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

Basel Alomair and Radha Poovendran

Network Security Lab (NSL)
University of Washington-Seattle
{alomair,rp3}@uw.edu

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 employment 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 can be a redundancy in the computations performed by the two primitives. If this turned out to be the case, removing such redundancy will improve the efficiency of the overall construction. In addition, computations performed by the encryption algorithm can be further utilized to improve the security of the authentication algorithm. In this work, we show how $\mathcal{E}$-MACs can be designed to reduce the amount of computations required by standard MACs based on universal hash functions, and show how $\mathcal{E}$-MACs can be secured against key-recovery attacks.

Key words: Confidentiality, authenticity, message authentication code (MAC), authenticated encryption, encrypt-and-authenticate, universal hash families

1 Introduction

There are two main approaches for the construction of secure cryptographic channels: 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., [6, 18, 23, 32, 35, 45]). In the generic approach, an authentication primitive is combined with an encryption primitive to provide message integrity and confidentiality (see, e.g., [14, 21, 51]).

Generic compositions can be constructed in three different ways: encrypt-and-authenticate ($E\&A$), encrypt-then-authenticate (EtA), and authenticate-then-encrypt (AtE). In the $E\&A$ composition, the plaintext is passed to the encryption algorithm to get the corresponding ciphertext, the plaintext is passed to the MAC algorithm to get the corresponding tag, and the resulting ciphertext-tag pair $(\mathcal{E}(M), \text{MAC}(M))$ is transmitted to the intended receiver. In the EtA composition, 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, and the resulting $(\mathcal{E}(M), \text{MAC}(\mathcal{E}(M)))$ is transmitted to the intended receiver. In the AtE composition, the plaintext is passed to the MAC algorithm to get a tag, the resulting tag is appended to the plaintext message and the result is passed to the encryption algorithm, and the resulting $(\mathcal{E}(M, \text{MAC}(M)))$ is transmitted to the intended receiver. The transport layer of SSH uses a variant of $E\&A$ [51], IPsec uses a variant of EtA [14], while SSL uses a variant of AtE [21].

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 [38]. 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 [38].

The $E\&A$ composition has a parallelizable advantage over the EtA and the AtE constructions. The fact that the encryption and authentication operations can be performed simultaneously can further increase the efficiency of the generic composition. On the other hand, the $E\&A$ composition imposes an extra requirement on the MAC algorithm. As opposed to the EtA and AtE compositions, the tag in the $E\&A$ composition is a function of the plaintext message (not the ciphertext as in EtA) and is sent in the clear (not encrypted as in AtE). 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. This implies that generic compositions are more involved than just combining an encryption algorithm and a MAC algorithm. Indeed, in [38] and [5], 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. 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, no effort has been made to design new primitives that utilize the special characteristics of the 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 $E\&A$ compositions. The driving motive behind this work was the intuition that MACs used in the generic composition 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 a redundancy in the computations performed by the two primitives. If this turned out to be the case, removing such redundancy can improve the efficiency of the overall composition.

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 Wegman-Carter style [13, 49] 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 [48].

Recently, however, Handschuh and Preneel [27] discovered a vulnerability in universal hashing based MACs. They demonstrated that once a collision in the hashing phase occurs, secret key information can be exposed, allowing subsequent forgeries to succeed with high probabilities. Their attack is not directed to a specific universal hash family and can be applied to all such MACs. The recommendations of the work in [27] are not to reuse the universal hash function key, thus going back to the impractical 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 propose the deployment of a new cryptographic primitive for the construction of secure channels using the $E\&A$ composition. 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 $E\&A$ system 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 show how $\mathcal{E}$-MACs can further utilize the special structure of the $E\&A$ system to improve the security of the authentication process. More specifically, 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 $E\&A$ structure.

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 [19] and the International Organization for Standardization ISO/IEC 9797-1 [29]. CMAC, a modified version of CBC-MAC, is presented in the NIST special publication 800-38B [15], which was based on OMAC of Iwata and Kurosawa [31]. Other block cipher based MACs include, but are not limited to, XOR-MAC [2] and PMAC [46]. The security of different MACs has been exhaustively studied (see, e.g., [3, 43]).

HMAC is a popular example of the use of iterated cryptographic hash functions to design MACs [1], which was adopted as a standard [20]. Another cryptographic hash function based MAC is the MDx-MAC of Preneel and Oorschot [42]. HMAC and two variants of MDx-MAC are specified in the International Organization for Standardization ISO/IEC 9797-2 [30]. 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 [11].

