Efficient Authentication for Mobile and Pervasive Computing

Basel Alomair and Radha Poovendran

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

Abstract. With today's technology, many applications rely on the existence of small devices that can exchange information and form communication networks. In a significant portion of such applications, the confidentiality and integrity of the communicated messages are of particular interest. In this work, we propose a novel technique for authenticating short encrypted messages that is more efficient than any message authentication code in the literature. By taking advantage of the fact that the message to be authenticated must also be encrypted, we propose a computationally secure authentication code that is as efficient as an unconditionally secure authentication, without the need for impractically long keys.

Keywords: Integrity, encryption, message authentication codes (MACs), efficiency.

1 Introduction and Related Work

Preserving the integrity of messages exchanged over public channels is one of the classic goals in cryptography and the literature is rich with message authentication code (MAC) algorithms that are designed solely for the purpose of preserving message integrity. Based on their security, MACs can be either unconditionally or computationally secure. Unconditionally secure MACs provide message integrity against forgers with unlimited computational power. On the other hand, computationally secure MACs are only secure when forgers have limited computational power.

A popular class of unconditionally secure authentication is based on universal hash-function families, pioneered by Wegman and Carter [50]. Since then, the study of unconditionally secure message authentication based on universal hash functions has been attracting research attention, both from the design and analysis standpoints (see, e.g., [47,2,9,1]). The basic concept allowing for unconditional security is that the authentication key can only be used to authenticate a limited number of exchanged messages. Since the management of one-time keys is considered impractical in many applications, computationally secure MACs have become the method of choice for most real-life applications. In computationally secure MACs, keys can be used to authenticate an arbitrary number

M. Soriano, S. Qing, and J. López (Eds.): ICICS 2010, LNCS 6476, pp. 186–202, 2010.

of messages. That is, after agreeing on a key, legitimate users can exchange an arbitrary number of authenticated messages with the same key. Depending on the main building block used to construct them, computationally secure MACs can be classified into three main categories: block cipher based, cryptographic hash function based, or universal hash-function family based.

CBC-MAC is one of the most known block cipher based MACs, specified in the Federal Information Processing Standards publication 113 [20] 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 [16], which was based on OMAC of [31].

HMAC is a popular example of the use of iterated cryptographic hash functions in the design of MACs [3], which was adopted as a standard [21]. Another cryptographic hash function based MAC is the MDx-MAC of Preneel and Oorschot [43]. HMAC and two variants of MDx-MAC are specified in the International Organization for Standardization ISO/IEC 9797-2 [30].

The use of universal hash-function families in the Wegman-Carter style is not restricted to the design of unconditionally secure authentication. Computationally secure MACs based on universal hashing can be constructed with two rounds of computations. In the first round, the message to be encrypted is compressed using a universal hash function. Then, in the second round, the compressed image is processed with a cryptographic function (typically a pseudorandom function[1]). Popular examples of computationally secure universal hashing based MACs include, but are not limited to, [8,26,17,10,7].

Indeed, universal hashing based MACs give better performance when compared to block cipher or cryptographic hashing based MACs. There are two main ideas behind the performance improvement of universal hashing based MACs. First, processing messages block by block using universal hash functions is faster than processing them block by block using block ciphers or cryptographic hash functions. Second, since the output of the universal hash function is much shorter than the entire message, processing the compressed image with a cryptographic function can be performed efficiently.

The main difference between unconditionally secure MACs based on universal hashing and computationally secure MACs based on universal hashing is the requirement to process the compressed image with a cryptographic primitive in the latter case. This round of computation is necessary to protect the secret key of the universal hash function. That is, since universal hash functions are not cryptographic functions, the observation of multiple message-image pairs can reveal the value of the hashing key. Since the hashing key is used repeatedly in computationally secure MACs, the exposure of the hashing key will lead to breaking the security of the MAC. This implies that unconditionally secure MACs based on universal hashing are more efficient than computationally secure ones. On the negative side, unconditionally secure universal hashing based

[1] Earlier designs used one-time pad encryption to process the compressed image. However, due to the difficulty to manage such on-time keys, recent designs resorted to computationally secure primitives (see, e.g., [10]).

MACs are considered impractical in most applications, due to the difficulty of managing one-time keys.

There are two important observations one can make about existing MAC algorithms. First, they are designed independently of any other operations required to be performed on the message to be authenticated. For instance, if the authenticated message must also be encrypted, existing MACs are not designed to utilize the functionalities that can be provided by the underlying encryption algorithm. Second, most existing MACs are designed for the general computer communication systems, independently of the properties that messages can possess. For example, one can find that most existing MACs are inefficient when the messages to be authenticated are short. (For instance, UMAC, the fastest reported message authentication code in the cryptographic literature [49], has undergone large algorithmic changes to increase its speed on short messages [36].)

