ATLANTIS PRESS

Advances in Engineering, volume 126

5th International Conference on Machinery, Materials and Computing Technology (ICMMCT 2017)

The FCM Scheme for Authenticated Encryption

Xiaomei Lei[1, a], Zhongdong Wu[1, b], Jiu Yong[1, c]

[1] School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China

[a]email: leixiaomei6629@sina.com; [b]email: 2395570019@qq.com; [c]email: 1527983509@qq.com

Keywords: Authenticated encryption, compression function, FMAC, Galois/Counter Mode

Abstract. We propose an advanced authenticated encryption with associated data (AEAD) scheme called FCM (compression Function/Counter Mode) based on a compression function required to be a pseudorandom function (PRF) against related key attacks. FCM adopts the stream cipher generated by parallel compression function to encrypt plaintexts, and then utilizes compression Function-based MAC (FMAC) to produce the authentication tag, in which FMAC is a variant of the Merkle-Damgård construction with a permutation. In this structure, FCM successfully avoids introducing the string representing lengths of plaintext and associated data in the generation of the authentication tag, which is a shortage in Galois/Counter Mode (GCM) especially for short message. Then we analyze the security of FCM from two aspects of encryption and authentication based on secure underlying primitives. At last features of FCM are summarized and compared with GCM.

I. Introduction

Authenticated encryption (AE) is a block cipher mode of operation in symmetric encryption, which provides confidentiality, integrity and authenticity assurances on the data. In many applications we have a mixture of secret and non-secret data, thus authenticated encryption with associated data (AEAD) based on a nonce was proposed to provide privacy for the secret data and authenticity for both types of data [1] [3]. For an AEAD scheme, the authenticated encryption algorithm AE_K with secret key K takes the non-secret data called an associated data A, a plaintext P, a nonce value IV as inputs, and outputs a ciphertext C and a tag T. That is $AE_K(A,P,IV)\rightarrow(C,T)$. Conversely, the decryption algorithm DE_K is expressed as $DE_K(A,C,IV,T)\rightarrow P/\perp$ to output either plaintext P or an error flag $\perp$ if the certification is not successful.

As a famous AEAD scheme, GCM (Galois/Counter Mode) [6] was proposed in 2004. It uses Counter mode which can output stream ciphers in parallel to encrypt plaintexts. In research of GCM, we found that the length-bits of the protected data are also involved as inputs. In GCM-AES128, the length-bits are 128 bits which are not so large accounted for in the entire data, but for short message this is inefficient. It is precisely because of this, that we propose a new authenticated encryption algorithm FCM (compression Function/Counter Mode) which still keeps parallel design in encryption but improves the authentication part in GCM.

In an excellent AEAD scheme, the generation of corresponding MAC (Message Authentication Code) is a very important part. We can construct a MAC from block ciphers just as CBC-MAC (Cipher Block Chaining-MAC) or hash functions just as HMAC (Hashed Message Authentication Code) [9], which is very popular in practice. However, the structure for HMAC invokes the underlying hash function twice. The drawback of this operation is its inefficiency for short messages. Inefficiency of HMAC may also come from the padding of the underlying hash function based on the Merkle-Damgård strengthening. Recently the efficient scheme FMAC [7] (compression Function-based MAC) was proposed based on a PRF with minimum padding, which is constructed from a compression function of a hash function instead of the hash function itself. Because of the simple structure and provable security of FMAC, we think out a scheme that uses FMAC to complete the authentication in GCM. Meanwhile, the underlying block cipher in Counter mode is replaced with a PRF, namely the compression function used in FMAC. The new proposed scheme is called FCM, which is efficient especially for short message.

This paper is organized as follows. Section II provides the detailed construction of FMAC, in addition to related operations, notations and definitions. Section III proposes a new AEAD scheme FCM, after analyzing the shortage in GCM. Section IV discusses the security of FCM, especially against the length extension attacks. The last Section V shows the conclusion and features of FCM.

II. Preliminary

A. Operations

In this paper, bitwise exclusive OR is denoted as $\oplus$. For two binary strings X and Y, $X \oplus Y$ is a string whose length is equal to length of the shorter string and calculated from left-most bits, for example $1001 \oplus 101 = 001$. Clearly, if X and Y have the same length then $X \oplus Y$ simply means their usual bitwise XOR.

The symbol $||$ indicates concatenation among strings, for example $101 || 1001 = 1011001$.