The use of universal hash families was pioneered by Wegman and Carter [13,49] in the context of designing unconditionally secure authentication. The use of universal hash functions for the design of computationally secure MACs appeared in [7–9,17,26,33,40]. 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.

Since in many practical applications both message confidentiality and authenticity are sought, 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 [5]. Establishing secure channels by means of generic constructions of authenticated encryption schemes was of particular interest. The security relations among different notions of security in authenticated encryption schemes was studied in detail in [5]. In [12], it was shown that EtA schemes build secure channels and, in [38], the security of the three generic construction methods is analyzed.

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 [22, 34, 41]. Such simple schemes, however, are known to be vulnerable to attacks [32]. Other dedicated schemes that combine encryption and message authenticity include [6, 18, 23, 32, 35, 45]. In [32], Jutla proposed the integrity aware parallelizable mode (IAPM), an encryption scheme with authentication. Gligor and Donescu proposed the XECB-MAC [23].

Rogaway *et al.* [45] proposed OCB: a block-cipher mode of operation for efficient authenticated encryption. Kohno *et al.* [35] proposed a high-performance conventional authenticated encryption mode (CWC), which the NIST standard Galois/Counter Mode (GCM) was based on [16].

3 Preliminaries

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. 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 K, M, and 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 returns the decision $d = \mathcal{V}(K, M, \tau)$ to $\mathcal{A}$.

The adversary's attack is a (q_s, q_v)-attack if during the course of the attack $\mathcal{A}$ makes no more than q_s signing queries and no more than q_v verify queries. The outcome of running the experiment in the presence of an adversary is used to define security. As in [5], 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 fast MACs, the proposed $\mathcal{E}$-MAC is based on universal hash-function families. 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., [13, 26, 36, 37, 44, 47, 49]). We give below a formal definition of one class of universal hash families called ϵ-almost universal [9].

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 $\Pr_{h \leftarrow \mathcal{H}}[h(M) = h(M')] \leq \epsilon$. $\mathcal{H}$ 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 $\Pr_{h \leftarrow \mathcal{H}}[h(M) = h(M')] \leq \epsilon$.*

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

4.1 Overview

Semantic security (or equivalently indistinguishability under chosen plaintext attacks (IND-CPA) [24]) is the only assumption we make on the underlying encryption algorithm. In fact, secure deterministic encryption algorithms suffices for our construction. However, since semantic security is a basic requirement in most applications, we will assume the use of a semantically secure encryption.

As in fast MACs in the literature, the proposed $\mathcal{E}$-MAC utilizes universal hash-function families in the Wegman-Carter style [13, 49]. However, as opposed to universal hash functions based MACs, we will show that $\mathcal{E}$-MACs can be secure without any post computation 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 [9, 27].) 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. In fact, any ϵ-almost-Δ-universal (ϵ-AΔU) hash family, such as the MMH family of Halevi and Krawczyk [26] and the NH family of Black *et al.* [9], will satisfy the security requirements detailed in Section 5. (The ϵ-AΔU is a stronger notion than ϵ-AU given in Definition 1; interested readers may refer to [26] 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. That is, whether the encryption is a stream cipher, cipher block chaining (CBC) mode block cipher, electronic code book (ECB) mode block cipher, etc., can have an impact on the design and performance of the composition. We only show here how the semantic security of the underlying encryption algorithm can be utilized to improve the efficiency and security of message authentication. Further improvements in $\mathcal{E}$-MACs performance using specific modes of operations is left for a continuing research in this direction.

4.2 Description

Instantiation. Fix an encryption primitive $\mathcal{E}$ that is semantically secure. Based on a security parameter N, legitimate users agree on an N-bit long prime integer p. Let $K = (k_1, k_2, \ldots, k_B)$, for k_i's drawn uniformly and independently from $\mathbb{Z}_p^*$, 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 can be long. One way to generate such a key is via a pseudorandom generator, e.g., [10, 28]. In such a case, only the seed of the pseudorandom generator is required to be distributed to the legitimate parties. 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 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.

