Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Review:

Survey of design and security evaluation of authenticated encryption algorithms in the CAESAR competition*

Fan ZHANG†‡[1,2,3,4], Zi-yuan LIANG[1,2,4], Bo-lin YANG[1], Xin-jie ZHAO[5], Shi-ze GUO[5], Kui REN[2,4]

[1]*College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China*

[2]*Institute of Cyberspace Research, Zhejiang University, Hangzhou 310027, China*

[3]*State Key Laboratory of Cryptology, Beijing 100878, China*

[4]*Alibaba-Zhejiang University Joint Institute of Frontier Technologies, Hangzhou 310027, China*

[5]*Institute of North Electronic Equipment, Beijing 100191, China*

†E-mail: fanzhang@zju.edu.cn

Received Sept. 17, 2018; Revision accepted Nov. 20, 2018; Crosschecked Dec. 17, 2018

Abstract: The Competition for Authenticated Encryption: Security, Applicability, and Robustness (CAESAR) supported by the National Institute of Standards and Technology (NIST) is an ongoing project calling for submissions of authenticated encryption (AE) schemes. The competition itself aims at enhancing both the design of AE schemes and related analysis. The design goal is to pursue new AE schemes that are more secure than advanced encryption standard with Galois/counter mode (AES-GCM) and can simultaneously achieve three design aspects: security, applicability, and robustness. The competition has a total of three rounds and the last round is approaching the end in 2018. In this survey paper, we first introduce the requirements of the proposed design and the progress of candidate screening in the CAESAR competition. Second, the candidate AE schemes in the final round are classified according to their design structures and encryption modes. Third, comprehensive performance and security evaluations are conducted on these candidates. Finally, the research trends of design and analysis of AE for the future are discussed.

Key words: CAESAR competition; Authenticated cipher; Block cipher; Stream cipher; Hash function; Security evaluation

https://doi.org/10.1631/FITEE.1800576 **CLC number:** TP309

1 Introduction

Traditional symmetric encryption algorithms provide confidentiality mainly for messages. Message authentication algorithms provide integrity separately. Many applications require that algorithms provide both message confidentiality and integrity. Therefore, there is an urgent need for authenticated encryption (AE). Integrating encryption and message authentication, AE algorithms have extensive research and application prospects. Sometimes, AE algorithms are also referred to as AE schemes or authenticated ciphers.

The characteristics of AE algorithms can be viewed from different angles. According to the times that a message is processed, AE can be divided into two categories.

The first category is called "one-pass AE," in which a message needs to be processed only once. In 2000, Jutla from IBM proposed integrity aware cipher block chaining (IACBC) and integrity aware parallelizable mode (IAPM), which are the earliest one-pass AE algorithms (Jutla, 2001). Shortly after

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 61472357 and 61571063), the Open Fund of State Key Laboratory of Cryptology, the Major Scientific Research Project of Zhejiang Lab, and the Alibaba-Zhejiang University Joint Institute of Frontier Technologies

ORCID: Fan ZHANG, http://orcid.org/0000-0001-6087-8243

that, Gligor and Donescu (2001) proposed new one-pass AE algorithms, XCBC and XECB. Rogaway et al. (2001) further improved IAPM. They designed a new type of one-pass AE, called "OCB," which can overcome the defects of the ECB model.

The other category is two-pass AE, in which a message needs to be processed twice. The counter with CBC-MAC (CCM) mode (Whiting et al., 2003) and the Galois/counter mode (GCM) (McGrew and Viega, 2004) are the two most classical two-pass AE algorithms. CCM has been adopted by the IEEE 802.11 wireless LAN standard, and has been selected as the recommended standard of AE mode for the block cipher Advanced Encryption Standard (AES) by NIST. GCM has also been adopted by several standardizations such as NIST, IPSec, SSL, and TLS.

According to the design principle, AE algorithms can be further classified into two mainstream types.

One type involves constructing AE algorithms by combining existing authentication and encryption algorithms. There are three methods of composition: encrypt-and-MAC plaintext, MAC-then-encrypt, and encrypt-then-MAC. Bellare and Namprempre (2008) showed that the encrypt-then-MAC method is the most secure one, whereas other methods, such as encrypt-and-MAC and MAC-then-encrypt, have some security vulnerabilities. Most of existing network communication security protocols, such as HTTPS, SSL, and IPSec, adopt encrypt-then-MAC as one of their design principles. In their designed constructions, the optional encryption algorithms they choose could be AES, 3DES, etc., and the optional authentication algorithms could be MAC or others. However, if the padding process is added between encryption and authentication, that is, when the encryption-padding-authentication scheme is adopted, adversaries can launch the padding oracle attack (Hwang and Lee, 2015), which can decrypt the protocol data without authorization.

The other type involves constructing AE algorithms with a dedicated new design that interleaves the authentication and encryption processes to share part or all of their calculations, which eventually results in a complete new AE algorithm. In the construction, authentication and encryption steps can access each other's inputs and intermediate results during calculations. The purpose of constructing a dedicated AE is to improve efficiency, reduce costs, and enhance security. In recent years, design and analysis of dedicated AEs has become a research hot spot in cryptography.

With the support of the National Institute of Standards and Technology (NIST), the Competition for Authenticated Encryption: Security, Applicability, and Robustness (CAESAR) competition calls for submissions of AE schemes worldwide. After the AES competition, the NESSIE project, and the SHA-3 competition, the CAESAR competition is considered as another significant milestone in the international cryptography community. The entire competition cycle started in 2013, and has a total of three rounds. As its name suggests, the CAESAR competition calls for AE schemes that can satisfy integrity, confidentiality, and robustness simultaneously. The comprehensive performance and security strength of the candidates should be higher than those of AES-GCM. Recently, the competition moved to the final round. There are seven finalists after three rounds of screening.

For years, the CAESAR competition has drawn extensive attention from the global cryptography research community. It produces a large number of excellent algorithms, generates new design techniques, and stimulates the growth of this research area. At the same time, these collected results may directly promote the standardization and large-scale application of AE algorithms.

After rigorous analysis by worldwide cryptographers, the candidates in the CAESAR competition have good security, real-time efficiency, and high research value for AE design and analysis. Our paper focuses mainly on the seven finalists, but also addresses the rest of the candidates in the third round. We introduce the basic requirements of the CAESAR candidates, and then discuss their designed structure, detailed features, and performance evaluation. We also collect related works about each candidate, and introduce their up-to-date research progress, hoping to provide certain reference to the community of AE-related research.

2 Background on AE algorithms

An authenticated cipher is composed of two procedures, authenticated encryption and verification

decryption, which can be depicted by two functions f_{E} and f_{D}, respectively.

1. AE function f_{E}: The inputs to f_{E} include the key K, the plaintext P, the associated data AD, and an arbitrary number N (nonce). AD is an optional header in plaintext that will not be encrypted, but will be covered by authenticity protection. The outputs from f_{E} are the ciphertext C and the tag T, which are the message authentication codes (MAC) of the AE algorithm. The AE function first encrypts on P with K, AD, and N to generate the corresponding C, and then produces the authenticated tag T with part of the input parameters and C.

2. Verification decryption function f_{D}: The input is the key K, the ciphertext C, the associated data AD, and the nonce N. After verification and decryption, the output is either the plaintext P or verification rejection.

Fig. 1 shows a schematic of the AE function f_{E}. The center AE function module is provided with these required inputs, including the plaintext, key, associated data, and the nonce, and outputs the corresponding ciphertext and authenticated tag. The diagram of the verification decryption function f_{D} is quite similar to that of f_{E}.

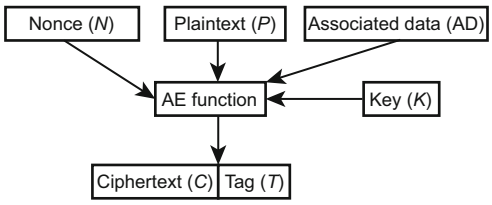

Fig. 1 Schematic of authenticated encryption functions

3 Design requirements of AE schemes and progress of CAESAR competition

The CAESAR competition has published necessary and optional requirements for the candidate schemes in detail. In this section, we discuss these design requirements and classify them into three aspects: inputs, security, and comprehensive performance. Because the CAESAR competition has finished all three rounds, we provide a brief introduction to the progress of CAESAR, and also list the finalists as well as other candidates in three rounds of screening.

3.1 Design requirements of the AE scheme

Most AE schemes adopt the standard AE construction mentioned in Section 2. Here are some of the specific requirements for CAESAR candidates.

1. Inputs. As introduced in Section 2, AE algorithms require several inputs, such as plaintexts, key, and associated data. The CAESAR competition requires that designers submit candidate AE schemes with five inputs and one output. The five inputs include a variable-length plaintext P, a variable-length associated data AD, a fixed-length secret message number SMN, a fixed-length public message number PMN, and a fixed-length key. The first four inputs have different security purposes. Note that the secret and public message numbers are new concepts presented in the CAESAR competition. The candidate scheme is required to output a ciphertext with variable length. The CAESAR competition also requires that, it must be possible to recover the plaintext and secret message number from the ciphertext, associated data, public message number, and key.

Table 1 shows the specific input requirements of the candidate AE schemes in the CAESAR competition.

2. Security. Security is the principal design criterion for candidate AE schemes. They need to fulfill the requirements of integrity, confidentiality, and robustness, and must resist classical cryptanalysis methods such as differential analysis, linear analysis, and algebraic analysis. The general designs of AE schemes have commonly diverged in two directions. One is that designers assume that the

Table 1 Specific input requirements for AE in the CAESAR Competition

Encryption input	Integrity	Confidentiality	Length	Option	Required to be used once
Plaintext	√	√	Variable	×	×
Associated data	√	×	Variable	√	×
Secret message number	√	√	Fixed	√	√
Public message number	√	×	Fixed	√	√
Seed key	√	√	Fixed	×	×

underlying cryptographic primitives used in AE (such as block cipher, e.g., AES) are based on strong pseudo-random permutation, and then prove the security of their new scheme. The other is that designers learn from previous work such as block cipher, stream cipher, hash function, MAC, and associated components, to construct a new AE scheme to resist the existing analysis methods.

3. Comprehensive performance. The candidate AE schemes have to satisfy several performance criteria, such as consumption, area overhead, throughput, and delay. The fundamental operations of AE include mainly bitwise XOR and modular operations (including modular addition and multiplication). The number of modular multiplication and cryptographic primitive calls is an important metric for evaluating the efficiency of AE schemes. The CAESAR competition requires candidate AE schemes have good software and hardware comprehensive performance. The finalists should have obvious advantages in terms of execution efficiency compared to AEC-GCM.

In addition, the candidate AE schemes are generally required to meet the following features:

1. Parallelizable. Most operations in AE schemes should be able to execute in parallel to fully use the computation power from underlying resources such as GPU.

2. Nonce-dependent or nonce-robust. The security of candidate AE schemes should directly depend on the randomness and uniqueness of the nonce, or it can be proved to be secure as independent from the nonce.

3. Online. The calculation of the i^{th} ciphertext is related only to the previous i plaintexts and the key. This requirement is not mandatory, but it is suggested that it be satisfied, considering the efficiency and throughput.

3.2 Progress of CAESAR

The CAESAR competition started in January 2013 and eventually lasted about five years. The latest yearly news of the competition was updated in the annual meeting called Directions In Authenticated Ciphers (DIAC). In March 2018, the finalists from the third round were announced. Table 2 lists part of the candidate schemes in the first, second, and third rounds. Note that the algorithms with crossed-out names were withdrawn by the designers themselves during the competition. The seven finalists were selected from the 15 candidate schemes in the third round. The finalists are ACORN (Wu, 2016), AEGIS (Wu and Preneel, 2013), Ascon (Dobraunig et al., 2016c), COLM (Andreeva et al., 2016a) superseded AES-COPA (Andreeva et al., 2015) and ELmD (Andreeva et al., 2016b), Deoxys (Jean et al., 2016), MORUS (Wu and Huang, 2016), and OCB (Krovetz and Rogaway, 2016). Note that Deoxys has two schemes, Deoxys-I and Deoxys-II. The latter is a finalist, while the former is only in the third round.

4 Analysis of the design characteristics for AE candidate schemes

In this section, we will discuss the design characteristics of the 15 candidates in the third round, including the seven finalists. We divide the candidates into four categories based on their design structures. We discuss the encryption mode each

Table 2 Third-round candidates and finalists in the CAESAR competition

Round	Name
First Round	++AE, ~~AES-CMCC~~, AES-COBRA, AES-CPFB, AVALANCHE, ~~Calico~~, CBA, ~~CBEAM~~, Enchilada, ~~FASER~~,~~HKC~~, iFeed[AES], Julius, KIASU, LAC, ~~Marble~~, ~~MeMambo~~, ~~PAES~~, ~~PANDA~~, POLAWIS, Prøst, Raviyoyla, Sablier, Silver, Wheesht, YAES
Second Round	HS1-SIV, ICEPOLE, Joltik, Minalpher, OMD, PAEQ, π-Cipher, POET, PRIMATEs, SCREAM, SHELL, STRIBOB, TriviA-ck
Third Round	AES-OTR (Minematsu, 2016), AES-JAMBU (Wu and Huang, 2014), AEZ (Hoang et al., 2014), CLOC (Minematsu et al., 2016), SILC (Iwata et al., 2014), Ketje (Berton et al., 2016), Keyak (Bertoni et al., 2015), NORX (Aumasson et al., 2015), Tiaoxin (Nikolić, 2016)
Finalists	ACORN (Wu, 2016), AEGIS (Wu and Preneel, 2013), Ascon (Dobraunig et al., 2016c), Deoxys (Jean et al., 2016), MORUS (Wu and Huang, 2016), OCB (Krovetz and Rogaway, 2016), COLM (Andreeva et al., 2016a), superseding AES-COPA (Andreeva et al., 2015), and superseding ELmD (Andreeva et al., 2016b)

candidate scheme has selected, and the features that the designers claimed in their submitted specification papers.

