Efficient Authentication for Mobile and Pervasive Computing

Basel Alomair · **Radha Poovendran**

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 are directed to meet the requirements of mobile and pervasive applications. By taking advantage of the fact that the message to be authenticated must also be encrypted, we propose provably secure authentication codes that are more efficient than any message authentication code in the literature. The key idea behind the proposed technique is to append a short random string to the plaintext message before encryption to facilitate a more efficient authentication.

Keywords Authentication · unconditional security · computational security · universal hash-function families · pervasive computing

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 for the sole 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 Carter and Wegman [27, 87, 28, 88]. 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., [20, 8, 43, 81, 21, 3, 6]). 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 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 [35] and the International Organization for Standardization ISO/IEC 9797-1 [47]. CMAC, a modified version of CBC-MAC, is presented in the NIST special publication 800-38B [31], which was based on the OMAC of [49]. Other block cipher based MACs include, but are not limited to, XOR-MAC [13] and PMAC [77]. The

A preliminary version of this paper appeared in the 12th International Conference on Information and Communications Security–ICICS'2010 [5]

Basel Alomair
Network Security Lab (NSL)
University of Washington, Seattle, Washington
E-mail: alomair@uw.edu

Radha Poovendran
Network Security Lab (NSL)
University of Washington, Seattle, Washington
E-mail: rp3@uw.edu

security of different MACs has been exhaustively studied (see, e.g., [14,74,75]).

The use of one-way cryptographic hash functions for message authentication was introduced by Tsudik in [85]. A popular example of the use of iterated cryptographic hash functions in the design of message authentication codes is HMAC, which was proposed by Bellare et al. in [12]. HMAC was later adopted as a standard [36]. Another cryptographic hash function based MAC is the MDx-MAC proposed by Preneel and Oorschot [73]. HMAC and two variants of MDx-MAC are specified in the International Organization for Standardization ISO/IEC 9797-2 [48]. 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 [25].

The use of universal hash-function families in the Carter-Wegman style is not restricted to the design of unconditionally secure authentication. Computationally secure MACs based on universal hash functions can be constructed with two rounds of computations. In the first round, the message to be authenticated 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, [19,62,42,32,22,52,18].

Indeed, universal hashing based MACs give better performance when compared to block cipher or cryptographic hashing based MACs. There are two main factors leading to the performance advantage 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 original message itself, processing the compressed image with a cryptographic function can be performed efficiently.

One of the main differences 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 class of MACs. 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. Thus, processing the compressed image with a cryptographic primitive is necessary for the security of this class of MACs. 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 MACs are considered impractical in most modern applications, due to the difficulty of managing one-time keys.

There are two important observations to 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 [84], has undergone large algorithmic changes to increase its speed on short messages [57].)

Nowadays, however, there is an increasing demand for the deployment of networks consisting of a collection of small devices. In many practical applications, the main purpose of such devices is to communicate short messages. A sensor network, for example, can be deployed to monitor certain events and report some collected data. In many sensor network applications, reported data consist of short confidential measurements. Consider, for instance, a sensor network deployed in a battlefield with the purpose of reporting the existence of moving targets or other temporal activities. In such applications, the confidentiality and integrity of reported events are of critical importance [1,70,69].

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 [78,50,68,9,11,10,7,4,2].

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 [92,86,83].

There have been significant efforts devoted to the design of hardware efficient implementations that suite such small devices. For instance, hardware efficient implementa-

[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., [22])

tions of block ciphers have been proposed in, e.g., [33,58, 46,72,23,60]. Implementations of hardware efficient cryptographic hash functions have also been proposed in, e.g., [65,80,24,54]. 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, with any secure encryption algorithm, to append a short random string to be used in the authentication process. Since the random strings used for different operations are independent, the authentication algorithm can benefit from the simplicity of unconditional secure authentication to allow for faster and more efficient authentication, without the difficulty to manage one-time keys.

ORGANIZATION. The rest of the paper is organized as follows. In Section 2 we list our notations and discuss some preliminaries. In Section 3 we describe the proposed authentication technique assuming messages do not exceed a maximum length, discuss its performance advantages over existing techniques, and prove its security. In Section 4, we propose a modification to the scheme of Section 3 that provides a stronger notion of integrity. In Section 5, we conclude the paper.

2 Notations and Preliminaries

2.1 Notations

- We use $\mathbb{Z}_p$ as the usual notation for the finite integer ring with the addition and multiplication operations performed modulo p.
- We use $\mathbb{Z}_p^*$ as the usual notation for the multiplicative group modulo p; i.e., $\mathbb{Z}_p^*$ contains the integers that are relatively prime to p.
- For two strings a and b of the same length, $(a \oplus b)$ denotes the bitwise exclusive-or (XOR) operation.
- For any two strings a and b, $(a||b)$ denotes the concatenation operation.
- For a nonempty set $\mathcal{S}$, the notation $s \overset{\$}{\leftarrow} \mathcal{S}$ denotes the operation of selecting an element from the set $\mathcal{S}$ uniformly at random and assigning it to s.

