Authenticated Encryption: How Reordering Can Impact Performance*

Basel Alomair

Computer Research Institute (CRI)
King Abdulaziz City for Science and Technology (KACST)
alomair@kacst.edu.sa

Abstract. In this work, we look at authenticated encryption schemes from a new perspective. As opposed to analyzing the *security* of different methods of constructing authenticated encryption schemes, we investigate the effect of the method used to construct an authenticated encryption scheme on the *performance* of the construction. We show that, by performing the authentication operation before the encryption operation, the security requirements on the authentication operation can be relaxed, leading to more efficient constructions, without affecting the security of the overall construction.

Keywords: Universal hash-function families, pseudorandom permutations, authenticated encryption, provable security.

1 Introduction

There are three different methods to generically compose an authenticated encryption scheme by combining an encryption algorithm with a MAC algorithm: Encrypt-and-MAC ($E\&M$), Encrypt-then-MAC (EtM), or MAC-then-Encrypt (MtE). Although significant efforts have been devoted to analyzing the security implications of different generic compositions (see, e.g., [10, 29]), little effort has been devoted to the study of the performance implications of different generic compositions [2]. Of particular interest to this work is the performance aspect of generic compositions when the encryption algorithm is block cipher based and the MAC algorithm is universal hash-function family based. (We focus on such constructions since block ciphers are the recommended building block for secure encryption [25] and since universal hash families based MACs are the fastest method for message authentication [39].)

In a typical EtM composition, the plaintext is broken into blocks. Each block is processed with a block cipher, resulting in a ciphertext block. The resulting ciphertext blocks are then authenticated using a MAC based on a universal hash-function family (in the Carter-Wegman style [15]). One of the most recent authenticated encryption schemes is the Carter-Wegman Counter (CWC) block cipher mode of authenticated encryption proposed by Kohno et al. in [27]. (The

* A more complete version of this paper can be found in [3].

F. Bao, P. Samarati, and J. Zhou (Eds.): ACNS 2012, LNCS 7341, pp. 84–99, 2012.

National Institute of Standards and Technology (NIST) has adopted the CWC mode of operation in the standardized Galois/Counter Mode (GCM) of authenticated encryption [17].) The CWC mode of operation gives high-performance authenticated encryption by combining the counter mode of encryption with a Wegman-Carter universal hash-function family for authentication.

The OKH Solution. In this work, we investigate the performance implications of the order in which the two operations, encryption and authentication, are performed. We describe the Odd Key Hashing (OKH) mode of authenticated encryption. The OKH mode is motivated by the CWC mode of authenticated encryption proposed by Kohno et al. [27]. However, unlike the CWC and the GCM schemes, the order of encrypt-then-authenticate is reversed in the OKH mode. That is, as opposed to applying the hashing operation on the ciphertext, it is applied on the plaintext, before block cipher encryption. The main result of this study is to show that, while the hash family used to construct a MAC in the EtM composition must be universal, this need not be the case in the MtE composition.[1] The performance implication of this result is that, since the hash family need not be universal, it can be computed faster than the fastest universal hash family in the cryptographic literature. The theoretical significance of this result is that relaxing the security requirements on the MAC algorithm does not affect the provable security of the overall authenticated encryption composition.

Background and Related Work. The notion of authenticated encryption was introduced independently by Katz and Yung in [26], and by Bellare and Rogaway in [11]. Since then, many authenticated encryption schemes have been proposed, such as, RPC of Katz and Yung [26], XECB of Gligor and Donescu [20], IAPM of Jutla [24], OCB of Rogaway et al. [37], EAX of Bellare et al. [12], and CWC of Kohno et al. [27]. Alomair and Poovendran showed that one can utilize the $E\&M$ composition to eliminate redundant computations in the MAC algorithm in order to come up with more efficient generic constructions [5, 6]. Stream cipher based authenticated encryption primitives have appeared in [19, 40]. However, these stream cipher based proposals have been analyzed and shown to be vulnerable to attacks [31, 33, 34, 41].

The use of universal hash-function families to construct MAC algorithms is due to Carter and Wegman [15]. Compared to block cipher based MACs, such as [9,16], and cryptographic hash function based MACs, such as [7,35], universal hashing based MACs lead to faster message authentication [14, 21, 28, 36]. The speed of a universal hash family based MAC relies mainly on the speed of the used universal hash family. The security of MACs based on universal hashing has been extensively studied. In [22], key recovery attacks against universal hash functions was introduced. In [4], it was shown that the security of universal hashing based on integer arithmetic is proportional to the smallest prime factor of the used modulus.

[1] Although the same result can be shown for the $E\&M$ composition, we restrict our discussion to the MtE composition for brevity.

There has been a significant effort for the design of fast universal hash families. In [28], Krawczyk introduced the cryptographic CRC which hashes in about 6 cycles/byte, as shown by Shoup in [38]. In [36], Rogaway proposed the bucket hashing which was the first hash family explicitly targeted for fast software implementation; it runs in about $1.5 - 2.5$ cycles/byte [14]. In [23], Johansson described bucket hashing with smaller key size. In [21], Halevi and Krawczyk proposed the MMH family, which hashes at about $1.2 - 3$ cycles/byte. In [18], Etzel et al. proposed the square hash, an MMH-variant that can be more efficient than MMH in certain settings [14]. In [13], Bernstein proposed floating-point arithmetic based hash function that achieves a peak speed of 2.4 cycles/byte. In [1], Afanassiev et al. described an application of hashing based on polynomial evaluation over finite fields. In [32], Nevelsteen and Preneel study the performance of several universal hash functions proposed for MACs. The speed champion of universal hash functions directed for software implementation is the NH family of Black et al. proposed in [14]. The NH family is an extension to the MMH family of [21]. The speed improvement comes from eliminating the non-trivial modular reduction required by the MMH family. The novelty of NH family is that it uses arithmetic modulo powers of two, i.e., "computations that computers like to do [14]." The NH family hashes at about 0.34 cycles/byte for 2^{-32} probability of message collision.