4.1 Analysis of the designed structure

According to the designed structure, AE candidate schemes can be grouped into four types: block cipher based AE (BC-AE), stream cipher based AE (SC-AE), sponge-based AE (SH-AE), and dedicated AE (D-AE). Table 3 depicts the structure design classification of the finalists and other third-round candidates in the CAESAR competition.

BC-AE is designed based on block ciphers and nonlinear transformations. It uses complex encryption schemes (such as AES round functions) to improve security.

SC-AE is designed based on stream ciphers. It uses bitwise operations to fully exploit the performance of underlying hardware to balance performance and security.

SH-AE is designed based on sponge functions. Because the sponge-based function is resistant to collision attacks, SH-AE can amplify the small differences that are produced by sponge functions and create high complexity in theory.

D-AE is designed on new dedicated structures. The designers can feel free to use different operations such as permutations, mixing, and AES round functions. The ultimate design goal is to satisfy the requirements of confidentiality and comprehensive performance. These requirements can be achieved by a hardware-acceleration instruction set such as AES-NI (Rott, 2010).

4.2 Analysis of encryption mode

Encryption mode is quite important in the design of AE schemes. We will discuss mainly two types of encryption modes in the candidate schemes: block cipher mode (BC mode) and authenticated encryption mode (AE mode).

4.2.1 Block cipher mode

Most candidate schemes in the third round are designed based on block cipher mode. In the following subsections we discuss the modes that were used, including OFR, OTR, CFB, EME, TAE, and XEX. Brief descriptions of the corresponding modes are also listed in Table 4.

4.2.2 Authenticated encryption mode

In the CAESAR competition, there are several different methods for adopting AE modes. The most straightforward method is to directly use the existing AE modes mentioned by NIST. A more challenging method is to design a totally new AE mode. Note that most candidates in the third round designed new AE modes. Another possible method that can achieve the design trade-off is to modify and improve some existing modes. We divide the AE modes into two categories: AE modes mentioned by NIST and new AE modes adopted by the designers.

Currently, 14 AE modes were mentioned in NIST discussions. Here is a brief discussion of the

Table 3 Classification of the designed structure for third-round candidates

Class	Finalist(s)	Third-round candidate(s)	Characteristic
BC-AE	COLM/AES -COPA/ELmD, Deoxys, OCB	AES-JAMBU, AES-OTR, AEZ, CLOC/SILC	BC-AE adopts reversible round functions, permutations, and hash functions. It makes the block cipher a black box to accomplish the AE function.
SC-AE	ACORN	–	SC-AE is based on the stream cipher mode. It learns from the idea of the block cipher. SC-AE supplements and improves the block cipher function, and adds the corresponding authentication part.
SH-AE	Ascon	Ketje, Keyak, NORX	SH-AE uses a hash function, encryption scheme, and message authentication scheme to satisfy the confidentiality, integrity, and robustness requirements of the secret message. All candidates in the third round adopt sponge functions as their hash functions.
D-AE	AEGIS, MORUS	Tiaoxin	D-AE updates new state values by the relationship between adjacent states. Then it increases the correlation between adjacent states and accomplishes data integrity.

Table 4 Encryption mode adopted by block cipher-based AE

Index	Name	Description
1	Output feedback mode (OFB) (Pub, 1980)	OFB divides the key streams into groups. It uses the initialization vector (IV) and the key stream blocks for encryption, and then feeds back the encryption results as the IV of the next round. At the same time, OFB XORs this round's results and the corresponding plaintext blocks to obtain the cipher blocks. OFB repeats the process until all plaintext blocks are encrypted.
2	Two-branch two-round feistel (OTR) (Minematsu, 2014)	OTR generates special masks to encrypt the block ciphers of each round. It iteratively feeds back the output of masking as the input of the next round. A two-round Feistel structure finally outputs the ciphertext. The main operations and initialization processes are mostly based on block ciphers, so it can be regarded as a special encryption mode of block ciphers.
3	Cipher feedback mode (CFB) (Pub, 1980)	The operations of CFB are similar to those of OFB, but the feedback part of CFB has to XOR each round's encryption results and the corresponding plaintext blocks. It does not need to XOR each round's IV and key streams.
4	ECB-mix-ECB mode (EME) (Halevi, 2004; Halevi and Rogaway, 2004)	EME consists of two ECB layers and a lightweight mixing layer between them. EME transforms n-bit block ciphers into mn-bit strings ($1 \leq m \leq n$). EME is a parallel mode, and has similar continuous computational efficiency to non-parallel CBC-mask-CBC (CMC) mode.
5	Tweakable authenticated encryption (TAE) (Liskov et al., 2002, 2011)	TAE has encryption processes that are similar to those of offset codebook mode (OCB). The difference is that TAE adopts tweakable block ciphers instead of basic block ciphers.
6	XOR-encrypt-XOR (XEX) (Rogaway, 2004)	XEX converts the block cipher into a tunable one, and makes sure that the tunability is within a certain range. XEX is a popular overall encryption mode, and is widely applicable in smart card devices.

AE modes covered in Table 5.

We also list all the new AE modes adopted by the finalists and other third-round candidates in the following:

1. MonkeyWrap (Berton et al., 2016)

MonkeyWrap is the AE mode adopted by Ketje. It is similar to SpongeWrap (Bertoni et al., 2011). One of their differences is that MonkeyWrap adopts MonkeyDuplex permutations. It first applies a mixing layer on the input data, XORs the message number, and then applies another mixing layer on the results to obtain the ciphertexts. Finally, MonkeyWrap iterates the message blocks to obtain the authenticated tag.

2. Motorist (Bertoni et al., 2015)

Motorist is a sponge-based AE mode adopted by Keyak. It supports more than one duplex operation in parallel. Motorist encrypts the plaintext blocks and message number blocks with a unique private vector, and then obtains the corresponding cipher blocks and authenticated tag. Motorist applies the same operation to all plaintext blocks to obtain all cipher blocks and tags.

3. Deoxys (Jean et al., 2016)

Deoxys is the AE mode based on Deoxys-BC (Jean et al., 2016). Deoxys aims at instance operations for Deoxys-BC. It combines the inputs of authenticated ciphers to obtain the ciphertexts and the authenticated tags. Deoxys-I is designed for the case of nonce-respecting mode, and Deoxys-II is designed for the case of nonce-misuse resistant mode.

4. CBC MAC & CFB (Iwata et al., 2014; Minematsu et al., 2016)

CBC MAC & CFB is the AE mode adopted by SILC and CLOC. It needs two CBC MACs to obtain the AD, and needs another two CBC MACs and one CFB to obtain the ciphertexts. CBC MAC & CFB separates CBC MAC and CFB logically. The operations can be finished separately. As a result, CBC MAC & CFB has high efficiency.

5. PMAC & XEX (Andreeva et al., 2015)

PMAC & XEX is the AE mode adopted by AES-COPA, and it can operate in parallel. PMAC & XEX can operate under PMAC-like mode to guarantee the integrity of the AD and plaintexts. It uses XEX-like masks to guarantee the confidentiality of messages.

Table 5 AE modes mentioned by NIST

Index	Name	Description
1	CBC-MAC mode with a counter (CCM) (Whiting et al., 2003)	CCM is block-cipher mode. It cannot operate in parallel. It is applicable only for 128-bit block ciphers, such as AES-128.
2	Cipher-state mode (Schroeppel et al., 2004)	CS uses the intermediate messages during encryption to provide efficient authentication. Because both encryption and authentication can be conducted in parallel, CS mode takes less computation cost. In addition, CS mode has good extendibility and can be applied to other block cipher schemes.
3	Carter-Wegman + CTR mode (CWC) (Kohno, 2003)	CWC inputs the nonce, associated data, and message number, and encrypts the messages in CTR counter mode to output ciphertexts. Then it inputs the AD, nonce, and ciphertexts in Carter-Wegman mode (Wegman and Carter, 1981) to output the authenticated tag.
4	Traditional AE mode (EAX) (Bellare et al., 2003)	EAX has two types: EAX and EAX2. EAX and EAX2 input the nonce and the header in the same way. Then they encrypt the message with a message number. After that, they apply a mixing step on the encryption results to output ciphertexts and finally output the authenticated tag using ciphertexts and headers.
5	Expanded EAX mode (EAX') (Moise et al., 2011)	EAX' inputs plaintexts, nonce, and the secret key, and then outputs the authenticated tag and ciphertexts. Compared with EAX, EAX' is significantly optimized in terms of nonce length, hardware encryption speed, etc.
6	Galois counter mode (GCM) (McGrew and Viega, 2004)	GCM adds a counter in the encryption step, and XORs plaintext blocks to output the cipher blocks. Then it inputs AD to module multiplication in GF (2^{128}), and the output XORs with the cipher blocks. The XOR results will be taken as the next round's associated data. Finally GCM XORs the results of the last plaintext block to output the authenticated tag.
7	Integrity aware cipher block chaining mode (IACBC) (Jutla, 2016a)	IACBC encrypts in CBC mode with random IV, and then XORs the block cipher results. It encrypts and calculates each round's IV with the message number, and then XORs the plaintext blocks with IV. Finally, IACBC XORs the two XOR results by iteration to output each round's ciphertexts.
8	Integrity aware parallel mode (IAPM) (Jutla, 2016b)	IAPM encrypts the random IVs, and divides the results into several vector pairs. The number of vector pairs must be equal to the number of plaintext blocks. Then IAPM encrypts the plaintext blocks using one of the corresponding vector pairs, and takes a modular operation with the other one to output the ciphertext.
9	Input and output chaining mode (IOC) (Recacha, 2016)	IOC uses two IVs in each round. One IV XORs with the plaintext blocks. The XOR results will be taken as the encryption input in this round and the XOR vector in the next round. The other vector XORs with the encryption results to output the cipher blocks. The XOR results will be the IV in the next round, which will XOR with next round's plaintext blocks to obtain the encryption inputs.
10	Offset codebook mode (OCB) (Rogaway, 2016)	OCB optimizes the IAPM mode. It first adds several zeros to initialize the encryption, then XORs the results and IV to obtain an immediate result and encrypts the result to obtain R. After that, OCB XORs the initialized results and R with some related parameters. The results will be each round's random vector pairs. Referring to the IAMP mode, OCB XORs the plaintext blocks and random vector pairs, and then encrypts to output the immediate cipher blocks. Finally, OCB XORs immediate cipher blocks to output the corresponding cipher blocks.
11	Propagating cipher feedback mode (Hellström and StreamSec, 2001)	PCFB is a strict stream AE mode that is based on bidirectional error propagation. Compared with CFB, PCFB encrypts the input vector before propagating and updating the messages, which improves the efficiency and complexity of cipher feedback.
12	Random key chaining mode (RKC) (Kaushal et al., 2012)	RKC uses hash functions and a deterministic nonce generator. The key of RKC is generated by the key streams. RKC first encrypts plaintext blocks with different round functions. Then RKC XORs the key and the nonce. The XOR result will be taken as the next round's key. RKC encrypts the results and the plaintext blocks, and then XORs the immidiate cipher blocks and plaintext blocks to obtain part of the output results. Finally, RKC chains the results in each round to obtain the ciphertexts. After a round of hash function operations, the results are taken as the authenticated tag.
13	Synthetic initialization vector mode (SIV) (Rogaway and Shrimpton, 2007)	The key of SIV consists of a pair of subkeys k_1, k_2. In CMAC mode, the message number and several headers encrypt with k_1 to output the authenticated tags. In CTR mode, they encrypt with k_2 to output the ciphertexts.
14	Extended cipher block chaining mode (XCBC) (Gligor, 2016)	There are three types of XCBC modes: XCBC$, XCBCC, and XCBCS. Both XCBC$ and XCBCS input only one parameter, and the IV of XCBCS participates in the encryption step. XCBCC inputs two parameters and uses an additional counter for encryption.

6. AEZ (Hoang et al., 2014)

AEZ is the AE mode adopted by the AEZ candidate scheme. It uses the decoding-coding method for authenticated encryption. For plaintexts less than 32 bytes, AEZ adopts AEZ-tiny based on FFX (Bellare et al., 2010; Dworkin, 2016). For larger lengths, AEZ adopts AEZ-core based on EME (Halevi, 2004; Halevi and Rogaway, 2004) and OTR (Minematsu, 2014).

