DE GRUYTER
OPEN

TATRA MOUNTAINS
Mathematical Publications

DOI: 10.1515/tmmp-2016-0038
Tatra Mt. Math. Publ. **67** (2016), 167–190

THE STATE OF THE AUTHENTICATED ENCRYPTION

Damian Vizár

ABSTRACT. Ensuring confidentiality and integrity of communication remains among the most important goals of cryptography. The notion of authenticated encryption marries these two security goals in a single symmetric-key, cryptographic primitive. A lot of effort has been invested in authenticated encryption during the fifteen years of its existence. The recent Competition for Authenticated Encryption: Security, Applicability, and Robustness (CAESAR) has boosted the research activity in this area even more. As a result, the area of authenticated encryption boasts numerous results, both theoretically and practically oriented, and perhaps even greater number of constructions of authenticated encryption schemes.

We explore the current landscape of results on authenticated encryption. We review the CEASAR competition and its candidates, the most popular construction principles, and various design goals for authenticated encryption, many of which appeared during the CAESAR competition. We also take a closer look at the candidate Offset Merkle-Damgård (OMD).

1. Introduction

Perhaps the two most fundamental goals of symmetric-key cryptography are providing *confidentiality* (privacy) and *authenticity* (together with integrity[1]) of messages that are being sent over an insecure channel. These two security properties of communication have traditionally been studied separately; they were formalized in separate notions [13], [14], and achieved by separate primitives (e.g., CBC mode for confidentiality and CBCMAC for authentication).

2010 Mathematics Subject Classification: 94A60.
Keywords: authenticated encryption, CAESAR competition.
[1]Although these two properties are not the same, we will use them interchangeably, as they are coupled together in the context of authenticated encryption.

However, such a separation seldom occurs in practice. On the contrary, in a vast majority of applications, authenticity is needed as an adjunct to confidentiality. The struggle to obtain integrity cheaply with privacy-only schemes, e.g., by encoding a redundancy into plaintexts, was more often than not without much success (e.g., the void integrity protection in WEP wireless security). It became evident, that the problem of simultaneously achieving privacy and integrity with a symmetric-key encryption scheme needed a systematic treatment.

To fill this gap, Bellare and Rogaway (and independently also Katz and Yung) proposed a formal approach to solving this problem in 2000, and coined the term *authenticated encryption* (AE) [16], [46]. In the same year, Bellare and Namprempre investigated the security of combining a conventional encryption scheme and a MAC to achieve AE (so called *generic composition*) [15]. Soon after that, dedicated, single-key AE schemes appeared, most notably OCB (2001) [65], CCM (2002) [31], [72] and GCM (2004) [51]. The relevance of efficient AE to real-world applications demonstrated itself by the number of standards that appeared in this period. The CCM appears in IEEE 802.11i, IPsec ESP and IKEv2; the GCM is appears in NIST SP 800-38D; the EAX mode is specified in ANSI C12.22; and ISO/IEC 19772:2009 defines six AE schemes (five dedicated AE designs and one generic composition method).

The CAESAR competition

After the first wave of dedicated AE schemes, multiple serious issues have been discovered. These were issues with security of established AE schemes (mistakes in the security proof, weak keys and problems with short tags in GCM [32], [58], [68]), or their performance (non-parallelizability and off-line computation of CCM), or issues with widely deployed security protocols (e.g., the padding-oracle attacks on SSL [70]). A collection of further such failures appears in the list of "Disasters" in symmetric cryptography [18]. In 2006 Rogaway and Shrimpton also pointed out that certain type of implementation errors can render many popular schemes (including CCM and GCM) completely insecure and formalized *misuse-resistant* AE [66].

All this indicated that there is still need for further research in the field of AE. The CAESAR competition for authenticated encryption was initiated in 2013 to encourage research activity and public discussion in the area of AE. The call for submissions proclaimed that the final portfolio will include schemes that "offer advantages over AES-GCM" (as specified in NIST SP 800-38D) and that "are suitable for widespread adoption" [17].

