ATLANTIS PRESS
Advances in Intelligent Systems Research, volume 133
2nd International Conference on Artificial Intelligence and Industrial Engineering (AIIE2016)

Improved Collision Cryptanalysis of Authenticated Cipher MORUS

Tairong Shi *, Jie Guan, Junzhi Li and Pei Zhang
Information Science and Technology Institute, Zhengzhou, Henan, China
*Corresponding author

***Abstract*—MORUS is an authenticated stream cipher designed by Wu et al. and submitted for the third-round of the CAESAR competition. The collision properties of MORUS-640-128 are studied. We propose the necessary conditions for an internal state collision after two-step update, i.e., the Hamming weight of the input difference is at least 5 and the difference is distributed in at least three 32-bit words, which provide the theoretical support for MORUS's resistance against collision attack.**

Keywords-CAESAR; MORUS; collision cryptanalysis; partition method

I. INTRODUCTION

Authenticated encryption algorithm [1] combines encryption with authentication, providing both confidentiality and authenticity simultaneously and has been widely used in the security protocols such as SSL/TLS [2], etc. In order to accelerate the development of authentication encryption, CAESAR competition [3] merges. CAESAR is a large-scale cryptographic competition supported by the US National Institute of Standards and Technology (NIST) like AES [4], e-STREAM [5]. With the advance of the competition, authenticated cipher receives more attentions and develops rapidly. The third-round candidates own excellent performance on security and efficiency which are competitive and attract many researchers [6], [7], [8].

MORUS [9] is a notable family of authenticated stream ciphers designed by H. Wu and T. Huang, which has been selected as a third-round candidate of CAESAR competition. By adopting the idea of block cipher, MORUS uses 5 registers of equal length in its internal state and update each state successively. Three MORUS versions: MORUS-640-128, MORUS-1280-128, and MORUS-1280-256 are recommended with different internal state and key sizes. Mileva et al. [10] have proposed a distinguisher attack on MORUS in non-reuse setting. And Zhang et al. [11] researched on the confusion and diffusion properties of the initialization of MORUS. The research on MORUS is relatively limited currently.

Collision attack [12], [13] is a typical method used in attacking the message authentication. The basic technique is to inject a message difference at certain encryption (or processing the associated date) step and cancel it at a later step.

In this paper, we evaluate the security of authentication by theory deduction. Observations reveal that 32-bit word are not confused completely with internal state. Based on the feature that the 32-bit words in the state element are independent of each other, we utilize the partition method and propose the necessary conditions for an internal state collision after two-step update. There results imply that MORUS resists against collision attack in theory.

This paper is organized as follows. In Section 2, the description of MORUS is provided. In Section 3, the distribution of word difference is given. Section 4 discusses the lower bound of difference weight for collision. Section 5 concludes the paper.

II. DESCRIPTION OF MORUS

MORUS operates four phases: initialization, processing of the authenticated data, encryption, and finalization. In this paper we only focus on the processing of associated data and encryption and study the security of MORUS-640-128. MORUS takes a 128-bit key, a 128-bit nonce, and a tag length less than or equal to 128. More details of MORUS-640-128 can be found in [9]. All the "MORUS" in the rest of this paper refers to "MORUS-640-128".

A. Notations

S^i : State at the beginning of *i*-th step, where $i \geq 0$;

S_k^i : State at the beginning of *k*-th round at *i*-th step, where $S_k^i = (S_{k,0}^i, S_{k,1}^i, S_{k,2}^i, S_{k,3}^i, S_{k,4}^i)$, $0 \leq k \leq 4$;

$S_{k,j}^i$: The *j*-th element of the state S_k^i. $S_{k,j}^i = (S_{k,j,1}^i, S_{k,j,2}^i, S_{k,j,3}^i, S_{k,j,4}^i)$, $0 \leq j \leq 4$;

$S_{k,j,l}^i$: *l*-th 32-bit word of the $S_{k,j}^i$, where $1 \leq l \leq 4$, $0 \leq j \leq 4$;

& : Bit-wise AND;

<<< : Rotation to the left;

>>> : Rotation to the right;

$\lceil x \rceil$: The smallest integer not less than x;

$Rotl(x,n)$: Divide the 128-bit block x into four 32-bit words and rotate each word left by n bits,

$Rotl^{-1}(x,n)$: Analogous to $Rotl(x,n)$, divide the 128-bit x into four 32-bit words and rotate each word right by n bits;

1^n (0^n): n bits of '1's ('0's);

W(α) : The hamming weight of α , namely, the number of 1 in α 's binary representation;

WB(Δm_t) : The number of nonzero word in Δm_t . Δm_t denotes the message difference, where $\alpha_i (1 \le i \le 4)$ represents a 32-bit word difference.

