Efficient Authentication for Mobile and Pervasive Computing

Basel Alomair, *Member*, *IEEE*, and Radha Poovendran, *Senior Member*, *IEEE*

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 two novel techniques 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 techniques is to utilize the security that the encryption algorithm can provide to design more efficient authentication mechanisms, as opposed to using standalone authentication primitives.

Index Terms—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 [1]. 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., [2], [3], [4], [5]). 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 [6] and the International Organization for Standardization ISO/IEC 9797-1 [7]. CMAC, a modified version of CBC-MAC, is presented in the NIST special publication 800-38B [8], which was based on the OMAC of [9]. Other block cipher-based MACs include, but are not limited to, XOR-MAC [10] and PMAC [11]. The security of different MACs has been exhaustively studied (see, e.g., [12], [13]).

The use of one-way cryptographic hash functions for message authentication was introduced by Tsudik [14]. 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. [15]. HMAC was later adopted as a standard [16]. Another cryptographic hash function-based MAC is the MDx-MAC proposed by Preneel and Van Oorschot [17]. HMAC and two variants of MDx-MAC are specified in the International Organization for Standardization ISO/IEC 9797-2 [18]. Bosselaers et al. [19] 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.

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

- *B. Alomair is with the Center for Cybersecurity (C4C), King Abdulaziz City for Science and Technology (KACST), King Abdullah Road, PO Box 6086, Riyadh 11442, Saudi Arabia. E-mail: alomair@uw.edu.*
- *R. Poovendran is with the Department of Electrical Engineering, University of Washington, 437 EE/CSE Building, Seattle, WA 98195-2500. E-mail: rp3@uw.edu.*

Manuscript received 17 May 2012; revised 17 Nov. 2012; accepted 30 Nov. 2012; published online 11 Dec. 2012.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2012-05-0254.
Digital Object Identifier no. 10.1109/TMC.2012.252.

[1] Earlier designs used one-time pad encryption to process the compressed image. However, due to the difficulty to manage such one-time keys, recent designs resorted to computationally secure primitives (see, e.g., [20])

Published by the IEEE CS, CASS, ComSoc, IES, & SPS

computationally secure universal hashing-based MACs include, but are not limited to, [20], [21], [22].

Indeed, universal hashing-based MACs give better performance when compared to block cipher or cryptographic hashing-based MACs. In fact, the fastest MACs in the cryptographic literature are based on universal hashing [23]. The main reason behind the performance advantage of universal hashing-based MACs is the fact that processing messages block by block using universal hash functions is orders of magnitude faster than processing them block by block using block ciphers or cryptographic hash functions.

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 functionality 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 [23], has undergone large algorithmic changes to increase its speed on short messages [24].)

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 [25], [26], [27].

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 (e.g., tags unique identifiers are 64-bit long in the EPC Class-1 Generation-2 standard [28]), 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 [29], [30], [31], [32].

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 [33], [34].

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, for example, [35], [36], [37]. Implementations of hardware efficient cryptographic hash functions have also been proposed in, for example, [38], [39]. 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.

1.1 Contributions

In this work, we pose the following research question: if there is an application in which messages that need to be exchanged are short and both their privacy and integrity need to be preserved, can one do better than simply encrypting the messages using an encryption algorithm and authenticating them using standard MAC algorithm? We answer the question by proposing two new techniques for authenticating short encrypted messages that are more efficient than existing approaches. In the first technique, we utilize the fact that the message to be authenticated is also encrypted, with any secure encryption algorithm, to append a short random string to be used in the authentication process. Since the random strings used for different operations are independent, the authentication algorithm can benefit from the simplicity of unconditional secure authentication to allow for faster and more efficient authentication, without the difficulty to manage one-time keys. In the second technique, we make the extra assumption that the used encryption algorithm is block cipher based to further improve the computational efficiency of the first technique. The driving motive behind our investigation is that using a general purpose MAC algorithm to authenticate exchanged messages in such systems might not be the most efficient solution and can lead to waste of resources already available, namely, the security that is provided by the encryption algorithm.

1.2 Organization

The remainder of the paper is organized as follows: In Section 2, we list our notations and discuss some preliminaries. In Section 3, we describe the first authentication technique assuming messages do not exceed a maximum length, discuss its performance advantages over existing techniques, and prove its security. In Section 4, we propose a modification to the scheme of Section 3 that provides a stronger notion of integrity. In Section 5, we describe the second technique assuming the encryption is block cipher based, discuss its performance, and prove its security. In Section 6, 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 $\mathrm{negl} : \mathbb{N} \rightarrow \mathbb{R}$ is said to be negligible if for any nonzero polynomial poly, there exists N_0 such that for all $N > N_0$, $|\mathrm{negl}(N)| < 1/|\mathrm{poly}(N)|$. 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 $\mathrm{Adv}_{\mathcal{E}}^{\mathrm{ind-cpa}}(\mathcal{A})$ 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

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

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 Block Ciphers

A block cipher mapping N-bit strings to N-bit strings is a family of permutations $\mathcal{F}$ specified by a finite set of keys $\mathcal{K}_e$. Each key $K \in \mathcal{K}_e$ defines a member of the family $\mathcal{F}_K \in \mathcal{F}$. As opposed to thinking of $\mathcal{F}$ as a set of functions mapping elements from $\{0,1\}^N$ to elements in $\{0,1\}^N$, it can be viewed as a single function $\mathcal{F} : \mathcal{K}_e \times \{0,1\}^N \rightarrow \{0,1\}^N$, whose first argument is usually written as a subscript. A random element $f \overset{\$}{\leftarrow} \mathcal{F}$ is determined by selecting a $K \overset{\$}{\leftarrow} \mathcal{K}_e$ uniformly at random and setting $f \leftarrow \mathcal{F}_K$.