2.2 Negligible Functions

Another term that will be used in the reminder of the paper is the definition of negligible functions. A function $\mathsf{negl} : \mathbb{N} \rightarrow \mathbb{R}$ is said to be negligible if for any nonzero polynomial poly, there exists N_0 such that for all $N > N_0$, $|\mathsf{negl}(N)| < 1/|\mathsf{poly}(N)|$. That is, the function is said to be negligible if it converges to zero faster than the reciprocal of any polynomial function [40].

2.3 Indistinguishability Under Chosen Plaintext Attacks

An important security notion for encryption algorithms that will be used in this paper is indistinguishability under chosen plaintext attacks (IND-CPA). Let $\mathcal{A}$ be an adversary who is given access oracle to an encryption algorithm, $\mathcal{E}$, and can ask the oracle to encrypt a polynomial number of messages to get their corresponding ciphertexts. The encryption algorithm is said to be IND-CPA secure if the adversary, after calling the encryption oracle a polynomial number of times, is given a ciphertext corresponding to one of two plaintext messages of her choice cannot determine the plaintext corresponding to the given ciphertext with an advantage significantly higher than $1/2$. Formally stated, let $\mathsf{Adv}_{\mathcal{E}}^{\text{ind-cpa}}(\mathcal{A})$ be the adversary's advantage of determining the plaintext corresponding to the given ciphertext. Then, $\mathcal{E}$ is said to be IND-CPA secure if

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

where N is a security parameter, typically the length of the secret key.

Note that IND-CPA security implies that the encryption algorithm must be probabilistic [41]. That is, encrypting the same message twice yields different ciphertexts. To see that, let the adversary call the encryption oracle on a message m_1 and receiving its ciphertext c_1. The adversary now chooses two messages, m_1 and m_2, ask the encryption oracle to encrypt them and receives the ciphertext corresponding to one of them. If the encryption is deterministic, the adversary can determine, with high confidence, to which plaintext the ciphertext corresponds by comparing it to c_1.

2.4 A Useful Result

The following lemma, a general result known in probability and group theory [79], 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 the proposed authentication scheme that can be used with any IND-CPA secure encryption algorithm. An important assumption we make is that messages to be authenticated are no longer than a predefined length. This includes applications in which messages are of fixed length that is known a priori, such as RFID systems in which tags need to authenticate their identifiers, sensor nodes reporting events that belong to certain domain or measurements within a certain range, etc. 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). Based 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., [16,56,15,61]).

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 [51]. For a message consisting of m blocks, the authenticated encryption of [51] 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 [39]. Rogaway et al. [76] 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 [17]. 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. [55] 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) [56]. 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 [56]. Indeed, popular authenticated encryption systems deployed in practice, such as SSH [91], SSL [38], IPsec [30], and TLS [29], use generic composition methods.

In the following section, we propose a novel method for authenticating messages encrypted with any IND-CPA secure encryption algorithm. The proposed method utilizes the existence of an IND-CPA 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 an N-bit long prime integer. (If N is too small to provide the desired security level, p can be chosen large enough to satisfy the required security level.) Choose an integer k_s uniformly at random from the multiplicative group $\mathbb{Z}_p^*$; k_s is the secret key of the scheme. The prime integer, p, and the secret key, k_s, are distributed to legitimate users and will be used for message authentication. Note that the value of p need not be secret, only k_s is secret.

Let $\mathcal{E}$ be any IND-CPA secure encryption algorithm. Let m be a short messages ($N-1$ bit or shorter) that is to be transmitted to the intended receiver in a confidential manner (by encrypting it with $\mathcal{E}$). Instead of authenticating the message using a traditional MAC algorithm, consider the following procedure. On input a message m, a random nonce $r \in \mathbb{Z}_p$ is chosen. (We overload m to denote both the binary string representing the message, and the integer representation of the message as an element of $\mathbb{Z}_p$. The same applies to k_s and r. The distinction between the two representations will be omitted when it is clear from the context.) We assume that integers representing distinct messages are also distinct, which can be achieved by appropriately encoding messages [22].

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

$$\tau \equiv mk_s + r \pmod{p}. \tag{2}$$

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

Since the generation of pseudorandom numbers can be considered expensive for computationally limited devices, there have been several attempts to design true random number generators that are suitable for RFID tags (see, e.g., [59, 44,45]) and for low-cost sensor nodes (see, e.g., [71,26, 37]). Thus, we assume the availability of such random number generators.

Now, the ciphertext $c = \mathcal{E}(m||r)$ and the authentication tag τ, computed according to equation (2), are transmitted to the intended receiver.

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

$$\tau \overset{?}{\equiv} mk_s + r \pmod p. \tag{3}$$

If the integrity check of equation (3) is satisfied, the message is considered authentic. Otherwise, the integrity of the message is denied.

Note, however, that the authentication tag is a function of the confidential message. Therefore, the authentication tag must not reveal information about the plaintext since, otherwise, the confidentiality of the encryption algorithm is compromised. Before we give formal security analysis of the proposed technique, we first discuss its performance compared to existing techniques.

3.3 Performance Discussion

There are three classes of standard message authentication codes (MACs) that can be used to preserve message integrity in mobile and pervasive computing. One can use a MAC based on block ciphers, a MAC based on cryptographic hash functions, or a MAC based on universal hash-function families. Since MACs based on universal hashing are known to be more computationally-efficient than MACs based on block ciphers and cryptographic hash function [84], we focus on comparing the proposed MAC to universal hash functions based MACs.