Authentication. Without loss of generality, we assume the message can be divided into B-1

blocks of length N-bits, that is $M = m_1||m_2||\ldots||m_{B-1}$. (We overload m_i to denote both the binary string in the i^{th} block and the integer representation of the i^{th} block as an element of $\mathbb{Z}_p$; the distinction between the two representations will be omitted when it is clear from the context.) For every message M to be encrypted and authenticated, the sender draws an integer r uniformly at random from $\mathbb{Z}_p$ anew for each message (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 receiver (the symbol "||" denotes the concatenation operation), along with the the N-bit long tag of message M computed as:

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

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

Remark 1. A misconception about universal hash-function families is that the authentication key needs to be as long as the longest message to be authenticated. Obviously, if this was true, universal hashing will be impractical for most applications. In the literature, there exist standard techniques to hash arbitrary-length messages using a fixed-length key. The first such technique was proposed by Wegman and Carter in [50], and later refined by Halevi and Krawczyk in [26]. The work of Black *et al.* [9] provides a different generic algorithm to transform any hash function that is ϵ-AU on *equal-length* messages, h, to a hash function that is ϵ-AU on *arbitrary-length* messages, h^*. However, for a lack of space and for a better continuity of the main ideas of the paper, we omit going into the details of such techniques. (Interested readers may refer to [9, 26, 50] for more information.) Therefore, we emphasize that the key $K = (k_1, k_2, \ldots, k_B)$ can be used to authenticate arbitrary-length messages.

Remark 2. Clearly, as will be formally proven in Section 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. 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 [36,39]. (See, e.g., [9] for a detailed use of Toeplitz-extension to increase the security of MACs based on universal hash functions.) Again, we omit describing this topic since it is out of the scope of this work and refer interested readers to [9, 26, 36, 39] for more details.

Verification. Upon receiving a ciphertext-tag pair, (c, τ), 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 $\sum_{i=1}^{B-1} k_i m_i + k_B r$ 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:

$$\tau \stackrel{?}{\equiv} \sum_{i=1}^{B-1} k_i m_i + k_B r \mod p. \tag{2}$$

Remark 3. 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.

5 Security Analysis

5.1 Security of the Proposed $\mathcal{E}$-MAC

Assume that message M has been encrypted with any semantically secure encryption scheme. For the rest of the paper, we will refer to M and τ as the message and the tag generated at the transmitter's end, respectively; while M' and τ' represent the message and the tag at the receiver's end, respectively. For ease of notation, we will refer to the plaintext message as the concatenation of M and r. 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.*

Lemma 1 is a direct consequence of the fact that, for a prime integer p, $\mathbb{Z}_p$ is a field.

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 $\delta_1 \in \mathbb{Z}_p^*$. Since every nonzero element in $\mathbb{Z}_p$ is invertible, 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 $1/(p-1)$, and the lemma follows. □

Lemma 3. *Authentication tags are statistically independent of their corresponding messages, and different authentication tags are mutually independent.*

The proof of Lemma 3 is provided in Appendix A. Now we can proceed with the proofs of the main claims of the proposed $\mathcal{E}$-MAC. Recall that applying a cryptographic function to the compressed image is an essential operation for the security of standard universal hash functions based MACs. Without such a cryptographic operation, the key of the universal hash function can be exposed (by chosen message attacks, for instance). We now formally prove two important claims about $\mathcal{E}$-MACs. Namely, the semantic security of the used encryption algorithm suffices to protect the key of the proposed $\mathcal{E}$-MAC, and tags do not reveal any information about the plaintext that is not revealed by the corresponding ciphertext.

Theorem 1. *An adversary able to extract any information about the proposed $\mathcal{E}$-MAC's secret key, or extract any information about the plaintext from authentication tags is able to break the semantic security of the underlying encryption algorithm.*

Proof. By Lemma 3, each tag is independent of its corresponding message and the secret key. Therefore, by only observing a single tag, the adversary cannot reveal any information about the authenticated message or the secret key. Furthermore, also by Lemma 3, 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. This holds as long as the coin tosses, the r's, remain secret. The transmitted r's, however, are generated internally and encrypted with the semantically secure algorithm $\mathcal{E}$. Therefore, no information about the proposed $\mathcal{E}$-MAC's key nor the authenticated messages can be exposed, unless the adversary can extract secret information about the r's, which can be done only by breaking the semantic security of the encryption algorithm. □

Remark 4. In addition to its important statement regarding the secrecy of transmitted messages, Theorem 1 presents an important statement regarding the secrecy of the nonce r. Clearly, if some r's are revealed, partial key information can be exposed. Other than attacking the system in a non-cryptographic way, Theorem 1 states that the only way to expose secrete information about the r's is by breaking the semantic security of the encryption algorithm. That is, from a cryptographic point of view, it is safe to assume that no information about the r's will be revealed.