The initiative met with a very strong response: 57 candidates were submitted to the first round. There is a great diversity of construction paradigms (blockcipher-based, permutation-based etc.), lower-level primitives (AES, Keccak permutation etc.) and security goals (basic AE security, misuse-resistant AE security etc.) among the submitted candidates. Beside the great number

of submitted schemes, CAESAR also induced a wave of research activity, including results on cryptanalysis, design of both cryptographic primitives and constructions, security analyses but also new security goals and formal models.

Organization of the paper
We briefly recall the notions of nonce-based AE with associated data (AEAD) and misuse resistant AE in Section 2. In Section 3, we review the CAESAR competition and submitted candidates, classifying them based on their construction principles and security goals. In Section 4 we take a closer look at the candidate OMD.

2. Authenticated encryption before CAESAR

Nonce-based AE with associated data (AEAD)
The goals of an AE scheme have been formally stated as early as in 2000, however the most widely accepted security notion for AE schemes appeared later in 2002 [64]. This notion, proposed by R o g a w a y differed from the initial ones in two key aspects.

The first was the way it strived to achieve strong, semantic security. Rather than randomized or stateful schemes, this notion considered schemes that have a deterministic encryption algorithm, but also take an additional, non-repeating initialization vector (aka. *nonce*) as input along with the message. The motivation for this was to minimize the requirements on implementations of the scheme, avoiding the need for generating random strings for every encryption and the necessity of storing a state between queries. The other aspect was addition of *associated data* (AD). This was inspired by the realization that in many situations, there are information we want to authenticate together with a message but which cannot be encrypted (e.g., a network packet header).

A nonce-based AEAD scheme Π is a triple $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ where $\mathcal{K}$ is the secret key space and $\mathcal{E}\colon \mathcal{K}\times\mathcal{N}\times\mathcal{A}\times\mathcal{M}\to\mathcal{C}\cup\{\perp\}$ and $\mathcal{D}\colon \mathcal{K}\times\mathcal{N}\times\mathcal{A}\times\mathcal{C}\to\mathcal{M}\cup\{\perp\}$ are deterministic encryption and decryption algorithms respectively. Both algorithms take four inputs: a secret key $K \in \mathcal{K}$, a nonce $N \in \mathcal{N}$, associated data (AD) $A \in \mathcal{A}$ and a plaintext (PT) $M \in \mathcal{M}$ in case of encryption, or a ciphertext (CT) $C \in \mathcal{C}$ in case of decryption. The encryption algorithm outputs a ciphertext while the decryption outputs either a plaintext or an error symbol $\perp$. It is required for every message $M \neq \perp$ that if $C = E(K, N, A, M)$, then $M = D(K, N, A, C)$ and that length of the ciphertext $|E(K, N, A, M)| = f(|A|, |M|)$ only depends on the length of the message and AD.[2]

[2]We typically have $|E(K, N, A, M)| = |M| + \tau$ for some positive constant τ.

The privacy goals of Π are captured by indistinguishability of ciphertexts from random strings under chosen plaintext attack. The privacy guarantees that Π offers against an adversary $\mathscr{A}$ are measured by the advantage function

$$\mathbf{Adv}_{\Pi}^{\mathbf{priv}} = \Pr\left[\mathscr{A}^{\mathcal{E}_K(\cdot,\cdot,\cdot)} \Rightarrow 1\right] - \Pr\left[\mathscr{A}^{\$(\cdot,\cdot,\cdot)} \Rightarrow 1\right].$$

The $\mathcal{E}_K(\cdot,\cdot,\cdot)$ denotes the encryption algorithm initialized by a random key and $\$(\cdot,\cdot,\cdot)$ denotes a dummy algorithm that returns a random string of $|\mathcal{E}_K(M)|$ bits. It required that every (encryption) query uses a unique nonce.