Nowadays, however, there is an increasing demand for the deployment of networks consisting of a collection of small devices. In many practical applications, the main purpose of such devices is to communicate short messages. A sensor network, for example, can be deployed to monitor certain events and report some collected data. In many sensor network applications, reported data consist of short confidential measurements. Consider a sensor network deployed in a battlefield with the purpose of reporting the existence of moving targets or other temporal activities. In such applications, the confidentiality and integrity of reported events can be critically important.

In another application, consider the increasingly spreading deployment of radio frequency identification (RFID) systems. In such systems, RFID tags need to identify themselves to authorized RFID readers in an authenticated way that also preserves their privacy. In such scenarios, RFID tags usually encrypt their identity, which is typically a short string, to protect their privacy. Since the RFID reader must also authenticate the identity of the RFID tag, RFID tags must be equipped with a message authentication mechanism. Another application that is becoming increasingly important is the deployment of body sensor networks. In such applications, small sensors can be embedded in the patient's body to report some vital signs. Again, in some applications the confidentiality and integrity of such reported messages can be important.

There have been significant efforts devoted to the design of hardware efficient implementations that suite such small devices. For instance, hardware efficient implementations of block ciphers have been proposed in, e.g., [18,11]. Implementations of hardware efficient cryptographic hash functions have also been proposed in, e.g., [39,46]. However, there has been little or no effort in the design of special algorithms that can be used for the design of message authentication codes that can utilize other operations and the special properties of such networks. In this paper, we provide the first such work.

Contributions. We propose a new technique for authenticating short encrypted messages that is more efficient than existing approaches. We utilize the fact that the message to be authenticated is also encrypted to append a short secret key that can be used for message authentication. Since the keys used for different operations are independent, the authentication algorithm can benefit from the simplicity of unconditional secure authentication to allow for faster and more efficient authentication, without the difficulty to manage one-time keys.

Organization. The rest of the paper is organized as follows. In Section 2 we discuss some preliminaries. In Section 3 we describe the details of the proposed authentication technique assuming messages do not exceed a maximum length. In Section 4, we give a detailed security analysis of the proposed authentication scheme. In Section 5, we propose a modification to the original scheme that provides a stronger notion of integrity. In Section 6, we give an extension to the basic scheme that can handle arbitrary-length messages. In Section 7, we give a brief discussion of the performance of the proposed technique. In Section 8, we conclude the paper.

2 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 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'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.

Another security notion that will be used in this paper is related to the security of encryption algorithms. Informally, an encryption algorithm is said to be semantically secure (or, equivalently, provides indistinguishability under chosen plaintext attacks (IND-CPA) [25]) if an adversary who is given a ciphertext corresponding to one of two plaintext messages of her choice cannot determine the plaintext corresponding to the given ciphertext with an advantage significantly higher than 1/2.

The following lemma, a general result known in probability and group theory [45], will be used in the proofs of this paper.

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

3 Authenticating Short Encrypted Messages

In this section, we describe a basic scheme assuming that messages to be authenticated are no longer than a predefined length. This includes applications in which messages are of fixed length, such as RFID systems where tags need to authenticate their identifiers, sensor nodes reporting events that belong to certain domain or measurements within a certain range, etc. In Section 6, we will describe an extension to this scheme that can take messages of arbitrary lengths. First, we discuss some background in the area of authenticated encryption systems.

3.1 Background

The proposed system is an instance of what is known in the literature as the "generic composition" of authenticated encryption. Generic compositions are constructed by combining an encryption primitive (for message confidentiality) with a MAC primitive (for message integrity). Depending on the order of performing the encryption and authentication operations, generic compositions can be constructed in one of three main methods: encrypt-then-authenticate (EtA), authenticate-then-encrypt (AtE), or encrypt-and-authenticate (E&A). The security of different generic compositions have been extensively studied (see, e.g., [5,35,4]).

A fundamentally different approach for building authenticated encryption schemes was pioneered by Jutla, where he put forth the design of integrity aware encryption modes to build single-pass authenticated encryption systems [32]. For a message consisting of m blocks, the authenticated encryption of [32] requires a total of $m + 2$ block cipher evaluations. Following the work of Jutla, variety of single-pass authenticated encryption schemes have been proposed. Gligor and Donescu proposed the XECB-MAC [24]. Rogaway *et al.* [44] 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. Bellare *et al.* proposed the EAX mode of operation for solving the authenticated encryption problem with associated data [6]. Given a message M, a header H, and a nonce N, their authenticated encryption requires $2\lceil |M|/n \rceil + \lceil |H|/n \rceil + \lceil |N|/n \rceil$ block cipher calls, where n is the block length of the underlying block cipher. Kohno *et al.* [34] proposed CWC, a high-performance conventional authenticated encryption mode.

Note, however, that the generic composition can lead to faster authenticated encryption systems when a fast encryption algorithm (such as stream ciphers) is combined with a fast message authentication algorithm (such as universal hash function based MACs) [35]. Generic compositions have also design and analysis advantages due to their modularity and the fact that the encryption and authentication primitives can be designed, analyzed, and replaced independently of each other [35]. Indeed, popular authenticated encryption systems deployed in practice, such as SSH [53], SSL [23], IPsec [15], and TLS [14], use generic composition methods.