This shows how $\mathcal{E}$-MACs can take advantage of the $E\&A$ structure to improve authentication efficiency and satisfy their secrecy requirement. All that is needed is to generate a random string, append it to the encrypted plaintext message, and use it to encrypt the authentication tag. Therefore, as claimed earlier, no post-processing of the compressed image is required and the secrecy requirement of $\mathcal{E}$-MACs can be achieved, without expensive computational effort. This result is not surprising. In fact, it supports the main motive behind this work, namely the intuition that post-processing the compressed image by a cryptographic function can be replaced by computations performed by the encryption algorithm (i.e., post-processing is redundant in such compositions).

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

Theorem 2. *Let Σ denotes the proposed $\mathcal{E}$-MAC and let $\mathcal{A}$ be an adversary making a (q_s, q_v)-attack on Σ. Given the semantic security of the underlying encryption scheme, $\mathcal{A}$'s advantage of successful forgery is at most*

$$\mathsf{Adv}_{\mathcal{A}}^{\Sigma} = \begin{cases} \dfrac{q_v}{p} & \text{if } \ q_s = 0 \\ \\ \dfrac{q_v}{p-1} & \text{if } \ q_s > 0. \end{cases} \tag{3}$$

Proof. From the proof of Lemma 3 (equation (16)), 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_s times and recorded $(M_1, \tau_1), \cdots, (M_{q_s}, \tau_{q_s})$. By Theorem 1, given the semantic security of the encryption algorithm, no information about the $\mathcal{E}$-MAC's secret key is revealed by the observed τ_i's.

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 $i = 1, \cdots, q_s$. We break the proof into two cases: queried tag and unqueried tag. For ease of notations, r_i will be denoted as the B^{th} block of the i^{th} message, that is, $r_i = m_{i_B}$.

QUERIED TAG $(M', \tau' = \tau_q)$: Assume that $\tau' = \tau_q$ for a $q \in \{1, \cdots, q_s\}$. 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:

$$\sum_{\ell=1}^{B} k_\ell \; m'_\ell \stackrel{?}{\equiv} \tau' \equiv \tau_q \equiv \sum_{\ell=1}^{B} k_\ell \; m_\ell \mod p, \tag{4}$$

where m'_ℓ denotes the ℓ^{th} block of M' and m_ℓ denotes the ℓ^{th} block of M_q (note that we write m_ℓ instead of m_{q_ℓ} for ease of notations since no distinction between different messages is necessary). We will analyze equation (4) by considering the following three cases: M' and M_q differ by a single block, M' and M_q differ by two blocks, or M' and M_q 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) is equivalent to

$$k_1 m'_1 \equiv k_1 m_1 \mod p. \tag{5}$$

   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 (4) is equivalent to

$$k_1\delta_1 + k_2\delta_2 \equiv 0 \mod p. \tag{6}$$

   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., $m'_i \equiv m_i + \delta_i \not\equiv m_i \mod p$; $\forall\ i \in I \subseteq \{1, 2, \cdots, B\}; |I| \geq 3$. Then, equation (4) is equivalent to

$$k_i\delta_i + \sum_{\substack{j\in I \\ j\neq i}} k_j\delta_j \equiv 0 \mod p, \tag{7}$$

   for some $i \in I$. Therefore, using Lemma 2 and the fact that $\sum_{j\in I, j\neq i} k_j\delta_j$ 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, $\sum_{j\in I, j\neq i} k_j\delta_j$ can still be congruent to zero modulo p.)

From the above three cases, the probability of successful forgery when the forged tag has been outputted by 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, $\tau' \neq \tau_q$ for all $q = 1, \cdots, q_s$. 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_q, for a $q \in \{1, \cdots, q_s\}$. Let $\tau' \equiv \tau_q + \gamma \mod p$ 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:

$$\sum_{\ell=1}^{B} k_\ell\ m'_\ell \stackrel{?}{\equiv} \tau' \equiv \tau_q + \gamma \equiv \sum_{\ell=1}^{B} k_\ell\ m_\ell + \gamma \mod p, \tag{8}$$

where m'_ℓ denotes the ℓ^{th} block of M' and m_ℓ denotes the ℓ^{th} block of M_q. Bellow we analyze equation (8) by considering two cases: M' and M_q differ by a single block, or M' and M_q differ by more than one block.