The operation $\mathrm{MSB}_t(X)$ indicates to keep the left-most t bits of X, for example $\mathrm{MSB}_5(00110111010) = 00110$.

B. Definitions and Abbreviations

The abbreviations and some definitions are defined below.

P	Plaintext	*w*	One input length of compression function
M	Message	*n*	The other input length of compression function
C	Ciphertext	*K*	*n*-bit secret key
T	Authentication tag	T_A	Authentication tag for associated data
A	Associated data	*len(X)*	The bit length of the bit string *X*
IV	Initialization vector	*[x]*	The least integer more than the real number *x*

Let $\Sigma = \{0,1\}$. We define Σ^n as the set of all Σ-sequences of length n, and Σ^* to represent any length of Σ-sequences. The definition of Σ^n is used to denote a compression function by $F : \Sigma^n \times \Sigma^w \to \Sigma^n$, where n and w are two positive integers. The function indicates that on input an n-bit chaining block and a w-bit message block, it outputs an n-bit string. We suppose that the compression function F is a PRF.

In this paper, we adopt the minimum padding method. 0^a represents the bit string that consists of a '0' bits. For the associated data, if the length $len(A)$ is not a multiple of w bits, a '1' bit is padded followed by the least number of '0' bits to make the length be multiple of w bits. This padding scheme is also used for the padding of ciphertexts.

C. FMAC

Merkle-Damgård (MD) construction is often used to design hash function iteratively. As the most popular hash-based MAC, HMAC is also based on the Merkle-Damgård structure. Unfortunately, the construction is not secure and up against length extension attacks if the underlying hash function is an MD hash function.

To construct secure schemes, Hirose proposed an improved MD construction MDP [8]. MDP (Merkle-Damgård with Permutation) is a variant of the MD construction with a permutation applied right before the processing of the last message block, which can produce a PRF if the underlying compression function is PRF against related-key attacks with respect to the permutations. Based on this, FMAC (compression Function-based MAC) is proposed as an application of MDP.

The proposed FMAC consists of the compression function $F : \Sigma^n \times \Sigma^w \to \Sigma^n$ and two distinct permutations π_1 and π_2 on Σ^n. Specifically the permutation π_i is defined as

$$\pi_i = x \oplus a_i,\ i = 0,1,2\cdots \tag{1}$$

In (1) a_i is a nonzero binary constant bit string with the same length of x which is the input of π_i.

In FMAC algorithm, there are only two inputs: the secret key K on Σ^n and the message M.

For any $M \in \Sigma^*$, M is denoted depending on whether the length is a multiple of *w* bits or not.

$$M = \begin{cases} M_1 \| M_2 \| \cdots \| M_{m-1} \| M_m^* & len(M_m) < w \\ M_1 \| M_2 \| \cdots \| M_{m-1} \| M_m & len(M_m) = w \end{cases} \tag{2}$$

In (2), the length of every block M_i is equal except the last block, namely $len(M_i) = w(1 \le i < m)$ and $m = [len(M)/w]$. The length of last block is shorter than *w*. We express the padded block $M_m^* = M_m \| 10^{w-len(M_m)-1}$ with the minimum padding method.

The input M of FMAC function is divided into *w*-block M_i to meet one input size of function $F : \Sigma^n \times \Sigma^w \to \Sigma^n$, the other *n*-bit input *K* satisfies $K \in \Sigma^n$. FMAC function is defined with two different permutations π_1 and π_2 as follows. For any $M \in \Sigma^*$, $K \in \Sigma^n$,

$$C^{F,\{\pi_1,\pi_2\}} : \Sigma^n \times \Sigma^* \to \Sigma^n = \begin{cases} I^{F,\pi_1}(K, M_1 \| M_2 \| \cdots \| M_{m-1} \| M_m^*) & len(M_m) < w \\ I^{F,\pi_2}(K, M_1 \| M_2 \| \cdots \| M_{m-1} \| M_m) & len(M_m) = w. \end{cases} \tag{3}$$

Equation (3) shows how to choose the permutation π_i. In the case where the message M needs to be padded, π_1 will be used; whereas in the case where M is not padded, π_2 will be used.

The structure of FMAC is shown in Fig.1. FMAC utilizes the function *F* to input the key *K* and block M_i in the same way until the last block. A permutation π_i is added to the front of last block. Then the output of permutation π_i and M_m or M_m^* are input into *F* to generate the tag *T*.