In the following section, we propose a novel method for authenticating messages encrypted with any secure encryption algorithm. The proposed method utilizes the existence of a secure encryption algorithm for the design of a highly efficient and highly secure authentication of short messages.

3.2 The Proposed System

Let $N - 1$ be an upper bound on the length, in bits, of exchanged messages. That is, messages to be authenticated can be no longer than $(N - 1)$-bit long. Choose p to be the smallest N-bit long prime integer. (If N is too small to provide the desired security level, p can be chosen large enough to satisfy the required security level.) Choose an integer k_s uniformly at random from the multiplicative group $\mathbb{Z}_p^*$; k_s is the secret key of the scheme. The prime integer, p, and the secret key, k_s, are distributed to legitimate users and will be used for message authentication. Note that the value of p need not be secret, only k_s is secret.

Let $\mathcal{E}$ be any semantically secure encryption algorithm. In fact, for our authentication scheme to be secure, we require a weaker notion than semantic security. Recall that semantic security implies that two encryptions of the same message should not be the same; that is, semantic security requires that the encryption algorithm must be probabilistic. Secure deterministic encryption algorithms are sufficient for the security of the proposed MAC. However, specially in RFID and sensor network applications, semantic security is usually a basic requirement (for example, for an RFID tag encrypting its identity, the encryption must be probabilistic to avoid illegal tracking).

Let m be a short messages that is to be transmitted to the intended receiver in a confidential manner (by encrypting it with $\mathcal{E}$). Instead of authenticating the message using a traditional MAC algorithm, consider the following procedure. On input a message m, a random nonce $k \in \mathbb{Z}_p$ is chosen. (We overload m to denote both the binary string representing the message, and the integer representation of the message as an element of $\mathbb{Z}_p$. The same applies to k_s and k. The distinction between the two representations will be omitted when it is clear from the context.)

Now, k is appended to the message and the resulting $m \parallel k$, where "$\|$" denotes the concatenation operation, goes to the encryption algorithm as an input. Then, the authentication tag of message m can be calculated as follows:

$$\tau \equiv mk_s + k \pmod{p}. \tag{1}$$

Remark 1. We emphasize that the nonce, k, is generated internally and is not part of the chosen message attack. In fact, k can be thought of as a replacement to the coin tosses that can be essential in many MAC algorithms. In such a case, the generation of k imposes no extra overhead on the authentication process. We also point out that, as opposed to one-time keys, k needs no special key management; it is delivered to the receiver as part of the encrypted ciphertext.

Since the generation of pseudorandom numbers can be considered expensive for computationally limited devices, there have been several attempts to design true random number generators that are suitable for RFID tags (see, e.g., [37,27,28]) and for low-cost sensor nodes (see, e.g., [42,12,22]). Thus, we assume the availability of such random number generators.

Now, the ciphertext $c = \mathcal{E}(m||k)$ and the authentication tag τ, computed according to equation (1), are transmitted to the intended receiver.

Upon receiving the ciphertext, the intended receiver decrypts it to extract m and k. Given τ, the receiver can check the validity of the message by performing the following integrity test:

$$\tau \overset{?}{\equiv} mk_s + k \pmod{p}. \tag{2}$$

If the integrity check of equation (2) is satisfied, the message is considered authentic. Otherwise, the integrity of the message is denied.

Note, however, that the authentication tag is a function of the confidential message. Therefore, the authentication tag must not reveal information about the plaintext since, otherwise, the confidentiality of the encryption algorithm is compromised. In the next section, we give formal security analysis of the proposed technique.

4 Security Analysis

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

4.1 Security of Authentication

As mentioned earlier, the authentication tag must satisfy two requirements: first, it must provide the required integrity and, second, it must not jeopardize the secrecy of the encrypted message. We start be stating an important lemma regarding the secrecy of k_s.

Lemma 2. *An adversary exposing information about the secret key, k_s, from authentication tags is able to break the semantic security of the encryption algorithm.*

Proof. Assume an adversary calling the signing oracle for q_s times and recording the sequence

$$\mathsf{Seq} = \{(m_1, \tau_1), \cdots, (m_{q_s}, \tau_{q_s})\} \tag{3}$$

of observed message-tag pairs. Recall that each authentication tag τ_i computed according to equation (1) requires the generation of a random nonce k. Recall further that k is generated internally and is not part of the chosen message attack. Now, if k is delivered to the receiver using a secure channel (e.g., out of band), then equation (1) is an instance of a perfectly secret (in Shannon's information theoretic sense) one-time pad cipher (encrypted with the one-time key k) and, hence, no information about k_s will be exposed. However, the k corresponding to each tag is delivered via the ciphertext. Therefore, the only way to expose secret information about k_s is to break the security of the encryption algorithm and infer information about the nonce k, and the lemma follows. □

We can now proceed with the main theorem formalizing the adversary's chances of successful forgery against the proposed authentication scheme.