In MACs based on universal hashing, two phases of computations are required: 1. a message compression phase using a universal hash function and, 2. a cryptographic phase in which the compressed image is processed with a cryptographic primitive (a block cipher or a cryptographic hash function). The compression phase is similar to the computation of equation (2) of the proposed MAC (in fact, the proposed MAC of equation (2) is an instance of strongly universal hash functions). As opposed to standard universal hash functions based MACs, however, there is no need to process the the result of equation (2) with a cryptographic function in the proposed technique.

When the messages to be authenticated are short, the modulus prime, p, can also be small. For a small modulus, the modular multiplication of equation (2) is not a time consuming operation. That is, for short messages, the cryptographic phase is the most time consuming phase. Since we target applications in which messages are short, eliminating the need to perform such a cryptographic operation will have a significant impact on the performance of the MAC operation. For instance, while the cryptographic hash function SHA-256 hashes at around 21 cycles/byte [64], the modular multiplication of equation (2) runs in about 1.5 cycles/byte [22], which illustrates the significance of removing the cryptographic phase from our MAC.

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

Compared to single-pass authenticated encryption algorithms, when combined with a stream cipher, the technique of Section 3.2 will be much faster (recall that single-pass authenticated encryption methods are block cipher based[2]). 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 r and the size of the block cipher might not require any additional block cipher calls, single-pass authenticated encryption methods typically require at least two additional block cipher calls.

3.4 Security Model

A message authentication scheme consists of a signing algorithm $\mathcal{S}$ and a verifying algorithm $\mathcal{V}$. The signing algorithm might be probabilistic, while the verifying one is usually not. Associated with the scheme are parameters ℓ and N describing the length of the shared key and the resulting authentication tag, respectively. On input an ℓ-bit key k and a message m, algorithm $\mathcal{S}$ outputs an N-bit string τ called the authentication tag, or the MAC of m. 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.

[2] Although stream cipher based authenticated encryption primitives have appeared in [34,89], such proposals have been analyzed and shown to be vulnerable to attacks [63,67,66,90].

In general, an adversary against 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 outcome of running the experiment in the presence of an adversary is used to define security.

3.5 Security Analysis

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

3.5.1 Data Privacy

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

$$\mathsf{Adv}_{\mathcal{E}}^{\text{ind-cpa}}(\mathcal{B}) = \Pr\left[b' = b\right]. \quad (4)$$

As stated in equation (1), $\mathcal{E}$ provides IND-CPA if the adversary has a negligible advantage of guessing the right bit over an adversary choosing a bit uniformly at random.

Now, let Σ denote the proposed authenticated encryption composition described in Section 3.2. Let $\mathcal{A}$ be an adversary against the privacy of Σ and let $\mathsf{Adv}_{\Sigma}^{\text{priv}}(\mathcal{A})$ denote adversary's $\mathcal{A}$ advantage in breaking the privacy of the system, where the privacy of the system is modeled as its indistinguishability under chosen plaintext attacks. One gets the following theorem.

Theorem 1 *Let Σ be the authenticated encryption composition described in Section 3.2 using $\mathcal{E}$ as the underlying encryption algorithm. Then given an adversary, $\mathcal{A}$, against the privacy of Σ, one can construct an adversary, $\mathcal{B}$, against $\mathcal{E}$ such that*

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

Theorem 1 states that an adversary breaking the privacy of the proposed system will also be able to break the IND-CPA of the underlying encryption algorithm. Therefore, if $\mathcal{E}$ provides IND-CPA, the adversary's advantage of exposing private information about the system is negligible. Note that private information here refers not only to the encrypted messages, but also the secret key, k_s, as well as the secret key of the encryption algorithm.

Proof (Proof of Theorem 1) Recall that each authentication tag, τ, computed according to equation (2) requires the generation of a random nonce, r. Recall further that r is generated internally and is not part of the chosen message attack. Now, if r is delivered to the receiver using a secure channel (e.g., out of band), then equation (2) is an instance of a perfectly secret (in Shannon's information theoretic sense) one-time pad cipher (encrypted with the one-time key r) and, hence, no information will be exposed. However, the r corresponding to each tag is delivered via the ciphertext. Therefore, the only way to expose private information is from the ciphertext.

Assume now that $\mathcal{A}$ is an adversary against the privacy of the system proposed in Section 3.2. Let $\mathcal{B}$ be an adversary with access oracle to the encryption algorithm $\mathcal{E}$ and let adversary $\mathcal{A}$ use adversary $\mathcal{B}$ to attack the privacy of observed ciphertexts. Then,

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

and the theorem follows. □

By Theorem 1, the privacy of the proposed technique is provably secure given the IND-CPA security of the underlying encryption algorithm, as desired.

