Cryptanalysis of the OKH Authenticated Encryption Scheme

Peng Wang[1] and Wenling Wu[2] and Liting Zhang[1]

[1] State Key Laboratory of Information Security
Institute of Information Engineering of Chinese Academy of Sciences
[2] Institution of Software of Chinese Academy of Sciences
wp@is.ac.cn, {wwl,zhangliting}@is.iscas.ac.cn

Abstract. Alomair proposed a new authenticated encryption scheme OKH at ACNS 2012, and proved its security, i.e. authenticity and privacy. Our research shows that it is not the case. We only need one query to break the authenticity of OKH with success probability of 1, and two queries to break the privacy of OKH with success probability of $1-1/2^n$, where n is the block-length of underlying blockcipher.

Keywords. authenticated encryption, universal hash function family, cryptanalysis.

1 Introduction

Authenticated encryption (AE) schemes achieve the security functions of message authentication codes and that of encryption schemes at the same time, i.e. authenticity and privacy. Simply speaking, authenticity guarantees that the ciphertext is really delivered from the sender and not modified by the adversary during the transmission. Privacy guarantees that the adversary can not gain any information (except the length) about plaintext from the view of ciphertext. Due to its wide applications, during the past few years, considerable effort has been made to construct AE schemes, e.g. IAPM [7], OCB [11], CCM [12], EAX [2], CWC [8], GCM [9].

A straightforward method to construct AE schemes is by composition of an encryption scheme and a message authentication code (MAC). Three generic compositions are involved: Encrypt-and-MAC (E&M), MAC-then-Encrypt (MtE), and Encrypt-then-MAC (EtM). CCM [12], EAX [2], CWC [8] and GCM [9] can be viewed as composed (two-pass) schemes with refinement of using only one key. The other method is constructing integrated (one-pass) schemes, such as IAPM [7] and OCB [11].

Recently, Alomair proposed a new composed AE scheme OKH [1] in E&M style. The main observation of OKH is that, in the E&M or EtM scheme, the security requirements of authenticity can be relaxed, which can improve the efficiency of the overall construction.

Typical MACs are based on blockciphers, such as CBC-MAC [5], CMAC [10] and PMAC [4], but more efficient MACs are based on universal hash function families, in which the message is first compressed into a fixed-length string by a universal hash function and then encrypted to be the tag, e.g. UMAC [3], and MACs in CWC [8] and GCM [9]. The universal hash function family is a group of hash functions without any cryptographic requirement, but satisfying some combinatorial properties. The MAC in OKH is also based on a hash function family called the Odd Key Hash Family, but as mentioned by the author [1] it does not even satisfy the basic property of universal hash family, i.e. property of being universal[3].

Alomair proved the security of OKH by the reduction method with the assumption that the underlying blockcipher is a pseudorandom permutation (PRP). Unfortunately it is not true.

Our Contributions. In this paper we show that both authenticity and privacy of OKH do not hold in the usual security models. As to authenticity, we only need to query a special message to the encryption algorithm of OKH, and then forgery a new ciphertext and its tag that can pass the decryption algorithm of OKH successfully with probability of 1. As to authenticity, we only need to query two special messages to distinguish the ciphertexts from the random strings with probability of $1 - 1/2^n$, where n is the block-length of underlying blockcipher.

2 Description of OKH

2.1 Notations

- For a binary string M, $|M|$ denotes the length of M in bits.
- For a non-empty set $\mathcal{S}$, we denote by $s \overset{\$}{\leftarrow} \mathcal{S}$ the selection of a member of $\mathcal{S}$ uniformly at random and assigning it to s.
- A *blockcipher* is a function $E : \{0,1\}^{kl} \times \{0,1\}^{bl} \to \{0,1\}^{bl}$, where bl and kl are the block-length and key-length respectively, and $E_K(\cdot) = E(K, \cdot)$ is a permutation for all $K \in \{0,1\}^{kl}$.

[3] A hash function family $F = \{f_K | K \in \mathcal{K}\}$ is ε-almost-universal if $\#\{K | f_K(X) = f_K(Y)\}/\#\mathcal{K} \leq \varepsilon$ for any $X \neq Y$.

- $X \oplus Y$ denotes the *exclusive or* (XOR) of two string X and Y. When the lengths of X and Y are not equal, we pad some 0s after the short one to make them equal and then do the usual XOR operation. E.g. $11 \oplus 1001 = 1100 \oplus 1001 = 0101$.
- We denote by $\cdot$ the *multiplication*, 0^n the n-bit string of all 0s, $\mathbb{Z}_{2^n} = \{0, 1, 2, \cdots, 2^{n-1}\}$ the set of all none-negative integers less than 2^n, and $\mathbb{Z}^*_{2^n}$ the set of all odd integers in $\mathbb{Z}_{2^n}$. Without confusion, we often use a string or a integer number interchangeably.