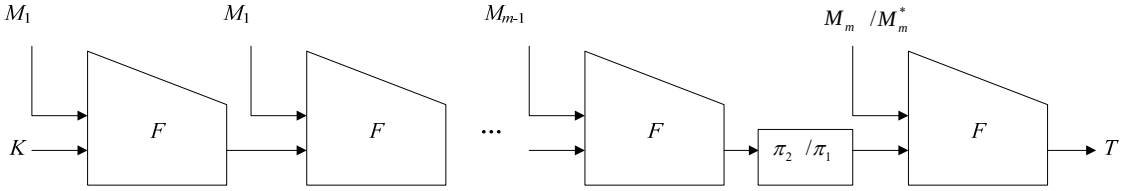

Fig.1. The structure of FMAC

III. FCM Scheme

A. GCM

Galois/Counter Mode (GCM) for AEAD is a block cipher mode of operation that uses universal hashing over a binary Galois field to provide authenticated encryption. GCM not only assure the confidentiality of message using a variation of the Counter mode of operation for encryption but also the authenticity of message using a universal hash function that is defined over a binary Galois field.

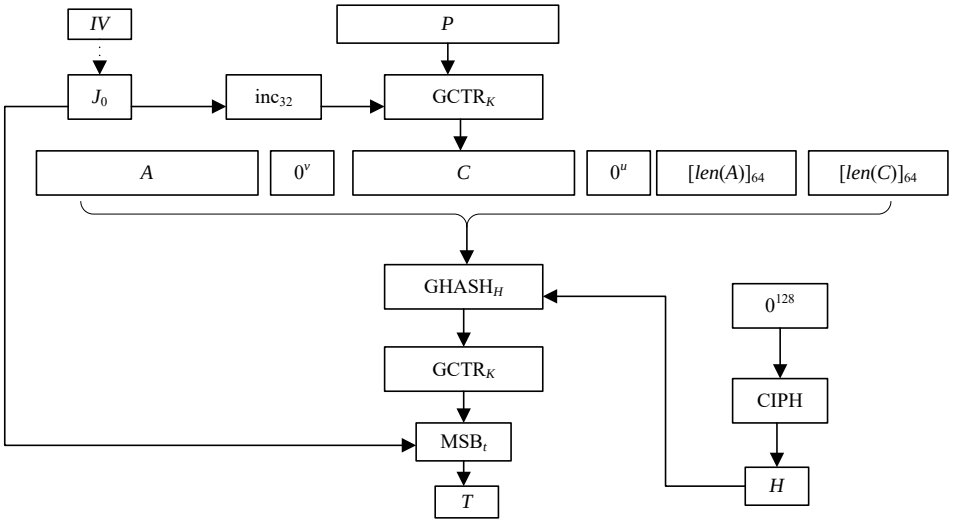

Fig.2. GCM-AE_K (*IV, P, A*) =(*C, T*)

Fig.2 illustrates GCM algorithm. From the structure of GCM, we can clearly see that the inputs

of GHASH must include ($[len(A)]_{64}$ and $[len(C)]_{64}$), in all 128 bits. This is not so optimal and efficient especially for the short message.

B. Authenticated Encryption Mode FCM

FCM utilizes mutually parallel compression functions in the layer $FCTR_{Ki}$ with initialization vector (*IV*) to produce the ciphertext *C*. Each *IV* value must be distinct, but need not have equal lengths, because it can be padded with the least '0' to be *w* bits. For the associated data *A*, it can be processed in the module FMAC to get T_A in advance. Then *C* and T_A are input to the other similar module FMAC to generate the final authentication tag *T* after truncated. Fig.3 shows the overview structure of FCM.

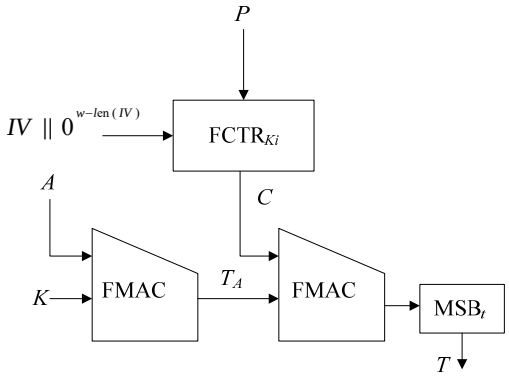

Fig.3. Overview structure of FCM

C. $FCTR_{Ki}$ for Encryption

In $FCTR_{Ki}$ we use the compression function $F:\Sigma^n\times\Sigma^w\to\Sigma^n$. In particular, the function here is recorded as $F(Ki,ctri)$ with an *n*-bit key and a *w*-bit counter value as inputs. We define $Ki=K\oplus b_i$, $b_i=b+i$, where b is a nonzero constant and $0\le i\le f$. Let $f=[len(P)/n]$, the plaintext P is denoted as $P=P_1\|P_2\|\cdots P_{f-1}\|P_f$, where $len(P_j)=n$, but for the last block P_f, $len(P_f)\le n$. The encryption steps are described as below:

$$\begin{aligned} &ctr0=IV\|0^{w-len(IV)}\\ &ctri=ctr(i-1)+1,\ 0\le i\le f\\ &C_j=P_j\oplus F(Ki,ctri),1\le j\le f \end{aligned} \tag{4}$$

The initialization vector (*IV*) is random or pseudorandom every time, padded to be $IV\|0^{w-len(IV)}$ as the initial value of the counter. Counter value *ctri* and *Ki* are input into *F* to produce *n*-bits stream cipher to do XOR with corresponding plaintext in parallel. In (4), if the length of last block $len(P_f)<n$, there is $len(C_f)=len(P_f)<n$ and no need to pad to be a full block. Finally, $FCTR_{Ki}$ outputs the ciphertext

$$C=C_1\|C_2\|\cdots C_{f-1}\|C_f \tag{5}$$

Then *C* is input to the authenticated layer FMAC. The $FCTR_{Ki}$ algorithm is shown in Fig.4.

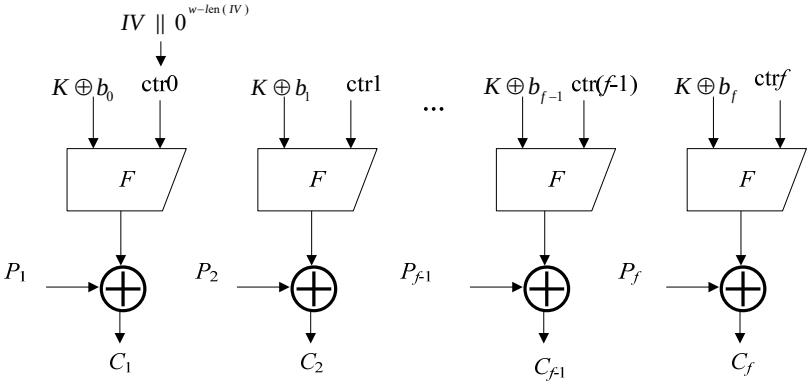

Fig.4. The $FCTR_{Ki}$ algorithm

D. Authentication based on FMAC

The authentication algorithm in FCM can use any secure and efficient compression function as underlying primitive such as SHA-3, SHA-256 and SHA-512. The compression function *F* here is the same with *F* in FCTR$_{Ki}$. We connect two FMAC skillfully to design the authentication algorithm. Fig.5 describes the authentication algorithm for FCM.

To satisfy one input length *w* of function *F*, the associated data *A* without length limitation is indicated as $A = A_1 \| A_2 \| \cdots A_{m-1} \| A_m$, where $m = [len(A)/w]$. The other input of *F* is the *n*-bit secret key. Firstly, the *w*-bit A_1 and *n*-bit *K* are input to the first function *F* and output an *n*-bit string as the input of the second function *F* with the other input A_2. The sequent associated data is compressed with the other input from the output of previous *F* in the same way except the last A_m. For A_m, there are two cases: $len(A_m)=w$ or $len(A_m)<w$. And before the last compression function for associated data, permutation π_i is used with an *n*-bit input and output. In the first case: $len(A_m)=w$, permutation π_2 is chosen to output an *n*-bit string. Then the *n*-bit string and the full block A_m are compressed by function *F* to generate the tag for associated data *A*,

$$T_A = \mathrm{FMAC}(K, A) \tag{6}$$

In the second case: $len(A_m)<w$, A_m is padded to be $A_m^* = A_m \| 10^{w-len(A_m)-1}$ and the permutation π_1 is chosen together with A_m^* to generate the tag T_A. We point that if *A* is fixed, T_A can be computed in advance.