Organization. In Section 2, we give some preliminaries. In Section 3, we describe the OK hash family. In Section 4, we describe the construction of the proposed OKH authenticated encryption scheme. In Section 5, we state and prove the authenticity and privacy theorems of the proposed scheme. In Section 6, we summarize the basic ideas behind the proposed mode of operation and provide performance comparisons. In Section 7, we conclude the paper.

2 Notations and Preliminaries

2.1 Notations

- If s is a binary string, $|s|$ denotes the length of s in bits.
- For a positive integer β, $\{0,1\}^\beta$ denotes a binary string of length β-bits, and $\{0,1\}^*$ denotes a binary string of arbitrary length.
- If b is a bit and β is a positive integer, we denote by b^β the concatenation of b with itself β times.
- For a non-empty set $\mathcal{H}$, we denote by $h \xleftarrow{\$} \mathcal{H}$ the selection of a member of $\mathcal{H}$ uniformly at random and assigning it to h.
- If x and n are positive integers so that $0 \leq x < 2^n$, we denote by $\mathsf{tostr}(x, n)$ the binary representation of x as an n-bit string (in a big-endian format).
- If s is a binary string, we denote by $\mathsf{toint}(s)$ the unsigned integer representation of s (in a big-endian format).
- If s is a binary string and ℓ is a positive integer, we denote by $\mathsf{setlen}(s, \ell)$ the truncation of s into its ℓ most significant bits. If $|s| < \ell$, then $\mathsf{setlen}(s, \ell)$ denotes the ℓ-bit long string $s||0^{\ell-|s|}$.

2.2 Universal Hash-Function Families

A family of hash functions $\mathcal{H}$ is specified by a finite set of keys $\mathcal{K}$. Each key $k \in \mathcal{K}$ defines a member of the family $\mathcal{H}_k \in \mathcal{H}$. As opposed to thinking of $\mathcal{H}$ as a set of functions from $\mathcal{D}$ to $\mathcal{R}$, it can be viewed as a single function $\mathcal{H} : \mathcal{K} \times \mathcal{D} \to \mathcal{R}$, whose first argument is usually written as a subscript. A random element $h \xleftarrow{\$} \mathcal{H}$ is determined by selecting a $k \xleftarrow{\$} \mathcal{K}$ uniformly at random and setting $h \leftarrow \mathcal{H}_k$. There are many classes of universal hash families, depending on their probability of message collision, we give below a formal definition of one class of universal hash families called ϵ-almost universal.

Definition 1. *Let $\mathcal{H} = \{h : \mathcal{D} \to \mathcal{R}\}$ be a family of hash functions and let $\epsilon \geq 0$ be a real number. $\mathcal{H}$ is said to be ϵ-almost universal, denoted ϵ-AU, if for all distinct $M, M' \in \mathcal{D}$, we have that $\Pr_{h \leftarrow \mathcal{H}}\left[h(M) = h(M')\right] \leq \epsilon$. $\mathcal{H}$ is said to be ϵ-almost universal on equal-length strings if for all distinct, equal-length strings $M, M' \in \mathcal{D}$, we have that $\Pr_{h \leftarrow \mathcal{H}}\left[h(M) = h(M')\right] \leq \epsilon$.*

2.3 Block Ciphers

A block cipher mapping ℓ-bit strings to ℓ-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 $\{0,1\}^\ell$ to elements in $\{0,1\}^\ell$, it can be viewed as a single function $\mathcal{F} : \mathcal{K}_e \times \{0,1\}^\ell \to \{0,1\}^\ell$, whose first argument is usually written as a subscript. A random element $f \xleftarrow{\$} \mathcal{F}$ is determined by selecting a $K \xleftarrow{\$} \mathcal{K}_e$ uniformly at random and setting $f \leftarrow \mathcal{F}_K$.

As in [27], we adopt the notion of security for block ciphers introduced in [30] and adopted for the concrete setting in [8]. Let $\mathcal{F} : \{0,1\}^L \times \{0,1\}^\ell \to \{0,1\}^\ell$, where L is the key length and ℓ is the block size of the block cipher, be a block cipher and let $\mathsf{Perm}(\ell)$ denote the set of all permutations on $\{0,1\}^\ell$. Let $\mathcal{A}$ be an adversary with access to an oracle and that returns a bit. Then,

$$\mathsf{Adv}^{\mathrm{prp}}_{\mathcal{F}}(\mathcal{A}) = \Pr\left[f \xleftarrow{\$} \mathcal{F} : \mathcal{A}^{f(\cdot)} = 1\right] - \Pr\left[\pi \xleftarrow{\$} \mathsf{Perm}(\ell) : \mathcal{A}^{\pi(\cdot)} = 1\right] \tag{1}$$

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 small.

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,

$$\mathsf{Adv}^{\mathrm{sprp}}_{\mathcal{F}}(\mathcal{A}) = \Pr\left[f \xleftarrow{\$} \mathcal{F} : \mathcal{A}^{f(\cdot), f^{-1}(\cdot)} = 1\right] - \Pr\left[\pi \xleftarrow{\$} \mathsf{Perm}(\ell) : \mathcal{A}^{\pi(\cdot), \pi^{-1}(\cdot)} = 1\right] \tag{2}$$

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, are believed to be secure SPRPs.

2.4 Authenticated Encryption Schemes

The authenticated encryption model that we use is similar to the one in [27,37]. A nonce-using, symmetric authenticated encryption scheme $\mathcal{AE} = (\mathcal{K}, \mathcal{SE}, \mathcal{VD})$ consists of three algorithms: the key generation algorithm ($\mathcal{K}$), the signed encryption algorithm ($\mathcal{SE}$), and the verified decryption algorithm ($\mathcal{VD}$). $\mathcal{AE}$ is defined over some key space KeySp, some nonce space $\mathsf{NonceSp} = \{0,1\}^{\mathsf{nl}}$, for a positive integer nl, and some message space $\mathsf{MsgSp} = \{0,1\}^*$. We require that membership in MsgSp can be efficiently tested and that if M, M' are two strings such that $M \in \mathsf{MsgSp}$ and $|M| = |M'|$, then $M' \in \mathsf{MsgSp}$.