The authenticity guarantee of Π against $\mathscr{A}$ is formalized as $\mathscr{A}$'s inability to forge a valid ciphertext and measured as $\mathbf{Adv}_{\Pi}^{\mathbf{auth}} = \Pr\left[\mathscr{A}^{\mathcal{E}_K(\cdot,\cdot,\cdot)} \text{ forges}\right]$. We say that $\mathscr{A}$ forges, if it issues a decryption query N, A, C that decrypts to an $M \neq \perp$ under key K and that was not obtained through encryption queries.

The notion of AEAD is of great significance, as it has become the most frequently targeted goal for design of practical AE schemes, both before and during the CAESAR competition. The CCM, GCM and OCB schemes all follow the AEAD notion.

Misuse resistant AE

In 2006, R o g a w a y and S h r i m p t o n pointed out, that even though nonce--based AE schemes are relatively easy to implement, nonces can still get reused, due to implementation errors, improper use of the schemes or due to unavoidable constraints (e.g., after cloning a virtual machine). The reuse of initialization vector would cause complete break in many deployed AEAD schemes (such as CCM and GCM). In order to cope with these issues, R o g a w a y and S h r i m p t o n proposed a security goal that offered more robustness towards *misuse* of AEAD schemes.

The security of a nonce-misuse resistant AE (MRAE) scheme is defined through indistinguishability of the real scheme from an idealized object; one that returns random strings on encryption queries and that refuses every adversarial forgery attempt. The security of an MRAE scheme Π against an adversary $\mathscr{A}$ is measured through the advantage function

$$\mathbf{Adv}_{\Pi}^{\mathbf{mrae}}(\mathscr{A}) = Pr\left[\mathscr{A}^{E_K(\cdot,\cdot,\cdot),D_K(\cdot,\cdot,\cdot)} \Rightarrow 1\right] - Pr\left[\mathscr{A}^{\$(\cdot,\cdot,\cdot),\perp(\cdot,\cdot,\cdot)} \Rightarrow 1\right].$$

The adversary is only required not to repeat queries as triplets (N, A, M), it can repeat nonces however. In the same work, the misuse-resistant AE scheme SIV was introduced.

The introduction of MRAE foreshadowed a new direction of research: AE schemes and notions that are *robust* to improper use or implementation errors. This line of work, which is relevant to practice, was picked up shortly before the start of CAESAR by L u c k s et al. [34] and further pursued and expanded during the CAESAR competition.

3. CAESAR competition

The **C**ompetition for **A**uthenticated **E**ncryption: **S**ecurity, **A**pplicability, and **R**obustness (CAESAR) was officially announced in January 2013. The main goal of the CAESAR competition is to "identify a portfolio of authenticated ciphers that (1) offer advantages over AES-GCM and (2) are suitable for widespread adoption." [17] The selection of submissions for the final portfolio is done in an iterative way in three rounds; at the end of each round, the list of surviving candidates is narrowed down and the candidates selected at the end of the third round will comprise the final portfolio. The end of CAESAR is planned for 15th December 2017 at the time of writing of this paper. The submissions to CAESAR must meet several requirements. These concern mainly the interface and security properties of the submitted schemes.

Another, indirect goal of CAESAR is to boost the research activity in the area of AE. This can be seen as a consequence of the series of security issues in symmetric cryptography mentioned in Section 1. The competition is a definitive success along this line: there were as many as 57 submissions to the first round. In July 2015, 29 candidates advanced to the second-round [1].

Syntax of AE schemes

All CAESAR candidates must be compatible with the following interface. The encryption algorithm must take as input

- a variable-length plaintext,
- variable-length associated data,
- a fixed-length secret message number (SMN),
- a fixed-length public message number (PMN),
- and a fixed-length key.

The encryption algorithm must output a variable-length ciphertext such that "It must be possible to recover the plaintext and the secret message number from the ciphertext, associated data, public message number, and key."