As in [42], we adopt the notion of security for block ciphers introduced in [43] and adopted for the concrete setting in [44]. Let $\mathcal{F} : \{0,1\}^\ell \times \{0,1\}^N \rightarrow \{0,1\}^N$, where ℓ is the key length and N is the block size of the block cipher, be a block cipher and let $\mathrm{Perm}(N)$ denote the set of all permutations on $\{0,1\}^N$. Let $\mathcal{A}$ be an adversary with access to an oracle and that returns a bit. Then,

$$\begin{aligned}\mathrm{Adv}_{\mathcal{F}}^{\mathrm{prp}}(\mathcal{A}) = \Pr\left[f \overset{\$}{\leftarrow} \mathcal{F} : \mathcal{A}^{f(\cdot)} = 1\right] \\ - \Pr\left[\pi \overset{\$}{\leftarrow} \mathrm{Perm}(N) : \mathcal{A}^{\pi(\cdot)} = 1\right]\end{aligned} \tag{2}$$

denotes the prp-advantage of $\mathcal{A}$ in distinguishing a random instance of $\mathcal{F}$ from a random permutation. Intuitively, we say that $\mathcal{F}$ is a secure prp, or a secure block cipher, if the prp-advantages of all adversaries using reasonable resources is a negligible function in the security parameter. A block cipher is said to be strong pseudorandom permutation (sprp) if it is indistinguishable from a random permutation even if the adversary is given an oracle access to the inverse function. Then,

$$\begin{aligned}\mathrm{Adv}_{\mathcal{F}}^{\mathrm{sprp}}(\mathcal{A}) = \Pr\left[f \overset{\$}{\leftarrow} \mathcal{F} : \mathcal{A}^{f(\cdot), f^{-1}(\cdot)} = 1\right] \\ - \Pr\left[\pi \overset{\$}{\leftarrow} \mathrm{Perm}(N) : \mathcal{A}^{\pi(\cdot), \pi^{-1}(\cdot)} = 1\right]\end{aligned} \tag{3}$$

denotes the sprp-advantage of $\mathcal{A}$ in distinguishing a random instance of $\mathcal{F}$ from a random permutation. Modern block ciphers, such as AES [45], are believed to be secure strong pseudorandom permutations.

2.5 A Useful Result

The following lemma, a general result known in probability and group theory [46], 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 our first authentication scheme that can be used with any IND-CPA secure encryption algorithm. An important assumption we make is that messages to be authenticated are no longer than a predefined length. This includes applications in which messages are of fixed length that is known a priori, such as RFID systems in which tags need to authenticate their identifiers, sensor nodes reporting events that belong to certain domain or measurements within a certain range and so on. The novelty of the proposed scheme is to utilize the encryption algorithm to deliver a random string and use it to reach the simplicity and efficiency of one-time pad authentication without the need to manage impractically long keys.

3.1 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 [20].

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:

$$\tau \equiv mk_s + r \pmod{p}. \tag{4}$$

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., [47], [48]) and for low-cost sensor nodes (see, e.g., [49], [50]). 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 (4), 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:

$$\tau \overset{?}{\equiv} mk_s + r \pmod{p}. \tag{5}$$

If the integrity check of (5) 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.

3.2 Performance Discussion

There are three classes of standard message authentication codes 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 [23], 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 (4) of the proposed MAC (in fact, the proposed MAC of (4) 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 result of (4) 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 (4) 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 functions SHA-256 and SHA-512 run in about 23.73 cycles/byte and 40.18 cycles/byte, respectively [51], the modular multiplication of (4) runs in about 1.5 cycles/byte [20], which illustrates the significance of removing the cryptographic phase from our MAC.

Another significant advantage of the proposed method, especially for low-power devices, is hardware efficiency. The hardware required to perform modular multiplication is less than the hardware required to perform sophisticated cryptographic operations. As a result, energy consumption is in turn reduced. For instance, while cryptographic hash functions consume 20-30 μJ/bit [52], modular multiplication can consume as low as 0.02 μJ/bit [53].

It remains to compare the proposed scheme with single-pass authenticated encryption primitives. However, since all secure authenticated encryption primitives are block cipher based,[2] while the scheme proposed here can be used alongside stream ciphers, we delay the comparison till Section 5, where we describe a more efficient authentication scheme assuming the encryption is block cipher based.

3.3 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.

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.

3.4 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.

3.4.1 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,

$$\mathrm{Adv}_{\mathcal{E}}^{\mathrm{ind-cpa}}(\mathcal{B}) = \Pr[b' = b]. \tag{6}$$

As stated in (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 3.1. Let $\mathcal{A}$ be an adversary against the privacy of Σ and let $\mathrm{Adv}_{\Sigma}^{\mathrm{priv}}(\mathcal{A})$ 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 3.1 using $\mathcal{E}$ as the underlying encryption algorithm. Then given an adversary, $\mathcal{A}$, against the privacy of Σ, one can construct an adversary, $\mathcal{B}$, against $\mathcal{E}$ such that*

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

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 (4) 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 (4) 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

[2] Although stream cipher based authenticated encryption primitives have appeared in [54] and [55], such proposals have been analyzed and shown to be vulnerable to differential attacks [56], [57].

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 3.1. Let $\mathcal{B}$ be an adversary with access oracle to the encryption algorithm $\mathcal{E}$ and let adversary $\mathcal{A}$ use adversary $\mathcal{B}$ to attack the privacy of observed ciphertexts. Then,

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

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.

3.4.2 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 3.1 and let $\mathrm{Adv}_{\Sigma}^{\mathrm{auth}}(\mathcal{A})$ denotes adversary's $\mathcal{A}$ advantage of successful forgery against Σ.

Theorem 2. *Let Σ denotes the proposed authenticated encryption composition of Section 3.1 in which the authentication tag is computed over 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*

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

Theorem 2 states that if the adversary's advantage in breaking the IND-CPA security of the underlying encryption algorithm is negligible, then so is her advantage in breaking the integrity of the scheme. That is, the integrity of the scheme of Section 3.1 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

$$\mathrm{Seq} = \{(m_1, \tau_1), \ldots, (m_q, \tau_q)\}, \tag{7}$$

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 $i \in \{1, \ldots, q\}$, since otherwise the adversary does not win by definition.

Let $m \equiv m_i + \epsilon \pmod p$ for any $i \in \{1, \ldots, q\}$, 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,

$$\tau \equiv m k_s + r \pmod p, \tag{8}$$

$$\equiv (m_i + \epsilon) k_s + (r_i + \delta) \pmod p, \tag{9}$$

$$\equiv \tau_i + \epsilon k_s + \delta \pmod p. \tag{10}$$

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 (4) is a well-studied example of a strongly universal hash family (see [58] for a definition of strongly universal hash families and detailed discussion showing that (4) is indeed strongly universal hash family). The only difference is that we restrict k_s to belong to the multiplicative group modulo p, whereas it can be equal to zero in unconditionally secure authentication. This is because, in unconditionally secure authentication, the keys can only be used once. In our technique, since k_s can be used to authenticate an arbitrary number of messages, it cannot be chosen to be zero. Otherwise, mk_s will always be zero and the system will not work. The novelty of our approach is to utilize the encryption primitive to reach the simplicity of unconditionally secure authentication, without the need for impractically long keys.