The randomized key generation algorithm $\mathcal{K}$ returns a key $K \in \mathsf{KeySp}$. The deterministic signed encryption algorithm $\mathcal{SE}$ takes as input a key $K \in \mathsf{KeySp}$, a nonce $N \in \mathsf{NonceSp}$, and a payload message $M \in \mathsf{MsgSp}$, and returns a ciphertext $\sigma \in \{0,1\}^*$. The deterministic verified decryption algorithm $\mathcal{VD}$ takes as input a key $K \in \mathsf{KeySp}$, a nonce $N \in \mathsf{NonceSp}$, a string $\sigma \in \{0,1\}^*$, and outputs a message $M \in \mathsf{MsgSp}$ or the special symbol $\mathsf{INVALID}$ on error. We ask for the basic validity requirement that if $\sigma = \mathcal{SE}_K(N, M)$ then it must be the case that $\mathcal{VD}_K(N, \sigma) = M$.

2.5 Adversarial Model

We adopt the standard adversarial model used in authenticated encryption schemes. The adversary is given oracle access to the signed encryption algorithm $\mathcal{SE}_K(N, M)$. The adversary can call the $\mathcal{SE}$ oracle on nonce-message pairs (N, M) of her choice and observing the outputs. After calling the $\mathcal{SE}$ oracle for q times, the adversary attempts a forgery by calling the verified decryption algorithm $\mathcal{VD}_K(N, \sigma)$ for an (N, σ) pair of her choice. Note that the adversary does not see the secret key K. If the verified decryption oracle returns the $\mathsf{INVALID}$ symbol, the adversary is considered unsuccessful; otherwise, the forgery attempt is said to be successful.

A standard assumption in authenticated encryption schemes is that the adversary is nonce-respecting. An adversary is said to be nonce-respecting if she never repeats a nonce. That is, after calling the signed encryption oracle on (N, M), the adversary never asks its oracle a query (N, M'), regardless of the oracle responses. We emphasize, however, that the nonce used in the forgery attempt may coincide with a nonce used in one of the adversary's queries.

2.6 Properties of Odd Integers

We state here two lemmas about odd integers in the finite integer ring $\mathbb{Z}_{2^n}$ that will be used for the rest of the paper.

Lemma 1. *For any nonzero integers α and β in $\mathbb{Z}_{2^n}$, 2^n divides $\alpha\beta$ only if both α and β are even integers. Formally, the following one-way implication must hold:*

$$\alpha\beta \equiv 0 \mod 2^n \quad \Rightarrow \quad \alpha \equiv \beta \equiv 0 \mod 2. \tag{3}$$

Lemma 2. *Let $\boldsymbol{X_1}$ be the random variable representing the experiment of drawing a number x_1 from the set of integers $\{0, 1, \cdots, 2^n - 1\}$ uniformly at random. Then, for any odd integer $k \in \mathbb{Z}_{2^n}$, the random variable $\boldsymbol{Y_1} = k \cdot \boldsymbol{X_1} \mod 2^n$ is uniformly distributed over the set $\{0, 1, \cdots, 2^n - 1\}$.*

Lemmas 1 and 2, along with the fact that there is a one-to-one correspondence between n-bit strings and the integer ring $\mathbb{Z}_{2^n}$, will be used to establish the results of this paper. The proofs of the two lemmas can be found in [3].

3 The Odd Key Hash Family

In this section, we give a description of the OK hash family that will be used in the construction of our OKH authenticated encryption. Fix an integer $n \geq 1$ (the "block size") and an integer $b \geq 1$ (the "number of blocks"). We define the family of functions $\mathsf{OK}[n, b]$ as follows. The domain is $\mathcal{D} = \{0,1\}^n \cup \{0,1\}^{2n} \cup \cdots \cup \{0,1\}^{bn}$ and the range is $\mathcal{R} = \{0,1\}^n$. Each function in $\mathsf{OK}[n, b]$ is defined by the b-tuple $K = (k_1, \cdots, k_b)$, where $k_i \in \mathbb{Z}^*_{2^n}$ for $i = 1, \cdots, b$. A random function in $\mathsf{OK}[n, b]$ is given by drawing the k_i's at random from the multiplicative group $\mathbb{Z}^*_{2^n}$. The function determined by K is written as $\mathsf{OK}_K(\cdot)$.

For an input message $M \in \mathcal{D}$, view M as a sequence of n-bit blocks, i.e., $M = m_1 \cdots m_\ell$, where $\ell \leq b$, and write each block in its unsigned integer representation in $\mathbb{Z}_{2^n}$ (in a big-endian format). Then, the compressed image of M is given by

$$\mathsf{OK}_K(M) = \sum_{i=1}^{\ell} k_i m_i \mod 2^n. \tag{4}$$

When the values of n and b are known, we will write OK instead of $\mathsf{OK}[n, b]$ to simplify the notations.

4 Description of the OKH Authenticated Encryption

As mentioned earlier, the key idea allowing for more efficient authentication over the CWC mode of operation is advancing the hashing phase to be applied on the plaintext instead of the ciphertext. The mode of operation used for encryption is similar to the counter mode (CTR) but with the the requirement that the plaintext is to be processed by the block cipher, as illustrated in Figure 1.

As in previous authenticated encryption schemes, we require the block cipher BC to be a strong pseudorandom permutation. Let BC: $\{0,1\}^{\mathsf{kl}} \times \{0,1\}^{\mathsf{bs}} \rightarrow \{0,1\}^{\mathsf{bs}}$ be the used block cipher, where kl and bs are the key length and block size of the block cipher, respectively. The authenticated encryption using BC for

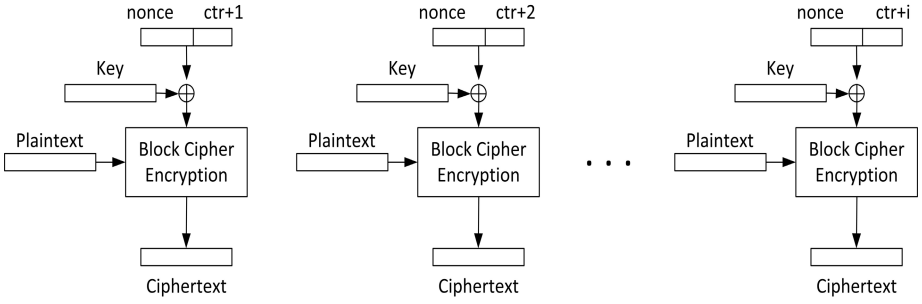

Fig. 1. The used mode of encryption to construct the OKH authenticated encryption. Note that the nonce-counter concatenation is XORed with the key, not the plaintext block. Therefore, given a nonce-respecting adversary, the encryption key of each block is different than all other blocks of the same message and different than all other blocks of different messages due to the use of nonce-counter concatenation. This observation is critical for the security of the proposed scheme.