Theorem 1. *An adversary making a (q_s, q_v)-attack on the proposed scheme can forge a valid tag with probability no more than $1/(p-1)$, provided the adversary's inability to break the encryption algorithm.*

Proof. Assume an adversary calling the signing oracle for q_s times and recording the sequence

$$\mathsf{Seq} = \{(m_1, \tau_1), \cdots, (m_{q_s}, \tau_{q_s})\} \tag{4}$$

of message-tag pairs. We aim to bound the probability that an (m, τ) pair of the adversary's choice will be accepted as valid, where $(m, \tau) \neq (m_i, \tau_i)$ for any $i \in \{1, \cdots, q_s\}$, since otherwise the adversary does not win by definition.

Let $m \equiv m_i + \epsilon \pmod p$ for any $i \in \{1, \cdots, q_s\}$, where ϵ can be any function of the recorded values. Similarly, let $k \equiv k_i + \delta \pmod p$, where δ is any function of the recorded values (k here represents the value extracted by the legitimate receiver after decrypting the ciphertext). Assume further that the adversary knows the values of ϵ and δ. Then,

$$\begin{align}
\tau &\equiv mk_s + k \pmod p \tag{5}\\
&\equiv (m_i + \epsilon)k_s + (k_i + \delta) \pmod p \tag{6}\\
&\equiv \tau_i + \epsilon k_s + \delta \pmod p. \tag{7}
\end{align}$$

Therefore, for (m, τ) to be validated, τ must be congruent to $\tau_i + \epsilon k_s + \delta$ modulo p. Now, by Lemma 2, k_s will remain secret as long as the adversary does not break the encryption algorithm. Hence, by Lemma 1, the value of ϵk_s is an unknown value uniformly distributed over the multiplicative group $\mathbb{Z}_p^*$ (observe that ϵ cannot be the zero element since, otherwise, m will be equal to m_i). Therefore, the adversary's probability of successful forgery is $1/(p-1)$, and the theorem follows. □

Remark 2. Observe that, if both k_s and k are used only once (i.e., one-time keys), the authentication tag of equation (1) is a well-studied example of a strongly universal hash family (see [48] for a definition of strongly universal hash families and detailed discussion showing that equation (1) is indeed strongly universal hash family). The only difference is that we restrict k_s to belong to the multiplicative group modulo p, whereas it can be equal to zero in unconditionally secure authentication. This is because, in unconditionally secure authentication, the keys can only be used once. In our technique, since k_s can be used to authenticate an arbitrary number of messages, it cannot be chosen to be zero. Otherwise, mk_s will always be zero and the system will not work. The novelty of our approach is to utilize the encryption primitive to reach the simplicity of unconditionally secure authentication, without the need for impractically long keys.

With the probability of successful forgery given in Theorem 1, we show next that the second requirement on the authentication tag is also satisfied. Namely, that the authentication tag does not reveal any information about the plaintext that is not revealed by the ciphertext.

Theorem 2. *An adversary exposing information about the encrypted message from the authentication tag is also able to break the semantic security of the encryption algorithm.*

The proof of Theorem 2 is similar to the proof of Lemma 2 and, thus, is omitted.

Theorems 1 and 2 imply that breaking the security of the authentication tag is reduced to breaking the semantic security of the underlying encryption algorithm. That is, the proposed method is provably secure, given the semantic security of the underlying encryption algorithm.

4.2 Security of the Authenticated Encryption Composition

In [5], two notions of integrity are defined for authenticated encryption systems: integrity of plaintext (INT-PTXT) and integrity of ciphertext (INT-CTXT). Combined with encryption algorithms that provide indistinguishability against chosen plaintext attacks (IND-CPA),[2] the security of different methods for constructing generic compositions is analyzed. Observe that our construction is an instance of the encrypt-and-authenticate (E&A) generic composition since the plaintext message goes to the encryption algorithm as an input, and the same plaintext message goes to the authentication algorithm as an input. Figure 1 illustrates the differences between the three methods for generically composing an authenticated encryption system.

[2] Recall that IND-CPA is equivalent to semantic security, as shown in [25].

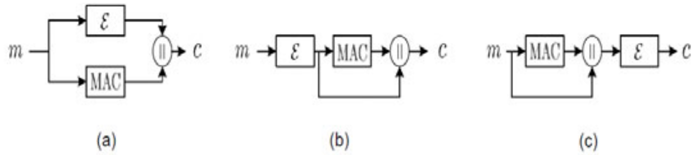

Fig. 1. A schematic of the three generic compositions; (a) Encrypt-and-Authenticate (E&A), (b) Encrypt-then-Authenticate (EtA), and (c) Authenticate-then-Encrypt (AtE)

It was shown in [5] that E&A generic compositions do not provide IND-CPA. This is mainly because there exist secure MAC algorithms that leak information about the authenticated message (a detailed example of such a MAC can be found in [5]). Obviously, if such a MAC is used to compose an E&A system, then the authenticated encryption does not provide IND-CPA. By Theorem 2, however, the proposed authentication code does not reveal any information about the plaintext message unless the adversary can break the security of the coupled encryption algorithm. Since the encryption algorithm is semantically secure, the resulting composition provides IND-CPA.