Note also that, unless further assumptions about the encryption algorithm is assumed (such as the pseudorandom permutation property as in Section 5), it is critical for the security of authentication to perform the multiplication modulo a prime integer. That is, it was shown in [4] that the security of authentication based on universal hash families similar to the one in (4) 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 [4].

It is also important to note that, although we do assume that message confidentiality is preserved, using an encryption algorithm, knowing the message does not lead to breaking the integrity of the proposed algorithms. As can be seen in the proof of Theorem 2, message integrity is proven to be secure even though the adversary is given the ability to launch chosen message attacks.

3.4.3 Security of the Authenticated Encryption Composition

In [59], 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

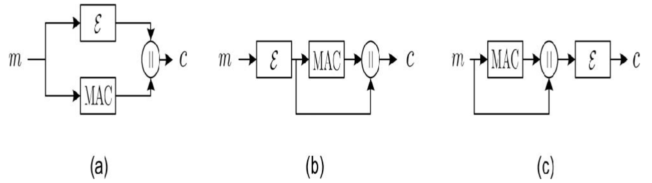

Fig. 1. A schematic of the three generic compositions: (a) encrypt-and-authenticate, (b) encrypt-then-authenticate (EtA), and (c) authenticate-then-encrypt.

message goes to the encryption algorithm as an input, and the same plaintext message goes to the authentication algorithm as an input. Fig. 1 illustrates the differences between the three methods for generically composing an authenticated encryption system.

It was shown in [59] 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 [59]). 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 [59] 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 [59]. 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 [60]). 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 [59], 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 $c = (s, PRF_x(s) \oplus (m \| r))$ 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, $c' = (s, PRF_x(s) \oplus (m\|k) \oplus s') = (s, PRF_x(s) \oplus (m\|k \oplus s'))$. 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

$$\tau \equiv mk_s + rk'_s \pmod p. \tag{11}$$

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

$$\sigma = mk_s \pmod p, \tag{12}$$

and transmitting an encrypted version of the result of (12) as the authentication tag. That is, since r is the main reason for the successful forgery illustrated above, instead of authenticating r as in (11), it is removed from the equation. However, since r was necessary for the privacy of the scheme of Section 3.1, it is required to encrypt the result of (12) 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 E&A composition of Section 3.1.

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

The proof that this modified scheme provides data privacy can be found in [59]. 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 [59]. 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 (12), 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. That is, for a successful forgery, the adversary must predict the correct ciphertext corresponding to σ. Recall, however, that by the definition of IND-CPA security, the adversary's chance of predicting the correct ciphertext is negligible. 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,

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

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

5 Encrypting with Pseudorandom Permutations (Block Ciphers)

In this section, we describe a message authentication approach that is faster than the one described in previous sections. The main idea of this approach is that the input-output relation of the used encryption operation can be realized as a pseudorandom permutation. In what follows, we will show how to utilize the pseudorandomness of block ciphers in a novel way to further improve the efficiency of the authentication algorithm of Section 3.

5.1 The Proposed System

Let $\mathcal{F} : \{0,1\}^N \rightarrow \{0,1\}^N$ be the function representing the bock cipher. We assume that $\mathcal{F}$ acts as a strong pseudorandom permutation, a typical assumption modern block ciphers are believed to satisfy [60]. Assume further that exchanged messages are N-bit long.

5.1.1 Message Encryption

Let m be a short message that is to be transmitted to the intended receiver in a confidential manner. For every message to be transmitted, a random nonce $r \in \mathbb{Z}_{2^N}$ is chosen. (We overload m to denote both the binary string representing the message, and the integer representation of the message as an element of $\mathbb{Z}_{2^N}$; the same applies to r. The distinction between the two representations will be omitted when it is clear from the context.)

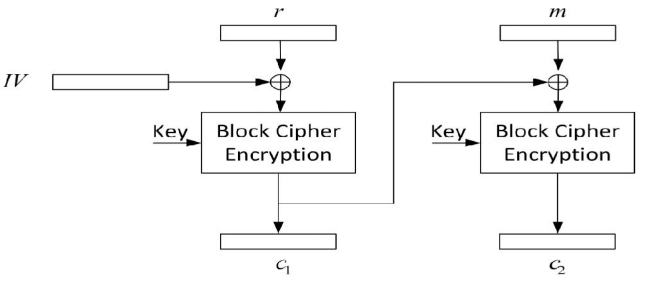

Fig. 2. The cipher block chaining mode of encryption used for message encryption. The random number, r, is treated as the first block of the plaintext.

Now, the concatenation of r and m goes to the encryption algorithm, call it $\mathcal{E}$, as an input. Ideally, we may desire $\mathcal{E}$ to be a strong pseudorandom permutation; however, since N can be sufficiently long (e.g., 128 or larger), constructing a block cipher that maps $2N$-bit strings to $2N$-bit strings can be expensive. Therefore, we resort to the well-studied cipher block chaining (CBC) mode of operation to construct $\mathcal{E}$ from $\mathcal{F}$, as illustrated in Fig. 2.[3]

Consider the CBC mode of operation depicted in Fig. 2. The nonce r is treated as the first plaintext block and is XORed with the initialization vector (IV) to insure IND-CPA security. The first ciphertext block,

$$c_1 = \mathcal{F}_{k_{\mathcal{E}}}(IV \oplus r), \tag{13}$$

is then XORed with the second plaintext block, m in our construction, to produce the second ciphertext block,

$$c_2 = \mathcal{F}_{k_{\mathcal{E}}}(c_1 \oplus m), \tag{14}$$

where $k_{\mathcal{E}}$ is the key corresponding to the block cipher. The resulting

$$c = \mathcal{E}(r, m) = IV \| c_1 \| c_2 \tag{15}$$

is then transmitted to the intended receiver as the ciphertext.

5.1.2 Message Authentication

With the encryption described above, authentication becomes simpler than the ones in previous sections; the authentication tag of message m is calculated as follows:

$$\tau \equiv m + r \ (\mathrm{mod}\ 2^N). \tag{16}$$

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

$$\tau \overset{?}{\equiv} m + r \ (\mathrm{mod}\ 2^N). \tag{17}$$

[3] Although other modes of operations can be used, we restrict our attention to the CBC mode for two reasons. First, the CBC mode of operation is sufficient to illustrate the main ideas of our construction. Second, it is a reasonable mode of operation for low-cost devices that are unable to perform parallel computing.