B. The State Update Function of MOURS

MORUS uses five 128-bit registers in its internal state. This function consists of five rounds with similar operations that update the state *S*.

$S^{i+1} = \text{StateUpdate}(S^i, m_i)$ is given as follows:

Round 1:

$$S^i_{1,0} = Rotl(S^i_{0,0} \oplus (S^i_{0,1} \& S^i_{0,2}) \oplus S^i_{0,3}, b_0);$$
$$S^i_{1,3} = S^i_{0,3} <<< w_0;$$
$$S^i_{1,1} = S^i_{0,1};$$
$$S^i_{1,2} = S^i_{0,2};$$
$$S^i_{1,4} = S^i_{0,4};$$

Round 2 to 5:

For *k*=1 to 4,

$$S^i_{(k+1)\bmod 5,k} = Rotl(S^i_{k,k} \oplus (S^i_{k,(k+1)\bmod 5} \& S^i_{k,(k+2)\bmod 5}) \oplus S^i_{k,(k+3)\bmod 5} \oplus m_i, b_k);$$
$$S^i_{(k+1)\bmod 5,(k+3)\bmod 5} = S^i_{k,(k+3)\bmod 5} <<< w_k;$$
$$S^i_{(k+1)\bmod 5,(k+1)\bmod 5} = S^i_{k,(k+1)\bmod 5};$$
$$S^i_{(k+1)\bmod 5,(k+2)\bmod 5} = S^i_{k,(k+2)\bmod 5};$$
$$S^i_{(k+1)\bmod 5,(k+4)\bmod 5} = S^i_{k,(k+4)\bmod 5}.$$

Generate S^{i+1} : For *k*=0 to 4: $S^{i+1}_k = S^i_k$.

The rotation constants $b_i, w_i, i = 0, \cdots, 4$ for each round are defined in [10].

C. Processing the Associated Data and Encryption

The Processing the associated data *AD* of MORUS can be described as follows. *adlen* means bit length of the associated data, where $u = \lceil adlen/128 \rceil$.

For *i*=0 to *u*-1: $S^i = \text{StateUpdate}(S^i, AD^{128}_i)$.

The encryption is described as follows. Let *msglen* represents the length of plaintext and $v = \lceil msglen/128 \rceil$.

For $i = 0$ to $v-1$:

$$C_i = P_i \oplus S^{u+i}_0 \oplus (S^{u+i}_1 <<< 96) \oplus (S^{u+i}_2 \& S^{u+i}_3);$$
$$S^{u+i+1} = \text{StateUpdate}(S^{u+i}, P_i);$$

III. The Distribution of Word-oriented Difference for Collision

In this section, we divide each message into four 32-bit words and give the representation of internal state differences.

Suppose that a message difference $\Delta m_t (\Delta m_t \ne 0)$ is injected at *i*-th encryption (or processing the associated date) step and $\Delta S^t = 0$, through the differential analysis, it's easy to get internal state differences in *t*+1-th step:

$$\Delta S^{t+1}_0 = 0$$
$$\Delta S^{t+1}_1 = Rotl(\Delta m_t, 31) <<< 64$$
$$\Delta S^{t+1}_2 = Rotl(\Delta m_t, 7) <<< 32$$
$$\Delta S^{t+1}_3 = Rotl(Rotl(\Delta m_t, 31) \oplus \Delta m_t, 22) <<< 32$$
$$\Delta S^{t+1}_4 = Rotl(S^{t+1}_0 \& \Delta S^{t+1}_1 \oplus (\Delta S^{t+1}_2 >>> 32) \oplus \Delta m_t, 13) <<< 64$$

Note: For convenience, we perform the left rotation for ΔS^{t+1}_3 and ΔS^{t+1}_4 in advance.