2.2 The Odd Key Hash function family

The Odd Key hash function family is a crucial component of OKH, which makes use of basic modular arithmetic operations within $\mathbb{Z}_{2^n}$. For an input message M with bit-length of multiples of n, partition it into a sequence of n-bit blocks, $M = M_1 M_2 \cdots M_l$, then the compressed image of M is given by

$$\text{OK-HASH}_{K_h}(M) = \sum_{i=1}^{l} K_i \cdot M_i \mod 2^n,$$

where the key $K_h = K_1 \cdots K_l$, $K_i \in \mathbb{Z}^*_{2^n}$, $i = 1, \cdots, l$.

Remark 1. As mentioned in [1], OK-HASH is not an almost universal hash function family, because $\text{OK-HASH}_{K_h}(0^{2n}) = \text{OK-HASH}_{K_h}(10^{n-1}10^{n-1})$ for any K_h.

2.3 OKH Authenticated Encryption

OKH is a nonce-based authenticated encryption scheme, combining an encryption scheme $\mathcal{SE} = (\mathcal{E}, \mathcal{D})$ and a special message authentication code OK-MAC based on the function family OK-HASH mentioned above. We can view OKH as an E&M composition AE scheme, and divide it into three components: OKH = (OKH.Key, OKH.Enc, OKH.Dec), where OKH.Key is a key generation algorithm, OKH.Enc is an encryption algorithm and OKH.Dec is a decryption algorithm.

Key Generation Algorithm. OKH has two keys, one for $\mathcal{SE}$, one for OK-MAC, denoted as K_e and K_h respectively, which are generated independently.

Encryption Algorithm. Both $\mathcal{SE}$ and OK-MAC only handle messages of full blocks. In order to treat arbitrary length messages, OKH.Enc pads the bit 1 and minimal bits of 0, making the length of the messages be multiples of block-length. We simply write the result of this procedure as $M10^*$. OKH uses $\mathcal{E}$ to get the ciphertext and OK-MAC to get the authentication tag:

$$\text{OKH.Enc}_{K_e,K_h}(N,M) = (\mathcal{E}_{K_e}(N,M), \text{OK-MAC}_{K_h,K_e}(N,M)),$$

where $\mathcal{E}$ encrypts the message block by block using the underlying blockcipher with different key $K_e \oplus (N||i)$,

$$\mathcal{E}_{K_e}(N,M) = E_{K_e \oplus (N||1)}(M_1)||E_{K_e \oplus (N||2)}(M_2)||\cdots||E_{K_e \oplus (N||l)}(M_l 10^*),$$

and

$$\text{OK-MAC}_{K_h,K_e}(N,M) = E_{K_e}(\text{OK-HASH}_{K_h}(M10^*) \oplus N),$$

when the length of input to the blockcipher E_{K_e} is less than one block, we pad some zeros to fill it.

Decryption Algorithm. OKH.Dec recovers the plaintext, and uses OK-MAC to regenerate the tag to decide whether to return the plaintext or not.

$$\text{OKH.Dec}_{K_e,K_h}(N,C,T) = \begin{cases} \perp & \text{if OK-MAC}_{K_h,K_e}(N,M) \neq T, \\ M & \text{else, where } M = \mathcal{D}_{K_e}(N,C), \end{cases}$$

where $\mathcal{D}$ is the inverse of $\mathcal{E}$,

$$\mathcal{D}_{K_e}(N,C) = D_{K_e \oplus (N||1)}(C_1)||D_{K_e \oplus (N||2)}(C_2)||\cdots||D_{K_e \oplus (N||l)}(C_l).$$

As a summery, we conclude OKH in pseudocodes as in fig. 1, and illustrate it in fig. 2.

```
Key Generation:

OKH.Key
    $K_e \xleftarrow{\$} \{0,1\}^{kl}$
    $K_i \xleftarrow{\$} \mathbb{Z}^*_{2^n}$, $i = 1, \cdots, l$
```

```
OKH Encryption:

OKH.Enc$_{K_e,K_h}(N,M)$
    $C \leftarrow \mathcal{E}_{K_e}(N,M)$
    $T \leftarrow$ OK-MAC$_{K_h,K_e}(N,M)$
    return $(C,T)$
```

```
OKH Decryption:
OKH.Dec$_{K_e,K_h}(N,C,T)$
    $M \leftarrow \mathcal{D}_{K_e}(N,C)$
    if $T \neq$ OK-MAC$_{K_h,K_e}(N,M)$
        then return $\perp$
    return $M$
```

Fig. 1. The pseudocodes of OKH Authenticated Encryption.