If the integrity check of (17) is satisfied, the message is considered authentic. Otherwise, the integrity of the message is denied.

5.2 Performance Discussion

First, we compare the scheme of this section to the scheme of Section 3, and then compare it to single-pass schemes. Assuming devices are already equipped with a secure block cipher to encrypt messages, the authentication technique of this section requires only one modular addition. While addition is performed in $O(n)$ time, the fastest integer multiplication algorithms typically require $O(n \ \log n \ \log\log n)$ time [61].[4] Therefore, as efficient as the scheme proposed in Section 3, the authentication technique of this section is at least $O(\log n \ \log\log n)$ faster.

Complexity analysis, however, can be inaccurate by absorbing large constants. This is indeed the case in comparing the basic scheme of Section 3 to the scheme of this section. For $n = 32$, the simple addition of this scheme runs in about 0.02 cycles/byte[5] as opposed to the 1.5 cycles/byte of the previous scheme. The reason that the improvement is better than $O(\log n \ \log\log n)$ is mainly due to the modular reduction. That is, while reduction modulo a prime integer is a nontrivial operation, reduction modulo 2^n can be performed by simply stopping at the nth bit.

To give performance comparisons with authenticated encryption primitives, we focus on two of the prominent single-pass authenticated encryption schemes, the IAPM of Jutla [62] and the OCB of Rogaway et al. [63]. Both IAPM and OCB require preprocessing (whitening) plaintext blocks before block cipher encryption. For instance, IAPM requires XORing plaintext blocks with pairwise differentially uniform sequences named s_i. Each s_i is generated by performing modular multiplication over the finite field $\mathbb{Z}_p$, similar to the multiplication of our first scheme of Section 3. In the OCB mode of operation, each message block, $M[i]$, is XORed with a string the authors denoted as $Z[i]$. The computation of each $Z[i]$ requires the generation of a Gray code γ_i (in which each γ_i and γ_{i+1} have a Hamming distance of one), multiply two polynomials over GF(2^n), and then take the reminder after dividing the multiplication result by a fixed irreducible polynomial. In the proposed scheme, plaintext blocks go to the block cipher without any preprocessing. To the best of our knowledge, the scheme proposed here is the first scheme that does not require multiplication operations either before block cipher encryption, such as single-pass authenticated encryption primitives, or after block cipher encryption, such as generically composed authenticated encryption systems.

Both IAPM and OCB also require the encryption of a nonce as the first block cipher call, which is similar to the first block cipher call in our scheme. Now, after the whitened plaintext blocks are encrypted, a check-sum of the resulting ciphertext blocks is computed and then an additional block cipher call is needed to encrypt the resulting check-sum. That is, in addition to block cipher calls required for encrypting the plaintext itself, both IAPM and OCB require two additional block cipher calls, as opposed to a single additional block cipher call in the proposed scheme. Therefore, in scenarios in which plaintext messages occupy only a single block, an extra block cipher call will contribute significantly to the total power consumption of the scheme. Note further that there is an extra saving due to the elimination of the plaintext whitening procedure.

Before we give formal security analysis of the proposed technique, we give a formal security model that will be used for the analysis.

5.3 Security Model

Recall that, to model the security of a message authentication scheme in the standard setup, a probabilistic polynomial time adversary, $\mathcal{A}$, is given oracle access to the signing and verifying algorithms, and challenged to generate a new *massage-tag* pair that will be accepted as valid, for a tag that has not been attached to the message by the signing oracle. Observe, however, that the message to be authenticated in our setup must also be encrypted. That is, what the intended user receives is a *ciphertext-tag* pair, as opposed to *plaintext-tag* pair in the standard model. This implies that the adversary must come up with a valid ciphertext-tag pair for a successful forgery. In what follows, we modify the standard model of Section 2 to address the difference between standard MACs and our MAC in which the message must be encrypted.

Let $\mathcal{E}$ be the underlying encryption algorithm. (We treat $\mathcal{E}$ as a black box that takes a plaintext message as an input and outputs its corresponding ciphertext.) The signing oracle internally calls the encryption algorithm and outputs a ciphertext-tag pair. That is, given an encryption algorithm $\mathcal{E}$, on input a key k and a message m, the signing algorithm $\mathcal{S}_{\mathcal{E}}(k, m)$ outputs (c, τ), where c is the ciphertext corresponding to m and τ is its authentication tag.[6]

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

As in the standard model, an adversary is a probabilistic polynomial time algorithm, $\mathcal{A}$. The adversary is given oracle access to algorithms $\mathcal{S}_{\mathcal{E}}(k, \cdot)$ and $\mathcal{V}_{\mathcal{D}}(k, \cdot, \cdot)$ for a random but hidden choice of k. $\mathcal{A}$ can query $\mathcal{S}_{\mathcal{E}}$ to generate a ciphertext-tag pair for a plaintext of her choice and ask the verifier $\mathcal{V}_{\mathcal{D}}$ to verify that (c, τ) is a valid ciphertext-tag pair. Formally, $\mathcal{A}$'s attack on the scheme is described by the following experiment:

1. A random string is selected as the shared secret, k.
2. Suppose $\mathcal{A}$ makes a signing query m. The oracle computes $(c, \tau) \leftarrow \mathcal{S}_{\mathcal{E}}(k, m)$, the ciphertext-tag pair, and returns it to $\mathcal{A}$. (Since $\mathcal{S}$ is typically probabilistic,

[4] A recent FFT-based algorithm reduced the complexity of integer multiplication to $n \log n 2^{O(\log^* n)}$ on a two-tape turing machine [61].

[5] Codes are written in the C programming language using a machine with 3.00-GHz Intel(R) Xeon(TM) 64-bit CPU running on UNIX operating system.

[6] For convenience, we write $\mathcal{E}$ as a subscript.

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 (c, τ). The oracle computes the decision $d = \mathcal{V}_{\mathcal{D}}(k, c, \tau)$ and returns it to $\mathcal{A}$.

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

The outcome of running the experiment in the presence of an adversary is used to define security. We say that $\mathcal{A}$ is successful if it makes a verify query (c, τ) which is accepted, for a (c, τ) that has not been outputted by the signing oracle $\mathcal{S}_{\mathcal{E}}$.