If and only if $\Delta S^{t+2}_i = 0 (0 \le i \le 4)$, the internal state collision occurs. Because we only concern that whether the difference is zero and external operation $Rotl(x, n)$ doesn't influence the results, we leave out $Rotl(x, n)$. Each element of state differences after the second step is shown as:

$$\Delta S^{t+2}_0 = (S^{t+1}_1 \& \Delta S^{t+1}_2) \oplus (\Delta S^{t+1}_1 \& S^{t+1}_2) \oplus (\Delta S^{t+1}_1 \& \Delta S^{t+1}_2) \oplus (\Delta S^{t+1}_3 >>> 32);$$

$$\Delta S^{t+2}_1 = \Delta S^{t+1}_1 \oplus (S^{t+1}_2 \& \Delta S^{t+1}_3) \oplus (\Delta S^{t+1}_2 \& S^{t+1}_3) \oplus (\Delta S^{t+1}_2 \& \Delta S^{t+1}_3) \oplus (\Delta S^{t+1}_4 >>> 64) \oplus \Delta m_{t+1};$$

$$\Delta S^{t+2}_2 = \Delta S^{t+1}_2 \oplus (S^{t+1}_3 \& \Delta S^{t+1}_4) \oplus (\Delta S^{t+1}_3 \& S^{t+1}_4) \oplus (\Delta S^{t+1}_3 \& \Delta S^{t+1}_4) \oplus \Delta m_{t+1};$$

$$\Delta S^{t+2}_3 = \Delta S^{t+1}_3 \oplus (\Delta S^{t+1}_4 \& S^{t+2}_0) \oplus \Delta m_{t+1};$$

$$\Delta S^{t+2}_4 = \Delta S^{t+1}_4 \oplus \Delta m_{t+1}.$$

It should be noted that $\Delta S^{t+2} = 0$ is a precondition, then we analyze the differences after the second step.

Now we discuss the difference distribution in the following situations with different values of $\text{WB}(\Delta m_t)$. Remember that $\alpha_i (1 \le i \le 4)$ represents a 32-bit word difference.

A. When $\text{WB}(\Delta m_t) = 1$

Now, we assume that there is only one nonzero word difference in Δm_t , $\Delta m_t = (\alpha_1, 0, 0, 0)$, $\alpha_1 \ne 0$.

$\Delta S^{t+1}_1 = 0\alpha'00$, where $\alpha' = \alpha <<< 31$;

$\Delta S^{t+1}_2 = 00\alpha''0$, where $\alpha'' = \alpha <<< 7, i = 1, 2$;

$\Delta S_3^{t+1} = 00\beta 0$, where $\beta = (\alpha \oplus \alpha') <<< 22$;

$\Delta S_4^{t+1} = 0\gamma 0u$, where $\gamma = (\alpha \oplus \alpha') <<< 13$, u is unknown difference.

Because of $\Delta S_4^{t+2} = \Delta S_4^{t+1} \oplus \Delta m_{t+1}$ and $\Delta S_4^{t+2} = 0$, we get $\Delta m_{t+1} = \Delta S_4^{t+1} = 0\gamma 0u$.

Through update function, we know the presentation of ΔS^{t+2}.

$$\begin{aligned}\Delta S_0^{t+2} &= S_1^{t+1} \,\&\, \Delta S_2^{t+1} \oplus \Delta S_1^{t+1} \,\&\, S_2^{t+1} \oplus \Delta S_1^{t+1} \,\&\, \Delta S_2^{t+1} \\ &\quad \oplus \Delta S_3^{t+1} >>> 32 \\ &= 0uu0 \oplus 000\beta = 0uu\beta\end{aligned}$$

To ensure $\Delta S_0^{t+2} = 0$, we deduce $\beta = 0$, namely $\alpha \oplus \alpha <<< 31 = 0$, then $\alpha = 1^{32}$ must be established. Therefore we can deduce $\alpha' = \alpha'' = 1^{32}$ and $\beta = \gamma = 0$, furthermore $\Delta S_3^{t+2} = 0000$, $\Delta S_4^{t+2} = \Delta m_{t+1} = 000u$.

Supposed $\Delta S_1^{t+2} = 0$, we can get

$$\begin{aligned}\Delta S_2^{t+2} &= \Delta S_2^{t+1} \oplus S_3^{t+1} \,\&\, \Delta S_4^{t+1} \oplus \Delta m_{t+1} \\ &= 00\alpha''0 \oplus 000u \oplus 000u = 00\alpha''u \neq 0\end{aligned}$$

It contradicts to $\Delta S_0^{t+2} = 0$, The above results also hold for $\Delta m_t = 0\alpha_1 00$, $\Delta m_t = 00\alpha_1 0$ and $\Delta m_t = 000\alpha_1$, so $\mathrm{WB}(\Delta m_t) \neq 1$.