Another result of [5] is that E&A compositions do not provide INT-CTXT. However, the authors also point out that the notion of INT-PTXT is the more natural requirement, while the main purpose of introducing the stronger notion of INT-CTXT is for the security relations derived in [5]. The reason why E&A compositions do not generally provide INT-CTXT is because there exist secure encryption algorithms with the property that the ciphertext can be modified without changing its decryption. Obviously, if such an encryption algorithm is combined with our MAC to compose an E&A composition, only INT-PTXT is achieved (since the tag in our scheme is a function of plaintext). A sufficient condition, however, for the proposed composition to provide INT-CTXT is to use a one-to-one encryption algorithm (most practical encryption algorithm are permutations, i.e., one-to-one [33]). To see this, observe that, by the one-to-one property, any modification of the ciphertext will correspond to changing its corresponding plaintext. By Theorem 1, a modified plaintext will go undetected with a negligible probability.

5 From Weak to Strong Unforgeability

As per [5], there are two notions of unforgeability in authentication codes. Namely, a MAC algorithm can be weakly unforgeable under chosen message attacks (WUF-CMA), or strongly unforgeable under chosen message attacks (SUF-CMA). A MAC algorithm is said to be SUF-CMA if, after launching chosen message attacks, it is infeasible to forge a message-tag pair that will be accepted as valid regardless of whether the message is “new” or not, as long as the tag has not been previously attached to the message by an authorized user.

If it is only hard to forge valid tags for "new" messages, the MAC algorithm is said to be WUF-CMA.

The authentication code, as described in Section 3, is only WUF-CMA. To see this, let $\mathcal{E}$ works as follows. On input a message m, generate a random string r, compute $PRF_x(r)$, where PRF_x is a pseudorandom function determined by a secret key x, and transmit $c = (r, PRF_x(r) \oplus m)$ as the ciphertext. Then, $\mathcal{E}$ is a semantically secure encryption. Applied to our construction, on input a message m, the ciphertext will be $c = \big(r, PRF_x(r) \oplus (m||k)\big)$ and the corresponding tag will be $\tau \equiv mk_s + k \pmod{p}$. Now, let s be a string of length equal to the concatenation of m and k. Then, $c' = \big(r, PRF_x(r) \oplus (m||k) \oplus s\big) = \big(r, PRF_x(r) \oplus (m||k \oplus s)\big)$. Let s be a string of all zeros except for the least significant bit, which is set to one. Then, either $\tau_1 \equiv mk_s + k + 1 \pmod{p}$ or $\tau_2 \equiv mk_s + k - 1 \pmod{p}$ will be a valid tag for m, when c' is transmitted as the ciphertext. That is, the same message can be authenticated using different tags with high probabilities.

While WUF-CMA can be suitable for some applications, it can also be impractical for other applications. Consider RFID systems, for instance. If the message to be authenticated is the tag's fixed identity, then WUF-CMA allows the authentication of the same identity by malicious users. In this section, we will modify the original scheme described in Section 3 to make it SUF-CMA, without incurring extra overhead.

As can be observed from the above example, the forgery is successful if the adversary can modify the value of k and predict its effect on the authentication tag τ. To rectify this problem, not only the message but also the nonce k must be authenticated. Obviously, this can be done with the use of another secret key k'_s and computing the tag as $\tau \equiv mk_s + kk'_s \pmod{p}$. This, however, requires twice the amount of shared key material and an extra multiplication operation. A similar, yet more efficient, way of achieving the same goal can be done as follows

$$\tau \equiv (m + k)k_s \pmod{p}. \tag{8}$$

The only difference between this case and the original scheme of Section 3 is that k is not allowed to be equal to $-m$ modulo p; otherwise, the authentication tag will be zero.

So, the description of the modified system is as follows. Assume the users have agreed on a security parameter N, exchanged an N-bit prime integer p, and a secret key $k_s \in \mathbb{Z}_p^*$. On input a message $m \in \mathbb{Z}_p$, a random nonce $k \in \mathbb{Z}_p$ is chosen so that $m + k \not\equiv 0 \pmod{p}$. The transmitter encrypts the concatenation of m and k, computes the authentication tag according to equation (8), and transmits the ciphertext $c = \mathcal{E}(m||k)$ along with the authentication tag τ to the intended receiver. Decryption and authentication are performed accordingly.

The proof that this modified system achieves weak unforgeability under chosen message attacks and the proof that the tag does not reveal information about the plaintext are the same as the proofs of Theorem 1 and Theorem 2, respectively. Below we show that the modified system described in this section is indeed strongly unforgeable under chosen message attacks.

