed in practice. Popular examples of such constructions are SSH [77], SSL [30] and TLS [22], which use variants of the E&A and AtE compositions that are known to be secure [8, 52, 55, 67].

So far, we have shown that $\mathcal{E}$-MACs can be used to replace standard MACs in the construction of E&A and AtE 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&A system to achieve better resilience to a new attack on universal hash functions based MACs; namely, the key-recovery attack [36].

7 $\mathcal{E}$-MACs and Key Recovery

Recently, Handschuh and Preneel [36] 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 [36] 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 paper. 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

$$k_1 \equiv \beta k_2 \mod p, \tag{30}$$

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 [36] 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 [36]. As per [36], 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&A and AtE systems to overcome the key-recovery problem discovered in [36]. Consider the $\mathcal{E}$-MAC proposed in Section 4, and recall that a random number $r \in_R \mathbb{Z}_p$ is generated internally in the E&A process. In the basic construction of Section 4, 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, the universal hashing key is $K = k_1||k_2||\cdots||k_B$ and the authentication tag is computed as:

$$\tau = \sum_{i=1}^{L} k_i m_i + k_B r \mod p. \tag{31}$$

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

$$\tau = \sum_{i=1}^{L} (k_i \oplus r) m_i + k_B r \mod p. \tag{32}$$

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,

$$k_1' \equiv \beta k_2' \mod p, \tag{33}$$

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 (32) instead of equation (31), 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.

8 Conclusion and Future Work

In this work, 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&A and AtE compositions. By taking advantage of the E&A and AtE 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.

References

1. Alomair, B., Clark, A., Poovendran, R.: The power of primes: security of authentication based on a universal hash-function family. Journal of Mathematical Cryptology **4**(2), 121–147 (2010)
2. Alomair, B., Poovendran, R.: Efficient Authentication for Mobile and Pervasive Computing. In: the 12th International Conference on Information and Communications Security – ICICS'10, *Lecture Notes in Computer Science*, vol. 6476, pp. 186–202. Springer (2010)
3. Alomair, B., Poovendran, R.: $\mathcal{E}$-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)
4. Bellare, M.: New proofs for NMAC and HMAC: Security without collision-resistance. In: Proceedings of the 16th Annual International Cryptology Conference – CRYPTO'06, *Lecture Notes in Computer Science*, vol. 4117, pp. 602–619. Springer (2006)
5. Bellare, M., Canetti, R., Krawczyk, H.: Keying hash functions for message authentication. In: Proceedings of the 16th Annual International Cryptology Conference – CRYPTO'96, *Lecture Notes in Computer Science*, vol. 1109, pp. 1–15. Springer (1996)
6. Bellare, M., Guerin, R., Rogaway, P.: XOR MACs: New methods for message authentication using finite pseudorandom functions. In: Proceedings of the 15th Annual International Cryptology Conference – CRYPTO'95, *Lecture Notes in Computer Science*, vol. 963, pp. 15–28. Springer (1995)
7. Bellare, M., Kilian, J., Rogaway, P.: The Security of the Cipher Block Chaining Message Authentication Code. Journal of Computer and System Sciences **61**(3), 362–399 (2000)
8. Bellare, M., Kohno, T., Namprempre, C.: 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)
9. Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. Journal of Cryptology **21**(4), 469–491 (2008)
10. Bellare, M., Pietrzak, K., Rogaway, P.: Improved security analyses for CBC MACs. In: Proceedings of the 25th Annual International Cryptology Conference – CRYPTO'05, *Lecture Notes in Computer Science*, vol. 3621, pp. 527–545. Springer (2005)
11. Bellare, M., Rogaway, P., Wagner, D.: The EAX mode of operation. In: Proceedings of the 11th International Workshop on Fast Software Encryption – FSE'04, *Lecture Notes in Computer Science*, vol. 3017, pp. 389–407. Springer (2004)
12. Bernstein, D.: Floating-point arithmetic and message authentication. Unpublished manuscript (2004). Available at `http://cr.yp.to/hash127.html`
13. Bernstein, D.: The Poly1305-AES message-authentication code. In: Proceedings of the 12th International Workshop on Fast Software Encryption – FSE'05, *Lecture Notes in Computer Science*, vol. 3557, pp. 32–49. Springer (2005)
14. Black, J., Halevi, S., Krawczyk, H., Krovetz, T., Rogaway, P.: UMAC: Fast and Secure Message Authentication. In: Proceedings of the 19th Annual International Cryptology Conference – CRYPTO'99, *Lecture Notes in Computer Science*, vol. 1666, pp. 482–499. Springer (1999)
15. Black, J., Rogaway, P.: 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, *Lecture Notes in Computer Science*, vol. 2332, pp. 384–397. Springer (2002)
16. Blum, L., Blum, M., Shub, M.: A Simple Unpredictable Pseudo-random Number Generator. SIAM Journal on Computing **15**(2), 364 (1986)
17. Bosselaers, A., Govaerts, R., Vandewalle, J.: Fast hashing on the Pentium. In: Proceedings of the 16th Annual International Cryptology Conference – CRYPTO'96, *Lecture Notes in Computer Science*, vol. 1109, pp. 298–312. Springer (1996)
18. Canetti, H., Krawczyk, H.: 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, *Lecture Notes in Computer Science*, vol. 2045, pp. 451–472. Springer (2001)
19. Carter, J., Wegman, M.: Universal classes of hash functions. In: Proceedings of the 9th ACM Symposium on Theory of Computing – STOC'77, pp. 106–112. ACM SIGACT (1977)