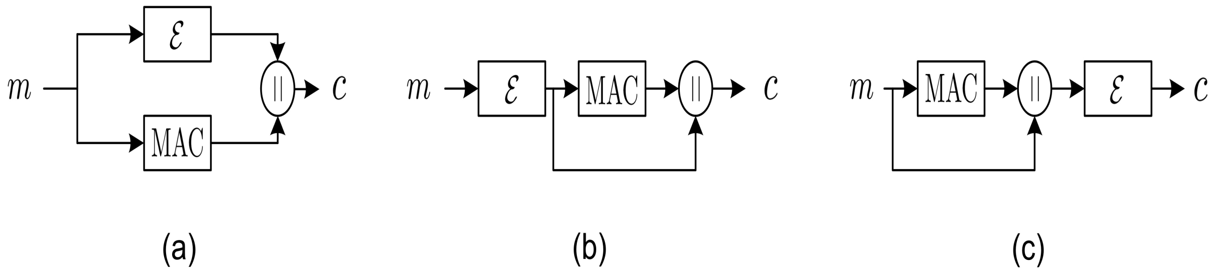

Fig. 1 A schematic of the three generic compositions; (a) Encrypt-and-Authenticate (E&A), (b) Encrypt-then-Authenticate (EtA), and (c) Authenticate-then-Encrypt (AtE).

3.5.2 Data Authenticity

We can now proceed with the main theorem formalizing the adversary's advantage of successful forgery against the proposed scheme. As before, let Σ denotes the proposed authenticated encryption composition of Section 3.2 and let $\mathsf{Adv}^{\text{auth}}_{\Sigma}(\mathcal{A})$ denotes adversary's $\mathcal{A}$ advantage of successful forgery against Σ.

Theorem 2 *Let Σ denotes the proposed authenticated encryption composition of Section 3.2 in which the authentication tag is computed over the the finite integer field $\mathbb{Z}_p$. Let $\mathcal{A}$ be an adversary making a q signing queries before attempting its forgery. Then, one can come up with an adversary, $\mathcal{B}$, against the IND-CPA security of the underlying encryption algorithm, $\mathcal{E}$, such that*

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

Theorem 2 states that if the adversary's advantage in breaking the IND-CPA security of the underlying encryption algorithm as negligible, then so is her advantage in breaking the integrity of the scheme. That is, the integrity of the scheme of Section 3.2 is provably secure provided the underlying encryption algorithm is IND-CPA secure.

Proof (Proof of Theorem 2) Assume an adversary calling the signing oracle for q times and recording the sequence

$$\mathsf{Seq} = \Big\{(m_1, \tau_1), \cdots, (m_q, \tau_q)\Big\} \tag{5}$$

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

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

$$\tau \equiv mk_s + r \pmod p \tag{6}$$
$$\equiv (m_i + \epsilon)k_s + (r_i + \delta) \pmod p \tag{7}$$
$$\equiv \tau_i + \epsilon k_s + \delta \pmod p. \tag{8}$$

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

Remark 2 Observe that, if both k_s and r are used only once (i.e., one-time keys), the authentication tag of equation (2) is a well-studied example of a strongly universal hash family (see [82] for a definition of strongly universal hash families and detailed discussion showing that equation (2) is indeed strongly universal hash family). The only difference is that we restrict k_s to belong to the multiplicative group modulo p, whereas it can be equal to zero in unconditionally secure authentication. This is because, in unconditionally secure authentication, the keys can only be used once. In our technique, since k_s can be used to authenticate an arbitrary number of messages, it cannot be chosen to be zero. Otherwise, mk_s will always be zero and the system will not work. The novelty of our approach is to utilize the encryption primitive to reach the simplicity of unconditionally secure authentication, without the need for impractically long keys.

Note also that, unless further assumptions about the encryption algorithm is assumed (such as the pseudorandom permutation property), it is critical for the security of authentication to perform the multiplication modulo a prime integer. That is, it was shown in [3] that the security of authentication based on universal hash families similar to the one in equation (2) is dependent on the used modulus. In particular, it was shown that the probability of successful forgery is proportional to the reciprocal of the smallest prime factor of the used modulus [3].

3.5.3 Security of the Authenticated Encryption Composition

In [16], Bellare and Namprempre defined two notions of integrity for authenticated encryption systems: the first is integrity of plaintext (INT-PTXT) and the second is integrity of ciphertext (INT-CTXT). Combined with encryption algorithms that provide indistinguishability under chosen plaintext attacks (IND-CPA), the security of different methods for constructing generic compositions is analyzed. Note 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.

It was shown in [16] that E&A compositions do not generally 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 [16]). Obviously, if such a MAC is used to compose an E&A system, then the authenticated encryption does not provide IND-CPA. By Theorem 1, however, the proposed authenticated encryption scheme is at least as private as the underlying encryption algorithm. Since the encryption algorithm is IND-CPA secure, the resulting composition provides IND-CPA.

Another result of [16] 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 [16]. 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 [53]). To see this, observe that, by the one-to-one property, any modification of the ciphertext will correspond to changing its corresponding plaintext and, by Theorem 2, a modified plaintext will go undetected with a negligible probability.

4 From Weak to Strong Unforgeability

As per [16], 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 s, compute $PRF_x(s)$, where PRF_x is a pseudorandom function determined by a secret key x, and transmit $c = (s, PRF_x(s) \oplus m)$ as the ciphertext. Then, $\mathcal{E}$ is an IND-CPA secure encryption. Applied to our construction, on input a message m, the ciphertext will be $c = \big(s, PRF_x(s) \oplus (m||r)\big)$ and the corresponding tag will be $\tau \equiv mk_s + r \pmod{p}$. Now, let s' be a string of length equal to the concatenation of m and r. Then, $c' = \big(s, PRF_x(s) \oplus (m||k) \oplus s'\big) = \big(s, PRF_x(s) \oplus (m||k \oplus s')\big)$. Let s' be a string of all zeros except for the least significant bit, which is set to one. Then, either $\tau_1 \equiv mk_s + r + 1 \pmod{p}$ or $\tau_2 \equiv mk_s + r - 1 \pmod{p}$ will be a valid tag for m, when c' is transmitted as the ciphertext. That is, the same message can be authenticated using different tags with high probabilities.