encryption and the OK family for hashing, $\mathrm{OKH} = (\mathcal{K}, \mathcal{SE}, \mathcal{VD})$, is defined as follows. The message spaces are

$$\mathsf{MsgSp} = \Big\{ M \in \{0,1\}^* : |M| \leq \mathsf{MaxLen} \Big\}, \tag{5}$$

$$\mathsf{KeySp} = \Big\{ (K_e, K_h) \in \{0,1\}^{\mathsf{kl}} \times \{0,1\}^{\mathsf{MaxLen}} \Big\}, \tag{6}$$

$$\mathsf{NonceSp} = \Big\{ N \in \{0,1\}^* : |N| \leq \mathsf{kl} - \log_2 \mathsf{MaxLen} \Big\}, \tag{7}$$

where MaxLen is the message maximum length. The size of the counter, CtrLen, in the mode of operation of Figure 1 is at least $\log_2 \mathsf{MaxLen}$. The concatenation of the nonce and the counter is of length kl-bits.

Informally speaking, the authentication tag is computed by dividing the plaintext message into blocks of n-bit long, hash it according to equation (4) using a member of the OK family, and encrypt the resulting n-bit hashed image (as shown in the MAC algorithm below). The size of the hashing images, n, is less than or equal to the block cipher size, bs. The encryption part is done the natural way (as shown in the $\mathcal{E}$ algorithm below).

Remark 1. There are two important points to note about the OK family. First, the OK family is defined over the domain $\mathcal{D}$ only. This issue, however, can be easily solved with an appropriate padding (e.g. with zeros as we do in this paper). Second, and more important, as in universal hash families, the OK family as described in Section 3 can only be used to authenticate equal-length messages. For example, a message block consisting of all zeros will not contribute to the value of the hashed image. Hence, it is easy to come up with two distinct messages that collide and, eventually, achieve a successful forgery. However, there are known techniques to make the hash function applicable to arbitrary-length messages. For instance, in [14] the authors proposed appending the length of

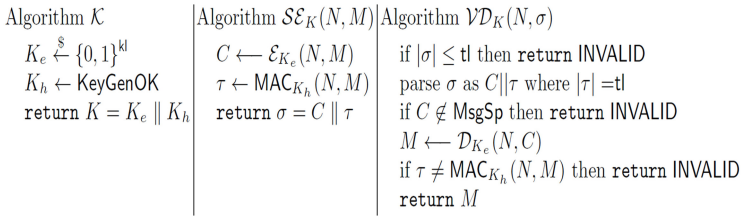

Fig. 2. Pseudocodes of the key generation ($\mathcal{K}$), signed encryption ($\mathcal{SE}$), and verified decryption ($\mathcal{VD}$) algorithms

the message at the end. In our case, it suffices to append the bit '1' at the end of the message since this will guarantee that changing the message length will change the hashed image in an unpredictable manner depending on the hashing key corresponding to the last message block.

Another important remark is related to the message length. For a message that is longer than MaxLen, it is treated as multiple chunks of length MaxLen or less and the corresponding tags are concatenated; that is, arbitrary long messages can be authenticated using the same fixed-length hashing key (this is actually the case for any universal hashing based MAC, not just the proposed one [14]). In typical settings, however, one need not worry about such messages since MaxLen is often sufficiently long. For instance, typical AES key-lengths are 128, 192, or 256. Assuming the shortest key-length of $\mathsf{kl} = 128$ bits, and setting $|N| = 88$ and $\mathsf{CtrLen} = 40$ as can be found in [27], MaxLen can be more than Tera bits long, while one is often dealing with much shorter messages (about third of the messages on the backbone of the Internet, for instance, are only 43 bytes [37]). For the rest of the paper, we will assume messages of length MaxLen or less for simplicity.

The OKH's key generation ($\mathcal{K}$), signed encryption ($\mathcal{SE}$), and verified decryption ($\mathcal{VD}$) algorithms are defined as in Figure 2. The rest of the algorithms (KeyGenOK, $\mathcal{E}$, $\mathcal{D}$, MAC, OK-HASH) are defined in Figure 3. Algorithm KeyGenOK handles the generation of the key that defines the used member of the OK hashing family. Algorithms $\mathcal{E}$ and $\mathcal{D}$ handle the encryption and decryption corresponding to the mode of operation depicted in Figure 1. Algorithm MAC handles the generation of authentication tags, which calls algorithm OK-HASH to compress the message.

5 Theorem Statements

In this section, we give the main security statements of the proposed scheme, formal security proofs can be found in the full version [3].

5.1 Security of Authentication

Fix an authenticated encryption scheme $\mathrm{OKH} = (\mathcal{K}, \mathcal{SE}, \mathcal{VD})$ and run an adversary $\mathcal{A}$ with an oracle $\mathcal{SE}_K(\cdot, \cdot)$ for some key K. The adversary $\mathcal{A}$

Algorithm KeyGenOK
  $\mathsf{Key} \xleftarrow{\$} \{0,1\}^{\mathsf{MaxLen}}$
  $\alpha \leftarrow |\mathsf{Key}|/\mathsf{tl}$
  break Key into tl-bit chunks K_i's
  for $i = 1$ to α do
    $K_i \leftarrow K_i \vee \mathsf{tostr}(1, \mathsf{tl})$
  return $K_h = K_1 || \cdots || K_\alpha$