7. OTR (Minematsu, 2016)

OTR is the AE mode adopted by AES-OTR, which was mentioned earlier.

8. Ascon (Dobraunig et al., 2016c)

Ascon is the AE mode adopted by the Ascon candidate scheme. It belongs to the AE mode family "$\text{Ascon}_{a,b}$-k-r." Its permutations are similar to those of MonkeyDuplex. The ciphertexts and authenticated tag are outputted after inputting the AD, nonce, secret key, and plaintexts.

9. Tiaoxin-346 (Nikolić, 2016)

Tiaoxin-346 is the AE mode based on stream ciphers and the nonce. Tiaoxin-346 has four processes: initialization, processing associated data, encryption, and finalization/tag production. First, Tiaoxin-346 divides and processes AD and plaintexts into groups. Second, it assigns the state values with keys and the nonce. Third, it updates the AD with the state values. Then Tiaoxin-346 XORs the plaintext blocks and the state values. Finally, it generates the authenticated tag with the previous state values.

10. JAMBU (Wu and Huang, 2014)

JAMBU is a lightweight AE mode. It converts lightweight block ciphers to lightweight AE schemes. During initialization, it encrypts two initial states, and the results will become the next round's input and the "external vectors" used to update the next state values. During encryption, JAMBU XORs the plaintext blocks and results of each state to output the cipher blocks. Then it XORs the final "external vectors" and two output vectors in the final round. The result is the authenticated tag.

11. PHASH & EME* (Andreeva et al., 2016a,b)

PHASH & EME* is the AE mode adopted by ELmD and COLM, and it can operate in parallel. It adopts the improved EME mode. To satisfy hardware requirements, it uses a lightweight linear mixing layer between two encryption processes instead of a nonlinear mixing layer in the standard EME mode. It adopts PHASH to satisfy the authentication requirements. Different from PMAC, PHASH can operate completely in parallel.

4.3 Features of AE candidate schemes

Because different candidate schemes have different design intensions and structures, they have different features in all aspects. Each candidate scheme lists its features in the submitted specifications. Detailed features of all candidates are outlines below:

1. ACORN

(1) Novel design. ACORN is a bit-based sequential authenticated cipher. The difference in ACORN is injected into the state for authentication for better performance.

(2) Parallel. In ACORN, 32 steps can be computed in parallel.

(3) One message bit is processed in each step.

(4) Length information concerning the associated data and plaintext/ciphertext is not needed.

(5) Efficient in both hardware and software.

2. AEGIS

(1) Efficient. On the latest Intel Haswell microprocessors, the speed of AEGIS128L is more than twice that of AES-GCM.

(2) Secure. AEGIS provides 128-bit authentication security, which is stronger than that of AES-GCM.

3. Ascon

(1) Lightweight and flexible in hardware. Ascon provides excellent characteristics in terms of size and speed for hardware implementation.

(2) Bitsliced in software. Ascon is designed to facilitate bitsliced software implementations.

(3) Easy integration of side-channel countermeasures. Ascon can be implemented efficiently on platforms and applications where side-channel resistance is important.

(4) Balanced design. Ascon is designed to provide lightweight implementation characteristics in both hardware and software, while still having good performance on both. Hence, Ascon is highly suited for scenarios where many lightweight devices communicate with a back-end server, a typical use case in the Internet of Things (IoT).

(5) Online. The Ascon cipher is online and can encrypt plaintext blocks before subsequent plaintexts or the plaintext length is known.

(6) Single-pass. For both encryption and decryption, just one pass over the data is required.

(7) Inverse-free. Ascon does not need to implement any inverse operation.

(8) High key agility. Ascon does not need a key schedule, or expand the key by any other means.

(9) Simplicity. Ascon is intuitively defined on 64-bit words using only the common bitwise Boolean functions AND, OR, XOR, NOT, and ROT (bitwise rotation).

(10) Robustness. Ascon is a nonce-based scheme.

4. AES-COPA

(1) Online.

(2) Nonce misuse resistance. AES-COPA is designed to maintain security when the nonce is reused. Specifically, it achieves the maximum possible security against nonce reuse for an online AE scheme.

(3) Efficiency. AES-COPA is designed to allow high-performance implementations in both software and hardware.

(4) Combination of well-known techniques. AES-COPA relies on the design principles of PMAC to achieve integrity of both the associated data and the plaintext.

(5) Encryption and decryption do not require both AES and its inverse.

5. ELmD

(1) Efficient.

(2) Nonce misuse resistant.

(3) Online.

(4) Fully pipeline implementable. ELmD has an encrypt-mix-encrypt structure and processes where the associated data and message are in identical format. It makes ELmD fully parallel and pipeline implementable.

(5) Resistant against block-wise adaptive adversaries. ELmD efficiently incorporates intermediate tags and provides security against blockwise adaptive adversaries.

(6) Provision for skipping intermediate tags during decryption. During decryption, the plaintext computation is independent of the intermediate tag computations. Hence, if intermediate verifications are not required, the extra computations required for verifying the intermediate tags can be skipped in ELmD.

(7) Robustness. ELmD works perfectly even if associated data is empty. ELmD performs well when used as a tweakable encryption scheme, IV-based stream-cipher, or MAC only. ELmD provides associated data integrity.

6. Deoxys

(1) Security margin. Deoxys has a good security margin for all the recommended parameters.

(2) Security proofs. The security arguments of Deoxys are directly inherited from the two modes used in its design.

(3) Software implementations. Deoxys achieves good performance for software implementations.

(4) Small messages. Deoxys is efficient for small messages, which is particularly important in many lightweight applications where the messages sent are usually composed of a few dozen bytes.

(5) Theoretical performances. The number of calls to the internal primitive is minimized.

(6) Well understood design. Deoxys also benefits from the vast research literature on AES cryptanalysis.

(7) Simplicity. Deoxys is simple for both the construction of the internal tweakable block cipher and the authentication mode.

(8) Flexibility. Deoxys has smooth parameter handling.

(9) Resistant to side-channel attacks. Deoxys can resist side-channel attacks with the same techniques as AES.

(10) Beyond-birthday-bound security. The nonce-misuse resistant mode, Deoxys-II, provides graceful degradation of security with the maximum number of nonce repetitions.

7. MORUS

(1) Efficient in software. The speed of MORUS-1280 is 0.69 cycles/byte (cpb) on Intel Haswell processors for long messages, around 30% faster than that of AES-GCM.

(2) Fast in hardware performance. In MORUS, the critical path to generate a keystream block is 3 AND gates and 8 XOR gates.

(3) Efficient across platforms. The MORUS family offers steady performance across platforms because its performance does not rely on the use of an AES-NI instruction set.

(4) Secure. MORUS provides 128-bit authentication security, which is stronger than that of AES-GCM.

8. OCB

(1) Fast. OCB is almost as fast as CTR.

(2) Provably secure. OCB is the result of more than a decade of research and development. It is

secure in the sense of a nonce-based AE scheme, if its underlying block cipher is a strong PRP.

(3) Parallel.

(4) Timing-attack resistant. There are no conditional computations in OCB that depend on secret data.

(5) Online.

(6) Static AD. When associated data is unchanging over a series of encryptions, the associated data's contribution does not need to be recalculated each time.

9. AES-OTR

(1) AES key.

(2) Inverse-free.

(3) Online.

(4) One-pass.

(5) Parallel.

(6) Rate-1 processing for both encryption and decryption.

(7) Provable security up to about $2^{n/2}$ input blocks, based on the assumption that E_K is a pseudorandom function (PRF).

10. AEZ

(1) Arbitrary key length, nonce length, and authenticator length.

(2) Nonce reuse resistant. Secure against nonce reuse in the strongest sense of the phrase.

(3) Unverified plaintext. It is fine to release unverified plaintext. This is one aspect of our notion of a robust AE.

(4) Parallel.

(5) Inverse-free.

(6) Static AD.

(7) Fast rejection. Invalid ciphertexts can be rejected far more quickly than valid ones being decrypted.

11. CLOC/SILC

(1) Block cipher only. It uses only the encryption of the block cipher for both encryption and decryption, and does not use bitwise operations.

(2) No precomputation is needed other than blockcipher key scheduling.

(3) Two state blocks. It works with two state blocks.

(4) Online.

(5) Static associated data can be processed efficiently if the corresponding intermediate state value is stored.

(6) Secure. Privacy and authenticity are proved based on the PRP assumption of the block cipher, assuming standard nonce-respecting adversaries.

12. JAMBU

(1) Lightweight. In addition to the registers used in the underlying block cipher, the JAMBU authenticated encryption mode requires only one additional register with half of the block size.

(2) Partial resistance against IV reuse. When the IV is accidentally reused under the same key, the security of encryption and authentication is not completely compromised.

13. Ketje

(1) Lightweight. All Ketje instances except Ketje major are lightweight in the sense that they have a small code and working memory footprint and require a relatively small amount of computation.

(2) Round function reuse. The implementation of the round function can be re-used for other symmetric cryptographic primitives, such as hashing. It further reduces the footprint compared to a solution with distinct primitives.

(3) Side channel resistant. Ketje lends itself well to protections against side-channel attacks, in both hardware and software.

(4) Session support. As a functional feature not present in most authenticated ciphers, Ketje supports sessions.

14. Kayak

(1) Session support.

(2) Efficiency. An important advantage of Kayak is its hardware efficiency, with better performance and cost trade-off compared to AES-GCM.

(3) Side channel resistant. The round function can be easily protected against different types of side-channel attacks.

15. NORX

(1) High security. NORX supports 128- and 256-bit keys and authentication tags of arbitrary size, thanks to its duplex construction.

(2) Efficiency. NORX is designed with 64-bit processors in mind, but is also compatible with smaller architectures like 8- to 32-bit platforms.

(3) Simplicity. NORX requires no S-Boxes, no Galois field operations, and no integer arithmetic.

(4) High key agility. NORX requires no key expansion when setting up a new key.

(5) Adjustable tag sizes. The NORX family uses a default tag size of $4w$ bits for our proposed

instances.

(6) Simple integration. NORX can be easily integrated into a protocol stack, because it supports flexible processing of arbitrary datagrams.

(7) Interoperability. Dedicated datagrams encode parameters of the cipher and encapsulate the protected data.

(8) Single-pass.

(9) Online.

(10) High data processing volume. NORX allows processing of very large data volumes from a single key–nonce pair.

(11) Minimum overhead. Payload encryption is non-expanding.

(12) Robustness against timing attacks. By avoiding data-dependent table look-ups, like S-boxes, and integer additions, the goal to harden software and hardware implementations of NORX against timing attacks should be easy to achieve.

(13) Moderate misuse resistance. NORX retains its security on nonce reuse as long as it can be guaranteed that header data is unique.

(14) Autonomy. NORX requires no external primitive.

(15) Diversity. The cipher does not depend on AES instructions.

(16) Moderate misuse resistance. NORX can be easily extended to support additional features.

16. Tiaoxin

(1) Original intension. It is a nonce-based, software-oriented design based on a stream cipher.

(2) Fewer AES rounds. It is the first AE scheme to use only three AES rounds per 16-byte message. More precisely, it uses six AES round calls per 32-byte message. All the six calls are fully parallelizable.

(3) Hardware efficiency. It achieves 0.28 cpb on Intel Haswell. Depending on the software platform, it is 3.5 to 6.5 times faster than the benchmark AES-GCM, and twice as fast as OCB3.

(4) Attack resistant. Tiaoxin has been analyzed against various types of attacks. Most of the design decisions were made to make the cipher secure. The security claims are the maximum expected in the framework of nonce-respecting adversaries. It provides full security against related-key attacks.

(5) Long messages. Tiaoxin accepts very long messages of sizes up to $2^{128}-1$ bits. There is no loss of security on long messages.

(6) Optimal state size. State sizes are found to be optimal among all the state sizes following the design strategy and security.

The AE mode and some features of the 15 candidates are depicted in Table 6, including whether the intermediate tags are supported in AE, whether AD is used, whether parallel computing is supported, whether the candidate is online, and other related features.

5 Comprehensive performance of the schemes

In this section we evaluate the comprehensive performances of seven finalists and eight other candidate AE schemes in the third round of the CAESAR competition. Their performances are listed below, compared with that of AEC-GCM.

1. ACORN

ACORN has better hardware performance than AES-GCM. ACORN 293-bit has a similar hardware speed to 288-bit TRIVIUM, but ACORN has more complex feedback loops. ACORN's 32 steps can run in parallel on hardware implementations. As for software implementations, if the encrypted message size increases from 64 to 4096 bytes, the comprehensive performance increases from 72.1 to 11.9 cpb.

2. AEGIS

AEGIS-128L has twice the running speed of AES-GCM, with less than half of AES-GCM's cost. The scheme provides a 128-bit authenticated tag for security, which is stronger than that of AES-GCM. AEGIS performs better than CCM, GCM, OCB3, ALE, and ASC-1. When encrypting a 4096-byte message, the performance of AEGIS-128L reaches 0.48 cpb, and the performance of AEGIS-128 and AEGIS-256 reaches about 0.7 cpb. Among these three versions of AEGIS, AEGIS-128L runs the fastest.