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

As can be observed from the above example, the forgery is successful if the adversary can modify the value of r and predict its effect on the authentication tag τ. To rectify this problem, not only the message but also the coin tosses, r, must be authenticated. Obviously, this can be done with the use of another secret key k'_s and computing the tag as

$$\tau \equiv mk_s + rk'_s \pmod{p}. \tag{9}$$

This, however, requires twice the amount of shared key material and an extra multiplication operation. A more efficient way of achieving the same goal can be done by computing the modular multiplication

$$\sigma = mk_s \pmod{p} \tag{10}$$

and transmitting an encrypted version of the result of equation (10) as the authentication tag. That is, since r is the main reason for the successful forgery illustrated above, instead of authenticating r as in equation (9), it is removed from the equation. However, since r was necessary for the privacy of the scheme of Section 3.2, it is required to encrypt the result of equation (10) before transmission to provide data privacy. This implies that the scheme described here is an instance of the Authenticate-then-Encrypt (AtE) composition as apposed to the Encrypt-and-Authenticate (E& A) composition of Section 3.2.

The description of the modified system is as follows. Assume the users have agreed on a security parameter N, exchanged an N-bit prime integer p, and a secret key $k_s \in \mathbb{Z}_p^*$. On input a message $m \in \mathbb{Z}_p$, compute the modular multiplication $\sigma = mk_s \pmod{p}$. The transmitter encrypts m and σ and transmits the ciphertext $c = \mathcal{E}(m, \sigma)$ to the intended receiver. The ciphertext can be the encryption of the plaintext message concatenated with σ, i.e. $\mathcal{E}(m||\sigma)$, or it can be the concatenation of the encryption of the message and the encryption of σ, i.e. $\mathcal{E}(m)||\mathcal{E}(\sigma)$. For ease of presentation, we will assume the latter scenario and call the ciphertext $c = \mathcal{E}(m)$ and the tag $\tau = \mathcal{E}(\sigma)$. Decryption and authentication are performed accordingly.

The proof that this modified scheme provides data privacy can be found in [16]. In particular, since the modified scheme of this section is an instance of AtE compositions, Bellare and Namprempre showed that if the underlying encryption algorithm is IND-CPA secure, then so is the generic AtE composition [16]. The proof that the modified scheme achieves weak unforgeability under chosen message attacks is similar to the proof of Theorem 2 and, thus, is omitted. 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 IND-CPA security of the underlying encryption algorithm.*

Proof Let (m, τ) be a valid message-tag pair recorded by the adversary. By equation (10), for the same m, the resulting σ will always be the same. Assume the adversary is attempting to authenticate the same message, m, with a different tag τ'. Since σ in both cases is the same, the difference between τ and τ' is due to the probabilistic behavior of the encryption algorithm. Therefore, the adversaries advantage of breaking the SUF-CMA security of the scheme is negligible provided the IND-CPA security of the encryption algorithm. That is,

$$\mathsf{Adv}_{\Sigma}^{\text{suf-cma}}(\mathcal{A}) \leq \mathsf{Adv}_{\mathcal{E}}^{\text{ind-cpa}}(\mathcal{B}) + \mathsf{negl}(N),$$

where $\mathsf{negl}(N)$ is a negligible function in the security parameter N, and the theorem follows. □

5 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 a random nonce to the intended receiver via the ciphertext. This allowed the design of an authentication code that benefits from the simplicity of unconditionally secure authentication without the need to manage one-time keys. In particular, it has been demonstrated in this 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. Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. Computer networks **38**(4), 393–422 (2002)
2. Alomair, B., Clark, A., Cuellar, J., Poovendran, R.: Scalable RFID Systems: a Privacy-Preserving Protocol with Constant-Time Identification. In: the 40th Annual IEEE/IFIP International Conference on Dependable Systems and Networks – DSN'10, pp. 1–10. IEEE, IEEE Computer Society (2010)
3. Alomair, B., Clark, A., Poovendran, R.: The Power of Primes: Security of Authentication Based on a Universal Hash-Function Family. Journal of Mathematical Cryptology **4**(2) (2010)
4. Alomair, B., Lazos, L., Poovendran, R.: Securing Low-cost RFID Systems: an Unconditionally Secure Approach. Journal of Computer Security – Special Issue on RFID System Security **19**(2), 229–256 (2011)
5. Alomair, B., Poovendran, R.: Efficient Authentication for Mobile and Pervasive Computing. In: The 12th International Conference on Information and Communications Security–ICICS'10. Springer (2010)
6. Alomair, B., Poovendran, R.: $\mathcal{E}$-MACs: Towards More Secure and More Efficient Constructions of Secure Channels. In: the 13th International Conference on Information Security and Cryptology – ICISC'10. Springer (2010)
7. Alomair, B., Poovendran, R.: Privacy versus Scalability in Radio Frequency Identification Systems. Computer Communication, Elsevier **33**(18), 2155–2163 (2010)
8. Atici, M., Stinson, D.: Universal Hashing and Multiple Authentication. In: Advances in Cryptology–CRYPTO'96, vol. 96, pp. 16–30. Lecture Notes in Computer Science, Springer (1996)
9. Avoine, G., Carpent, X., Martin, B.: Strong Authentication and Strong Integrity (SASI) is not that Strong. In: Workshop on RFID Security – RFIDSec'10, *Lecture Notes in Computer Science*, vol. 6370, pp. 50–64. Springer (2010)
10. Avoine, G., Coisel, I., Martin, T.: Time Measurement Threatens Privacy-Friendly RFID Authentication Protocols. In: Workshop on RFID Security – RFIDSec'10, *Lecture Notes in Computer Science*, vol. 6370, pp. 138–157. Springer (2010)
11. Avoine, G., Martin, B., Martin, T.: Tree-Based RFID Authentication Protocols Are Definitively Not Privacy-Friendly. In: Workshop on RFID Security – RFIDSec'10, *Lecture Notes in Computer Science*, vol. 6370, pp. 103–122. Springer (2010)
12. Bellare, M., Canetti, R., Krawczyk, H.: Keying Hash Functions for Message Authentication. In: Advances in Cryptology–CRYPTO'96, vol. 96, pp. 1–15. Lecture Notes in Computer Science, Springer (1996)