20. Carter, L., Wegman, M.: Universal hash functions. Journal of Computer and System Sciences **18**(2), 143–154 (1979)
21. Contini, S., Yin, Y.: 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, *Lecture Notes in Computer Science*, vol. 4284, pp. 37–53. Springer (2006)
22. Dierks, T., Rescorla, E.: RFC 5246: The transport layer security (TLS) protocol version 1.2. Tech. rep., Internet Engineering Task Force (2008)
23. Doraswamy, N., Harkins, D.: IPSec: the new security standard for the Internet, intranets, and virtual private networks. Prentice Hall (2003)
24. Dworkin, M.: Recommendation for block cipher modes of operation: The CMAC mode for authentication. National Institute of Standards and Technology (NIST) Special Publication 800-38B (2005)
25. Dworkin, M.: 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)
26. Etzel, M., Patel, S., Ramzan, Z.: Square hash: Fast message authentication via optimized universal hash functions. In: Proceedings of the 19th Annual International Cryptology Conference – CRYPTO'99, *Lecture Notes in Computer Science*, vol. 1666, pp. 234–251. Springer (1999)
27. Ferguson, N., Whiting, D., Schneier, B., Kelsey, J., Kohno, T.: Helix: Fast encryption and authentication in a single cryptographic primitive. In: Proceedings of the 10th International Workshop on Fast Software Encryption – FSE'03, *Lecture Notes in Computer Science*, vol. 2887, pp. 330–346. Springer (2003)
28. FIPS 113: Computer Data Authentication. Federal Information Processing Standards Publication (1985)
29. FIPS 198: The Keyed-Hash Message Authentication Code (HMAC). Federal Information Processing Standards Publication (2002)
30. Freier, A., Karlton, P., Kocher, P.: The SSL Protocol Version 3.0. Internet Engineering Task Force (IETF) (2011)
31. Gligor, V., Donescu, P.: Integrity-Aware PCBC Encryption Schemes. In: Proceedings of the 7th International Workshop on Security Protocols – SP'99, *Lecture Notes in Computer Science*, vol. 1796, pp. 153–168. Springer (2000)
32. Gligor, V., Donescu, P.: Fast Encryption and Authentication: XCBC Encryption and XECB Authentication Modes. In: Proceedings of the 9th International Workshop on Fast Software Encryption – FSE'01, *Lecture Notes in Computer Science*, vol. 2355, pp. 1–20. Springer (2002)
33. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences **28**(2), 270–299 (1984)
34. Gubner, J.: Probability and random processes for electrical and computer engineers. Cambridge University Press (2006)
35. Halevi, S., Krawczyk, H.: MMH: Software message authentication in the Gbit/second rates. In: Proceedings of the 4th International Workshop on Fast Software Encryption – FSE'97, *Lecture Notes in Computer Science*, vol. 1267, pp. 172–189. Springer (1997)
36. Handschuh, H., Preneel, B.: 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, pp. 144–161. Springer (2008)