B. When $\mathrm{WB}(\Delta m_t) = 2$

There are two cases:

Case 1 $\Delta m_t = 00\alpha_1\alpha_2 (\alpha_i \neq 0, i = 1,2)$

$\Delta S_1^{t+1} = \alpha_1'\alpha_2' 00$, where $\alpha_i' = \alpha_i <<< 31, i = 1,2$;

$\Delta S_2^{t+1} = 0\alpha_1''\alpha_2''0$, where $\alpha_i'' = \alpha_i <<< 7, i = 1,2$;

$\Delta S_3^{t+1} = 0\beta_1\beta_2 0$, where $\beta_i = (\alpha_i \oplus \alpha_i') <<< 22, i = 1,2$;

$\Delta S_4^{t+1} = \gamma_1\gamma_2 uu$, $\gamma_i = (\alpha_i \oplus \alpha_i') <<< 13, i = 1,2$. u refers to an unknown difference,

$\Delta m_{t+1} = \Delta S_4^{t+1} = \gamma_1\gamma_2 uu$.

Now we analyze the ΔS_0^{t+2}.

$$\begin{aligned}\Delta S_0^{t+2} &= S_1^{t+1} \,\&\, \Delta S_2^{t+1} \oplus \Delta S_1^{t+1} \,\&\, S_2^{t+1} \oplus \Delta S_1^{t+1} \,\&\, \Delta S_2^{t+1} \\ &\quad \oplus \Delta S_3^{t+1} >>> 32 \\ &= uuu0 \oplus 00\beta_1\beta_2 = uuu\beta_2\end{aligned}$$

To ensure $\Delta S_0^{t+2} = 0$, we deduce $\beta_2 = 0$, it's easy to get $\alpha' = \alpha'' = 1^{32}, \gamma_2 = 0$. So $\Delta S_3^{t+1} = 0\beta_1 00, \Delta S_4^{t+1} = \Delta m_{t+1} = \gamma_1 0uu$.

Supposed $\Delta S_1^{t+2} = \Delta S_2^{t+2} = 0$, we can obtain

$$\begin{aligned}\Delta S_3^{t+2} &= \Delta S_3^{t+1} \oplus \Delta S_4^{t+1} \,\&\, S_0^{t+2} \oplus \Delta m_{t+1} \\ &= 0\beta_1 00 \oplus u0uu = u\beta_1 uu\end{aligned}$$

To ensure $\Delta S_3^{t+2} = 0$, we deduce $\beta_1 = 0$, $\alpha' = \alpha'' = 1^{32}$ and $\gamma_1 = 0$, therefore $\Delta S_3^{t+1} = 0000$, $\Delta S_4^{t+1} = \Delta m_{t+1} = 00uu$. As for ΔS_2^{t+2}, we get that

$$\begin{aligned}\Delta S_2^{t+2} &= \Delta S_2^{t+1} \oplus S_3^{t+1} \,\&\, \Delta S_4^{t+1} \oplus \Delta m_{t+1} \\ &= 0\alpha_1''\alpha_2''0 \oplus 00uu = 0\alpha_1''uu \neq 0\end{aligned}$$

It contradicts to $\Delta S_2^{t+2} = 0$. The above results also hold for $\Delta m_t = 0\alpha_1\alpha_2 0$, $\Delta m_t = 00\alpha_1\alpha_2$ and $\Delta m_t = \alpha_1 00\alpha_2$.

Case 2 $\Delta m_t = 0\alpha_1 0\alpha_2 (\alpha_i \neq 0, i = 1,2)$

$\Delta S_1^{t+1} = 0\alpha_2' 0\alpha_1'$, where $\alpha_i' = \alpha_i <<< 31, i = 1,2$;

$\Delta S_2^{t+1} = \alpha_1''0\alpha_2''0$, where $\alpha_i'' = \alpha_i <<< 7, i = 1,2$;

$\Delta S_3^{t+1} = \beta_1 0\beta_2 0$, where $\beta_i = (\alpha_i \oplus \alpha_i') <<< 22, i = 1,2$;

$\Delta S_4^{t+1} = 0u0u$. u denotes unknown difference.