1. Without loss of generality, assume that M' and M_q differ in the first block only. That is $m'_1 \equiv m_1 + \delta \not\equiv m_1 \mod p$ and $m'_i \equiv m_i \mod p$ for all $i = 2, \cdots, B$. Then, equation (8) is equivalent to

$$k_1\delta \equiv \gamma \mod p. \tag{9}$$

   Therefore, by Lemma 2, the probability of success is at most $1/(p-1)$.

2. Assume now that M' and M_q differ by more than one block. That is, $m'_i \equiv m_i + \delta_i \neq m_i \mod p$; $\forall i \in I \subseteq \{1, 2, \cdots, B\}; |I| \geq 2$. Then, equation (8) is equivalent to

$$\sum_{i \in I} k_i \delta_i \equiv \gamma \mod p. \tag{10}$$

   By Lemma 2 and the fact that $\sum_{i \in I} k_i \delta_i$ 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 outputted by the signing oracle 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 5. Observe that 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.

Observe further that the proposed $\mathcal{E}$-MAC is 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 [5]. To see this, 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'. 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 Theorem 1, however, the adversary cannot expose the $\mathcal{E}$-MAC's key. Therefore, Theorem 2 holds whether or not the message is new, as long as the tag has not been attached to the message by the signing oracle.

5.2 Security of the *E&A* Composition

In [5], 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*.

Although the work of [5] shows that the $E\&A$ composition is generally insecure, the results do not apply to all variants of $E\&A$ constructions. For instance, the $E\&A$ composition does not provide indistinguishability under chosen plaintext attacks (IND-CPA) because there exist secure MACs that reveal information about the plaintext ([5] provides 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 $\mathcal{E}$-MAC does not reveal any information about the plaintext that is not revealed by the ciphertext.

Another result of [5] is that the generic $E\&A$ does not provide INT-CTXT. (Although the notion of INT-PTXT is the more natural security requirement [5] while the interest of the stronger INT-CTXT notion is more in the security implications shown in [5].) The reason why $E\&A$ 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 [5]. Obviously, when such an encryption algorithm is combined with the proposed $\mathcal{E}$-MAC to construct an $E\&A$ 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 an $E\&A$ system that does 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 2, modified ciphertexts can only be accepted with negligible probabilities. Indeed, secure $E\&A$ systems have been constructed in practice. A popular example of such constructions is SSH [51], which uses a variant of $E\&A$ the has been proven to be secure in [4].

So far, we have shown that $\mathcal{E}$-MACs can be used to replace standard MACs in the construction of $E\&A$ systems with two additional properties: they can have provable confidentiality and they can be more efficient (observe that the tag of the proposed $\mathcal{E}$-MAC is the output of the universal hash function; no post-processing was performed). 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 [27].

6 $\mathcal{E}$-MACs and Key Recovery

Recently, Handschuh and Preneel [27] 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 [27] 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 $k_1 m_1 + k_2 m_2 \equiv k_2 m_1 + k_1 m_2 \mod p$).

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

$$k_1 \equiv \beta k_2 \mod p, \tag{11}$$

where $\beta = (\alpha m_1 - m_2)(m_1 - m_2)^{-1}$ is exposed. 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 $k_1(\beta^{-1}m_2) + k_2(\beta m_1) = k_2 m_2 + k_1 m_1$ regardless of values of m_1 and m_2.

Handschuh and Preneel [27] 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 [27]. As per [27], 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 $E\&A$ system to overcome the key-recovery problem discovered in [27]. 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:

$$\tau = \sum_{i=1}^{B-1} k_i m_i + k_B r \mod p. \tag{12}$$

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

$$\tau = \sum_{i=1}^{B-1} (k_i \oplus r) m_i + k_B r \mod p. \tag{13}$$

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,

$$k_1' \equiv \beta k_2' \mod p, \tag{14}$$

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 (13) instead of equation (12), 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 6. 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.