37. Hastad, J., Impagliazzo, R., Levin, L., Luby, M.: A Pseudorandom Generator from Any One-Way Function. SIAM Journal on Computing **28**(4), 1364–1396 (1999)
38. Herstein, I.: Abstract algebra. Macmillan New York (1986)
39. ISO/IEC 9797-1: Information technology – Security techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher (1999)
40. ISO/IEC 9797-2: Information technology – Security techniques – Message Authentication Codes (MACs) – Part 2: Mechanisms using a dedicated hash-function (2002)
41. Iwata, T., Kurosawa, K.: omac: One-key cbc mac. In: Proceedings of the 10th International Workshop on Fast Software Encryption – FSE'03, *Lecture Notes in Computer Science*, vol. 2887, pp. 129–153. Springer (2003)
42. Jaulmes, E., Joux, A., Valette, F.: 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, *Lecture Notes in Computer Science*, vol. 2365, pp. 613–616. Springer (2002)
43. Jonsson, J.: On the security of CTR+ CBC-MAC. In: Proceedings of the 9th Annual International Workshop on Selected Areas in Cryptography – SAC'02, *Lecture Notes in Computer Science*, vol. 2595, pp. 76–93. Springer (2002)
44. Jutla, C.: Encryption modes with almost free message integrity. Journal of Cryptology **21**(4), 547–578 (2008)
45. Kaps, J., Yuksel, K., Sunar, B.: Energy scalable universal hashing. IEEE Transactions on Computers **54**(12), 1484–1495 (2005)
46. Katz, J., Lindell, Y.: Introduction to modern cryptography. Chapman & Hall/CRC (2008)
47. Kim, J., Biryukov, A., Preneel, B., Hong, S.: 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, *Lecture Notes in Computer Science*, vol. 4116, pp. 242–256. Springer (2006)
48. Kohl, J., Neuman, C.: The Kerberos Network Authentication Service (V5). Tech. rep., RFC 1510 (1993)
49. Kohno, T., Viega, J., Whiting, D.: CWC: A high-performance conventional authenticated encryption mode. In: Proceedings of the 11th International Workshop on Fast Software Encryption – FSE'04, *Lecture Notes in Computer Science*, vol. 3017, pp. 408–426. Springer (2004)
50. Krawczyk, H.: LFSR-based hashing and authentication. In: Proceedings of the 14th Annual International Cryptology Conference – CRYPTO'94, *Lecture Notes in Computer Science*, vol. 839, pp. 129–139. Springer (1994)
51. Krawczyk, H.: New hash functions for message authentication. In: Proceedings of the 15th Annual International Cryptology Conference – CRYPTO'95, *Lecture Notes in Computer Science*, vol. 921, pp. 301–310. Springer (1995)
52. Krawczyk, H.: 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, *Lecture Notes in Computer Science*, vol. 2139, pp. 310–331. Springer (2001)
53. Krovetz, T.: `http://fastcrypto.org/umac/`
54. Mansour, Y., Nisan, N., Tiwari, P.: The computational complexity of universal hashing. In: Proceedings of the 22nd ACM Symposium on Theory of Computing – STOC'90, pp. 235–243. ACM SIGACT (1990)
55. Maurer, U., Tackmann, B.: 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, pp. 505–515. ACM SIGSAC (2010)