13. Bellare, M., Guerin, R., Rogaway, P.: XOR MACs: New methods for message authentication using finite pseudorandom functions. In: Advances in Cryptology–CRYPTO'95, vol. 963, pp. 15–28. Lecture Notes in Computer Science, Springer (1995)
14. Bellare, M., Kilian, J., Rogaway, P.: The Security of the Cipher Block Chaining Message Authentication Code. Journal of Computer and System Sciences **61**(3), 362–399 (2000)
15. 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)
16. Bellare, M., Namprempre, C.: Authenticated Encryption: Relations Among Notions and Analysis of the Generic Composition Paradigm. Journal of Cryptology **21**(4), 469–491 (2008)
17. Bellare, M., Rogaway, P., Wagner, D.: The EAX mode of operation. In: Proceedings of Fast Software Encryption–FSE'04, vol. 3017, pp. 389–407. Lecture Notes in Computer Science, Springer (2004)
18. Bernstein, D.: Floating-point arithmetic and message authentication. Available at `http://cr.yp.to/hash127.html` (2004)
19. Bernstein, D.: The Poly1305-AES message-authentication code. In: Proceedings of Fast Software Encryption–FSE'05, vol. 3557, pp. 32–49. Lecture Notes in Computer Science, Springer (2005)
20. Bierbrauer, J.: A2-codes from universal hash classes. In: Advances in Cryptology–EUROCRYPT'95, vol. 921, pp. 311–318. Lecture Notes in Computer Science, Springer (1995)
21. Bierbrauer, J.: Universal hashing and geometric codes. Designs, Codes and Cryptography **11**(3), 207–221 (1997)
22. Black, J., Halevi, S., Krawczyk, H., Krovetz, T., Rogaway, P.: UMAC: Fast and Secure Message Authentication. In: Advances in Cryptology–CRYPTO'99, vol. 1666, pp. 216–233. Lecture Notes in Computer Science, Springer (1999)
23. Bogdanov, A., Knudsen, L., Leander, G., Paar, C., Poschmann, A., Robshaw, M., Seurin, Y., Vikkelsoe, C.: PRESENT: An Ultra-Lightweight Block Cipher. In: Cryptographic Hardware and Embedded Systems–CHES'07, vol. 4727, pp. 450–466. Lecture Notes in Computer Science, Springer (2007)
24. Bogdanov, A., Leander, G., Paar, C., Poschmann, A., Robshaw, M., Seurin, Y.: Hash Functions and RFID Tags : Mind The Gap. In: Proceedings of the 10th International Workshop Cryptographic Hardware and Embedded Systems–CHES'08, *Lecture Notes in Computer Science*, vol. 5154, pp. 283–299. Springer (2008)
25. Bosselaers, A., Govaerts, R., Vandewalle, J.: Fast hashing on the Pentium. In: Advances in Cryptology-CRYPTO'96, vol. 1109, pp. 298–312. Lecture Notes in Computer Science, Springer (1996)
26. 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)
27. Carter, J., Wegman, M.: Universal classes of hash functions. In: Proceedings of the ninth annual ACM symposium on Theory of computing–STOC'77, pp. 106–112. ACM (1977)
28. Carter, L., Wegman, M.: Universal hash functions. Journal of Computer and System Sciences **18**(2), 143–154 (1979)
29. Dierks, T., Rescorla, E.: The transport layer security (TLS) protocol version 1.2. Tech. rep., RFC 5246 (2008)
30. Doraswamy, N., Harkins, D.: IPSec: the new security standard for the Internet, intranets, and virtual private networks. Prentice Hall (2003)
31. Dworkin, M.: Recommendation for block cipher modes of operation: The CMAC mode for authentication (2005)
32. Etzel, M., Patel, S., Ramzan, Z.: Square hash: Fast message authentication via optimized universal hash functions. In: Advances in Cryptology–CRYPTO'99, vol. 1666, pp. 234–251. Lecture Notes in Computer Science, Springer (1999)