3. Ascon

The initial state of Ascon is only 320-bit long, which means that Ascon has less hardware cost than AES-GCM. It has less overhead and better performance than AES-GCM in both hardware and software implementations. One of Ascon's disadvantages is that it cannot operate in parallel and cannot use high-performance tools like the AES-NI instruction set.

4. COLM/AES-COPA/ELmD

Table 6 Some features of the third-round candidates

Candidate	Design construction	AE mode	Tag	AD	Parallel Enc/Dec	Block cipher mode	Design prototype	Mask design feature	Online	Original intension
ACORN	SC-AE	–	–	√	√/√	–	ACORN	–	√	Lightweight
AEGIS	D-AE	–	–	√	√/–	–	AES-round	–	√	Fast/AES-NI
Ascon	SH-AE	Ascon	–	√	–/–	–	Ascon	Duplex	√	Lightweight
COLM	BC-AE	PHASH &EME*	–	√	√/√	EME	AES	Doubling	√	Fast/AES-NI
AES-COPA	BC-AE	PMAC &XEX	–	√	√/√	XEX	AES	Doubling	√	Fast/AES-NI
ELmD	BC-AE	PHASH &EME*	√	√	√/√	EME	AES	Doubling	√	Fast/AES-NI
Deoxys	BC-AE	Deoxys	–	√	√/√	EME/TAE	Deoxys-BC	–	√	Lightweight
MORUS	D-AE	–	–	√	–/–	–	MORUS	–	√	Fast
OCB	BC-AE	OCB	–	√	√/√	XEX	AES	Doubling	√	Fast
AES-JAMBU	BC-AE	JAMBU	–	√	–/–	OFB	AES	–	√	Lightweight
AES-OTR	BC-AE	OTR	–	√	√/√	OTR	AES	Doubling	√	Fast
AEZ	BC-AE	AEZ	–	√	√/√	OTR	AES4/10	–	–	Fast/Low power
CLOC	BC-AE	CBCMAC &CFB	–	√	–/–	CFB	AES	–	√	Low-overhead
SILC	BC-AE	CBCMAC &CFB	–	√	–/√	CFB	AES	–	√	Lightweight
Ketje	SH-AE	Monkey-Wrap	√	√	–/–	–	Keccak-f	Duplex	√	Based on Keccak
Keyak	SH-AE	Motorist	√	√	√/√	–	Keccak-f	Duplex	√	Based on Keccak
NORX	SH-AE	–	–	√	√/√	–	Ng	Duplex	√	Lightweight
Tiaoxin	D-AE	Toxin-346	–	√	√/√	–	AES-round	–	√	Fast

*Ng: present modes are not used. √: the character is provided; –: not mentioned.

COLM has better parallel capability than COPA and ELmD. Compared with AES-GCM, COPA has better resistance against nonce-misuse attacks, and runs faster than AES in some cases. Different from AES-GCM, AES-COPA has no weak keys and does not require multiplication in GF(2^{128}). ELmD is an authentication cipher that resists nonce misuse. Compared with AES-GCM, it can provide online security when resisting nonce-reuse attacks. Each time a cipher block needs to be calculated, AES-GCM needs to consider the hardware overhead including both AES block cipher operations and field multiplication operations, but ELmD needs only to consider AES block cipher operations. ELmD is also superior to AES-GCM on software.

5. Deoxys

Deoxys uses a lot of AES-based designs and can operate AES-NI instructions in parallel. The nonce-respecting mode is named Deoxys-I, and the nonce-misuse mode is named Deoxys-II. Both have better comprehensive performance than AES-GCM. Deoxys is based on tweakable block ciphers and performs well for short messages.

6. MORUS

The software comprehensive performance of MORUS-1280 reaches 0.69 cpb, which is 30% faster than AES-GCM. On hardware, only three AND gates and eight XOR gates are used to generate block keys. As for long messages, the comprehensive performance of MORUS-640 and MORUS-1280 can reach 1.11 and 0.69 cpb, respectively, whereas that of AES-GCM reaches only 1.03 cpb.

7. OCB

OCB has comprehensive performance similar to CTR, and a security level and feature settings similar to their counterparts of AES-GCM. In contrast with AES-GCM, the tag length of OCB can be truncated or reduced. Under the same conditions, OCB has better comprehensive performance than GCM, but slightly worse performance than CTR.

8. AES-JAMBU

AES-JAMBU needs only one additional half-blocksize register, while AES-GCM needs two additional blocksize registers. AES-GCM requires a lot of memory for a lookup table, which makes it not applicable for lightweight encryption devices. In contrast,

AES-JAMBU might be applicable for lightweight devices. When dealing with a 4096-byte message under the same conditions, the comprehensive performance of AES-JAMBU reaches 9.98 cpb, whereas that of AES-GCM reaches only about 2.07 cpb.

9. AES-OTR

AES-OTR may reduce the comprehensive performance in block ciphers. However, when using the AES-NI instruction set, AES-OTR basically has the same comprehensive performance as AES-OCB.

10. AEZ

The comprehensive performances of AEZ and AES-CTR are basically the same. The best performance of AEZ can reach 0.63 cpb on Intel Skylake and 1.3 cpb on Apple A9 ARM. It has better comprehensive performance than AES-GCM.

11. CLOC/SILC

CLOC can be applied to embedded processors and lightweight encryption equipment. It does not use the whole Galois field (GF) in the mixing procedure. As a result, it costs less when using AES operations. In contrast, AES-GCM is not applicable to embedded devices. SILC does not use GF in the mixing procedure either. It performs well on hardware. Compared with CLOC and SILC, AES-GCM requires a lot of gate circuits to complete mixing on GF.

12. Ketje

Compared with AES-GCM, Ketje's round function can be applied to other symmetric encryption schemes. Ketje adopts a function-based design structure and supports session mode. It is also applicable to lightweight scenarios.

13. Keyak

Keyak has better comprehensive performance than AES-GCM. It adopts SHA-3 as the basis of its design. When calculating the hash, it allows reuse of the computational resources to improve operation efficiency.

14. NORX

NORX is designed for 64-bit architecture, but is also applicable for 8- and 32-bit architectures. It takes advantage of some features in microprocessor architecture. It is applicable to any parallel situation during payload processing. Similar to AES-GCM, NORX does not need additional key expansion while generating a new key. It uses queries to select keys instead.

15. Tiaoxin

Tiaoxin uses only six AES round functions for every 32-byte message, and all these round functions can be operated in parallel. In counter mode, it runs twice as fast as AES-128, 3.5 to 6.5 times faster than AES-GCM, and twice as fast as OCB3. The overall performance of Tiaoxin reaches 0.28 cpb, and its efficiency can be further improved.

It is obvious that most of these candidates have improved comprehensive performance to some extent as compared to AES-GCM. However, due to the diverse application scenarios, different AEs have different requirements in the running environment and parameter settings. As a result, the evaluation standards of AEs are different, which makes it more complicated to compare the comprehensive performances of different AE schemes in the CAESAR competition.

6 Security analysis of finalists

The seven finalists can be categorized into different groups according to their designed structures, as mentioned in Section 4. Three finalists use block-cipher-based authenticated encryption schemes. ACORN is a stream-cipher-based scheme, and Ascon is a sponge-based scheme. The other algorithms, AEGIS and MORUS, are dedicated AE schemes.

6.1 Security analysis of BC-AE schemes

6.1.1 Analysis of COLM/AES-COPA/ELmD

COLM was designed by Andreeva et al. (2016a). It is based on encryption-mixing-encryption mode, which means there is a simple linear mixing layer between the two encryption layers. COLM can achieve online misuse resistance and protect against blockwise adaptive adversaries. The comprehensive performance of COLM is similar to the best performance of COPA and ELmD. COLM supports parallel authentication, mixing encryption, and AE with intermediate tags. AES-COPA was designed by Andreeva et al. (2015). It is based on AES structure with the AES-NI instruction set. COPA has good hardware performance and resists chosen plaintext attacks (CPAs) and forgery attacks. ELmD was designed by Andreeva et al. (2016b). It is based on the encrypt-linear mix-decrypt mode. ELmD can also resist blockwise adaptive adversaries.

Dobraunig et al. (2016a) presented a statistical fault attack on AES-COPA. AES-COPA uses an XEX-like structure for encryption. Dobraunig et al. (2017) operated on collections of faulty ciphertexts. They found that part of the master key can be recovered by two statistical fault attacks. The statistical fault attack is also applicable to ELmD. Lu (2015) presented a universal birthday-bound forgery attack on COPA. The results showed that the security claim of AES-COPA against tag guessing might not be correct. Lu (2017) further improved the idea. Bossuet et al. (2016) found that ELmD could achieve nonce misuse resistance. When a nonce is reused, ELmD can achieve online resistance and overall confidentiality. They showed that theoretically ELmD had good integrity and confidentiality for private data. Bay et al. (2016) provided universal forgery attacks and key recovery attacks. Their key recovery attacks can reduce the effective key strength by more than 60 bits.

Nandi (2015) discussed COPA with respect to the direction of cryptanalysis and provable security. Dobraunig et al. (2016b) performed a statistical fault attack on AES-COPA. The complexity and the required fault number of their attacks are the same as their counterparts of the attacks on simple AES. Kotegawa et al. (2016) discussed the hardware implementations of several CAESAR authenticated ciphers, including COLM. Deshpande and Gaj (2017) characterized the CAESAR competition candidates for the first time and discussed parallel suitability when processing multiple blocks of associated data, messages, and ciphertexts. Note that they chose AES-COPA as a detailed example for analysis. Forler et al. (2017) summarized previous forgery attacks on AES-COPA and presented their own opinions.

6.1.2 Analysis of Deoxys

Deoxys was designed by Jean et al. (2016). It is a new tweakable block cipher based AE scheme. There are two versions of Deoxys, Deoxys-I and Deoxys-II. They are applicable to nonce-respecting and nonce-misuse cases, respectively. Note that only Deoxys-II became one of the finalists. Deoxys can be used for lightweight AE. Dobraunig et al. (2016b) proved that Deoxys is vulnerable to statistical fault attacks, and the adversaries can recover the last round key of Deoxys. Deoxys (Jean et al., 2016) runs for at least four rounds, which makes it resistant against differential attacks. Furthermore, as a new AE mode, Deoxys can resist meet-in-the-middle attacks.

Koteshwara et al. (2017) provided an evaluation of Deoxys using the Altera Cyclone V family of FPGAs. They described simplified flow diagrams and presented a detailed summary of the timing performance, area, memory, and energy requirements of AES-GCM and Deoxys. Their analysis showed that Deoxys requires 10% less energy per bit and 25% fewer LUTs than AES-GCM.

Cid et al. (2017) provided the first independent security analysis of Deoxys. They showed that it is possible to attack 10 rounds of Deoxys-BC-256 and 13 rounds of Deoxys-BC-384. Sasaki (2018) improved Cid's attack. They reduced the complexities of 8- and 9-round related-tweakey boomerang distinguishers against Deoxys-BC-256 to 2^{28} and 2^{98}, respectively.

Mehrdad et al. (2018) described several impossible differential cryptanalyses on the round-reduced variants of Deoxys-BC-256. They presented the first third-party cryptanalysis of Deoxys-BC-256 in the single key model. Mehrdad et al. (2018) presented a key-recovery attack on Deoxys-I. However, their attack cannot be applied to Deoxys-II.

6.1.3 Analysis of OCB

OCB is an RFC 7253 scheme designed by Krovetz and Rogaway (2016). In the submitted version, the authors explained the scheme's parameter settings and performance. The purpose of OCB is to achieve two security properties, confidentiality and authenticity, using high-mixing pseudo-random function (PRF) permutation, such as AES-like block ciphers. During the OCB process, the difference during execution must be eliminated to resist timing attacks. Furthermore, OCB is not designed to resist nonce reuse, which means it is required to use a never-used nonce each time to ensure that the nonce is sufficiently random. In terms of security analysis, Sun et al. (2012) pointed out that it is hard for OCB to resist the collision attack, and proposed an exemplary collision attack.

Bhaumik and Nandi (2017) improved the integrity bound of OCB3. When the number of encryption query blocks is not larger than the birthday bound, the adversary may fail to halt the integrity of OCB3. Zhang P et al. (2016) introduced two modified schemes, OCB-IC and OCB-IPC, in the nonce-

misuse setting. OCB-IC and OCB-IPC are proven INT-RUP up to the birthday bound in the nonce-misuse setting, if the underlying tweakable block cipher is a secure mixed tweakable pseudo-random permutation (MTPRP). Ertaul et al. (2016) presented the Data Vault, which is an Android data storage application, to store the data securely using OCB. Their work showed that OCB is quite suitable for mobile applications. Clift (2014) presented a hardware model of OCB3 in System Verilog hardware description language, to show that the hardware approach also performs better than AES-GCM.