Subsequently, the ciphertext *C* from the layer FCTR$_{Ki}$ and T_A are processed in a similar FMAC structure. Same as the associated data, the ciphertext *C* in (5) is also partitioned into *w*-bit block,

$$C = C_{l1} \| C_{l2} \| \cdots C_{ls-1} \| C_{ls} \tag{7}$$

where $ls = [len(C)/w]$. Firstly, the first ciphertext block C_{l1} and the tag T_A are compressed in *F* function. The subsequent operation is similar to the above. There are also two cases: $len(C_{ls})=w$ or $len(C_{ls})<w$ and the choice of permutation π_i is the same. If $len(C_{ls})=w$, namely no padding, π_2 is adopted; if $len(C_{ls})<w$, the block C_{ls} needs to be padded as $C_{ls}^* = C_{ls} \| 10^{w-len(C_{ls})-1}$, and π_1 is used. After iterative computation, we obtain authentication tag *T*,

$$T = \mathrm{MSB}_t[\mathrm{FMAC}(T_A, C)] \tag{8}$$

Then the data (*A*, *IV*, *C*, *T*) is sent to the receiving end.

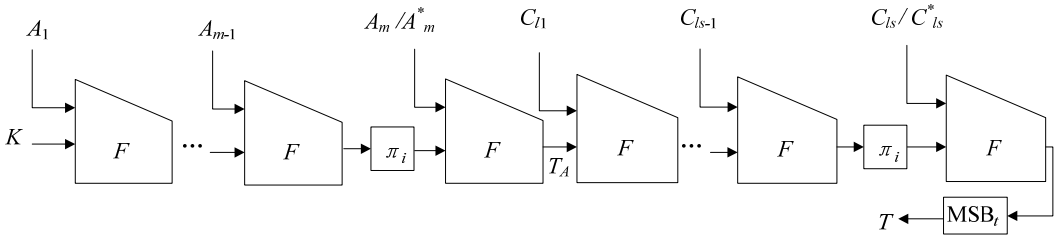

Fig.5. The authentication algorithm for FCM

E. The Decryption and Verification of FCM

The decryption is similar to the encryption except that the ciphertext block *C* is XORed with the stream cipher produced by $F(Ki, ctri)$ to compute the corresponding plaintext *P*.

For verification, a tag *T'* is generated in the same way, using the received data (*A*, *C*, *IV*). Before the output of plaintexts, the generated *T'* is compared with the received *T*. If two tags match, the receiver will output original plaintext *P*.

IV. The Security Analysis

In this section, we analyze the security of FCM which is on the basis of well-established security

properties of the underlying encryption mode and authentication method.

A. The Security Analysis of FCM

In encryption of FCM, we adopt parallel compression functions to produce ciphertexts under the assumption that the compression function is a PRF. Although the $FCTR_{Ki}$ structure is varied from Counter mode (CTR) based on a block cipher, the security of $FCTR_{Ki}$ does not weaken. It is similar to the security of CTR mode which has been proven by Bellare et. al. [2] using the assumption that the block cipher is indistinguishable from a pseudorandom permutation (PRP).

B. The Security of Message Authentication

Usually the Merkle-Damgård (MD) structure is up against the length extension attack. It is a type of attack where an attacker can use the output of message M_1: hash(M_1) and the length of M_1 to calculate the output of the message $M_1 \| M_2$: hash($M_1 \| M_2$). This attack can be used to sign a message altered by attacker when a MD based hash function is used as a message authentication code. In FMAC the introduction of a permutation cleverly avoids the length extension attack. We explain how to maintain security against length extension attack in FMAC in below.

We suppose that the adversary B attempts to add extra message M_2 through intercepting the MAC of message M_1 to obtain a MAC of (M_1, M_2), and then use the MAC to do authentication for message $M_1 \| M_2$. We use the definition of FMAC to analyze whether B can succeed below.

M_1 and M_2 are assumed to be a multiple of *w* in order to simplify the situation, so that only the permutation π_2 is used in FMAC function (3). The followings are specific steps.

a) B intercepted an authentication tag: $I^{F,\pi_2}(K, M_1)$ and the message M_1.

b) B used a pair $[I^{F,\pi_2}(K, M_1), M_2]$ as inputs and calculated the tag of message (M_1, M_2).

$$T_{M_1,M_2} = I^{F,\pi_2}[I^{F,\pi_2}(K, M_1), M_2] \tag{9}$$

c) B sent T_{M_1,M_2} and $M_1 \| M_2$ to corresponding receiver to authenticate.

d) B can't pass authentication and failed. Because the correct authentication tag of $M_1 \| M_2$ is

$$T_{M_1\|M_2} = I^{F,\pi_2}(K, M_1 \| M_2) \tag{10}$$

From Fig.1, we can judge easily that $T_{M_1,M_2} \neq T_{M_1\|M_2}$ easily because of the introduction of the permutation π_2. But B mistook T_{M_1,M_2} for $T_{M_1\|M_2}$, so B failed. From the above analysis, we prove that authentication algorithm for FCM can prevent length extension attack.