$\Delta m_{t+1} = \Delta S_4^{t+1} = 0u0u$.

Supposed $\Delta S_0^{t+2} = \Delta S_1^{t+2} = \Delta S_2^{t+2} = 0$, we analyze ΔS_3^{t+2}.

$$\begin{aligned}\Delta S_3^{t+2} &= \Delta S_3^{t+1} \oplus \Delta S_4^{t+1} \,\&\, S_0^{t+2} \oplus \Delta m_{t+1} \\ &= \beta_1 0\beta_2 0 \oplus 0u0u = \beta_1 u\beta_2 u\end{aligned}$$

To ensure ΔS_3^{t+2}, we get $\beta_1 = \beta_2 = 0$ and $\gamma_1 = \gamma_2 = 0$, therefore $\Delta S_3^{t+1} = 0000$. Take consideration of ΔS_2^{t+2},

$$\begin{aligned}\Delta S_2^{t+2} &= \Delta S_2^{t+1} \oplus S_3^{t+1} \,\&\, \Delta S_4^{t+1} \oplus \Delta m_{t+1} = \alpha_1''0\alpha_2''0 \oplus 0u0u \\ &= \alpha_1''u\alpha_2''u \neq 0\end{aligned}$$

It contradicts to $\Delta S_2^{t+2} = 0$, The same results also hold for $\Delta m_t = 0\alpha_2 0\alpha_4$, so $\mathrm{WB}(\Delta m_t) \neq 2$.

C. When $\mathrm{WB}(\Delta m_t) = 3$

Assume that $\Delta m_t = (0, \alpha_1, \alpha_2, \alpha_3)(\alpha_i \neq 0, i = 1,2,3)$, we can deduce the followings.

$\Delta S_1^{t+1} = \alpha_2'\alpha_3' 0\alpha_1'$, where $\alpha_i' = \alpha_i <<< 31, i = 1,2,3$;

$\Delta S_2^{t+1} = \alpha_1''\alpha_2''\alpha_3''0$, where $\alpha_i'' = \alpha_i <<< 7, i = 1,2,3$;

$\Delta S_3^{t+1} = \beta_1\beta_2\beta_3 0$, where $\beta_i = (\alpha_i \oplus \alpha_i') <<< 22, i = 1,2,3$;

$\Delta S_4^{t+1} = \gamma_1 uuu$, $\gamma_i = (\alpha_i \oplus \alpha_i') <<< 13, i = 1,2,3$, u denotes unknown difference.

$\Delta m_{t+1} = \Delta S_4^{t+2} = \gamma_1 uuu$.

We can obtain the internal state difference ΔS^{t+2} as follows:

$$\Delta S_0^{t+2} = S_1^{t+1} \,\&\, \Delta S_2^{t+1} \oplus \Delta S_1^{t+1} \,\&\, S_2^{t+1} \oplus \Delta S_1^{t+1} \,\&\, \Delta S_2^{t+1} \oplus \Delta S_3^{t+1} >>> 32 = uuuu$$

$$\Delta S_1^{t+2} = \Delta S_1^{t+1} \oplus S_2^{t+1} \,\&\, \Delta S_3^{t+1} \oplus \Delta S_2^{t+1} \,\&\, S_3^{t+1} \oplus \Delta S_2^{t+1} \,\&\, \Delta S_3^{t+1} \oplus \Delta S_4^{t+1} >>> 64 \oplus \Delta m_{t+1} = uuuu$$

$$\Delta S_2^{t+2} = \Delta S_2^{t+1} \oplus S_3^{t+1} \,\&\, \Delta S_4^{t+1} \oplus \Delta S_3^{t+1} \,\&\, S_4^{t+1} \oplus \Delta S_3^{t+1} \,\&\, \Delta S_4^{t+1} \oplus \Delta m_{t+1} = uuuu$$

$$\Delta S_3^{t+2} = \Delta S_3^{t+1} \oplus \Delta S_4^{t+1} \,\&\, S_0^{t+2} \oplus \Delta m_{t+1} = uuuu$$

Each 128-bit element of ΔS^{t+2} could be 0, therefore $\mathrm{WB}(\Delta m_t) = 3$ is possible.