Algorithm $\mathcal{E}_{K_e}(N, M)$
  $\ell \leftarrow$ min int so that bs divides $|M||0^\ell|$
  $M \leftarrow M||1||0^{\ell-1}$
  $\alpha \leftarrow |M|/\mathsf{bs}$
  break M into bs-bit chunks M_i's
  for $i = 1$ to α do
    $K_{e_i} \leftarrow K_e \oplus \big(N||\mathsf{tostr}(i, \mathsf{CtrLen})\big)$
    $c_i \leftarrow \mathsf{BC}_{K_{e_i}}(M_i)$
  return $C = c_1||\cdots||c_\alpha$

Algorithm $\mathcal{D}_{K_e}(N, C)$
  $\alpha \leftarrow |C|/\mathsf{bs}$
  break C into bs-bit chunks c_i's
  for $i = 1$ to α do
    $K_{e_i} \leftarrow K_e \oplus \big(N||\mathsf{tostr}(i, \mathsf{CtrLen})\big)$
    $M_i \leftarrow \mathsf{BC}^{-1}_{K_{e_i}}(c_i)$
  return $M = M_1 \,\|\, \cdots || M_\alpha$

Algorithm $\mathsf{MAC}_{K_h}(N, M)$
  $\gamma' \leftarrow \mathsf{OK\text{-}HASH}_{K_h}(M)$
  $\ell \leftarrow \mathsf{bs} - n$
  $\gamma \leftarrow \gamma'||0^\ell$
  $\tau' \leftarrow \mathsf{BC}_{K_e}\big(\gamma \oplus \mathsf{setlen}(N, \mathsf{bs})\big)$
  $\tau \leftarrow \mathsf{setlen}(\tau', \mathsf{tl})$
  return τ

Algorithm $\mathsf{OK\text{-}HASH}_{K_h}(M)$
  $\ell \leftarrow$ min int so that n divides $|M||0^\ell|$
  $M \leftarrow M||1||0^{\ell-1}$
  $\alpha \leftarrow |M|/n$
  break M into n-bit chunks M_i's
  for $i = 1$ to α do
    $m_i \leftarrow \mathsf{toint}(M_i)$
    $k_i \leftarrow \mathsf{toint}(K_i)$
  $\gamma \leftarrow \sum_{i=1}^{\alpha} k_i m_i \mod 2^n$
  return γ

Fig. 3. Pseudocodes of the KeyGenOK, $\mathcal{E}$, $\mathcal{D}$, MAC, OK-HASH

successfully forges in this run if $\mathcal{A}$ is nonce-respecting, $\mathcal{A}$ outputs a pair (N, σ) where $\mathcal{VD}_K(N, \sigma) \neq \mathsf{INVALID}$, and $\mathcal{A}$ made no earlier query (N, M) which resulted in the response σ. Let $\mathsf{Adv}^{\mathrm{auth}}_{\mathsf{OKH}[\mathsf{BC},\mathsf{OK}]}(\mathcal{A}) = \Pr\left[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathcal{SE}_K(\cdot,\cdot)} \text{ forges}\right]$ be the adversary's advantage of successful forgery against the scheme OKH that uses BC as a block cipher for encryption and the OK family for hashing.

We give here information-theoretic bounds on the authenticity of the scheme of Section 4 assuming the use of a true random permutation, $\mathsf{Perm}(\ell)$, for encryption.

Theorem 1. *Fix an* $\mathsf{OK}[n, b]$ *hash family and let* $\mathsf{Perm}(\ell) : \{0,1\}^\ell \to \{0,1\}^\ell$ *be a true random permutation and let* tl *be the desired tag length. Let* $\mathcal{A}$ *be a nonce-respecting adversary that asks* q *queries and then makes its forgery attempt against the* OKH *of Section 4. Then,* $\mathcal{A}$*'s advantage of successful forgery is bounded by*

$$\mathsf{Adv}^{\mathrm{auth}}_{\mathsf{OKH}[\mathsf{Perm}(\ell),\mathsf{OK}_{[n,b]}]}(\mathcal{A}) \leq 2^{-n} + 2^{-\mathsf{tl}}.$$

It is standard to pass a complexity-theoretic analog of Theorem 1, but in doing this one will need access to a BC^{-1} oracle in order to verify a forgery attempt, which translates into needing the strong pseudorandom permutation assumption. One gets the following. Fix an $\mathsf{OK}[n, b]$ hash family and a block cipher $\mathsf{BC} : \mathcal{K} \times \{0,1\}^\ell \to \{0,1\}^\ell$. Let $\mathcal{A}$ be a nonce-respecting adversary that asks q queries totaling at most λ bits of payload and then makes its forgery

attempt. Then, there is an adversary $\mathcal{B}$ attacking the block cipher in which

$$\mathsf{Adv}^{\mathrm{auth}}_{\mathrm{OKH}[\mathsf{BC},\mathsf{OK}]}(\mathcal{A}) \leq \mathsf{Adv}^{\mathrm{sprp}}_{\mathsf{BC}}(\mathcal{B}) + 2^{-n} + 2^{-\mathsf{tl}}.$$

Furthermore, adversary $\mathcal{B}$ takes the same time adversary $\mathcal{A}$ takes and makes at most $\lambda/\ell + q$ oracle queries.

Before we proceed with the security analysis, we give the intuition behind the choices of the OK hash family and the mode of encryption in Figure 1. First, observe that the OK family is not a universal hash family. To see this, let the OK family be defined over the ring $\mathbb{Z}_{2^n}$. Let $M = m_1||m_2$ be a message of two n-bit blocks (for the rest of the paper, we overload m_i to denote both the n-bit binary string of the i^{th} message block and its integer representation as an element of $\mathbb{Z}_{2^n}$ in a big-endian format; the distinction between the two representations will be omitted as long as it is clear from the context). Consider now the message $M' = m'_1||m'_2 = m_1 + 2^{n-1}||m_2 + 2^{n-1} \neq M$. Then,

$$\begin{aligned} k_1 m'_1 + k_2 m'_2 &= k_1\big(m_1 + 2^{n-1}\big) + k_2\big(m_2 + 2^{n-1}\big) \\ &\equiv k_1 m_1 + k_2 m_2 \mod 2^n, \end{aligned} \tag{8}$$

where equation (8) holds since both k_1 and k_2 are odd integers by design.