5.4 Security Analysis

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

5.4.1 Data Privacy

Recall that two pieces of information are transmitted to the intended receiver (the ciphertext and the authentication tag), both of which are functions of the private plaintext message. Now, when it comes to the authentication tag, observe that the nonce r serves as a one-time key (similar to the role r plays in the construction of Section 3). The formal analysis that the authentication tag does not compromise message privacy is the same as the one provided in Section 3.4.1 and, thus, is omitted.

The ciphertext of (13), on the other hand, is a standard CBC encryption and its security is well studied; thus, we give the theorem statement below without a formal proof (interested readers may refer to textbooks in cryptography, for example, [58], [60], [64] for more details). Let the privacy of the system be modeled as its indistinguishability from a pseudorandom permutation, one gets the following.

Theorem 4. *Let $\mathcal{E}$ be the encryption algorithm of Fig. 2 and let $\mathcal{F}$ be the block cipher used to construct $\mathcal{E}$. Then given an adversary $\mathcal{A}$ against the privacy of $\mathcal{E}$, one can construct an adversary $\mathcal{B}$ against the pseudorandomness of $\mathcal{F}$ such that*

$$\mathrm{Adv}_{\mathcal{E}}^{\mathrm{priv}}(\mathcal{A}) \leq \mathrm{Adv}_{\mathcal{F}}^{\mathrm{prp}}(\mathcal{B}).$$

Furthermore, the experiment for $\mathcal{B}$ takes the same time as the experiment for $\mathcal{A}$ and, if $\mathcal{A}$ makes at most q_e oracle queries, then $\mathcal{B}$ makes at most $2q_e$ oracle queries.

Theorem 4 states that an adversary breaking the privacy of the encryption algorithm of Fig. 2 is also able to break the pseudorandomness of the underlying block cipher. Therefore, the adversary's advantage of breaking the privacy of the encryption algorithm is negligible, provided the use of a secure block cipher.

5.4.2 Data Authenticity

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

MACs in an our authenticated encryption composition, on the other hand, are fundamentally different than standard MACs. The intended receiver in an authenticated encryption system receives a ciphertext-tag pair as opposed to message-tag pair. This implies that, for an attempted forgery to be successful, the adversary must come up with a *ciphertext-tag* pair that will be accepted as valid, not a *message-tag* pair. (Note, however, that we do not hide messages from the adversary. In fact, we assume the adversary's ability to launch chosen message attacks, as can be seen in the security model and the formal proof below.)

Remark 3. We emphasize that this security model can also be used for the analyses of previous sections (since it is also the case that the intended user receives a ciphertext-tag pair). The reason for not using this security model in previous sections is that the security can be proven using the standard model. For the technique proposed in this section, however, security cannot be proven without the modified model (as can be seen in the proof below).

Following the standard convention in cryptography, we give below information-theoretic bound on the adversary's probability of successful forgery assuming the block cipher is a true random permutation (the complexity-theoretic analogy is given after the theorem). Let Σ denote the proposed composition of Section 5.1 and let $\mathrm{Adv}_{\Sigma}^{\mathrm{auth}}(\mathcal{A})$ denote adversary's $\mathcal{A}$ advantage of successful forgery against Σ.

Theorem 5. *Let $\mathcal{F} : \{0,1\}^N \rightarrow \{0,1\}^N$ be a true random permutation used to construct an encryption algorithm, $\mathcal{E}$, in the cipher block chaining mode, as depicted in Fig. 2. Let Σ denote the proposed composition of Section 5.1 and let $\mathcal{A}$ be an adversary calling the signing oracle q times before making a forgery attempt. Then,*

$$\mathrm{Adv}_{\Sigma}^{\mathrm{auth}}(\mathcal{A}) \leq 2^{1-N}.$$

To pass a complexity-theoretic analog of Theorem 5, one will need access to an $\mathcal{F}^{-1}$ oracle to verify a forgery attempt, which translates into needing the strong pseudorandom permutation assumption. One gets the following. Fix a block cipher $\mathcal{F} : \mathcal{K} \times \{0,1\}^N \rightarrow \{0,1\}^N$ that is used to construct the mode of encryption of Fig. 2. Let $\mathcal{A}$ be an adversary that asks q signing queries then attempting its forgery. Then, there is an adversary $\mathcal{B}$ attacking the block cipher in which

$$\mathrm{Adv}_{\Sigma}^{\mathrm{auth}}(\mathcal{A}) \leq \mathrm{Adv}_{\mathcal{F}}^{\mathrm{sprp}}(\mathcal{B}) + 2^{1-N}, \tag{18}$$

where $\mathrm{Adv}_{\mathcal{F}}^{\mathrm{sprp}}(\mathcal{B})$ is as defined in (3). Furthermore, adversary $\mathcal{B}$ takes the same time adversary $\mathcal{A}$ takes, minus the time of generating the coin tosses and the generation and authentication of tags, and makes at most $2(q+1)$ oracle queries.

Equation (18) implies that if the adversary's advantage in breaking the sprp security of the underlying block cipher is negligible, then so is her advantage in breaking the integrity of the scheme. That is, the integrity of the scheme of Section 5.1 is provably secure provided the underlying block cipher is sprp secure.

Proof (Proof of Theorem 5). When $q = 0$ it is rather straightforward. It follows directly from the fact that each value of the authentication tag is equally probable (by Lemma 1).

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

$$\mathrm{Seq} = \{(m_1, c_1, \tau_1), \ldots, (m_q, c_q, \tau_q)\}. \tag{19}$$

$\mathcal{A}$ then calls the verify oracle with (c, τ), where $(c, \tau) \neq (c_i, \tau_i)$ for any $i = 1, \ldots, q$ since otherwise $\mathcal{A}$ does not win by definition. We aim to bound the probability that (c, τ) will be validated. Let r and m be the nonce and the message corresponding to the decryption of c, respectively. There are two possible strategies for forgery:

1. attempt to forge a valid ciphertext-tag pair corresponding to a specific plaintext of $\mathcal{A}$'s choice,
2. attempt to authenticate a ciphertext-tag pair regardless of their corresponding plaintext (i.e., modify a recorded ciphertext-tag pair in a way undetected by the legitimate receiver).

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

To bound the probability of $\mathrm{forgery}_1$, assume $\mathcal{A}$ attempts to falsely authenticate a plaintext $r \| m \neq r_i \| m_i$ for any $i = 1, \ldots, q$. If $r \neq r_i$, the adversary must predict the two ciphertext blocks and the probability of successful forgery is 2^{-2N} (since $\mathcal{F}$ is a true random permutation). If $r = r_i$, the adversary must predict the ciphertext block corresponding to m, which is equal to 2^{-N}. Therefore, the probability of $\mathrm{forgery}_1$ is at most 2^{-N}.