V. Conclusion

We presented the structure of FCM and analyzed the security of encryption and authentication on FCM. All of these show that FCM is a secure and efficient AEAD scheme. The scheme has the following features:

- In the process of generating authentication tag, FCM avoids introducing the length bits of associated data and ciphertext compared with GCM. This improves the efficiency of FCM, especially for short message.
- FCM can resist length extension attack which is the problem faced by many hash function based on Merkle-Damgård structure.
- FCM is very flexible. Users can select the underlying compression function such as SHA-256 and SHA-512 according to demands.
- The usage of parallel compression function makes FCM parallelizable and relatively fast in hardware.
- The authentication tag for additional data can be pre-computed if the data is fixed.
- FCM requires only the forward direction because of the stream cipher in Counter mode.
- The authentication of the protected data can be verified independently from the decryption of the confidential data.

Acknowledgement

In this paper, the research was supported by the Basic Research Service Fund of University in Gansu Province under Grant No. 213056.

References

[1] P. Rogaway, "Authenticated-encryption with associated-data," in the 9th ACM conference on Computer and communications security. ACM, 2002, pp. 98-107.

[2] M. Bellare, et al, "A Concrete Security Treatment of Symmetric Encryption: Analysis of the DES Modes of Operation," in Proceedings of the 38th Symposium on Foundations of Computer Science, IEEE, 1997.

[3] T. Kohno, J. Viega, D. Whiting, "CWC: A high-performance conventional authenticated encryption mode," in Fast Software Encryption, vol. 3017, Springer Berlin Heidelberg, 2004, pp. 408-426.

[4] K. Yasuda, "Boosting Merkle-Damgård hashing for message authentication," in International Conference on the Theory and Application of Cryptology and Information Security, Springer Berlin Heidelberg, 2007, pp. 216-231.

[5] M. Nandi, "Fast and secure CBC-type MAC algorithms," in Fast Software Encryption. Springer Berlin Heidelberg, 2009, pp. 375-393.

[6] D. McGrew and J. Viega. "The Galois/counter mode of operation (GCM)," Submission to NIST, 2004.

[7] S. Hirose and Y. Atsushi, "A Tweak for a PRF Mode of a Compression Function and Its Applications," in the 9th International Conference on Security for Information Technology and Communications, 2016.

[8] S. Hirose, J. H. Park, and A. Yun, "A simple variant of the Merkle-Damgård scheme with a permutation," International Conference on the Theory and Application of Cryptology and Information Security. Springer Berlin Heidelberg, 2007.

[9] H. Krawczyk, R. Canetti, and M. Bellare. (1997) HMAC: Keyed-hashing for message authentication. RFC. [online]. Available: http://dl.acm.org/citation.cfm?id=RFC2104

[10] M. Bellare, "New proofs for NMAC and HMAC: Security without collision-resistance," in Annual International Cryptology Conference, Springer Berlin Heidelberg, 2006, pp. 113-129.

[11] D. Gligoroski, "Length extension attack on narrow-pipe SHA-3 candidates," in International Conference on ICT Innovations, Springer Berlin Heidelberg, 2010, pp. 5-10.

[12] H. Wu, "The hash function JH," Submission to NIST (round 3), 2011.

[13] K. Kurosawa, "Power of a public random permutation and its application to authenticated encryption," IEEE Transactions on Information Theory, 56.10, pp.5366-5374, 2010.

[14] D. Chakraborty and P. Sarkar, "A general construction of tweakable block ciphers and different modes of operations," IEEE Transactions on Information Theory, 54.5, pp. 1991-2006, 2008.

[15] S. Cogliani, et al. "OMD: a compression function mode of operation for authenticated encryption," International Workshop on Selected Areas in Cryptography, Springer International Publishing, 2014, pp. 112-128.

[16] E. Andreeva, et al. "Parallelizable and authenticated online ciphers," in International Conference on the Theory and Application of Cryptology and Information Security, Springer Berlin Heidelberg, 2013, pp. 424-443.