Remark 2. Equation (8) implies that the OK family, unlike universal hash families, cannot be used to construct standard MACs since forgers can easily find two colliding messages. However, one key idea of the proposed OKH is that finding two messages that collide does not translate into a successful forgery (since the adversary must also predict the correct ciphertext corresponding to the colliding messages). The other key idea behind the design of OKH is that the effect of modifying an observed ciphertext block will result in modifying its corresponding plaintext block randomly (assuming the block cipher is a strong pseudorandom permutation). For that, the following lemma addresses the adversary's chances of causing a collision in the hashing phase by modifying the ciphertext.

Lemma 3. *Let $C \neq C'$ be any two distinct ciphertexts and let $M \neq M'$ be the plaintext messages corresponding to C and C', respectively. Then, assuming the use of a random permutation for block encryption,* $\Pr_{h \leftarrow \mathsf{OK}[n,b]}\Big[h(M) = h(M')\Big] \leq 2^{-n}$.

Proof. Let c_i and c'_i denote the i^{th} blocks of C and C', respectively. Similarly, let m_i and m'_i denote the i^{th} blocks of M and M', respectively. Since $M \neq M'$, they must be different in at least one block. Assume M and M' differ in a single block only. Without loss of generality, let $m'_1 = m_1 + \epsilon \neq m_1$ and the rest of the blocks are the same. Then, since k_1 is an odd integer, by Lemma 1, $k_1 \cdot \epsilon \not\equiv 0 \mod 2^n$ and the probability $\Pr_{h \leftarrow \mathsf{OK}[n,b]}\Big[h(M) = h(M')\Big] = 0$.

Assume now that M and M' differ by more than one block. Write $m'_i = m_i + \epsilon_i \neq m_i$ for each block i in which the two messages differ. Since a random permutation is used for encryption, even if c'_i differs with c_i by a known constant,

ϵ_i will be a random element of $\mathbb{Z}_{2^n}$ (for any user with no knowledge of the encryption key). Therefore, by Lemma 2, $\Pr_{h \leftarrow \mathsf{OK}[n,b]}\left[h(M) = h(M')\right] = 2^{-n}$ and the lemma follows. □

Remark 3. Lemma 3 illustrates the significance of restricting the hashing keys to the set of odd integers. To see this, let the keys be drawn from $\mathbb{Z}_{2^n}$ as opposed to $\mathbb{Z}^*_{2^n}$. Assume that k_i, for some i, happened to be equal to 2^{n-1}. Then, using the fact that the used block cipher can be modeled as a strong pseudorandom permutation, any modification of the i^{th} ciphertext block will go undetected with a probability 1/2 (this is because $\epsilon \cdot 2^{n-1}$ is congruent to zero modulo 2^n for any even ϵ). In general, if k_i is equal to $2^{n-\ell}$, for any positive integer $\ell < n$, then any modification of the i^{th} ciphertext block will go undetected with a probability $1/2^\ell$.

With Remark 2 and Lemma 3 in mind, one can proceed with the formal proof of Theorem 1, which can be found in [3].

5.2 Security of Encryption

In this section, we show that the privacy of the proposed scheme is provably secure assuming the used block cipher is a secure pseudorandom permutation. Consider an adversary $\mathcal{A}$ who has one of two types of oracles: a real encryption oracle and a fake encryption oracle. The real encryption oracle $\mathcal{E}_K(\cdot,\cdot)$ takes as input a pair (N, M) and returns a ciphertext $C \xleftarrow{\$} \mathcal{E}_K(N, M)$. Assume that the length of the ciphertext depends only on the length of the plaintext, that is, $|C| = l(|M|)$. The fake encryption oracle, $\$(\cdot,\cdot)$, takes as input a pair (N, M) and returns a random string $C \xleftarrow{\$} \{0,1\}^{l(|M|)}$. Given adversary $\mathcal{A}$ and authenticated encryption scheme $\text{OKH} = (\mathcal{K}, \mathcal{SE}, \mathcal{VD})$, define

$$\mathsf{Adv}^{\text{priv}}_{\text{OKH}[\mathsf{BC},\mathsf{OK}]}(\mathcal{A}) = \Pr\left[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathcal{SE}_K(\cdot,\cdot)} = 1\right] - \Pr\left[\mathcal{A}^{\$(\cdot,\cdot)} = 1\right]$$

to be $\mathcal{A}$'s advantage of breaking the privacy of the authenticated encryption scheme using BC as a block cipher and OK for hashing. That is, as in previous authenticated encryption proposals (e.g., [27, 37]), the strong model of distinguishing the ciphertext from a random string is used to model the privacy of encryption.

Theorem 2. *Let* OKH[BC;OK] *be the authenticated encryption scheme described in Section 4 using the* OK *hash family for compression and the block cipher* BC *for encryption. Then given a nonce-respecting adversary,* $\mathcal{A}$*, against* OKH[BC;OK]*, one can construct an adversary* $\mathcal{B}$ *against* BC *such that*

$$\mathsf{Adv}^{\text{priv}}_{\text{OKH}[\mathsf{BC};\mathsf{OK}]}(\mathcal{A}) \leq \mathsf{Adv}^{\text{prp}}_{\mathsf{BC}}(\mathcal{B}). \tag{9}$$

Furthermore, the experiment for $\mathcal{B}$ *takes the same time as the experiment for* $\mathcal{A}$ *and, if* $\mathcal{A}$ *makes at most* q *oracle queries totaling at most* μ *bits of payload data, then* $\mathcal{B}$ *makes at most* $\mu/\ell + q$ *oracle queries.*

Table 1. Performance comparison of the MMH hash family of Halevi and Krawczyk [21], the polynomial-evaluation (POLY) hash family of Bernstein [13], the NH hash family of Black et al. [14], and the proposed OK family.

	MMH family	POLY family	NH family	OK family
Collision probability	2^{-30}	2^{-96}	2^{-64}	2^{-64}
Hashed image (bits)	32	128	128	64
Speed (cycles/byte)	1.2	2.4	0.84	0.27

Theorem 2 states that, if BC is a secure pseudorandom permutation, then the proposed authenticated encryption scheme provides data privacy; the formal proof can be found in [3].