56. McGrew, D., Viega, J.: 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, *Lecture Notes in Computer Science*, vol. 3348, pp. 343–355. Springer (2004)
57. Meyer, C., Matyas, S.: Cryptography: A New Dimension in Computer Data Security. John Wiley & Sons (1982)
58. Muller, F.: Differential attacks against the Helix stream cipher. In: Proceedings of the 11th International Workshop on Fast Software Encryption – FSE'04, *Lecture Notes in Computer Science*, vol. 3017, pp. 94–108. Springer (2004)
59. Nakajima, J., Matsui, M.: 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, *Lecture Notes in Computer Science*, vol. 2332, pp. 165–180. Springer (2002)
60. Paul, S., Preneel, B.: 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, *Lecture Notes in Computer Science*, vol. 3797, pp. 90–103. Springer (2005)
61. Paul, S., Preneel, B.: Solving systems of differential equations of addition. In: Proceedings of the 10th Australasian Conference on Information Security and Privacy – ICISP'05, *Lecture Notes in Computer Science*, vol. 3574, pp. 75–88. Springer (2005)
62. Preneel, B., van Oorschot, P.: On the security of two MAC algorithms. In: Proceedings of the 15th International Conference on the Theory and Application of Cryptographic Techniques – EUROCRYPT'96, *Lecture Notes in Computer Science*, vol. 1070, pp. 19–32. Springer (1996)
63. Preneel, B., Van Oorschot, P.: MDx-MAC and building fast MACs from hash functions. In: Proceedings of the 15th Annual International Cryptology Conference – CRYPTO'95, *Lecture Notes in Computer Science*, vol. 963, pp. 1–14. Springer (1995)
64. Preneel, B., Van Oorschot, P.: On the security of iterated message authentication codes. IEEE Transactions on Information theory **45**(1), 188–199 (1999)
65. Rogaway, P.: Bucket hashing and its application to fast message authentication. Journal of Cryptology **12**(2), 91–115 (1999)
66. Rogaway, P., Bellare, M., Black, J., Krovetz, T.: 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, pp. 196–205. ACM SIGSAC (2001)
67. Stallings, W.: Cryptography and network security: principles and practice. Prentice Hall (2010)
68. Stinson, D.: Universal hashing and authentication codes. Designs, Codes and Cryptography **4**(3), 369–380 (1994)
69. Stinson, D.: Cryptography: Theory and Practice. CRC Press (2006)
70. van Tilborg, H.: Encyclopedia of cryptography and security. Springer (2005)
71. Tsudik, G.: Message authentication with one-way hash functions. ACM Computer Communication Review **22**(5), 38 (1992)
72. Wegman, M., Carter, J.: New classes and applications of hash functions. In: Proceedings of the 20th Annual IEEE Symposium on Foundations of Computer Science – FOCS'79, pp. 175–182. IEEE Computer Society (1979)
73. Wegman, M., Carter, L.: New hash functions and their use in authentication and set equality. Journal of Computer and System Sciences **22**(3), 265–279 (1981)
74. Whiting, D., Schneier, B., Lucks, S., Muller, F.: Phelix – fast encryption and authentication in a single cryptographic primitive. ECRYPT Stream Cipher Project, Report 2005/020, `www.ecrypt.eu.org/stream` (2005)
75. Wu, H., Preneel, B.: Differential-linear attacks against the stream cipher Phelix. In: Proceedings of the 14th International Workshop on Fast Software Encryption – FSE'07, *Lecture Notes in Computer Science*, vol. 4593, pp. 87–100. Springer (2007)
76. Yasuda, K.: Multilane HMAC– Security beyond the birthday limit. In: Proceedings of the 8th International Conference on Cryptology in India – INDOCRYPT'07, *Lecture Notes in Computer Science*, vol. 4859, pp. 18–32. Springer (2007)
77. Ylonen, T., Lonvick, C.: The Secure Shell (SSH) Transport Layer Protocol. Tech. rep., RFC 4253 (2006)