33. Feldhofer, M., Dominikus, S., Wolkerstorfer, J.: Strong Authentication for RFID Systems using the AES Algorithm. In: Cryptographic Hardware and Embedded Systems–CHES'04, vol. 3156, pp. 357–370. Lecture Notes in Computer Science, Springer (2004)
34. Ferguson, N., Whiting, D., Schneier, B., Kelsey, J., Kohno, T.: Helix: Fast encryption and authentication in a single cryptographic primitive. In: Proceedings of Fast Software Encryption–FSE'03, vol. 2887, pp. 330–346. Lecture notes in computer science, Springer (2003)
35. FIPS 113: Computer Data Authentication. Federal Information Processing Standards Publication, 113 (1985)
36. FIPS 198: The Keyed-Hash Message Authentication Code (HMAC). Federal Information Processing Standards Publication **198** (2002)
37. 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'07, pp. 1–7. Citeseer (2007)
38. Freier, A., Karlton, P., Kocher, P.: The SSL Protocol Version 3.0 (1996)
39. Gligor, V., Donescu, P.: Fast Encryption and Authentication: XCBC Encryption and XECB Authentication Modes. In: Fast Software Encryption–FSE'02, vol. 2355, pp. 1–20. Lecture Notes in Computer Science, Springer (2002)
40. Goldreich, O.: Foundations of Cryptography. Cambridge University Press (2001)
41. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences **28**(2), 270–299 (1984)
42. Halevi, S., Krawczyk, H.: MMH: Software message authentication in the Gbit/second rates. In: Proceedings of Fast Software Encryption–FSE'97, vol. 1267, pp. 172–189. Lecture notes in computer science, Springer (1997)
43. Helleseth, T., Johansson, T.: Universal hash functions from exponential sums over finite fields and Galois rings. In: Advances in cryptology–CRYPTO'96, vol. 1109, pp. 31–44. Lecture Notes in Computer Science, Springer (1996)
44. 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'07 (2007)
45. 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)
46. Hong, D., Sung, J., Hong, S., Lim, J., Lee, S., Koo, B., Lee, C., Chang, D., Lee, J., Jeong, K., Kim, H., Kim, J., Chee, S.: HIGHT: A New Block Cipher Suitable for Low-Resource Device. In: Cryptographic Hardware and Embedded Systems–CHES'06, *Lecture Notes in Computer Science*, vol. 4249, pp. 46–59. Springer (2006)
47. ISO/IEC 9797-1: Information technology – Security techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher (1999)
48. ISO/IEC 9797-2: Information technology – Security techniques – Message Authentication Codes (MACs) – Part 2: Mechanisms using a dedicated hash-function (2002)
49. Iwata, T., Kurosawa, K.: omac: One-key cbc mac. In: Fast Software Encryption–FSE'03, vol. 2887, pp. 129–153. Lecture notes in computer science, Springer (2003)
50. Juels, A.: RFID security and privacy: A research survey. IEEE Journal on Selected Areas in Communications **24**(2), 381–394 (2006)
51. Jutla, C.: Encryption modes with almost free message integrity. Journal of Cryptology **21**(4), 547–578 (2008)
52. Kaps, J., Yuksel, K., Sunar, B.: Energy scalable universal hashing. IEEE Transactions on Computers **54**(12), 1484–1495 (2005)
53. Katz, J., Lindell, Y.: Introduction to modern cryptography. Chapman & Hall/CRC (2008)
54. Kavun, E.B., Yalcin, T.: A Lightweight Implementation of Keccak Hash Function for Radio-Frequency Identification Applications. In: Workshop on RFID Security–RFIDSec'10 (2010)