D. *When* $\mathrm{WB}(\Delta m_t) = 4$

We know $\Delta m_t = (\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ $(\alpha_i \neq 0, i = 1,2,3,4)$. After one step update, each 32-bit word of ΔS_1^{t+1} and ΔS_2^{t+1} are non-zero, meanwhile, each 32-bit word of ΔS_3^{t+1} and ΔS_4^{t+1} is uncertain, through our analysis, each 128-bit element of ΔS^{t+2} could be 0,therefore $\mathrm{WB}(\Delta m_t) = 4$ is possible.

We can draw the conclusion below.

Conclusion 1. For MORUS, if the message difference $\Delta m_t (\Delta m_t \neq 0)$ is injected at *i*-th encryption (or processing the associated date) step and gets eliminated at the next step, then $\mathrm{WB}(\Delta m_t) \geq 3$ must be true.

IV. The Lower Bound of Difference Weight for Collision

Because the collision probability is related to the weight of message difference, we discuss the bit-oriented necessary conditions in this section. Through theory deduction, we complete exhaustion when the weight of message difference is less than or equal to 5.

In the end the experiment results indicate the following conclusion.

Conclusion 2. For MORUS, when the weight of message difference is less than or equal to 5, there is no collision.

Table I. compares the result of this paper with that of designers on collision cryptanalysis.

TABLE I. RESULTS COMPARISON

Analyst	Analysis method	Idea	Difference distribution	Weight of difference
Designers	Experiment	—	—	$\mathrm{W}(\Delta m_t) \geq 26$
This paper	Theory deduction	Partition method	$\mathrm{WB}(\Delta m_t) \geq 3$	$\mathrm{W}(\Delta m_t) \geq 5$

V. Conclusions

In this work, we focus on the security of message authentication in MORUS. Through partition method, we found out the distribution of input difference is determined, namely $\mathrm{WB}(\Delta m_t) \geq 3$. Finally, by experiments, we deduce the lower bound of difference is 5.

The new idea we used to find the distribution of word-oriented difference for collision is universal, which can be applied to other authenticated ciphers. Moreover, we will evaluate the ability of MORUS in resisting other attacks.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant No. 61572516, 61272041 and 61272488).

References

[1] M. Bellare et al., “Authenticated encryption: Relations among notions and analysis of the generic composition paradigm,” *Journal of Cryptology*, vol. 21, no. 4, Jan. 2008, pp. 469-491.

[2] K. Hickman et al., “The SSL protocol,” Netscape Communications Corp, 1995, 501.

[3] D. J. Bernstein, “Caesar: Competition for authenticated encryption: Security, applicability, and robustness,” 2015.

[4] J. Daemen et al., “AES the advanced encryption standard,” The Design of Rijndael, 2002.

[5] ECRYPT, European Network of Excellence in Cryptology: “The Stream Cipher Project: eSTREAM,” IST-2002-507932. http://www.ecrypt.eu.org/stream.

[6] C. Dobrauning et al., “Heuristic Tool for Linear Cryptanalysis with Applications to CAESAR Candidates,” ASIACRYPT, Auckland, New Zealand, 2015, pp.490-509.

[7] T. Peyrin et al., “Cryptanalysis of JAMBU,” FSE 2015, Istanbul, Turkey, Mar. 2015, pp. 264-281.

[8] M. Salam et al., “Investigating Cube Attack on the Authenticated Encryption Stream Cipher ACORN,” ATIS 2016, Cairns, QLD, Australia, Oct. 2016, pp. 15-26

[9] H. Wu et al., “The authenticated cipher MORUS (v1),” CAESAR submissions. http://competitions.cr.yp.to/ round2/morusv11.pdf, 2015.

[10] A. Mileva et al., “Analysis of the Authenticated Cipher MORUS (v1),” Int. Conf. BalkanCryptSec 2015, Koper, Slovenia, Sept. 2015, pp. 45-59.

[11] P. Zhang et al., “Research on the confusion and diffusion properties of the initialization of MORUS,” J. Cryptologic Research, 2015, vol. 2, no. 6, pp. 536-548.

[12] T. Fuhr et al., “Collision Attacks Against CAESAR Candidates – Forgery and Key-Recovery Against AEZ and Marble,” *Advances in Cryptology-ASIACRYPT 2015*, Auckland, New Zealand, Nov. 2015, pp. 510-532.

[13] T. Peyrin, “Collision Attack on Grindahl,” J. Cryptology, vol. 28, no. 4, January 2015, pp. 879-898.