To bound the probability of $\mathrm{forgery}_2$, denote by $\mathrm{Collision}$ the event that $m + r \equiv m_i + r_i \pmod{2^N}$ for some $i \in \{1, \ldots, q\}$. That is, the tag corresponding to the modified ciphertext, τ, collides with τ_i, one of the recorded tags in the sequence of (19). Also, we use $\overline{\mathrm{Collision}}$ as the typical notation for the complement of $\mathrm{Collision}$.

Obviously, when the event $\mathrm{Collision}$ occurs, $(c, \tau_i) \neq (c_i, \tau_i)$ will pass the integrity check, leading to successful forgery. Recall, however, that $\mathcal{F}$ is a true random permutation; hence, m and r, the message and the nonce corresponding to c, cannot be correlated to m_i and r_i, the plaintext and the nonce corresponding to c_i. That is, from the adversary's standpoint, m and r are random elements of $\mathbb{Z}_{2^N}$. Therefore, the probability that the plaintext-nonce pair corresponding to c (the modified version of c_i), will result in a τ that collides with τ_i is

$$\Pr[\mathrm{Collision}] = \Pr\left[m + r \equiv m_i + r_i 2^N\right] \tag{20}$$

$$\leq 2^{-N}. \tag{21}$$

Assume now that the event $\overline{\mathrm{Collision}}$ is true. If no collision has occurred, then the adversary's probability of successful forgery is bounded by the probability of predicting the plaintext message corresponding to c. That is, similar to the probability of $\mathrm{forgery}_1$,

$$\Pr[\mathrm{forgery}_2 | \overline{\mathrm{Collision}}] \leq 2^{-N}. \tag{22}$$

By (21) and (22), it follows that the probability of $\mathrm{forgery}_2$ can be bounded by

$$\begin{aligned}\Pr[\mathrm{forgery}_2] = {} & \Pr[\mathrm{forgery}_2 | \mathrm{Collision}] \cdot \Pr[\mathrm{Collision}] \\ & + \Pr[\mathrm{forgery}_2 | \overline{\mathrm{Collision}}] \cdot \Pr[\overline{\mathrm{Collision}}]\end{aligned} \tag{23}$$

$$\leq \Pr[\mathrm{Collision}] + \Pr[\mathrm{forgery}_2 | \overline{\mathrm{Collision}}] \tag{24}$$

$$\leq 2^{-N} + 2^{-N}. \tag{25}$$

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

5.4.3 *Security of the Authenticated Encryption Composition*

The same discussion of Section 3.4.3 applies here.

6 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. When devices are equipped with block ciphers to encrypt messages, a second technique that utilizes the fact that block ciphers can be modeled as strong pseudorandom permutations is proposed to authenticate messages using a single modular addition. The proposed schemes are shown to be orders of magnitude faster, and consume orders of magnitude less energy than traditional MAC algorithms. Therefore, they are more suitable to be used in computationally constrained mobile and pervasive devices.

Acknowledgments

A preliminary version of this paper appeared in the 12th International Conference on Information and Communications Security-ICICS 2010 [65].

References

[1] L. Carter and M. Wegman, "Universal Hash Functions," *J. Computer and System Sciences,* vol. 18, no. 2, pp. 143-154, 1979.

[2] T. Helleseth and T. Johansson, "Universal Hash Functions from Exponential Sums over Finite Fields and Galois Rings," *Proc. 16th Ann. Int'l Cryptology Conf. Advances in Cryptology (CRYPTO '96),* pp. 31-44, 1996.

[3] V. Shoup, "On Fast and Provably Secure Message Authentication Based on Universal Hashing," *Proc. 16th Ann. Int'l Cryptology Conf. Advances in Cryptology (CRYPTO '96),* pp. 313-328, 1996.

[4] B. Alomair, A. Clark, and R. Poovendran, "The Power of Primes: Security of Authentication Based on a Universal Hash-Function Family," *J. Math. Cryptology,* vol. 4, no. 2, 2010.

[5] B. Alomair and R. Poovendran, "E-MACs: Towards More Secure and More Efficient Constructions of Secure Channels," *IEEE Trans. Computers,* 2012.

[6] *Federal Information Processing Standards (FIPS) Publication 113, Computer Data Authentication,* FIPS, 1985.

[7] *ISO/IEC 9797-1:1999 Standard, Information Technology - Security Techniques - Message Authentication Codes (MACs) - Part 1: Mechanisms Using a Block Cipher,* ISO/IEC, 1999.

[8] M. Dworkin, "Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication," 2005.

[9] T. Iwata and K. Kurosawa, "OMAC: One-Key CBC MAC," *Proc. Int'l Conf. Fast Software Encryption (FSE '03),* pp. 129-153, 2003.

[10] M. Bellare, R. Guerin, and P. Rogaway, "XOR MACs: New Methods for Message Authentication Using Finite Pseudorandom Functions," *Proc. 15th Ann. Int'l Cryptology Conf. Advances in Cryptology (CRYPTO '95),* pp. 15-28, 1995.

[11] P. Rogaway and J. Black, "PMAC," *Proposal to NIST for a Parallelizable Message Authentication Code,* 2001.

[12] M. Bellare, J. Kilian, and P. Rogaway, "The Security of the Cipher Block Chaining Message Authentication Code," *J. Computer and System Sciences,* vol. 61, no. 3, pp. 362-399, 2000.

[13] B. Preneel and P. Van Oorschot, "On the Security of Iterated Message Authentication Codes," *IEEE Trans. Information Theory,* vol. 45, no. 1, pp. 188-199, Jan. 1999.

[14] G. Tsudik, "Message Authentication with One-Way Hash Functions," *ACM SIGCOMM Computer Comm. Rev.,* vol. 22, no. 5, pp. 29-38, 1992.

[15] M. Bellare, R. Canetti, and H. Krawczyk, "Keying Hash Functions for Message Authentication," *Proc. 16th Ann. Int'l Cryptology Conf. Advances in Cryptology (CRYPTO '96),* pp. 1-15, 1996.

[16] *Federal Information Processing Standards (FIPS) Publication 198, The Keyed-Hash Message Authentication Code (HMAC),* FIPS, 2002.