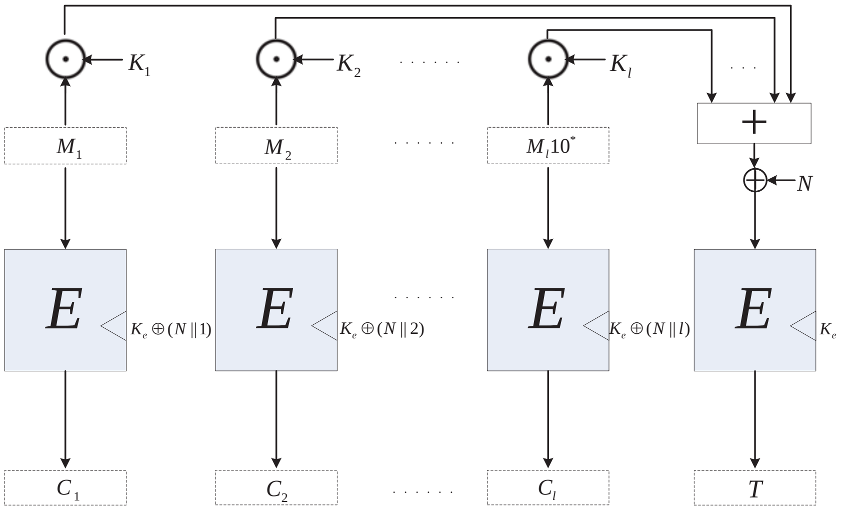

Fig. 2. The OKH Authenticated Encryption Scheme, when the block-length of the underlying blockcipher and that in the OK-HASH are equal.

Remark 2. In the original description of OKH [1], the block-length of the underlying blockcipher and that in the OK-HASH may not be equal, the former is no less than the later. In the following discussion, we only consider the situation that the two lengths are equal (i.e. $bl = n$), just as illustrated in fig. 2.

3 Security Models

We adopt the standard security models as those mentioned in [1].

Authenticity Model. The adversary $\mathcal{A}$ is given oracle access to the encryption algorithm OKH.Enc. $\mathcal{A}$ queries OKH.Enc with a pair of nonce and message with restriction that he never repeats the nonce, or in other words he is nonce-respecting, observing the output. After some queries (current query may depend on past queries), he returns a triple of nonce, ciphertext and tag (N, C, T), which does not appear before in the previous answers to the queries. If (N, C, T) is valid, i.e. OKH.Dec$(N, C, T) \neq \perp$, we say that $\mathcal{A}$ makes a successful forgery. Formally, the advantage of $\mathcal{A}$ is defined by

$$\mathbf{Adv}^{auth}_{\mathrm{OKH}}(\mathcal{A}) = \Pr[\mathcal{A}^{\mathrm{OKH.Enc}(\cdot,\cdot)} \text{ forges}].$$

Privacy Model. The nonce-respecting adversary $\mathcal{B}$ is also given oracle access to the encryption algorithm OKH.Enc. $\mathcal{B}$ queries OKH.Enc with pairs of nonce and message, observing the outputs, trying to distinguish it from random bits. Formally, the advantage of $\mathcal{B}$ is defined by

$$\mathbf{Adv}^{priv}_{\mathrm{OKH}}(\mathcal{B}) = |\Pr[\mathcal{B}^{\mathrm{OKH.Enc}(\cdot,\cdot)} \Rightarrow 1] - \Pr[\mathcal{B}^{\$(\cdot,\cdot)} \Rightarrow 1]|,$$

where $\$(N, M)$ returns a random string with the same length of OKH.Enc(N, M).

4 Cryptanalysis of OKH

4.1 Some Properties

We first notice some properties of a special binary integer number 10^{n-1} in $\mathbb{Z}_{2^n}$.

Property 1. For any odd integers K_i and K_j,

$$10^{n-1} \cdot K_i \equiv 10^{n-1} \mod 2^n, \tag{1}$$

$$10^{n-1} \cdot K_i + 10^{n-1} \cdot K_j \equiv 0^n \mod 2^n. \tag{2}$$

Using these properties, we construct two pairs of messages which have the same authentication tag under OK-MAC with the same or different nonces.