6.2 Security analysis of SC-AE schemes

6.2.1 Analysis of ACORN

ACORN was designed by Wu (2016). It is a stream cipher based AE scheme. ACORN calculates with three basic functions and performs well on hardware.

Liu and Lin (2014) pointed out that ACORN v1 is vulnerable to slide attacks. Wu (2016) had two basic assumptions about AE schemes. One is that the encryption nonce cannot be reused. The other is that the scheme cannot output decrypted plaintexts if authentication fails. Chaigneau et al. (2015) analyzed these assumptions. Their results showed that these two assumptions could not be both satisfied. By solving the system linear equations, the 128-bit key of ACORN could be recovered. Salam et al. (2016b) showed that ACORN v1 has risks against state collision attacks, and the risks could be used in forgery attacks. Salam et al. (2016a) proposed cube attacks to a round-reduced version of ACORN. They showed that some ACORN linear equations can be easily generated, which can lead to state recovery attacks with a complexity of about $2^{72.8}$. Josh and Sarkar (2015) perceived some outcomes on the key stream bits of ACORN v1. They observed that bitwise XOR of the first key stream bits with a fixed key and IV and different associated data becomes 0. Lafitte et al. (2016) studied ACORN's security against a SAT-based cryptanalysis. They provided the first practical and efficient attacks on the first and the last versions of ACORN. More precisely, they achieved state recovery, key recovery, state collision, and forgery attacks. Dey et al. (2016a) presented a hardware fault attack on ACORN.

Dalai and Roy (2017) proposed a state recovery attack on ACORN with 2^{120} complexity. The attack can recover the state of the encryption phase of ACORN. In their attack, the adversary needs to inject 326 faults and to obtain 10 known plaintext bits. Zhang XJ et al. (2017, 2018) proposed a fault attack on ACORN v2 and v3. They assumed that the random fault is injected into the initial state of ACORN v2 and v3. Their research showed that compared with ACORN v2, the tweaked version, ACORN v3, was more vulnerable against the fault attack. Siddhanti et al. (2017) mounted a DFA on ACORN v3 that requires nine faults to recover the state. As for ACORN v3, they recovered the secret key once the state was known. Dwivedi et al. (2016) investigated ACORN, aiming at new state recovery attacks using the SAT solver as a main tool. Their analysis revealed that the ACORN scheme has strong resistance against SAT-based state recoveries.

6.3 Security analysis of SH-AE schemes

6.3.1 Analysis of Ascon

Ascon was designed by Dobraunig et al. (2016c). It is a sponge-based AE. The scheme combines standards such as AES, SHA-3, and eStream, which provides large security boundaries for Ascon. Ascon has good efficiency and security performance.

Dobraunig et al. (2016c) used cube-like, differential, and linear cryptanalysis to evaluate Ascon security. They proposed practical key-recovery attacks on round-reduced versions of Ascon-128, where the initialization step is reduced to 5 out of 12 rounds, whereas theoretical key recovery attacks are possible for 6 rounds of the initialization step. They also presented a practical forgery attack for 3 rounds of the finalization step, a theoretical forgery attack for 4-round finalization. They also proposed zero-sum distinguishers for the full 12-round Ascon permutation. Groß et al. (2015) presented hardware implementations of Ascon for high performance. For instance, they showed that their implementation is already enough to encrypt a Gigabit Ethernet connection. Ascon is fast and small, and it can also be easily protected against differential power attacks (DPAs). Jovanovic et al. (2014) analyzed the sponge function of Ascon. The sponge function can achieve $2^{c/2}$ security, where c is its capacity. They showed that sponge-based constructions for AE can achieve

a significantly higher bound. Their results are effective for Ascon.

Farahmand et al. (2018) provided true lightweight implementations of the selected ciphers at round 3, including Ascon-128 and Ascon-128a. Dobraunig et al. (2017) proposed a re-keying approach, and presented a symmetric AE scheme that is secure against DPA attacks. Agrawal et al. (2017) proposed a new way to handle a long ciphertext with a small buffer size by storing and releasing only one intermediate state. They applied their generalized technique of storing a single intermediate state to all the CAESAR submissions, and found that only Ascon satisfied the limited memory constraint using their technique. Gross et al. (2017) implemented Ascon in hardware and optimized Ascon to fully explore its design space for different typical applications. Yalla and Kaps (2017) evaluated the lightweight package in two case studies. They developed the first lightweight implementations of Ascon-128 and Ascon-128a. Samwel and Daemen (2017) presented and applied the first CPA attack on Ascon. Unterluggauer et al. (2018) proposed MEAS, the first memory encryption and authentication scheme against DPA attacks. They gave a concrete MEAS instance based on lightweight Ascon. Through investigating six AE schemes (ACORN, Ascon-128a, Ketje Jr, ICEPOLE-128a, MORUS, and NORX-32), Dwivedi et al. (2016) aimed at state recovery attacks using the SAT solver as a main tool. They concluded that these schemes provide strong resistance against SAT-based state recoveries. Li et al. (2017) evaluated the security level of Ascon against a cube-like attack.

6.4 Security analysis of D-AE schemes

6.4.1 Analysis of AEGIS

AEGIS was designed by Wu and Preneel (2013). It is a D-AE scheme constructed from the AES encryption round function and the AES-NI instruction set.

AEGIS is implemented using AES rounds (Abdellatif et al., 2017). It performs well in hardware, and has a fast encryption/decryption speed and high computational complexity. The FlexRay network (Xue, 2016) can provide security protection for AEGIS, and improves its authentication capability without reducing its operation speed. AEGIS can satisfy 128-bit authentication security, which is stronger than AES-GCM. AEGIS has better confidentiality against statistical attacks and internal collisions. Similar to sponge-based AE (Minaud, 2014), AEGIS uses part of a secret key stream to update its internal state values. Their results showed that the secret key stream has linear leakage. Dey et al. (2016b) proposed differential fault analysis of Tiaoxin and the AEGIS family of ciphers in a nonce-reuse setting. Their analysis showed that the states of AEGIS-128, AEGIS-256, and AEGIS-128L can be recovered with 384, 512, and 512 single-bit faults, respectively.

Mary and Begum (2017) proposed an AEGIS scheme that can be used to moderate DoS attacks in web applications. Their proposed work explained the DoS attack and then tested it in a simulated environment. The outcomes are examined during the early phase of the research. They refined the AEGIS scheme based on investigation and can identify the various types of DoS attack patterns.

6.4.2 Analysis of MORUS

MORUS is a D-AE scheme designed by Wu and Huang (2016). It has good software and hardware implementation efficiency. MORUS protects privacy data through cryptography operations such as permutation.

Mileva et al. (2015) presented several observations of MORUS v1, but the above presented results do not threaten MORUS's security. From the perspective of completeness and differential diffusivity (Zhang P et al., 2015), the secret key can be recovered if the adversary can recover the IV. They proposed statistical attacks and internal collisions on MORUS (Wu and Huang, 2016), and proved that MORUS still has good security.

Dwivedi et al. (2016) investigated the MORUS security margin. They proposed a new key recovery approach, and Dwivedi et al. (2017) also verified the resistance of MORUS against internal differential and rotational cryptanalysis. Their analysis revealed that MORUS has a solid security margin. Shi et al. (2016) proposed the necessary conditions for an internal state collision after two-step update. Salam et al. (2017) investigated the application of cube attacks on MORUS. They applied the cube attacks to a version of MORUS where the initialization phase was reduced from 16 to 4 steps. Their

analysis showed that the cube attack could successfully recover the secret key of MORUS-640 with a total complexity of about 2^{10} for this reduced version. Their attack can similarly break MORUS-1280 with complexity 2^9. Salam et al. (2018b) also investigated the application of fault attacks on MORUS. Ashur et al. (2018) analyzed the components of MORUS and reported several results. They showed a 3-round forgery attack, and a 10-round key-recovery attack in the nonce-misuse setting.

7 Security analysis of other candidates in round 3

There remain eight AE schemes in the third round that are not finalists. Similar to the seven finalists, these eight schemes can be categorized into different groups based on their designed structure, according to Section 4. Four of them are block-cipher-based AE schemes, three are sponge-based AE schemes, and the Tiaoxin is a dedicated AE scheme. There is no scheme that follows the stream-cipher-based design.

7.1 Security analysis of BC-AE schemes

7.1.1 Analysis of AES-JAMBU

JAMBU was proposed by Wu and Huang (2014). It is a nonce-based AE mode that can be applied to any block cipher. The CAESAR candidate, AES-JAMBU, uses AES-128 as the internal operation primitive and JAMBU as the AE mode.

One of the security claims of JAMBU is nonce-misuse resistance. Peyrin et al. (2015) showed that this claim can be broken. They pointed out that there may exist possible attacks that might require only about 2^{32} encryption queries and computations. They also showed how their attack can be extended in the nonce-respecting scenario. They finally discussed how JAMBU could be patched to resist these attacks they mentioned.

Wang et al. (2017) discussed the limitation of AES-JAMBU by giving security proofs under both nonce-respecting and nonce-misuse cases. They proved that in the nonce-respecting case, JAMBU had slightly worse security than the birthday bound of n bits, and in the nonce-misuse case, JAMBU had a tight security bound of $n/2$ bits.

7.1.2 Analysis of AES-OTR

OTR was designed by Minematsu (2016). It uses a balanced two-round Feistel network for encryption, and is based on AES.

Sadeghi and Alizadeh (2014) proposed forgery attacks against AES-OTR with observations. By intercepting part of the input plaintexts, they forged the filtering and selecting operations under different execution conditions, and presented the success rate of their forgery attacks.

Dobraunig et al. (2016b) pointed out that AES-OTR is vulnerable to statistical fault analysis attacks. AES-OTR uses only two rounds of the balanced Feistel network, which makes it possible to recover the key by statistical analysis. Bost and Sanders (2016) showed that AES-OTR does not achieve such a property for a large number of parameters. They described the collisions between the input masks and explained a practical attack against AES-OTR.

Deshpande and Gaj (2017) implemented a two-stage inner-round pipeline for all the candidate algorithms including AES-OTR to improve the throughput.

Banik et al. (2016) investigated implementation in a compact fashion using the 8-bit serialized AES circuit. They investigated three AE modes: CLOC, SILC, and AES-OTR. Dobraunig et al. (2016a) presented the first practical fault attack on several nonce-based AE modes for AES including AES-OTR.

Vaudenay and Vizár (2017) described attacks with birthday complexity and nonce reuse for each of the candidates including AES-OTR. Kotegawa et al. (2016) performed hardware implementations of CAESAR candidates including AES-OTR with VIVADO high-level synthesis. Then they showed various techniques to optimize the speed, area, and clock frequency.

Al Mahri et al. (2016) investigated the security of OTR mode against forgery attacks. They showed that in the current instantiation, some forgeries might be constructed. In addition, they proposed a new way to instantiate OTR so that the masking coefficients are distinct, thus generalizing OTR without weakening its security.

7.1.3 Analysis of AEZ

AEZ was designed by Hoang et al. (2014). It is a lightweight AE based on AES. AEZ satisfies the RAE (robust AE) conditions, and has good confidentiality and robustness. While implementing AE, AEZ maintains strong robustness and can prevent parameter misuse attacks effectively.

Hoang et al. (2016) proved that AEZ remains secure when a nonce is repeated. A nonce-reuse misuse-resistant AE scheme must make two passes over the data, and it cannot be online. AEZ challenges the presumption that two-pass AE schemes have an intrinsic problem. Hoang et al. (2015) proved that AEZ adopted RAE mode, so it satisfies the robustness requirements. Fuhr et al. (2014) proposed a collision attack against AEZ v2 and v3. Then the designers modified AEZ and proposed AEZ v4.1. However, Chaigneau and Gilbert (2016) showed that AEZ v4.1 remains vulnerable to key-recovery attacks by presenting an attack on AEZ.

Bonnetain (2017) showed that all the versions of AEZ are completely broken against a quantum adversary. They proposed a generalization for the quantum period, and found that it is possible to build efficient attacks. Al Mahri et al. (2017) investigated differential fault attacks against AEZ v4.2. Shi et al. (2018) considered the security of AEZ-prf for AEZ v4.2, which is the latest version of AEZ. They found collision-associated data, and then launched collision attacks under different assumptions. Mennink (2017) observed that the tweakable block cipher used in AEZ suffers from structural design issues. One of the three 128-bit subkeys can possibly become zero.

7.1.4 Analysis of CLOC/SILC

Compact low-overhead CFB (CLOC) was proposed initially by Minematsu et al. (2016) at the FSE conference in 2014. Compared with the new version submitted to the CAESAR competition, the old version in FSE is different in terms of the minimum byte size and block size. Simple Lightweight CFB (SILC) (Iwata et al., 2014) is designed based on CLOC. It aims at optimizing the hardware overhead of CLOC.

Both CLOC and SILC adopt serial encryption, so both of them are applicable for encrypting and authenticating short messages. Dobraunig et al. (2016b) proposed statistical fault attacks against CFB mode. Because CLOC and SILC are both designed based on CFB mode, Dobraunig et al. (2016c) claimed that their attacks are applicable to the CLOC/SILC scheme.