[17] B. Preneel and P.V. Oorschot, "MDx-MAC and Building Fast MACs from Hash Functions," *Proc. 15th Ann. Int'l Cryptology Conf. Advances in Cryptology (CRYPTO '95),* vol. 963, pp. 1-14, 1995.

[18] *ISO/IEC 9797-2:2002 Standard, Information Technology - Security Techniques - Message Authentication Codes (MACs) - Part 2: Mechanisms Using a Dedicated Hash-Function,* ISO/IEC 2002.

[19] A. Bosselaers, R. Govaerts, and J. Vandewalle, "Fast Hashing on the Pentium," *Proc. 16th Ann. Int'l Cryptology Conf. Advances in Cryptology (CRYPTO '96),* pp. 298-312, 1996.

[20] J. Black, S. Halevi, H. Krawczyk, T. Krovetz, and P. Rogaway, "UMAC: Fast and Secure Message Authentication," *Proc. 19th Ann. Int'l Cryptology Conf. Advances in Cryptology (CRYPTO '99),* pp. 216-233, 1999.

[21] D. Bernstein, "The Poly1305-AES Message-Authentication Code," *Proc. 12th Int'l Conf. Fast Software Encryption (FSE '05),* pp. 32-49, 2005.

[22] S. Halevi and H. Krawczyk, "MMH: Software Message Authentication in the Gbit/Second Rates," *Proc. Int'l Conf. Fast Software Encryption (FSE '97),* pp. 172-189, 1997.

[23] H. van Tilborg, *Encyclopedia of Cryptography and Security.* Springer, 2005.

[24] T. Krovetz, "UMAC: Fast and Provably Secure Message Authentication," http://fastcrypto.org/umac, 2006.

[25] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks,* vol. 38, no. 4, pp. 393-422, 2002.

[26] A. Perrig, J. Stankovic, and D. Wagner, "Security in Wireless Sensor Networks," *Comm. ACM,* vol. 47, no. 6, pp. 53-57, 2004.

[27] B. Alomair, A. Clark, J. Cuellar, and R. Poovendran, "Toward a Statistical Framework for Source Anonymity in Sensor Networks," *IEEE Trans. Mobile Computing,* vol. 12, no. 2, pp. 248-260, doi:10.1109/TMC.2011.267, Feb. 2013.

[28] *Class-1 Generation-2 UHF Radio Frequency Identification Protocol Standard Version 1.2.0,* EPCglobal, Inc., 2008.

[29] S. Sarma, S. Weis, and D. Engels, "RFID Systems and Security and Privacy Implications," *Proc. Fourth Int'l Workshop Cryptographic Hardware and Embedded Systems (CHES '02),* pp. 1-19, 2003.

[30] A. Juels, "RFID Security and Privacy: A Research Survey," *IEEE J. Selected Areas in Comm.,* vol. 24, no. 2, pp. 381-394, Feb. 2006.

[31] B. Alomair and R. Poovendran, "Privacy versus Scalability in Radio Frequency Identification Systems," *Computer Comm.,* vol. 33, no. 18, pp. 2155-2163, 2010.

[32] B. Alomair, A. Clark, J. Cuellar, and R. Poovendran, "Scalable RFID Systems: A Privacy-Preserving Protocol with Constant-Time Identification," *IEEE Trans. Parallel and Distributed Systems,* vol. 23, no. 8, pp. 1536-1550, Aug. 2012.

[33] K. Venkatasubramanian, A. Banerjee, and S. Gupta, "EKG-Based Key Agreement in Body Sensor Networks," *Proc. IEEE INFOCOM,* pp. 1-6, 2008.

[34] C. Tan, H. Wang, S. Zhong, and Q. Li, "Body Sensor Network Security: An Identity-Based Cryptography Approach," *Proc. First ACM Conf. Wireless Network Security,* pp. 148-153, 2008.

[35] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, "Strong Authentication for RFID Systems Using the AES Algorithm," *Proc. Workshop Cryptographic Hardware and Embedded Systems (CHES '04),* pp. 357-370, 2004.

[36] D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B. Koo, C. Lee, D. Chang, J. Lee, K. Jeong, H. Kim, J. Kim, and S. Chee, "HIGHT: A New Block Cipher Suitable for Low-Resource Device," *Proc. Workshop Cryptographic Hardware and Embedded Systems (CHES '06),* pp. 46-59, 2006.

[37] A. Bogdanov, L. Knudsen, G. Leander, C. Paar, A. Poschmann, M. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT: An Ultra-Lightweight Block Cipher," *Proc. Workshop Cryptographic Hardware and Embedded Systems (CHES '07),* pp. 450-466, 2007.

[38] A. Shamir, "SQUASH-A New MAC with Provable Security Properties for Highly Constrained Devices Such as RFID Tags," *Proc. Int'l Conf. Fast Software Encryption (FSE '08),* pp. 144-157, 2008.

[39] A. Bogdanov, G. Leander, C. Paar, A. Poschmann, M. Robshaw, and Y. Seurin, "Hash Functions and RFID Tags: Mind the Gap," *Proc. 10th Int'l Workshop Cryptographic Hardware and Embedded Systems (CHES '08),* pp. 283-299, 2008.

[40] O. Goldreich, *Foundations of Cryptography.* Cambridge Univ., 2001.

[41] S. Goldwasser and S. Micali, "Probabilistic Encryption," *J. Computer and System Sciences,* vol. 28, no. 2, pp. 270-299, 1984.

[42] T. Kohno, J. Viega, and D. Whiting, "CWC: A High-Performance Conventional Authenticated Encryption Mode," *Proc. Int'l Conf. Fast Software Encryption (FSE '04),* pp. 408-426, 2004.

[43] M. Luby and C. Rackoff, "How to Construct Pseudorandom Permutations from Pseudorandom Functions," *SIAM J. Computing,* vol. 17, pp. 373-386, 1988.

[44] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway, "A Concrete Security Treatment of Symmetric Encryption," *Proc. 38th Ann. Symp. Foundation of Computer Science (FOCS '97),* pp. 394-403, 1997.

[45] J. Daemen and V. Rijmen, *The Design of Rijndael: AES - The Advanced Encryption Standard.* Springer Verlag, 2002.

[46] S. Schwarz, "The Role of Semigroups in the Elementary Theory of Numbers," *Math. Slovaca,* vol. 31, no. 4, pp. 369-395, 1981.