Theorem 3. *The proposed scheme is strongly unforgeable under chosen message attacks (SUF-CMA), provided the adversary's inability to break the encryption algorithm.*

Proof. Let (m, τ) be a valid message-tag pair recorded by the adversary. Assume the adversary is trying to authenticate the same message, m, with a different tag τ'. This implies that the nonce k must be different (otherwise, the tag will be the same). Now, assume the adversary can modify the ciphertext and predict the effect on the nonce k; that is, k becomes $k + \delta$ for some nonzero $\delta \in \mathbb{Z}_p^*$ of the adversary's choice. Then,

$$\tau' \equiv (m + k + \delta)k_s \equiv \tau + \delta k_s \pmod{p}. \tag{9}$$

Therefore, for (m, τ') to be accepted as valid, the adversary must predict the correct value of δk_s. Since, by Lemma 2, k_s will remain secret and, by Lemma 1, the value of δk_s is uniformly distributed over $\mathbb{Z}_p^*$, the probability of authenticating the same message with a different tag is $1/(p - 1)$, and the theorem follows. □

Remark 3. We emphasize that the adversary cannot query the signing oracle twice to get τ and τ' according to equation (9), leading to solving for k_s and breaking the system. Recall that, on input a message m, the oracle draws a random nonce k that the adversary does not control nor observe. The above proof deals with an adversary calling the signing oracle and interacting with the intended receiver, not an adversary calling the signing oracle twice.

6 Authenticating Arbitrary-Length Messages

In Sections 3 and 5, we described how to authenticate messages that are shorter than a per-specified maximum length. Recall that the authentication tag is computed as $\tau \equiv (m + k)k_s \pmod{p}$. Consequently, any message that is different than m with multiples of p, i.e., $m_\ell = m + \ell p$ for any integer ℓ, will have the same authentication tag. That is why it was critical for the security of authentication to restrict messages to be less than p. In this section, we show how to authenticate messages when their maximum length is not known a priori.

Given a desired level of integrity, a security parameter N is chosen and a secret N-bit long key k_s is given to authorized parties. For every message to be authenticated, the transmitter selects an N-bit prime integer p, that is not equal to k_s, and generates a fresh random nonce $k \in \mathbb{Z}_p$ so that $k + m \not\equiv 0 \pmod{p}$. The plaintext is a concatenation of the message m, the random nonce k, and the prime integer p. That is, the transmitted ciphertext is $c = \mathcal{E}(m||k||p)$, where $\mathcal{E}$ is the underlying semantically secure encryption algorithm and "$||$" denotes the concatenation operation. The sender then can authenticate the message m as follows:

$$\tau \equiv (m + k)k_s \pmod{p}. \tag{10}$$

Decryption and authentication are done the natural way .

Note that the main difference between this approach and the one described in Section 5 is that the prime modulus p in this case varies in different operations and is not public. Just like the nonce k, it is delivered via the ciphertext.

Now, assume an adversary, after calling the signing oracle on m and receiving its tag τ, is attempting to forge a valid tag for a message $m' \neq m$. Write $m' = m + \epsilon$, for some nonzero ϵ. There are two possible scenarios here: either ϵ is a multiple of p or not. Let ϵ be an integer that is not a multiple of p; i.e., $\epsilon \not\equiv 0 \pmod{p}$. Then, the valid authentication tag for m' will be

$$\tau' \equiv (m' + k)k_s \equiv \tau + \epsilon k_s \pmod{p}. \tag{11}$$

Since, by Lemma 1, ϵk_s is uniformly distributed over $\mathbb{Z}_p^*$ and, by Lemma 2, k_s will remain secret, the probability of predicting the correct authentication tag corresponding to m' is bounded by $1/(p-1)$. The proof that the MAC proposed here is also strongly unforgeable under chosen message attacks is the same as the proof of Theorem 3.

On the other hand, if the adversary can guess the prime integer p, forgery can be successful by replacing m with $m + \ell p$ for any integer ℓ. However, even if the adversary is assumed to know the length of the prime integer, say N-bits, the prime number theorem shows that the number of primes less than 2^N can be approximated by [13]:

$$\pi(2^N) \approx \frac{2^N}{N \ln(2)}, \tag{12}$$

where $\pi(x)$ is the prime-counting function. That is, the probability of guessing the used prime integer, without breaking the semantic security of the underlying encryption algorithm, is an exponentially decreasing function in N.

7 Performance

Compared to standard block cipher based and cryptographic hash function based, the proposed technique involves a single addition and a single modular multiplication. Even for long messages, dividing the message into blocks and performing modular multiplications is faster than block cipher or cryptographic hash operations (this is actually how universal hashing is performed). Since we target application in which the messages to be authenticated are short strings, multiplication can be performed even faster.

Compared to universal hashing based MACs, our technique can be considered as a single block of a universal hash function with one important advantage. Namely, unlike standard universal hashing based MACs, there is no need to process the compressed image with a cryptographic primitive in our design. That is, we utilized the computations performed by the encryption algorithm to eliminate the post-processing round of computation in universal hashing base MACs.

Another advantage of the proposed method is hardware efficiency. The hardware required to perform modular multiplication is less than the hardware required to perform sophisticated cryptographic operations. This advantage is particularly important for low-cost devices.