Property 2. For arbitrary blocks $M_i \in \{0, 1\}^n$ $(i = 1, \cdots, l)$, we have

$$\text{OK-MAC}(N, M_1 \cdots M_l 10^{n-1}10^{n-1}) = \text{OK-MAC}(N, M_1 \cdots M_l), \tag{3}$$

$$\text{OK-MAC}(N, M_1 \cdots M_l 10^{n-1}) = \text{OK-MAC}(N', M_1 \cdots M_l), \tag{4}$$

where $N \oplus N' = 10^{nl-1}$, nl is the length of the nonce.

Proof. It is easy to verify the following two equations about OK-HASH,

$$\text{OK-HASH}(M_1 \cdots M_l 10^{n-1}10^{n-1}10^{n-1}) = \text{OK-HASH}(M_1 \cdots M_l 10^{n-1}),$$

$$\text{OK-HASH}(M_1 \cdots M_l 10^{n-1}10^{n-1}) = \text{OK-HASH}(M_1 \cdots M_l 10^{n-1}) \oplus 10^{n-1}.$$

By the definition of OK-MAC, the equations of (3) and (4) follow. □

So if we look at the authentication code in OKH solely, OK-MAC is not a secure MAC, due to the fact that OK-HASH is not almost universal. We can query the MAC using one message, and get the tag, then the other message and the tag constitute a successful forgery immediately. But breaking authenticity of AE scheme is slightly different, what the adversary tries to find is a valid triple of nonce, ciphertext and tag which does not appear before. But we notice that in equation (3) $M_1 \cdots M_l$ is the prefix of $M_1 \cdots M_l 10^{n-1}10^{n-1}$, which will help us to break the authenticity of OKH.

4.2 Breaking Authenticity of OKH

We give the following authenticity attacking algorithm. This attack only makes one special query to the encryption oracle $\text{OKH.Enc}_{K_e,K_h}(\cdot,\cdot)$, then returns a valid triple of nonce, ciphertext and tag which does not appear before.

Authenticity attacking algorithm $\mathcal{A}$:

1) Query $(N, M_1 \cdots M_l 10^{n-1}10^{n-1})$ to the encryption oracle, where M_i $(i = 1, \cdots, l)$ are arbitrary blocks, and get $(C_1C_2 \cdots C_{l+3}, T)$, where C_i $(i = 1, \cdots, l+3)$ are ciphertext blocks, T is the tag.
2) Return $(N, C_1C_2 \cdots C_{l+1}, T)$.

Analysis of algorithm $\mathcal{A}$. The ciphertext blocks to the query are $C_i = E_{K_e \oplus (N||i)}(M_i)$, $i = 1, \cdots, l$, $C_{l+1} = E_{K_e \oplus (N||(l+1))}(10^{n-1})$, $j = 1, 2, 3$. So the corresponding plaintext blocks of C_i $(i = 1, \cdots, l)$ and C_{l+1} under same nonce N are M_i $(i = 1, \cdots, l)$ and 10^{n-1}. 10^{n-1} is interpreted as the padding, therefore the final plaintext is $M_1 \cdots M_l$. Equation (3) shows that the tags of $(N, M_1 \cdots M_l 10^{n-1}10^{n-1})$ and $(N, M_1 \cdots M_l)$ are the same. So $(N, C_1C_2 \cdots C_{l+1}, T)$ is valid, which does not appear before. Therefore $\mathbf{Adv}^{auth}_{\text{OKH}}(\mathcal{A}) = 1$.

Remark 3. In the proof of authenticity in [1], the author did not consider the situation that one plaintext may be the prefix of the other. The security proof lies on the fact that the corresponding plaintexts of two different ciphertext differ in single block or several blocks. This is obvious not true.

4.3 Breaking Privacy of OKH

In equation (4), two messages have the same authentication tag under different nonces. Therefore we can make two nonce-respecting queries, resulting in two equal tags, which can be used to distinguish ciphertexts from random strings.

Privacy attacking algorithm $\mathcal{B}$:

1) Query $(N, M_1 \cdots M_l 10^{n-1})$, and get $(C_1C_2 \cdots C_{l+2}, T)$.
2) Query $(N', M_1 \cdots M_l)$ where $N \oplus N' = 10^{nl-1}$, and get $(C'_1C'_2 \cdots C'_{l+1}, T')$.
3) If $T = T'$, then return 1, else return 0.

Analysis of algorithm $\mathcal{B}$. If the oracle is $\text{OKH.Enc}(\cdot,\cdot)$. By equation (4), we know that $T = T'$ always holds. If the oracle is $\$(\cdot,\cdot)$, T and T' are two random strings. The probability of $T = T'$ is $1/2^n$, therefore $\mathbf{Adv}^{priv}_{\text{OKH}}(\mathcal{B}) = 1 - 1/2^n$.