Banik et al. (2016) provided a low-area hardware implementation of CLOC and SILC. Roy et al. (2016, 2017) presented single fault based almost-universal forgeries on both CLOC and SILC. They also proposed new constructions that can resist the fault-based forgery, assuming that the underlying block cipher is fault resistant.

7.2 Security analysis of SH-AE schemes

7.2.1 Analysis of Ketje/Keyak

Ketje and Keyak were both designed by Bertoni et al. (2015, 2016), and are based on Keccak. Ketje adopts MonkeyWrap and uses MonkeyDuplex permutation to separate two bits in each group instead of direct permutation with a single bit, which is adopted by SpongeWrap. Keyak adopts Motorist. It calculates with a larger bit number, and performs better on hardware.

The round-reduced Keccak sponge function is the basis of Ketje and Keyak. Dinur et al. (2015) briefly introduced round-reduced Keccak. Stoffelen (2015) proved that the nonlinear element in both Ketje and Keyak is the same as that in the hash function Keccak. This element has a very efficient hardware implementation. Moreover, they proved that the S-box of Ketje and Keyak has a multiplicative complexity of 5. Morawiecki et al. (2015) applied key-recovery cube-attack-like attacks and balanced attacks, and then proved that Keyak has a higher analysis complexity.

Fuhr et al. (2018) studied the security of Ketje against divide-and-conquer attacks. They showed that under some conditions, Ketje Jr becomes vulnerable to divide-and-conquer attacks with time complexities of 271.5 for the original version and 282.3 for the current tweaked version, both with a 96-bit key. Dong et al. (2017) gave the first attacks on 6/7-round reduced Ketje Sr. According to their attack, it could be claimed that 7-round reduced Ketje Sr v2 is weaker than Ketje Sr v1 against cube-like attacks. For Ketje Sr v1, the time complexities of 6- and 7-round attacks are 265.6 and 211.3, respectively. For Ketje Sr. v2, the time complexity of a

7-round attack is 297.

Samwel and Daemen (2017) performed side-channel analysis on hardware implementations of Keyak. They presented the first DPA attack on Keyak. To ensure security against side-channel attacks of Keyak, Meyers et al. (2017) presented a masked implementation of Keyak on an ARM Cortex M4, which is nonce-reuse. However, the implementation has not been tested yet, so they cannot claim that it offers protection against first-order DPA.

Liu and Liu (2017) proposed an efficient universal forgery attack on Keyak. They also proposed an efficient key recovery attack that can be implemented in $O(c)$. Their attacks showed that Keyak is completely broken in the quantum model. Bi et al. (2017) evaluated the security level of the river Keyak against cube-like attacks. They extended the key-recovery attack on river Keyak to eight rounds within the time complexity 2^{81}.

Wetzels and Bokslag (2015) presented an overview of the algorithms and design components underlying the Keccak cryptographic primitive and Keyak. They aimed to familiarize readers with the basic principles of AE, the Sponge and Duplex constructions, and the permutation functions underlying Keccak and Keyak.

Song et al. (2017) found the best attack against Keyak with 128-bit keys in the nonce-respecting setting, and 9 rounds of Keyak can be attacked if the key size is 256 bits.

7.2.2 Analysis of NORX

NORX was designed by Aumasson et al. (2015). It is a sponge-based AE based on MonkeyDuplex structure. NORX supports parallel and random-length authenticated tags.

Aumasson et al. (2014b) proved that NORX has a unique parallel and scalable architecture, so it can be applied to various encryption scenarios. Aumasson et al. (2014a) presented a thorough analysis of NORX, focusing on differential and rotational properties. They gave upper bounds on the differential probability, and discussed some rotational properties of the core permutation. NORX is due to a domain separation method that relies on the intangibility of the inner part of the state (Mennink et al., 2015). In related analysis, NORX has better privacy and reliability than other sponge-based functions. Das et al. (2015) proposed the higher-order differential properties of NORX. Bagheri et al. (2016) presented state/key recovery attacks for both NORX32 and NORX64, and gave the corresponding time and data complexities of their attack. Furthermore, they showed a state recovery attack against NORX in parallel using internal differential attacks. They also presented a practical distinguisher for the keystream of NORX64 based on two rounds of the permutation in parallel using an internal differential-linear attack.

Biryukov et al. (2017) analyzed the core permutation. Their results showed that under the Markov assumption, up to 2.125 rounds of the special F function of NORX32 and NORX64 can be distinguished. Kumar et al. (2018) proposed an optimized NORX, which is 40.81% faster and 18.01% smaller compared with the state-of-the-art NORX implementation. Their scheme improved the throughput per area by 76.9% compared with state-of-the-art NORX. Huang and Wu (2018) showed that the NORX core permutation is non-ideal if defending against a new distinguishing attack. Specifically, they could distinguish a NORX64 permutation with 248.5 queries and distinguish a NORX32 permutation with 264.7 queries using differential linear attacks.

7.3 Security analysis of D-AE schemes

7.3.1 Analysis of Tiaoxin

Tiaoxin was designed by Nikolić (2016). It is a nonce-based dedicated AE scheme. Tiaoxin uses only three AES rounds per 16-byte message (six rounds per 32-byte message). All the six calls are fully in parallel, which can improve the performance of Tiaoxin.

As the designers claimed, the adversary who analyzes differential and linear trails of Tiaoxin does not have access to the precise values of the state bytes. As a result, linear attacks and differential analysis will not be a threat to the security of Tiaoxin. Tiaoxin can also resist rotational attacks, internal differentials, and fixed-point attacks (Nikolić, 2016).

Dey et al. (2016b) proposed differential fault analysis of Tiaoxin. Their results showed that the secret key of Tiaoxin can be recovered with 384 single-bit faults. In addition, Salam et al. (2018a) described two different fault injection attacks on Tiaoxin-346.

Their first attack is similar to Dey's attack, and the second attack uses a random fault model to recover the secret key of the cipher. Their result showed that a successful attack has a computational complexity of 2^{36}.

8 Development trend analysis

When discussing design and implementation of AE schemes, it is important to find the balance between efficiency and security. To improve efficiency, many existing AE schemes use AES-NI, pipeline, super-scalar architecture, and SIMD instructions. Thus, the existing processor architecture can be fully used. To provide a communication platform, the CAESAR competition holds Directions In Authenticated Ciphers (DIAC) once a year to discuss the candidate schemes. At DIAC conferences, academia and industry experts evaluate these candidate schemes in terms of security, hardware, and software performance.

Lightweight is one of the most promising development trends for authenticated ciphers. Some AE schemes in the CAESAR competition aim at lightweight application scenarios, which are quite efficient in software and hardware implementation. Today, mobile phones are much more commonly used in daily life, and more attention has been paid to mobile phone information protection, especially on Android platforms (Wang et al., 2018). Applying lightweight authenticated ciphers to mobile phone security systems (Zhang WZ et al., 2016) can be taken into consideration. It can be predicted that new authenticated ciphers can improve mobile phone security to a certain degree. Moreover, in recent years, as the use of IoT and artificial intelligence (AI) has continuously increased, there are more and more terminals trying to adapt AI techniques to IoT systems (Zhang T et al., 2017). AE schemes in a resource-constrained environment can maintain the security of these terminals without reducing their efficiency, and lightweight AE is therefore in urgent demand.

NIST has also paid much attention to lightweight cryptography (McKay et al., 2017). In April 2018, NIST issued the first call for lightweight cryptography to protect small electronics. It is important to protect the data created by innumerable tiny networked devices such as those in the IoT, which will need a new class of cryptographic defense against cyber attacks. The scope of NIST's lightweight cryptography project includes all cryptographic primitives and modes that are needed in constrained environments. The AE scheme is an initial focus of the project. It seems reasonable that lightweight candidate schemes submitted to the CAESAR competition might possibly participate in the NIST competition for lightweight cryptography.

Note that to ensure security, many AE schemes are based on the block cipher and have provable security. For example, AES-JAMBU, AES-OTR, and AEZ are designed based on AES, whereas others such as Ascon and Ketje/Keyak adopt the core function in the SHA-3 standard. The security of these core functions has been evaluated by many cryptographers who can guarantee relative provable security of AE schemes. In addition, research into AE schemes in the CAESAR competition further promotes the encryption mode. The candidate schemes in the last round mostly adopt new AE modes. Some submitters even proposed a dedicated AE scheme. Security analysis and evaluation of those new modes and schemes has also become a popular international research topic.

In practical application scenarios, the security of AE schemes is not the same as hardware implementation security. While designing and implementing AE schemes, the submitters must consider their schemes' security against side-channel leakage, fault injection, and other attacks. There has been much related research on security evaluation of AE schemes.

On the one hand, some researchers have proposed differential and other power analyses of these schemes to evaluate their resistance to side-channel attacks such as power analysis. Compared with the power analysis of traditional schemes, it is more important to select a suitable power leakage point. Most power analysis needs several rounds.

On the other hand, some researchers have proposed differential fault analysis, statistical fault analysis, and other fault analysis to attack new AE schemes. Most of their fault injections target the initialization step, because it will cost less and have better practical feasibility.

In addition, some researchers have focused on resistance against real fault attacks. They presented security strategies such as masks, out-of-order execution, balanced circuit, and parity checking (Rivain and Prouff, 2010; Veyrat-Charvillon et al., 2012) to

improve the security of AE schemes.

Other researchers have proposed AE schemes that resist leakage (Pereira et al., 2015; Berti et al., 2016). They considered side-channel attacks at the beginning of their design. Their schemes can ensure provable security even when leakages exist.

9 Conclusions

In this study, we introduce and discuss the CAESAR competition, which looked for new authenticated ciphers that could provide stronger security and better performance than the present AE schemes. We review the progress of the CAESAR competition, which was funded by NIST in 2013 and lasted five years. First, we give the introduction of authenticated ciphers. Second, we introduce the requirements and the progress of the CAESAR competition. Third, we classify and analyze the finalists and the remaining candidates in the third round. The corresponding features, performance, and security are elaborated. Finally, we predict the development trend of authenticated ciphers in the future.

Authenticated ciphers are proposed to provide both encryption and authentication. Compared with the existing network security protocols which adopt different schemes during implementation, new AE schemes have several advantages. They have lower costs, better confidentiality, and improved integrity. They can also reduce the complexity of key management and reduce the risks caused by the simple and direct concatenation of encryption and authentication. Therefore, related research into constructing new AE schemes from the design stage is becoming more and more promising, and collecting and discussing these works is quite meaningful.

References

Abdellatif KM, Chotin-Avot R, Mehrez H, 2017. AES-GCM and AEGIS: efficient and high speed hardware implementations. *J Signal Proc Syst*, 88(1):1-12. http://doi.org/10.1007/s11265-016-1104-y

Agrawal M, Chang D, Sanadhya SK, 2017. A new authenticated encryption technique for handling long ciphertexts in memory constrained devices. *Int J Appl Cryptogr*, 3(3):236-261. http://doi.org/10.1504/IJACT.2017.086223

Al Mahri HQ, Simpson L, Bartlett H, et al., 2016. Tweaking generic OTR to avoid forgery attacks. Proc 6th Int Conf on Applications and Techniques in Information Security, p.41-53. https://doi.org/10.1007/978-981-10-2741-3_4

Al Mahri HQ, Simpson L, Bartlett H, et al., 2017. A fault-based attack on AEZ v4.2. Proc IEEE Trustcom/BigDataSE/ICESS, p.634-641. https://doi.org/10.1109/trustcom/bigdatase/icess.2017.294

Andreeva E, Bogdanov A, Luykx A, et al., 2015. AES-COPA v.2. CAESAR Submission.

Andreeva E, Bogdanov A, Luykx A, et al., 2016a. AES-COPA v.1. Submission to the CAESAR competition.

Andreeva E, Bogdanov A, Datta N, 2016b. ELmD v2.1. CAESAR Third Round Submission.

Ashur T, Eichlseder M, Lauridsen MM, et al., 2018. Cryptanalysis of MORUS. Int Conf on the Theory and Application of Cryptology and Information Security, p.35-64.

Aumasson JP, Jovanovic P, Neves S, 2014a. Analysis of NORX: investigating differential and rotational properties. Proc 3rd Int Conf on Cryptology and Information Security in Latin America, p.306-324. https://doi.org/10.1007/978-3-319-16295-9_17

Aumasson JP, Jovanovic P, Neves S, 2014b. NORX: parallel and scalable AEAD. Proc 19th European Symp on Research in Computer Security, p.19-36. https://doi.org/10.1007/978-3-319-11212-1_2

Aumasson JP, Jovanovic P, Neves S, 2015. NORX v3.0. Submission to CAESAR (2016).