6 Design and Performance Discussions

In this section, we discuss the main design ideas behind the proposed OKH scheme and compare its performance to other authentication codes in the cryptographic literature.

First, note that there is a one-to-one correspondence between the set of n-bit sequences and the integer ring $\mathbb{Z}_{2^n}$. Hence, the integer ring $\mathbb{Z}_{2^n}$ is the natural choice when performing arithmetic on binary sequences. From a computational-efficiency point of view, the integer ring $\mathbb{Z}_{2^n}$ has an advantage over other finite integer rings in that modular reduction can simply be realized by truncating what is beyond the n^{th} bit (no nontrivial modular reduction is required). From a mathematical point of view, the integer ring $\mathbb{Z}_{2^n}$ possesses the unique property that an element $a \in \mathbb{Z}_{2^n}$ is invertible if and only if it is an odd integer. That is, the multiplicative group $\mathbb{Z}_{2^n}^*$ consists of all odd integers less than 2^n, and nothing else. Consequently, for a random number ϵ drawn uniformly from $\mathbb{Z}_{2^n}$ and an odd key k, the value of $\epsilon \cdot k \mod 2^n$ is uniformly distributed over $\mathbb{Z}_{2^n}$ (by Lemma 2). This fact, along with the fact that block ciphers can be realized as strong pseudorandom permutations, are the main principles behind the design of the OKH authenticated encryption composition. That is, by advancing the hashing phase to be applied to the plaintext, before block cipher encryption, and restricting the hashing keys to the set of odd integers, one can show that a successful forgery by causing a collision in the hashing phase can occur with a negligible probability (by Theorem 1).

In the literature, without advancing the hashing phase to be performed before block cipher encryption, the objective of guaranteeing that a forgery attempt by causing a collision in the hashing phase can succeed with a negligible probability has been achieved by restricting the hash function to be universal. Intuitively, removing such a restriction on the hash function should only increase its speed.

In what follows, we give a detailed performance comparison between the OK family and the NH family of Black et al. [14], the fastest universal hash family in the cryptographic literature for software implementations. Comparison with other known hash families is summarized in Table 1. As before, let M be a message to be authenticated and write M as a sequence of n-bit strings; i.e.,

$M = m_1||\cdots||m_\ell$, where $|m_i| = n$. Similarly, let the key $K = k_1||\cdots||k_\ell$ be the hashing key. Let NH_K be a random member of the NH family determined by the key K. Then, the compressed image of M is computed as

$$\mathsf{NH}_K(M) = \sum_{i=1}^{\ell/2} \Big((k_{2i-1} + m_{2i-1} \mod 2^n) \cdot (k_{2i} + m_{2i} \mod 2^n) \Big) \mod 2^{2n}. \tag{10}$$

On the other hand, the hash image of M as computed by the OK family is

$$\mathsf{OK}_K(M) = \sum_{i=1}^{\ell} k_i \cdot m_i \mod 2^n. \tag{11}$$

That is, when the block size is n, the OK computations are performed over $\mathbb{Z}_{2^n}$ while the NH computations are performed over the larger integer ring $\mathbb{Z}_{2^{2n}}$.

To give a numerical example, let $n = 64$ bits. Then, the OK family requires 64-bit computations while the NH family requires 128-bit computations. When using a 64-bit machine, this implies that NH computations must be split over two registers, while OK computations are performed using a single register. Splitting operations over two registers can slow down the speed by about 63%. More importantly, in standard compilers, there is no integer data type of size 128-bit. Therefore, to multiply two 64-bit integers, one needs to split each integer into two 32-bit parts and multiply with appropriate shifts. Using a 64-bit machine with 3.00GHz Intel(R) Xeon(TM) CPU running on UNIX operating system, the NH family runs at 0.87 cycles/byte while the OK family runs at 0.27 cycles/byte (All codes are written in C).

7 Conclusion

In this paper, we proposed the OKH authenticated encryption scheme. By advancing the hashing phase before block cipher encryption, we showed how a hashing function that is not universal can be used without affecting security of authentication. Since the hash function does not have to be universal, it can be computed faster than universal hash function in the literature of cryptography.

References

1. Afanassiev, V., Gehrmann, C., Smeets, B.: Fast Message Authentication Using Efficient Polynomial Evaluation. In: Biham, E. (ed.) FSE 1997. LNCS, vol. 1267, pp. 190–204. Springer, Heidelberg (1997)
2. Alomair, B.: Towards Authenticated and Private Computer and Wireless Communications. PhD thesis, University of Washington (2011)
3. Alomair, B.: Authenticated Encryption: How Reordering Can Impact Performance. In: Bao, F., Samarati, P., Zhou, J. (eds.) ACNS 2012. LNCS, vol. 7341, pp. 84–99. Springer, Heidelberg (2012), `http://cri.kacst.edu.sa/goto.php?link=alomair`