7 Conclusion and Future Work

In this work, we studied the encrypt-and-authenticate generic composition of secure channels. We introduced $\mathcal{E}$-MACs, a new symmetric-key cryptographic primitive that can be used in the construction of $E\&A$ compositions. By taking advantage of the $E\&A$ structure, 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, two security objectives have been achieved. First, the random string is used to encrypt the authentication tag so that the secrecy of the plaintext is not

compromised by its tag. Second, the random string can be used to randomize the secret key of the used $\mathcal{E}$-MAC so that it will be secure against key-recovery attacks.

Since this is only the first work in this direction, bringing more research can only contribute positively towards the design of more efficient and more secure authentication. One specific direction that is yet to be investigated is the use of encryption algorithms that provide more than just semantic security. In particular, since most secure block ciphers are pseudorandom permutations, using block ciphers operated in different modes is a promising direction for more improvements in the design of $\mathcal{E}$-MACs.

References

1. M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. In *Advances in Cryptology–CRYPTO'96*, volume 96, pages 1–15. Lecture Notes in Computer Science, Springer, 1996.
2. M. Bellare, R. Guerin, and P. Rogaway. XOR MACs: New methods for message authentication using finite pseudorandom functions. In *Advances in Cryptology–CRYPTO'95*, volume 963, pages 15–28. Lecture Notes in Computer Science, Springer, 1995.
3. 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.
4. 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.
5. 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.
6. M. Bellare, P. Rogaway, and D. Wagner. The EAX mode of operation. In *Proceedings of Fast Software Encryption–FSE'04*, volume 3017, pages 389–407. Lecture Notes in Computer Science, Springer, 2004.
7. D. Bernstein. Floating-point arithmetic and message authentication. Unpublished manuscript, 2004. Available at `http://cr.yp.to/hash127.html`.
8. D. Bernstein. The Poly1305-AES message-authentication code. In *Proceedings of Fast Software Encryption–FSE'05*, volume 3557, pages 32–49. Lecture Notes in Computer Science, Springer, 2005.
9. J. Black, S. Halevi, H. Krawczyk, T. Krovetz, and P. Rogaway. UMAC: Fast and Secure Message Authentication. In *Advances in Cryptology–CRYPTO'99*, volume 1666, pages 216–233. Lecture Notes in Computer Science, Springer, 1999.
10. L. Blum, M. Blum, and M. Shub. A Simple Unpredictable Pseudo-random Number Generator. *SIAM Journal on Computing*, 15:364, 1986.
11. A. Bosselaers, R. Govaerts, and J. Vandewalle. Fast hashing on the Pentium. In *Advances in Cryptology-CRYPTO'96*, volume 1109, pages 298–312. Lecture Notes in Computer Science, Springer, 1996.
12. H. Canetti and H. Krawczyk. Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels. In *Advances in cryptology–EUROCRYPT'01*, volume 2045, pages 451–472. Lecture Notes in Computer Science, Springer, 2001.
13. J. Carter and M. Wegman. Universal classes of hash functions. In *Proceedings of the ninth annual ACM symposium on Theory of computing–STOC'77*, pages 106–112. ACM, 1977.
14. N. Doraswamy and D. Harkins. *IPSec: the new security standard for the Internet, intranets, and virtual private networks*. Prentice Hall, 2003.
15. M. Dworkin. Recommendation for block cipher modes of operation: The CMAC mode for authentication, 2005.
16. M. Dworkin. NIST Special Publication SP800-38D defining GCM and GMAC, 2007.
17. M. Etzel, S. Patel, and Z. Ramzan. Square hash: Fast message authentication via optimized universal hash functions. In *Advances in Cryptology–CRYPTO'99*, volume 1666, pages 234–251. Lecture Notes in Computer Science, Springer, 1999.
18. N. Ferguson, D. Whiting, B. Schneier, J. Kelsey, and T. Kohno. Helix: Fast encryption and authentication in a single cryptographic primitive. In *Proceedings of Fast Software Encryption–FSE'03*, volume 2887, pages 330–346. Lecture notes in computer science, Springer, 2003.
19. FIPS 113. Computer Data Authentication. *Federal Information Processing Standards Publication, 113*, 1985.
20. FIPS 198. The Keyed-Hash Message Authentication Code (HMAC). *Federal Information Processing Standards Publication*, 198, 2002.
21. A. Freier, P. Karlton, and P. Kocher. The SSL Protocol Version 3.0, 1996.
22. V. Gligor and P. Donescu. Integrity-Aware PCBC Encryption Schemes. *Security Protocols: 7th International Workshop, Cambridge, Uk, April 19-21, 1999: Proceedings*, 2000.