Bagheri N, Huang T, Jia KT, et al., 2016. Cryptanalysis of reduced NORX. Proc 23rd Int Conf on Fast Software Encryption, p.554-574. https://doi.org/10.1007/978-3-662-52993-5_28

Banik S, Bogdanov A, Minematsu K, 2016. Low-area hardware implementations of CLOC, SILC and AES-OTR. IEEE Int Symp on Hardware Oriented Security and Trust, p.71-74. https://doi.org/10.1109/HST.2016.7495559

Bay A, Ersoy O, Karakoç F, 2016. Universal forgery and key recovery attacks on ELmD authenticated encryption algorithm. Proc 22nd Int Conf on the Theory and Application of Cryptology and Information Security, p.354-368. https://doi.org/10.1007/978-3-662-53887-6_13

Bellare M, Namprempre C, 2008. Authenticated encryption: relations among notions and analysis of the generic composition paradigm. *J Cryptol*, 21(4):469-491. http://doi.org/10.1007/s00145-008-9026-x

Bellare M, Rogaway P, Wagner D, 2003. A conventional authenticated-encryption mode. Manuscript.

Bellare M, Rogaway P, Spies T, 2010. The FFX mode of operation for format-preserving encryption. NIST Submission.

Berti F, Koeune F, Pereira O, et al., 2016. Leakage-resilient and misuse-resistant authenticated encryption. IACR Cryptology ePrint Archive: Report 2016/996.

Bertoni G, Daemen J, Peeters M, et al., 2011. Duplexing the sponge: single-pass authenticated encryption and other applications. Int Workshop on Selected Areas in Cryptography, p.320-337. https://doi.org/10.1007/978-3-642-28496-0_19

Bertoni G, Daemen J, Peeters M, et al., 2015. Keyak v2. CAESAR Submission.

Berton G, Daemen J, Peeters M, et al., 2016. Ketje v2. CAESAR Submission.

Bhaumik R, Nandi M, 2017. Improved security for OCB3. Proc 23rd Int Conf on the Theory and Application of Cryptology and Information Security, p.638-666. https://doi.org/10.1007/978-3-319-70697-9_22

Bi WQ, Li Z, Dong XY, et al., 2017. Conditional cube attack on round-reduced River Keyak. *Des Code Cryptogr*, 86(6):1295-1310. https://doi.org/10.1007/s10623-017-0396-7

Biryukov A, Udovenko A, Velichkov V, 2017. Analysis of the NORX Core Permutation. IACR Cryptology ePrint Archive: Report 2017/034.

Bonnetain X, 2017. Quantum key-recovery on full AEZ. Proc 24th Int Conf on Selected Areas in Cryptography, p.394-406. https://doi.org/10.1007/978-3-319-72565-9_20

Bossuet L, Datta N, Mancillas-López C, et al., 2016. ELmD: a pipelineable authenticated encryption and its hardware implementation. *IEEE Trans Comp*, 65(11):3318-3331. http://doi.org/10.1109/TC.2016.2529618

Bost R, Sanders O, 2016. Trick or tweak: on the (in)security of OTR's tweaks. Proc 22nd Int Conf on the Theory and Application of Cryptology and Information Security, p.333-353. https://doi.org/10.1007/978-3-662-53887-6_12

Chaigneau C, Gilbert H, 2016. Is AEZ v4.1 sufficiently resilient against key-recovery attacks? *IACR Trans Symmetr Cryptol*, 2016(1):114-133. https://doi.org/10.13154/tosc.v2016.i1.114-133

Chaigneau C, Thomas F, Gilbert H, 2015. Full key-recovery on ACORN in nonce-reuse and decryption-misuse settings. Posed on the Crypto-Competition Mailing List.

Cid C, Huang T, Peyrin T, et al., 2017. A security analysis of deoxys and its internal tweakable block ciphers. *IACR Trans Symmetr Cryptol*, 2017(3):73-107. https://doi.org/10.13154/tosc.v2017.i3.73-107

Clift P, 2014. Hardware Implementation of Offset Codebook Mode3 (OCB3). MS Thesis, California State University, Sacramento, USA.

Dalai DK, Roy D, 2017. A state recovery attack on ACORN-v1 and ACORN-v2. Proc 11th Int Conf on Network and System Security, p.332-345. https://doi.org/10.1007/978-3-319-64701-2_24

Das S, Maitra S, Meier W, 2015. Higher order differential analysis of NORX. IACR Cryptology ePrint Archive: Report 2015/186.

Deshpande S, Gaj K, 2017. Analysis and inner-round pipelined implementation of selected parallelizable CAESAR competition candidates. Euromicro Conf on Digital System Design, p.274-282. https://doi.org/10.1109/DSD.2017.80

Dey P, Rohit RS, Adhikari A, 2016a. Full key recovery of ACORN with a single fault. *J Inform Secur Appl*, 29:57-64. http://doi.org/10.1016/j.jisa.2016.03.003

Dey P, Rohit RS, Sarkar S, et al., 2016b. Differential fault analysis on Tiaoxin and AEGIS family of ciphers. Proc 4th Int Symp on Security in Computing and Communication, p.74-86. https://doi.org/10.1007/978-981-10-2738-3_7

Dinur I, Morawiecki P, Pieprzyk J, et al., 2015. Cube attacks and cube-attack-like cryptanalysis on the round-reduced Keccak sponge function. Proc 34th Annual Int Conf on the Theory and Applications of Cryptographic Techniques, p.733-761. https://doi.org/10.1007/978-3-662-46800-5_28

Dobraunig C, Eichlseder M, Korak T, et al., 2016a. Practical fault attacks on authenticated encryption modes for AES. IACR Cryptology ePrint Archive: Report 2016/616.

Dobraunig C, Eichlseder M, Korak T, et al., 2016b. Statistical fault attacks on nonce-based authenticated encryption schemes. Proc 22nd Int Conf on the Theory and Application of Cryptology and Information Security, p.369-395. https://doi.org/10.1007/978-3-662-53887-6_14

Dobraunig C, Eichlseder M, Mendel F, et al., 2016c. Ascon v1.2. Submission to the CAESAR Competition.

Dobraunig C, Eichlseder M, Mangard S, et al., 2017. ISAP—towards side-channel secure authenticated encryption. *IACR Trans Symmetr Cryptol*, 2017(1):80-105. https://doi.org/10.13154/tosc.v2017.i1.80-105

Dong XY, Li Z, Wang XY, et al., 2017. Cube-like attack on round-reduced initialization of Ketje Sr. *IACR Trans Symmetr Cryptol*, 2017(1):259-280. https://doi.org/10.13154/tosc.v2017.i1.259-280

Dwivedi AD, Klouček M, Morawiecki P, et al., 2016. SAT-based cryptanalysis of authenticated ciphers from the CAESAR competition. IACR Cryptology ePrint Archive: Report 2016/1053.

Dwivedi AD, Morawiecki P, Wójtowicz S, 2017. Differential and rotational cryptanalysis of round-reduced MORUS. Proc 14th Int Joint Conf on e-Business and Telecommunications, p.275-284. http://doi.org/10.5220/0006411502750284

Dwivedi AD, Klouček M, Morawiecki P, et al., 2016. SAT-based cryptanalysis of authenticated ciphers from the CAESAR competition. IACR Cryptology ePrint Archive: Report 2016/1053.

Dworkin M, 2016. Recommendation for block cipher modes of operation: methods for format-preserving encryption. NIST Special Publication 800-38G.

Ertaul L, Sravya KL, Sanka N, 2016. Implementation of authenticated encryption algorithm offset code book (OCB). Proc Int Conf on Wireless Networks, p.78-84.

Farahmand F, Diehl W, Abdulgadir A, et al., 2018. Improved lightweight implementations of CAESAR authenticated ciphers. Cryptology ePrint Archive: Report 2018/573.

Forler C, List E, Lucks S, et al., 2017. Reforgeability of authenticated encryption schemes. Proc 22nd Australasian Conf on Information Security and Privacy, p.19-37. https://doi.org/10.1007/978-3-319-59870-3_2

Fuhr T, Leurent G, Suder V, 2014. Collision attacks against CAESAR candidates. Proc 21st Int Conf on the Theory and Application of Cryptology and Information Security, p.510-532. https://doi.org/10.1007/978-3-662-48800-3_21

Fuhr T, Naya-Plasencia M, Rotella Y, 2018. State-recovery attacks on modified Ketje Jr. *IACR Trans Symmetr Cryptol*, 2018(1):29-56. https://doi.org/10.13154/tosc.v2018.i1.29-56

Gligor VDP, 2016. Extended cipher block chaining encryption. Submission to NIST.

Gligor VD, Donescu P, 2001. Fast encryption and authentication: XCBC encryption and XECB authentication modes. Int Workshop on Fast Software Encryption, p.92-108.

Groß H, Wenger E, Dobraunig C, et al., 2015. Suit up!-Made-to-measure hardware implementations of ASCON. Euromicro Conf on Digital System Design, p.645-652. https://doi.org/10.1109/DSD.2015.14

Gross H, Wenger E, Dobraunig C, et al., 2017. ASCON hardware implementations and side-channel evaluation. *Microprocess Microsyst*, 52:470-479. http://doi.org/10.1016/j.micpro.2016.10.006

Halevi S, 2004. EME*: extending EME to handle arbitrary-length messages with associated data. Proc 5th Int Conf on Cryptology in India, p.315-327. https://doi.org/10.1007/978-3-540-30556-9_25

Halevi S, Rogaway P, 2004. A parallelizable enciphering mode. Cryptographers' Track at the RSA Conf, p.292-304. https://doi.org/10.1007/978-3-540-24660-2_23

Hellström H, StreamSec H, 2001. Propagating cipher feedback mode. Proc 2nd NIST Modes of Operation Workshop.

Hoang VT, Krovetz T, Rogaway P, 2014. AEZ v1: authenticated-encryption by enciphering. CAESAR 1st Round.

Hoang VT, Krovetz T, Rogaway P, 2015. Robust authenticated-encryption AEZ and the problem that it solves. Proc 34th Annual Int Conf on the Theory and Applications of Cryptographic Techniques, p.15-44. https://doi.org/10.1007/978-3-662-46800-5_2

Hoang VT, Krovetz T, Rogaway P, 2016. AEZ v4. 2: authenticated encryption by enciphering. CAESAR Submission.

Huang T, Wu HJ, 2018. Distinguishing attack on NORX permutation. *IACR Trans Symmetr Cryptol*, 2018(1):57-73. https://doi.org/10.13154/tosc.v2018.i1.57-73

Hwang S, Lee C, 2015. Padding Oracle attack on block cipher with CBC|CBC-double mode of operation using the BOZ-PAD. *J Soc e-Buss Stud*, 20(1):89-97. https://doi.org/10.7838/jsebs.2015.20.1.089

Iwata T, Minematsu K, Guo J, et al., 2014. SILC: simple lightweight CFB. Submission to the CAESAR Competition.

Jean J, Nikolić I, Peyrin T, et al., 2016. Deoxys v1.41. Submitted to CAESAR.

Josh RJ, Sarkar S, 2015. Some observations on ACORN v1 and Trivia-SC. Lightweight Cryptography Workshop, p.20-21.

Jovanovic P, Luykx A, Mennink B, 2014. Beyond $2^{c/2}$ security in sponge-based authenticated encryption modes. Proc 20th Int Conf on the Theory and Application of Cryptology and Information Security, p.85-104. https://doi.org/10.1007/978-3-662-45611-8_5

Jutla CS, 2001. Encryption modes with almost free message integrity. Int Conf on the Theory and Applications of Cryptographic Techniques, p.529-544.

Jutla CS, 2016a. Integrity aware cipher block chaining. Submission to NIST.

Jutla CS, 2016b. Integrity aware parallelizable mode. Submission to NIST.

Kaushal PK, Sobti R, Geetha G, 2012. Random Key Chaining (RKC): AES mode of operation. *Int J Appl Inform Syst*, 1(5):39-45. http://doi.org/10.5120/ijais12-450184

Kohno T, 2003. Carter Wegman (authentication) with Counter (encryption). http://csrc.nist.gov/CryptoToolkit/modes/proposedmodes/cwc/cwc-spec.pdf

Kotegawa M, Iwai K, Tanaka H, et al., 2016. Optimization of hardware implementations with high-level synthesis of authenticated encryption. *Bull Netw Comput Syst Soft*, 5(1):26-33.

Koteshwara S, Das A, Parhi KK, 2017. FPGA implementation and comparison of AES-GCM and Deoxys authenticated encryption schemes. IEEE Int Symp on Circuits and Systems, p.1-4. https://doi.org/10.1109/ISCAS.2017.8050315

Krovetz T, Rogaway P, 2016. OCB (v1.1). https://competitions.cr.yp.to/round3/ocbv11.pdf

Kumar S, Haj-Yahya J, Chattopadhyay A, 2018. Efficient hardware accelerator for NORX authenticated encryption. IEEE Int Symp on Circuits and Systems, p.1-5. https://doi.org/10.1109/ISCAS.2018.8351145

Lafitte F, Lerman L, Markowitch O, et al., 2016. SAT-based cryptanalysis of ACORN. IACR Cryptology ePrint Archive: Report 2016/521.