4. Alomair, B., Clark, A., Poovendran, R.: The power of primes: security of authentication based on a universal hash-function family. Journal of Mathematical Cryptology 4(2), 121–147 (2010)
5. Alomair, B., Poovendran, R.: Efficient Authentication for Mobile and Pervasive Computing. In: Soriano, M., Qing, S., López, J. (eds.) ICICS 2010. LNCS, vol. 6476, pp. 186–202. Springer, Heidelberg (2010)
6. Alomair, B., Poovendran, R.: $\mathcal{E}$-MACs: Towards More Secure and More Efficient Constructions of Secure Channels. In: Rhee, K.-H., Nyang, D. (eds.) ICISC 2010. LNCS, vol. 6829, pp. 292–310. Springer, Heidelberg (2011)
7. Bellare, M., Canetti, R., Krawczyk, H.: Keying Hash Functions for Message Authentication. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 1–15. Springer, Heidelberg (1996)
8. Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A concrete security treatment of symmetric encryption. In: Proceedings of the 38th IEEE Symposium on Foundations of Computer Science – FOCS 1997, pp. 394–403. IEEE Computer Society Press (1997)
9. Bellare, M., Guérin, R., Rogaway, P.: XOR MACs: New Methods for Message Authentication Using Finite Pseudorandom Functions. In: Coppersmith, D. (ed.) CRYPTO 1995. LNCS, vol. 963, pp. 15–28. Springer, Heidelberg (1995)
10. Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. Journal of Cryptology 21(4), 469–491 (2008)
11. Bellare, M., Rogaway, P.: Encode-Then-Encipher Encryption: How to Exploit Nonces or Redundancy in Plaintexts for Efficient Cryptography. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 317–330. Springer, Heidelberg (2000)
12. Bellare, M., Rogaway, P., Wagner, D.: The EAX Mode of Operation. In: Roy, B., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 389–407. Springer, Heidelberg (2004)
13. Bernstein, D.: Floating-point arithmetic and message authentication (2004) (unpublished manuscript), `http://cr.yp.to/hash127.html`
14. Black, J., Halevi, S., Krawczyk, H., Krovetz, T., Rogaway, P.: UMAC: Fast and Secure Message Authentication. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 216–499. Springer, Heidelberg (1999)
15. Carter, J., Wegman, M.: Universal classes of hash functions. In: Proceedings of the 9th ACM Symposium on Theory of Computing – STOC 1977, pp. 106–112. ACM SIGACT (1977)
16. Dworkin, M.: Recommendation for block cipher modes of operation: The CMAC mode for authentication. National Institute of Standards and Technology (NIST) Special Publication 800-38B (2005)
17. Dworkin, M.: Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. National Institute for Standards and Technology (NIST) Special Publication 800-38D (2007)
18. Etzel, M., Patel, S., Ramzan, Z.: SQUARE HASH: Fast Message Authentication via Optimized Universal Hash Functions. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 234–251. Springer, Heidelberg (1999)
19. Ferguson, N., Whiting, D., Schneier, B., Kelsey, J., Lucks, S., Kohno, T.: Helix: Fast Encryption and Authentication in a Single Cryptographic Primitive. In: Johansson, T. (ed.) FSE 2003. LNCS, vol. 2887, pp. 330–346. Springer, Heidelberg (2003)

20. Gligor, V.D., Donescu, P.: Fast Encryption and Authentication: XCBC Encryption and XECB Authentication Modes. In: Matsui, M. (ed.) FSE 2001. LNCS, vol. 2355, pp. 92–108. Springer, Heidelberg (2002)
21. Halevi, S., Krawczyk, H.: MMH: Software Message Authentication in the Gbit/Second Rates. In: Biham, E. (ed.) FSE 1997. LNCS, vol. 1267, pp. 172–189. Springer, Heidelberg (1997)
22. Handschuh, H., Preneel, B.: Key-Recovery Attacks on Universal Hash Function Based MAC Algorithms. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 144–161. Springer, Heidelberg (2008)
23. Johansson, T.: Bucket Hashing with a Small Key Size. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 149–162. Springer, Heidelberg (1997)
24. Jutla, C.: Encryption modes with almost free message integrity. Journal of Cryptology 21(4), 547–578 (2008)
25. Katz, J., Lindell, Y.: Introduction to modern cryptography. Chapman & Hall/CRC (2008)
26. Katz, J., Yung, M.: Unforgeable Encryption and Chosen Ciphertext Secure Modes of Operation. In: Schneier, B. (ed.) FSE 2000. LNCS, vol. 1978, pp. 284–299. Springer, Heidelberg (2001)
27. Kohno, T., Viega, J., Whiting, D.: CWC: A High-Performance Conventional Authenticated Encryption Mode. In: Roy, B., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 408–426. Springer, Heidelberg (2004)
28. Krawczyk, H.: LFSR-Based Hashing and Authentication. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 129–139. Springer, Heidelberg (1994)
29. Krawczyk, H.: The Order of Encryption and Authentication for Protecting Communications (or: How Secure Is SSL?). In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 310–331. Springer, Heidelberg (2001)
30. Luby, M., Rackoff, C.: How to Construct Pseudorandom Permutations from Pseudorandom Functions. SIAM Journal on Computing 17(2), 373–386 (1988)
31. Muller, F.: Differential Attacks against the Helix Stream Cipher. In: Roy, B., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 94–108. Springer, Heidelberg (2004)
32. Nevelsteen, W., Preneel, B.: Software Performance of Universal Hash Functions. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 24–41. Springer, Heidelberg (1999)
33. Paul, S., Preneel, B.: Near Optimal Algorithms for Solving Differential Equations of Addition with Batch Queries. In: Maitra, S., Veni Madhavan, C.E., Venkatesan, R. (eds.) INDOCRYPT 2005. LNCS, vol. 3797, pp. 90–103. Springer, Heidelberg (2005)
34. Paul, S., Preneel, B.: Solving Systems of Differential Equations of Addition. In: Boyd, C., González Nieto, J.M. (eds.) ACISP 2005. LNCS, vol. 3574, pp. 75–88. Springer, Heidelberg (2005)
35. Preneel, B., van Oorschot, P.C.: MDx-MAC and Building Fast MACs from Hash Functions. In: Coppersmith, D. (ed.) CRYPTO 1995. LNCS, vol. 963, pp. 1–14. Springer, Heidelberg (1995)
36. Rogaway, P.: Bucket hashing and its application to fast message authentication. Journal of Cryptology 12(2), 91–115 (1999)
37. Rogaway, P., Bellare, M., Black, J., Krovetz, T.: OCB: A block-cipher mode of operation for efficient authenticated encryption. In: Proceedings of the 8th ACM Conference on Computer and Communications Security – CCS 2001, pp. 196–205. ACM SIGSAC (2001)

38. Shoup, V.: On Fast and Provably Secure Message Authentication Based on Universal Hashing. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 313–328. Springer, Heidelberg (1996)
39. van Tilborg, H.: Encyclopedia of cryptography and security. Springer (2005)
40. Whiting, D., Schneier, B., Lucks, S., Muller, F.: Phelix – fast encryption and authentication in a single cryptographic primitive. ECRYPT Stream Cipher Project, Report 2005/020 (2005), `http://www.ecrypt.eu.org/stream`
41. Wu, H., Preneel, B.: Differential-Linear Attacks Against the Stream Cipher Phelix. In: Biryukov, A. (ed.) FSE 2007. LNCS, vol. 4593, pp. 87–100. Springer, Heidelberg (2007)