23. V. Gligor and P. Donescu. Fast Encryption and Authentication: XCBC Encryption and XECB Authentication Modes. In *Fast Software Encryption–FSE'02*, volume 2355, pages 1–20. Lecture Notes in Computer Science, Springer, 2002.
24. S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.
25. J. Gubner. *Probability and random processes for electrical and computer engineers*. Cambridge University Press, 2006.
26. S. Halevi and H. Krawczyk. MMH: Software message authentication in the Gbit/second rates. In *Proceedings of Fast Software Encryption – FSE'97*, volume 1267, pages 172–189. Lecture notes in computer science, Springer, 1997.
27. H. Handschuh and B. Preneel. Key-Recovery Attacks on Universal Hash Function Based MAC Algorithms. In *Proceedings of the 28th Annual conference on Cryptology: Advances in Cryptology–CRYPTO*, pages 144–161. Springer, 2008.
28. 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.
29. ISO/IEC 9797-1. Information technology – Security techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher, 1999.
30. ISO/IEC 9797-2. Information technology – Security techniques – Message Authentication Codes (MACs) – Part 2: Mechanisms using a dedicated hash-function, 2002.
31. T. Iwata and K. Kurosawa. omac: One-key cbc mac. In *Fast Software Encryption–FSE'03*, volume 2887, pages 129–153. Lecture notes in computer science, Springer, 2003.
32. C. Jutla. Encryption modes with almost free message integrity. *Journal of Cryptology*, 21(4):547–578, 2008.
33. J. Kaps, K. Yuksel, and B. Sunar. Energy scalable universal hashing. *IEEE Transactions on Computers*, 54(12):1484–1495, 2005.
34. J. Kohl and C. Neuman. The Kerberos Network Authentication Service (V5). Technical report, RFC 1510, September 1993, 1993.
35. T. Kohno, J. Viega, and D. Whiting. CWC: A high-performance conventional authenticated encryption mode. In *Fast Software Encryption–FSE'04*, volume 3017, pages 408–426. Lecture Notes in Computer Science, Springer, 2004.
36. H. Krawczyk. LFSR-based hashing and authentication. In *Advances in Cryptology–CRYPTO'94*, volume 839, pages 129–139. Lecture Notes in Computer Science, Springer, 1994.
37. H. Krawczyk. New hash functions for message authentication. *Advances in cryptology, CRYPTO'95*, 921:301–310, 1995.
38. H. Krawczyk. The order of encryption and authentication for protecting communications(or: How secure is SSL?). In *Advances in Cryptology–CRYPTO'01*, volume 2139, pages 310–331. Lecture Notes in Computer Science, Springer, 2001.
39. Y. Mansour, N. Nisan, and P. Tiwari. The computational complexity of universal hashing. In *Proceedings of the twenty-second annual ACM symposium on Theory of Computing–STOC'90*, pages 235–243. ACM, 1990.
40. D. McGrew and J. Viega. The security and performance of the Galois/Counter Mode (GCM) of operation. In *Progress in Cryptology-INDOCRYPT'04*, volume 3348, pages 343–355. Lecture notes in computer science, Springer, 2004.
41. C. Meyer and S. Matyas. *Cryptography: A New Dimension in Computer Data Security*. John Wiley & Sons, 1982.
42. B. Preneel and P. Van Oorschot. MDx-MAC and building fast MACs from hash functions. In *Advances in Cryptology-CRYPTO'95*, volume 963, pages 1–14. Lecture Notes in Computer Science, Springer, 1995.
43. B. Preneel and P. Van Oorschot. On the security of iterated message authentication codes. *IEEE Transactions on Information theory*, 45(1):188–199, 1999.
44. P. Rogaway. Bucket hashing and its application to fast message authentication. *Journal of Cryptology*, 12(2):91–115, 1999.
45. P. Rogaway, M. Bellare, and J. Black. OCB: A Block-Cipher Mode of Operation for Efficient Authenticated Encryption. *ACM Transactions on Information and System Security*, 6(3):365–403, 2003.
46. P. Rogaway and J. Black. PMAC: Proposal to NIST for a parallelizable message authentication code, 2001.
47. D. Stinson. Universal hashing and authentication codes. *Designs, Codes and Cryptography*, 4(3):369–380, 1994.
48. H. van Tilborg. *Encyclopedia of cryptography and security*. Springer, 2005.
49. M. Wegman and J. Carter. New classes and applications of hash functions. In *20th Annual Symposium on Foundations of Computer Science–FOCS'79*, pages 175–182. IEEE, 1979.
50. 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.
51. T. Ylonen and C. Lonvick. The Secure Shell (SSH) Transport Layer Protocol. Technical report, RFC 4253, 2006.