55. Kohno, T., Viega, J., Whiting, D.: CWC: A high-performance conventional authenticated encryption mode. In: Fast Software Encryption–FSE'04, vol. 3017, pp. 408–426. Lecture Notes in Computer Science, Springer (2004)
56. Krawczyk, H.: The order of encryption and authentication for protecting communications(or: How secure is SSL?). In: Advances in Cryptology–CRYPTO'01, vol. 2139, pp. 310–331. Lecture Notes in Computer Science, Springer (2001)
57. Krovetz, T.: `http://fastcrypto.org/umac/` (2006)
58. Lim, C.H., Korkishko, T.: mCrypton - A Lightweight Block Cipher For Security of Low-Cost RFID Tags and Sensors. In: Workshop on Information Security Applications–WISA'05, *Lecture Notes in Computer Science*, vol. 3786, pp. 243–258. Springer (2005)
59. Liu, Z., Peng, D.: True Random Number Generator in RFID Systems Against Traceability. In: IEEE Consumer Communications and Networking Conference–CCNS,06, vol. 1, pp. 620–624. IEEE (2006)
60. Macé, F., Standaert, F.X., Quisquater, J.J.: ASIC Implementations of the Block Cipher SEA for Constrained Applications. In: Workshop on RFID Security–RFIDSec'07 (2007)
61. Maurer, U., Tackmann, B.: On the soundness of authenticate-then-encrypt: formalizing the malleability of symmetric encryption. In: Proceedings of the 17th ACM conference on Computer and communications security – CCS'10, pp. 505–515. ACM (2010)
62. McGrew, D., Viega, J.: The security and performance of the Galois/Counter Mode (GCM) of operation. In: Progress in Cryptology-INDOCRYPT'04, vol. 3348, pp. 343–355. Lecture notes in computer science, Springer (2004)
63. Muller, F.: Differential attacks against the Helix stream cipher. In: Fast Software Encryption–FSE'04, vol. 3017, pp. 94–108. Lecture Notes in Computer Science, Springer (2004)
64. Nakajima, J., Matsui, M.: Performance analysis and parallel implementation of dedicated hash functions. In: Advances in Cryptology–EUROCRYPT 2002, pp. 165–180. Springer (2002)
65. O'Neill (McLoone), M.: Low-Cost SHA-1 Hash Function Architecture for RFID Tags. In: Workshop on RFID Security–RFIDSec'08 (2008)
66. Paul, S., Preneel, B.: Near Optimal Algorithms for Solving Differential Equations of Addition with Batch Queries. In: Progress in Cryptology–INDOCRYPT'05, vol. 3797, pp. 90–103. Lecture Notes in Computer Science, Springer (2005)
67. Paul, S., Preneel, B.: Solving systems of differential equations of addition. In: Australasian Conference on Information Security and Privacy–ICISP'05, vol. 3574, pp. 75–88. Lecture Notes in Computer Science, Springer (2005)
68. Peris-Lopez, P., Hernandez-Castro, J., Estevez-Tapiador, J., Ribagorda, A.: RFID systems: A survey on security threats and proposed solutions. In: Personal Wireless Communications, pp. 159–170. Springer (2006)
69. Perrig, A., Stankovic, J., Wagner, D.: Security in wireless sensor networks. Communications of the ACM **47**(6), 53–57 (2004)
70. Perrig, A., Szewczyk, R., Tygar, J., Wen, V., Culler, D.: SPINS: Security protocols for sensor networks. Wireless networks **8**(5), 521–534 (2002)
71. 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)
72. Poschmann, A., Leander, G., Schramm, K., Paar, C.: A Family of Light-Weight Block Ciphers Based on DES Suited for RFID Applications. In: Workshop on RFID Security–RFIDSec'06. Ecrypt (2006)
73. Preneel, B., Van Oorschot, P.: MDx-MAC and building fast MACs from hash functions. In: Advances in Cryptology-CRYPTO'95, vol. 963, pp. 1–14. Lecture Notes in Computer Science, Springer (1995)
74. Preneel, B., Van Oorschot, P.: On the security of iterated message authentication codes. IEEE Transactions on Information theory **45**(1), 188–199 (1999)
75. Rogaway, P.: Comments on NISTs RMAC Proposal (2002)
76. 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)
77. Rogaway, P., Black, J.: PMAC: Proposal to NIST for a parallelizable message authentication code (2001)
78. Sarma, S., Weis, S., Engels, D.: RFID systems and security and privacy implications. Cryptographic Hardware and Embedded Systems-CHES 2002 pp. 1–19 (2003)
79. Schwarz, S.: The role of semigroups in the elementary theory of numbers. Math. Slovaca **31**(4), 369–395 (1981)
80. Shamir, A.: SQUASH–A New MAC with Provable Security Properties for Highly Constrained Devices Such as RFID Tags. In: Fast Software Encryption–FSE'08, vol. 5086, pp. 144–157. Lecture Notes in Computer Science, Springer (2008)
81. Shoup, V.: On fast and provably secure message authentication based on universal hashing. In: Advances in Cryptology–CRYPTO'96, vol. 1109, pp. 313–328. Lecture Notes in Computer Science, Springer (1996)
82. Stinson, D.: Cryptography: Theory and Practice. CRC Press (2006)
83. Tan, C., Wang, H., Zhong, S., Li, Q.: Body sensor network security: an identity-based cryptography approach. In: Proceedings of the first ACM conference on Wireless network security, pp. 148–153. ACM (2008)
84. van Tilborg, H.: Encyclopedia of cryptography and security. Springer (2005)
85. Tsudik, G.: Message authentication with one-way hash functions. ACM SIGCOMM Computer Communication Review **22**(5), 38 (1992)
86. Venkatasubramanian, K., Banerjee, A., Gupta, S.: Ekg-based key agreement in body sensor networks. In: INFOCOM Workshops 2008, IEEE, pp. 1–6. IEEE (2008)
87. Wegman, M., Carter, J.: New classes and applications of hash functions. In: 20th Annual Symposium on Foundations of Computer Science–FOCS'79, pp. 175–182. IEEE (1979)
88. 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)
89. 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, `www.ecrypt.eu.org/stream` (2005)
90. Wu, H., Preneel, B.: Differential-linear attacks against the stream cipher Phelix. In: Fast Software Encryption–FSE'07, vol. 4593, pp. 87–100. Lecture Notes in Computer Science, Springer (2007)
91. Ylonen, T., Lonvick, C.: The Secure Shell (SSH) Transport Layer Protocol. Tech. rep., RFC 4253 (2006)
92. Zhang, Y.: A design proposal of security architecture for medical body sensor networks. In: Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on, pp. 4–90. IEEE (2006)