Unlike the well-established syntax of AEAD, the newly proposed CAESAR syntax features two initialization vector-like inputs, the PMN and the SMN. The PMN alone appears to have a very similar role as the nonce: to act as a possibly non-repeating initialization vector (IV) and to get authenticated. The SMN cannot however be directly linked to any component of an AEAD scheme defined in Section 2: it appears to be an IV in essence, but it's encrypted and authenticated as well. Some clues about the purpose of the SMN can be found on the Cryptographic competitions website [19]: "The traditional view is that message numbers are not secret... Adding secrecy to message numbers requires changing this data flow: the authenticated ciphertext expands to include

(and perhaps to generate) the message number... An authenticated cipher that includes message numbers in ciphertexts can save bandwidth by not repeating the session number."

Also, the nonce-requirement of the CAESAR AE syntax is different, requiring that (SMN, PMN) never repeat as a pair. Namprempre et al. [55] pointed out that this modification could potentially lead to a divergent understanding about what exactly should be the security goals for such a scheme (i.e., how should one formulate the security notion). However, submitted designs were allowed not to support the secret message number and merely 5 out of 57 first-round candidates made use of the SMN. Only 2 out of these 5 advanced to the second round.

Security requirements

CAESAR submissions are required to provide integrity protection for all four non-key inputs (plaintext, associated data, secret message number, public message number) and confidentiality for the plaintext and the secret message number. All submissions are required to clearly state if they

1. require that public and secret message numbers be unique (as a pair) for every encryption query made with the same key for the security guarantees to hold (similarly to the nonce in AEAD), or
2. guarantee the MRAE-like security, or
3. "provide some intermediate level of robustness against message-number reuse, in which case this section must specify what that level of robustness is."

The research of robustness of AE schemes has increased during and shortly before CAESAR. The nonce-misuse resistance of online-computable schemes was disputed [5], [41]. Andreeva et al. formalized the security of AE schemes when the decryption algorithm leaks an unverified plaintext [9]. A strong notion of Robust AE (RAE) which implies resistance to both nonce-reuse and release of unverified plaintext was put forward by Hoang et al. [40].

3.1. CAESAR candidates

There were 57 submissions to the CAESAR competition. Nine submissions were withdrawn between the beginning of the first round (15th March 2014) and the beginning of the second round (7th July 2015) of the competition, after the discovery of serious flaws (all announced in the crypto-competitions mailing list [1]):

- 20th March 2014: M. O. Saarinen points out that forgeries for HKC are trivial.
- 20th March 2014: S. Neves identifies a differential property in McMambo that enables one to forge ciphertexts with high probability.
- 21st March 2014: M. Nandi announces a forgery attack on AES-COBRA.

- 24th March 2014: Y. Sasaki and L. Wang identify nonce-reusing forgery attacks on PAES and PANDA, breaking their nonce-misuse resistance claims.
- 2nd April 2014: F. Mendel, B. Mennink, V. Rijmen and E. Tischhauser announce a forgery attack on Calico.
- 3rd April 2014: CBEAM is withdrawn after a forgery attack by B. Minaud (attack uses a property attack of the underlying primitive).
- 11th April 2014: X. Feng announces a practical key-recovery attack on FASER.
- 14th January 2014: T. Fuhr, G. Leurent and V. Suder announce a generic forgery attack on the Marble mode of operation.

The remaining 48 CAESAR candidates show a great diversity in the applied construction principles, used building blocks and targeted security goals. We briefly look at which building blocks are used in the first round candidates and which security notions do they target. The AE Zoo [7] website and a paper of Abed et al. [6] compile useful overviews and we draw on the data they collected in the following paragraphs. The website of CAESAR holds submission documentation of all the candidates [2].

Building blocks and functionalities

The 48 first-round CAESAR candidates that were not withdrawn can be classified into 7 classes based on the primitive they rely on.