[47] Z. Liu and D. Peng, "True Random Number Generator in RFID Systems against Traceability," *Proc. IEEE Consumer Comm. and Networking Conf. (CCNS '06),* pp. 620-624, 2006.

[48] D. Holcomb, W. Burleson, and K. Fu, "Power-Up SRAM State as an Identifying Fingerprint and Source of True Random Numbers," *IEEE Trans. Computers,* vol. 58, no. 9, pp. 1198-1210, Sept. 2009.

[49] C. Petrie and J. Connelly, "A Noise-Based IC Random Number Generator for Applications in Cryptography," *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications,* vol. 47, no. 5, pp. 615-621, May 2000.

[50] S. Callegari, R. Rovatti, and G. Setti, "Embeddable ADC-Based True Random Number Generator for Cryptographic Applications Exploiting Nonlinear Signal Processing and Chaos," *IEEE Trans. Signal Processing,* vol. 53, no. 2, pp. 793-805, Feb. 2005.

[51] J. Nakajima and M. Matsui, "Performance Analysis and Parallel Implementation of Dedicated Hash Functions," *Proc. Int'l Conf. Theory and Applications of Cryptographic Techniques: Advances in Cryptology (EUROCRYPT),* pp. 165-180, 2002.

[52] B. Preneel, "Using Cryptography Well," http://secappdev.org/handouts/2010/Bart, 2010.

[53] J. Großschädl, R. Avanzi, E. Savaş, and S. Tillich, "Energy-Efficient Software Implementation of Long Integer Modular Arithmetic," *Proc. Seventh Int'l Conf. Cryptographic Hardware and Embedded Systems (CHES '05),* pp. 75-90, 2005.

[54] N. Ferguson, D. Whiting, B. Schneier, J. Kelsey, and T. Kohno, "Helix: Fast Encryption and Authentication in a Single Cryptographic Primitive," *Proc. Int'l Conf. Fast Software Encryption (FSE '03),* pp. 330-346, 2003.

[55] D. Whiting, B. Schneier, S. Lucks, and F. Muller, "Phelix-Fast Encryption and Authentication in a Single Cryptographic Primitive, eSTREAM," ECRYPT Stream Cipher Project, Report 2005/020, http://www.ecrypt.eu.org/stream, 2005.

[56] F. Muller, "Differential Attacks against the Helix Stream Cipher," *Proc. Int'l Conf. Fast Software Encryption (FSE '04),* pp. 94-108, 2004.

[57] H. Wu and B. Preneel, "Differential-Linear Attacks Against the Stream Cipher Phelix," *Proc. Int'l Conf. Fast Software Encryption (FSE '07),* pp. 87-100, 2007.

[58] D. Stinson, *Cryptography: Theory and Practice.* CRC, 2006.

[59] M. Bellare and C. Namprempre, "Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm," *J. Cryptology,* vol. 21, no. 4, pp. 469-491, 2008.

[60] J. Katz and Y. Lindell, *Introduction to Modern Cryptography.* Chapman & Hall/CRC, 2008.

[61] M. Fürer, "Faster Integer Multiplication," *Proc. ACM Symp. Theory of Computing (STOC '07),* p. 66, 2007.

[62] C. Jutla, "Encryption Modes with Almost Free Message Integrity," *J. Cryptology,* vol. 21, no. 4, pp. 547-578, 2008.

[63] P. Rogaway, M. Bellare, and J. Black, "OCB: A Block-Cipher Mode of Operation for Efficient Authenticated Encryption," *ACM Trans. Information and System Security,* vol. 6, no. 3, pp. 365-403, 2003.

[64] A. Menezes, P.V. Oorschot, and S. Vanstone, *Handperiodical of Applied Cryptography.* CRC, 1997.

[65] B. Alomair and R. Poovendran, "Efficient Authentication for Mobile and Pervasive Computing," *Proc. 12th Int'l Conf. Information and Comm. Security (ICICS '10),* 2010.

Basel Alomair received the bachelor's, master's, and PhD degrees from King Saud University, Riyadh, Saudi Arabia; University of Wisconsin, Madison; and University of Washington, Seattle, respectively. He is an assistant professor and founding director of the Center for Cybersecurity (C4C) at King Abdulaziz City for Science and Technology (KACST), an affiliate professor in the Electrical Engineering Department at the University of Washington-Seattle, and an information security consultant for the Technology Control Company (TCC). He was recognized by the IEEE Technical Committee on Fault-Tolerant Computing (TC-FTC) and the IFIP Working Group on Dependable Computing and Fault Tolerance (WG 10.4) through the 2010 IEEE/IFIP William C. Carter Award for his significant contributions in the area of dependable computing. His network security and cryptology research was recognized by the 2011 Outstanding Research Award from the University of Washington. He was also the recipient of the 2012 Distinguished Dissertation Award from the Center for Information Assurance and Cybersecurity (CIAC) for his fundamental contributions in the area of securing energy-constrained RFID systems. His main research interests are in information security and applied cryptography. He is a member of the IEEE.

Radha Poovendran is a professor and founding director of the Network Security Lab in the Department of Electrical Engineering (EE) at the University of Washington (UW). He has received the NSA Rising Star Award (1999) and Faculty Early Career Awards including the National Science Foundation CAREER (2001), ARO YIP (2002), ONR YIP (2004), and PECASE (2005) for his research contributions to multiuser, wireless security. He has received the Outstanding Teaching Award and Outstanding Research Advisor Award from UW EE (2002), Graduate Mentor Award from Office of the Chancellor at University of California San Diego (2006), and Pride@-Boeing award (2009). He has coauthored papers recognized with IEEE PIMRC Best Paper Award (2007), IEEE&IFIP William C. Carter Award (2010) and AIAA/IEEE Digital Avionics Systems best session paper award (2010). He has coedited a periodical titled Secure Localization and Time Synchronization in Wireless Ad Hoc and Sensor Networks and has served as a coguest editor for an *IEEE JSAC special issue on wireless ad hoc networks security*. He has cochaired many conferences and workshops including the first ACM Conference on Wireless Network Security (WiSec) in 2008 and NITRD-NSF National workshop on high-confidence transportation cyber-physical systems in 2009, trustworthy aviation information systems at the 2010 and 2011 AIAA Infotech@Aerospace and 2011 IEEE Aerospace. He is chief editor for the forthcoming *Proc. IEEE special issue on cyber-physical systems*. He was a Kavli fellow of the National Academy of Sciences (2007) and is a senior member of the IEEE.