Li Z, Dong XY, Wang XY, 2017. Conditional cube attack on round-reduced ASCON. *IACR Trans Symmetr Cryptol*, 2017(1):175-202. https://doi.org/10.13154/tosc.v2017.i1.175-202

Liskov M, Rivest RL, Wagner D, 2002. Tweakable block ciphers. Proc 22nd Annual Int Cryptology Conf, p.31-46. https://doi.org/10.1007/3-540-45708-9_3

Liskov M, Rivest RL, Wagner D, 2011. Tweakable block ciphers. *J Cryptol*, 24(3):588-613. http://doi.org/10.1007/s00145-010-9073-y

Liu FB, Liu FM, 2017. Universal forgery and key recovery attacks: application to FKS, FKD and Keyak. Cryptology ePrint Archive: Report 2017/691.

Liu MC, Lin DD, 2014. Cryptanalysis of lightweight authenticated cipher ACORN. Posed on the Crypto-Competition Mailing List.

Lu JQ, 2015. On the security of the COPA and marble authenticated encryption algorithms against (almost) universal forgery attack. IACR Cryptology ePrint Archive: Report 2015/079.

Lu JQ, 2017. Almost universal forgery attacks on the COPA and marble authenticated encryption algorithms. Proc ACM Asia Conf on Computer and Communications Security, p.789-799. https://doi.org/10.1145/3052973.3052981

Mary DSN, Begum AT, 2017. An algorithm for moderating DoS attack in web based application. Int Conf on Technical Advancements in Computers and Communications, p.26-31. https://doi.org/10.1109/ICTACC.2017.17

McGrew D, Viega J, 2004. The Galois/counter mode of operation (GCM). Submission to NIST Modes of Operation Process.

McKay KA, Bassham LE, Turan MS, et al., 2017. Report on lightweight cryptography. NIST.

Mehrdad A, Moazami F, Soleimany H, 2018. Impossible differential cryptanalysis on deoxys-BC-256. Cryptology ePrint Archive, Report 2018/048.

Mennink B, 2017. Weak keys for AEZ, and the external key padding attack. Cryptographers' Track at the RSA Conf, p.223-237. https://doi.org/10.1007/978-3-319-52153-4_13

Mennink B, Reyhanitabar R, Vizár D, 2015. Security of full-state keyed sponge and duplex: applications to authenticated encryption. Int Conf on the Theory and Application of Cryptology and Information Security, p.465-489. https://doi.org/10.1007/978-3-662-48800-3_19

Meyers M, Daemen J, Batina L, 2017. Side channel protected Keyak on ARM cortex-M4. http://www.cs.ru.nl/bachelors-theses/2017/Martin_Meyers_4497899_Side_channel_protected_Keyak_on_ARM_Cortex-M4.pdf

Mileva A, Dimitrova V, Velichkov V, 2015. Analysis of the authenticated cipher MORUS (v1). Proc 2nd Int Conf on Cryptography and Information Security in the Balkans, p.45-59. https://doi.org/10.1007/978-3-319-29172-7_4

Minaud B, 2014. Linear biases in AEGIS keystream. Proc 21st Int Conf on Selected Areas in Cryptography, p.290-305. https://doi.org/10.1007/978-3-319-13051-4_18

Minematsu K, 2014. Parallelizable rate-1 authenticated encryption from pseudorandom functions. Proc 33rd Annual Int Conf on the Theory and Applications of Cryptographic Techniques, p.275-292. https://doi.org/10.1007/978-3-642-55220-5_16

Minematsu K, 2016. AES-OTR v3.1. Japan Submission to CAESAR, NEC Corporation.

Minematsu K, Guo J, Kobayashi E, 2016. CLOC and SILC. https://competitions.cr.yp.to/round3/clocsilcv3.pdf

Moise A, Beroset E, Phinney T, et al., 2011. EAX' Cipher Mode. NIST.

Morawiecki P, Pieprzyk J, Straus M, et al., 2015. Applications of key recovery cube-attack-like. IACR Cryptology ePrint Archive: Report 2015/1009.

Nandi M, 2015. Revisiting security claims of XLS and COPA. IACR Cryptology ePrint Archive: Report 2015/444.

Nikolić I, 2016. Tiaoxin v2.1. CAESAR Third Round Submission.

Pereira O, Standaert FX, Vivek S, 2015. Leakage-resilient authentication and encryption from symmetric cryptographic primitives. Proc 22nd ACM SIGSAC Conf on Computer and Communications Security, p.96-108. https://doi.org/10.1145/2810103.2813626

Peyrin T, Sim SM, Wang L, et al., 2015. Cryptanalysis of JAMBU. Proc 22nd Int Workshop on Fast Software Encryption, p.264-281. https://doi.org/10.1007/978-3-662-48116-5_13

Pub F, 1980. DES Modes of Operation. https://csrc.nist.gov/publications/detail/fips/81/archive/1980-12-02

Recacha F, 2016. Input and output chaining. Submission to NIST.

Rivain M, Prouff E, 2010. Provably secure higher-order masking of AES. Proc 12th Int Workshop on Cryptographic Hardware and Embedded Systems, p.413-427. https://doi.org/10.1007/978-3-642-15031-9_28

Rogaway P, 2004. Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. Proc 10th Int Conf on the Theory and Application of Cryptology and Information Security, p.16-31. https://doi.org/10.1007/978-3-540-30539-2_2

Rogaway P, 2016. Offset codebook. Submission to NIST.

Rogaway P, Shrimpton T, 2007. The SIV mode of operation for deterministic authenticated-encryption (key wrap) and misuse-resistant nonce-based authenticated-encryption. http://web.cs.ucdavis.edu/~rogaway/papers/siv.pdf

Rogaway P, Bellare M, Black J, et al., 2001. OCB Mode. IACR Cryptology ePrint Archive.

Rott J, 2010. Intel® Advanced Encryption Standard Instructions (AES-NI). Technical Report, Intel.

Roy DB, Chakraborti A, Chang D, et al., 2016. Fault based almost universal forgeries on CLOC and SILC. Proc 6th Int Conf on Security, Privacy, and Applied Cryptography Engineering, p.66-86. https://doi.org/10.1007/978-3-319-49445-6_4

Roy DB, Chakraborti A, Chang D, et al., 2017. Two efficient fault-based attacks on CLOC and SILC. *J Hardw Syst Secur*, 1(3):252-268. http://doi.org/10.1007/s41635-017-0022-1

Sadeghi H, Alizadeh J, 2014. A forgery attack on AES-OTR.

Salam I, Simpson L, Bartlett H, et al., 2017. Investigating cube attacks on the authenticated encryption stream cipher MORUS. IEEE Trustcom/BigDataSE/ICESS, p.961-966. https://doi.org/10.1109/trustcom/bigdatase/icess.2017.337

Salam I, Al Mahri HQ, Simpson L, et al., 2018a. Fault attacks on Tiaoxin-346. Proc Australasian Computer Science Week Multiconf, Article 5. https://doi.org/10.1145/3167918.3167940

Salam I, Simpson L, Bartlett H, et al., 2018b. Fault attacks on the authenticated encryption stream cipher MORUS. *Cryptography*, 2(1), Article 4. https://doi.org/10.3390/cryptography2010004

Salam MI, Bartlett H, Dawson E, et al., 2016a. Investigating cube attacks on the authenticated encryption stream cipher ACORN. Proc 6th Int Conf on Applications and Techniques in Information Security, p.15-26. https://doi.org/10.1007/978-981-10-2741-3_2

Salam MI, Wong KKH, Bartlett H, et al., 2016b. Finding state collisions in the authenticated encryption stream cipher ACORN. Proc Australasian Computer Science Week Multiconf, Article 36.

Samwel N, Daemen J, 2017. DPA on hardware implementations of Ascon and Keyak. Proc Computing Frontiers Conf, p.415-424. https://doi.org/10.1145/3075564.3079067

Sasaki Y, 2018. Improved related-tweakey boomerang attacks on deoxys-BC. Progress in Cryptology-AFRICACRYPT, p.87-106. https://doi.org/10.1007/978-3-319-89339-6_6

Schroeppel RC, Anderson WE, Beaver CL, et al., 2004. Cipher-state (CS) mode of operation for AES. Submission to NIST.

Shi TR, Guan J, Li JZ, et al., 2016. Improved collision cryptanalysis of authenticated cipher MORUS. Proc 2nd Int Conf on Artificial Intelligence and Industrial Engineering, p.429-432. http://doi.org/10.2991/aiie-16.2016.98

Shi TR, Jin CH, Guan J, 2018. Collision attacks against AEZ-PRF for authenticated encryption AEZ. *China Commun*, 15(2):46-53. http://doi.org/10.1109/CC.2018.8300271

Siddhanti A, Sarkar S, Maitra S, et al., 2017. Differential fault attack on grain v1, ACORN v3 and lizard. Proc 7th Int Conf on Security, Privacy, and Applied Cryptography Engineering, p.247-263. https://doi.org/10.1007/978-3-319-71501-8_14

Song L, Guo J, Shi DP, et al., 2017. New MILP modeling: improved conditional cube attacks on Keccak-based constructions. Cryptology ePrint Archive: Report 2017/1030.

Stoffelen K, 2015. Intrinsic Side-Channel Analysis Resistance and Efficient Masking. MS Thesis, Radboud University, Nijmegen, the Netherlands.

Sun ZL, Wang P, Zhang LT, 2012. Collision attacks on variant of OCB mode and its series. Proc 8th Int Conf on Information Security and Cryptology, p.216-224. https://doi.org/10.1007/978-3-642-38519-3_14

Unterluggauer T, Werner M, Mangard S, 2018. MEAS: memory encryption and authentication secure against side-channel attacks. *J Cryptogr Eng*, 2018(1):1-22. https://doi.org/10.1007/s13389-018-0180-2

Vaudenay S, Vizár D, 2017. Under pressure: security of Caesar candidates beyond their guarantees. Cryptology ePrint Archive: Report 2017/1147.

Veyrat-Charvillon N, Medwed M, Kerckhof S, et al., 2012. Shuffling against side-channel attacks: a comprehensive study with cautionary note. Proc 18th Int Conf on the Theory and Application of Cryptology and Information Security, p.740-757. https://doi.org/10.1007/978-3-642-34961-4_44

Wang G, Zhang HY, Liu FM, 2017. Security proof of JAMBU under nonce respecting and nonce misuse cases. Cryptology ePrint Archive: Report 2017/831.

Wang HR, He H, Zhang WZ, 2018. Demadroid: object reference graph-based malware detection in Android. *Secur Commun Netw*, Article 7 064 131. https://doi.org/10.1155/2018/7064131

Wegman MN, Carter JL, 1981. New hash functions and their use in authentication and set equality. *J Comp Syst Sci*, 22(3):265-279. http://doi.org/10.1016/0022-0000(81)90033-7

Wetzels J, Bokslag W, 2015. Sponges and engines: an introduction to Keccak and Keyak. http://arxiv.org/abs/1510.02856

Whiting D, Housley R, Ferguson N, 2003. Counter with CBC-MAC (CCM). Network Working Group.

Wu HJ, 2016. ACORN: a lightweight authenticated cipher (v3). Candidate for the CAESAR Competition.

Wu HJ, Huang T, 2014. JAMBU lightweight authenticated encryption mode and AES-JAMBU. CAESAR Competition Proposal.

Wu HJ, Huang T, 2016. The authenticated cipher MORUS (v2). https://competitions.cr.yp.to/round3/morusv2.pdf

Wu HJ, Preneel B, 2013. AEGIS: a fast authenticated encryption algorithm. Proc 20th Int Conf on Selected Areas in Cryptography, p.185-201. https://doi.org/10.1007/978-3-662-43414-7_10

Xue L, 2016. A Novel Approach for Flexray Protocol with Confidentiality and Authenticity. MS Thesis, National University of Singapore, Singapore.

Yalla P, Kaps JP, 2017. Evaluation of the CAESAR hardware API for lightweight implementations. Int Conf on ReConFigurable Computing and FPGAs (ReConFig), p.1-6. https://doi.org/10.1109/RECONFIG.2017.8279790

Zhang P, Guan J, Li JZ, et al., 2015. Research on the confusion and diffusion properties of the initialization of MORUS. *J Cryptol Res*, 2(6):536-548 (in Chinese). http://doi.org/10.13868/j.cnki.jcr.000100

Zhang P, Wang P, Hu HG, 2016. The INT-RUP security of OCB with intermediate (Parity) checksum. IACR Cryptology ePrint Archive: Report 2016/1059.

Zhang T, Li Q, Zhang CS, et al., 2017. Current trends in the development of intelligent unmanned autonomous systems. *Front Inform Technol Electron Eng*, 18(1):68-85. http://doi.org/10.1631/FITEE.1601650

Zhang WZ, Li X, Xiong NX, et al., 2016. Android platform-based individual privacy information protection system. *Pers Ubiq Comp*, 20(6):875-884. http://doi.org/10.1007/s00779-016-0966-0

Zhang XJ, Feng XT, Lin DD, 2017. Fault attack on the authenticated cipher ACORN v2. *Secur Commun Netw*, Article 3 834 685. https://doi.org/10.1155/2017/3834685

Zhang XJ, Feng XT, Lin DD, 2018. Fault attack on ACORN v3. *Comp J*, 61(8):1166-1179. http://doi.org/10.1093/comjnl/bxy044