Compared to single-pass authenticated encryption algorithms, when combined with a stream cipher, our construction will be much faster (recall that single-pass authenticated encryption methods are block cipher based[3]). Furthermore, our construction is an instance of the encrypt-and-authenticate (E&A) generic composition. That is, the encryption and authentication operations can be performed in parallel. If the underlying encryption algorithm is a block cipher based, the time to complete the entire operation will be the time it takes for encryption only. Even with the added time to encrypt the nonce, which depending on the length of k and the size of the block cipher might not require any additional block cipher calls, single-pass authenticated encryption methods typically require at least two additional block cipher calls.

8 Conclusion

In this work, a new technique for authenticating short encrypted messages is proposed. The fact that the message to be authenticated must also be encrypted is used to deliver an authentication key to the intended receiver via the ciphertext. This allowed the design of an authentication code that benefits from the simplicity of unconditionally secure authentication without the need to manage one-time keys. In particular, it has been demonstrated in this paper that authentication tags can be computed with one addition and a one modular multiplication. Given that messages are relatively short, addition and modular multiplication can be performed faster than existing computationally secure MACs in the literature of cryptography.

References

1. Aloamir, 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) (2010)
2. Atici, M., Stinson, D.: Universal Hashing and Multiple Authentication. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 16–30. Springer, Heidelberg (1996)
3. Bellare, M., Canetti, R., Krawczyk, H.: Keying Hash Functions for Message Authentication. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 1–15. Springer, Heidelberg (1996)
4. 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)
5. Bellare, M., Namprempre, C.: Authenticated Encryption: Relations Among Notions and Analysis of the Generic Composition Paradigm. Journal of Cryptology 21(4), 469–491 (2008)

[3] Although stream cipher based authenticated encryption primitives have appeared in [19,51], such proposals have been analyzed and shown to be vulnerable to attacks [38,41,40,52].

6. Bellare, M., Rogaway, P., Wagner, D.: The EAX mode of operation. In: Roy, B., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 389–407. Springer, Heidelberg (2004)
7. Bernstein, D.: Floating-point arithmetic and message authentication (2004), `http://cr.yp.to/hash127.html`
8. Bernstein, D.: The Poly1305-AES message-authentication code. In: Gilbert, H., Handschuh, H. (eds.) FSE 2005. LNCS, vol. 3557, pp. 32–49. Springer, Heidelberg (2005)
9. Bierbrauer, J.: Universal hashing and geometric codes. Designs, Codes and Cryptography 11(3), 207–221 (1997)
10. Black, J., Halevi, S., Krawczyk, H., Krovetz, T., Rogaway, P.: UMAC: Fast and Secure Message Authentication. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 216–233. Springer, Heidelberg (1999)
11. Bogdanov, A., Knudsen, L., Leander, G., Paar, C., Poschmann, A., Robshaw, M., Seurin, Y., Vikkelsoe, C.: PRESENT: An Ultra-Lightweight Block Cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (2007)
12. Callegari, S., Rovatti, R., Setti, G.: Embeddable ADC-based true random number generator for cryptographic applications exploiting nonlinear signal processing and chaos. IEEE Transactions on Signal Processing 53(2 Part 2), 793–805 (2005)
13. Cormen, T., Leiserson, C., Rivest, R.: Introduction to Algorithms. McGraw-Hill, New York (1999)
14. Dierks, T., Rescorla, E.: The transport layer security (TLS) protocol version 1.2. Technical report, RFC 5246 (2008)
15. Doraswamy, N., Harkins, D.: IPSec: the new security standard for the Internet, intranets, and virtual private networks. Prentice Hall, Englewood Cliffs (2003)
16. Dworkin, M.: Recommendation for block cipher modes of operation: The CMAC mode for authentication (2005)
17. Etzel, M., Patel, S., Ramzan, Z.: Square hash: Fast message authentication via optimized universal hash functions. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 234–251. Springer, Heidelberg (1999)
18. Feldhofer, M., Dominikus, S., Wolkerstorfer, J.: Strong Authentication for RFID Systems using the AES Algorithm. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 357–370. Springer, Heidelberg (2004)
19. Ferguson, N., Whiting, D., Schneier, B., Kelsey, J., Kohno, T.: Helix: Fast encryption and authentication in a single cryptographic primitive. In: Johansson, T. (ed.) FSE 2003. LNCS, vol. 2887, pp. 330–346. Springer, Heidelberg (2003)
20. FIPS 113. Computer Data Authentication. Federal Information Processing Standards Publication, 113 (1985)
21. FIPS 198. The Keyed-Hash Message Authentication Code (HMAC). Federal Information Processing Standards Publication, 198 (2002)
22. Francillon, A., Castelluccia, C., Inria, P.: TinyRNG: A cryptographic random number generator for wireless sensors network nodes. In: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks–WiOpt 2007, pp. 1–7. Citeseer (2007)
23. Freier, A., Karlton, P., Kocher, P.: The SSL Protocol Version 3.0 (1996)
24. Gligor, V., Donescu, P.: Fast Encryption and Authentication: XCBC Encryption and XECB Authentication Modes. In: Matsui, M. (ed.) FSE 2001. LNCS, vol. 2355, pp. 1–20. Springer, Heidelberg (2002)
25. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences 28(2), 270–299 (1984)