Remark 4. In the current real-or-random privacy model, OKH is totally insecure. We note that even in a more general left-or-right privacy model, OKH is not secure. In this model, the adversary can query $(N, M), (N, M')$ with restriction that $|M| = |M'|$, the oracle only returns left or right ciphertext, and after several queries the adversary must guess this one-bit information about left-or-right. In this model, the adversary can attack as following: 1) Query $(N, M_1 \cdots M_l 10^{n-1}), (N, M_1 \cdots M_l 0^n)$; 2) Query $(N', M_1 \cdots M_l), (N', M_1 \cdots M_l)$ where $N \oplus N' = 10^{nl-1}$; 3) If the two returned tags are equal, the adversary guesses it is left, else guesses right. It is easy to verify that the success probability is 1.

Remark 5. The proof of privacy in [1], the security lies only on the pseudorandom of the underlying blockcipher, which assumes that once the key of the blockcipher is randomly selected, the blockcipher is indistinguishable from a uniformly random permutation, i.e. the blockcipher is a pseudorandom permutation (PRP). The encryption component of OKH is similar to the ECB mode, with exception that the keys to the underlying blockcipher are $K_e \oplus (N||i)$ $(i = 1, \cdots, l)$, which are all related by the key K_e. The only assumption of PRP can not guarantee the independence between the blockcipher invocations. With the assumption of PRP, it is easy to construct a new block cipher like [6], which is also a PRP, but the same under two different related keys such as $K_e \oplus (N||1)$ and $K_e \oplus (N||2)$. Then first two block encryptions of OKH are the same, which also can be used to break the privacy of OKH. Algorithm $\mathcal{B}$ only makes use of the weakness of OKH-MAC.

5 Conclusion

Although the security proofs were given in [1], the OKH authenticated encryption scheme is not secure at all. Both authenticity and privacy of OKH do not hold in the common security models. We only need one or two queries to break the security of OKH with success probability of 1 or almost 1.

References

1. Alomair, B.: Authenticated encryption: How reordering can impact performance. In: Bao, F., Samarati, P., Zhou, J. (eds.) ACNS. Lecture Notes in Computer Science, vol. 7341, pp. 84–99. Springer (2012)
2. Bellare, M., Rogaway, P., Wagner, D.: The EAX mode of operation. In: Roy, B.K., Meier, W. (eds.) FSE. Lecture Notes in Computer Science, vol. 3017, pp. 389–407. Springer (2004)

3. Black, J., Halevi, S., Krawczyk, H., Krovetz, T., Rogaway, P.: UMAC: Fast and secure message authentication. In: Wiener, M.J. (ed.) CRYPTO. Lecture Notes in Computer Science, vol. 1666, pp. 216–233. Springer (1999)
4. Black, J., Rogaway, P.: A block-cipher mode of operation for parallelizable message authentication. In: Knudsen, L.R. (ed.) EUROCRYPT. Lecture Notes in Computer Science, vol. 2332, pp. 384–397. Springer (2002)
5. FIPS-133: Federal information processing standards publication (FIPS 133). computer data authentication (1985)
6. Iwata, T., Kurosawa, K.: On the correctness of security proofs for the 3GPP confidentiality and integrity algorithms. In: Paterson, K.G. (ed.) Cryptography and Coding. Lecture Notes in Computer Science, vol. 2898, pp. 306–318. Springer (2003)
7. Jutla, C.S.: Encryption modes with almost free message integrity. In: Pfitzmann, B. (ed.) EUROCRYPT. Lecture Notes in Computer Science, vol. 2045, pp. 529–544. Springer (2001)
8. Kohno, T., Viega, J., Whiting, D.: CWC: A high-performance conventional authenticated encryption mode. In: Roy, B.K., Meier, W. (eds.) FSE. Lecture Notes in Computer Science, vol. 3017, pp. 408–426. Springer (2004)
9. McGrew, D.A., Viega, J.: The galois/counter mode of operation (GCM) (2004), http://csrc.nist.gov/groups/ST/toolkit/BCM/
10. NIST: Recommendation for block cipher modes of operation: The CMAC mode for authentication. NIST Special Publication 800-38B (2005), http://csrc.nist.gov/publications/nistpubs/800-38B/SP_800-38B.pdf
11. Rogaway, P., Bellare, M., Black, J., Krovetz, T.: OCB: a block-cipher mode of operation for efficient authenticated encryption. In: Reiter, M.K., Samarati, P. (eds.) ACM Conference on Computer and Communications Security. pp. 196–205. ACM (2001)
12. Whiting, D., Housley, R., Ferguson, N.: Counter with CBC-MAC (CCM) (2002), http://csrc.nist.gov/groups/ST/toolkit/BCM/