A Proof of Lemma 3

Proof. Throughout this proof, 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. Let the secret key $K = k_1||k_2||\cdots||k_B$ be fixed. Then, for any tag $\tau \in \mathbb{Z}_p$ computed according to equation (1), and any plaintext message M, the following holds:

$$\Pr(\boldsymbol{\tau} = \tau | \boldsymbol{M} = M) = \Pr\left(\boldsymbol{r} = (\tau - \sum_{i=1}^{B-1} k_i m_i)\ k_B^{-1}\right) = \frac{1}{p}, \tag{15}$$

where m_i denotes the i^{th} block of the message M. Equation (15) 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 guaranteed since k_B is not the zero element. Furthermore, as a direct consequence of the fact that $\mathbb{Z}_p$ is a field, 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:

$$\Pr(\boldsymbol{\tau} = \tau) = \frac{1}{p}. \tag{16}$$

Combining Bayes' theorem [25] with equations (15) and (16) yields:

$$\Pr(\boldsymbol{M} = M | \boldsymbol{\tau} = \tau) = \frac{\Pr(\boldsymbol{\tau} = \tau | \boldsymbol{M} = M) \Pr(\boldsymbol{M} = M)}{\Pr(\boldsymbol{\tau} = \tau)} = \Pr(\boldsymbol{M} = M). \tag{17}$$

Equation (17) 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. 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:

$$\Pr(\boldsymbol{\tau_1} = \tau_1, \cdots, \boldsymbol{\tau_\ell} = \tau_\ell) = \sum_{M_1,\cdots,M_\ell} \Pr(\boldsymbol{\tau_1} = \tau_1, \cdots, \boldsymbol{\tau_\ell} = \tau_\ell | \boldsymbol{M_1} = M_1, \cdots, \boldsymbol{M_\ell} = M_\ell) \Pr(\boldsymbol{M_1} = M_1, \cdots, \boldsymbol{M_\ell} = M_\ell)$$

$$= \sum_{M_1,\cdots,M_\ell} \Pr\left(\boldsymbol{r}_1 = (\tau_1 - \sum_{i=1}^{B-1} k_i m_{1_i})\ k_B^{-1}, \cdots, \boldsymbol{r}_\ell = (\tau_\ell - \sum_{i=1}^{B-1} k_i m_{\ell_i})\ k_B^{-1}\right) \Pr(\boldsymbol{M_1} = M_1, \cdots, \boldsymbol{M_\ell} = M_\ell) \tag{18}$$

$$= \sum_{M_1,\cdots,M_\ell} \Pr\left(\boldsymbol{r}_1 = (\tau_1 - \sum_{i=1}^{B-1} k_i m_{1_i})\ k_B^{-1}\right) \cdots \Pr\left(\boldsymbol{r}_\ell = (\tau_\ell - \sum_{i=1}^{B-1} k_i m_{\ell_i})\ k_B^{-1}\right) \Pr(\boldsymbol{M_1} = M_1, \cdots, \boldsymbol{M_\ell} = M_\ell) \tag{19}$$

$$= \sum_{M_1,\cdots,M_\ell} \frac{1}{p} \cdots \frac{1}{p} \Pr(\boldsymbol{M_1} = M_1, \cdots, \boldsymbol{M_\ell} = M_\ell) \tag{20}$$

$$= \Pr(\boldsymbol{\tau_1} = \tau_1) \cdots \Pr(\boldsymbol{\tau_\ell} = \tau_\ell), \tag{21}$$

where m_{j_i} denotes the i^{th} block of the j^{th} message M_j. Equation (19) holds due to the independence of the $\boldsymbol{r}_i$'s; equation (20) holds due to the uniform distribution of the $\boldsymbol{r}_i$'s; and equation (21) holds due to the uniform distribution of the $\boldsymbol{\tau}_i$'s. Therefore, authentication tags are mutually independent, and the lemma follows. □