26. Halevi, S., Krawczyk, H.: MMH: Software message authentication in the Gbit/second rates. In: Biham, E. (ed.) FSE 1997. LNCS, vol. 1267, pp. 172–189. Springer, Heidelberg (1997)
27. Holcom, D., Burleson, W., Fu, K.: Initial SRAM state as a Fingerprint and Source of True Random Numbers for RFID Tags. In: Workshop on RFID Security–RFIDSec 2007 (2007)
28. Holcomb, D., Burleson, W., Fu, K.: Power-up SRAM State as an Identifying Fingerprint and Source of True Random Numbers. IEEE Transactions on Computers 58(9) (2009)
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. Iwata, T., Kurosawa, K.: Omac: One-key cbc mac. In: Johansson, T. (ed.) FSE 2003. LNCS, vol. 2887, pp. 129–153. Springer, Heidelberg (2003)
32. Jutla, C.: Encryption modes with almost free message integrity. Journal of Cryptology 21(4), 547–578 (2008)
33. Katz, J., Lindell, Y.: Introduction to modern cryptography. Chapman & Hall/CRC (2008)
34. Kohno, T., Viega, J., Whiting, D.: CWC: A high-performance conventional authenticated encryption mode. In: Roy, B., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 408–426. Springer, Heidelberg (2004)
35. Krawczyk, H.: The order of encryption and authentication for protecting communications (or: How secure is SSL?). In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 310–331. Springer, Heidelberg (2001)
36. Krovetz, T. (2006), http://fastcrypto.org/umac/
37. Liu, Z., Peng, D.: True Random Number Generator in RFID Systems Against Traceability. In: IEEE Consumer Communications and Networking Conference–CCNS 2006, vol. 1, pp. 620–624. IEEE, Los Alamitos (2006)
38. Muller, F.: Differential attacks against the Helix stream cipher. In: Roy, B., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 94–108. Springer, Heidelberg (2004)
39. O'Neill (McLoone), M.: Low-Cost SHA-1 Hash Function Architecture for RFID Tags. In: Workshop on RFID Security–RFIDSec 2008 (2008)
40. Paul, S., Preneel, B.: Near Optimal Algorithms for Solving Differential Equations of Addition with Batch Queries. In: Maitra, S., Veni Madhavan, C.E., Venkatesan, R. (eds.) INDOCRYPT 2005. LNCS, vol. 3797, pp. 90–103. Springer, Heidelberg (2005)
41. Paul, S., Preneel, B.: Solving systems of differential equations of addition. In: Boyd, C., González Nieto, J.M. (eds.) ACISP 2005. LNCS, vol. 3574, pp. 75–88. Springer, Heidelberg (2005)
42. Petrie, C., Connelly, J.: A noise-based IC random number generator for applications in cryptography. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications 47(5), 615–621 (2000)
43. Preneel, B., Van Oorschot, P.: MDx-MAC and building fast MACs from hash functions. In: Coppersmith, D. (ed.) CRYPTO 1995. LNCS, vol. 963, pp. 1–14. Springer, Heidelberg (1995)
44. Rogaway, P., Bellare, M., Black, J.: OCB: A Block-Cipher Mode of Operation for Efficient Authenticated Encryption. ACM Transactions on Information and System Security 6(3), 365–403 (2003)

45. Schwarz, S.: The role of semigroups in the elementary theory of numbers. Math. Slovaca 31(4), 369–395 (1981)
46. Shamir, A.: SQUASH–A New MAC with Provable Security Properties for Highly Constrained Devices Such as RFID Tags. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 144–157. Springer, Heidelberg (2008)
47. Stinson, D.: Universal hashing and authentication codes. Designs, Codes and Cryptography 4(3), 369–380 (1994)
48. Stinson, D.: Cryptography: Theory and Practice. CRC Press, Boca Raton (2006)
49. van Tilborg, H.: Encyclopedia of Cryptography and Security. Springer, Heidelberg (2005)
50. 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)
51. Whiting, D., Schneier, B., Lucks, S., Muller, F.: Phelix-fast encryption and authentication in a single cryptographic primitive, eSTREAM. ECRYPT Stream Cipher Project, Report 2005/020 (2005), http://www.ecrypt.eu.org/stream
52. Wu, H., Preneel, B.: Differential-linear attacks against the stream cipher Phelix. In: Biryukov, A. (ed.) FSE 2007. LNCS, vol. 4593, pp. 87–100. Springer, Heidelberg (2007)
53. Ylonen, T., Lonvick, C.: The Secure Shell (SSH) Transport Layer Protocol. Technical report, RFC 4253 (2006)