Blockcipher-based: Out all first-round candidates, 24 use a blockcipher as a lower-level primitive. The most frequently used blockcipher is AES (17 candidates). Among the remaining candidates, two use 4-round AES (AEZ and Marble), and three use tweakable blockciphers based on the Tweakey framework [43]. The blockcipher-based candidates being in fact blockcipher modes of operation, the majority either internally uses or modifies one of the previously known blockcipher modes of operations, such as CTR (3 candidates), CFB (2 candidates), ECB (6 candidates), CBC (1 candidate), OTR (2 candidates), or EME (6 candidates).

Stream cipher-based: Seven candidates make use of either an existing and well analysed (ChaCha, Trivium) or a dedicated stream cipher.

Permutation-based: Three candidates are using dedicated, keyless permutations as a lower-level primitive. The candidates in this category do not use the permutation in a sponge-like mode but apply other techniques (e.g., derivations of the Even-Mansour construction).

Sponge-based: Nine candidates are using a keyless permutation in a sponge-like mode of operation. Among these, two candidates use the Keccak-f

permutation used in the SHA3 hash function, others rely on dedicated permutations.

Compression function-based: A single candidate (OMD) uses compression functions from the SHA256 and SHA512 hash functions.

Based on dedicated primitives: 3 candidates are based on primitives that do not fall into the previous categories.

Not based on a primitive: A single candidate (POLAWIS) is not based on any typical symmetric-key primitive.

A b e d et al. also list a few desirable functional characteristics related to the "applicability" of the AE schemes and indicate which candidates have which features. We list these properties.

Parallelizability: Some candidates are fully parallelizable, i.e., both encryption and decryption algorithm can be executed on several independent computational units (e.g., OCB, OTR). Other candidates introduce mechanisms that introduce parallelism into otherwise serial modes (NORX, Keyak). Other schemes targeting stronger security notions are not parallelizable, however their internal structure allows one to leverage pipelined execution of primitive calls (POET).

Online-ness: We say that an (encryption) algorithm is "online" if it can process the input data in a single pass, with constant memory and constant latency (a very desirable feature for hardware implementations). In different words, encryption algorithm is online if it can output ith CT block after having read only the first i plaintext blocks. All candidates apart from those targeting the full nonce-misuse resistant security (AEZ, iFeed, AES-CMCC, Julius, HS1-SIV) and Trivia-ck seem to have online encryption.

Inverse freeness: AE schemes that do not need to evaluate both forward and inverse calls to the lower-level primitive (e.g., blockcipher and its inverse) save memory in software and area in hardware implementations. Fourteen blockcipher-based candidates have this property and only three non-blockcipher-based candidates are not inverse-free.

Incremental encryption/AD processing: Some AE schemes are constructed in a way that lets them encrypt a query (N, A, M) much more efficiently if they have previously encrypted a query (N', A', M) that differs from (N, A, M) only slightly (e.g., by a few bits) in 1) the message (incremental AE), or 2) AD (incremental AD). Fourteen candidates can process AD incrementally and only PAEQ can incrementally process both AD and message.

Static AD reuse: Certain constructions allow to speed up processing of a query, if it has the same AD as a previously processed query. 15 candidates benefit from this feature.

Intermediate tags: Unlike the majority, some submissions support the option of including intermediate authentication tags in regular intervals in ciphertexts. While increasing the consumption of resources related to ciphertext expansion, this allows an early discovery of forgery attempts for long messages. Seven submission have this option.

Security goals

All CAESAR candidates are *required* to ensure authenticity for all, non-key inputs to the encryption algorithm, and privacy for the SMN and the plaintext if the PMN, SMN pair never repeats. When stated formally, this corresponds (after exclusion of the SMN) to what is captured by the notion for nonce-based AEAD, as defined in Section 2. Most of the candidates provide these security guarantees.

Candidates are however free to offer additional security. Several security notions (apart from MRAE) regarding the security of AE schemes in presence of a "misuse" have been proposed and targeted by CAESAR candidates. We list the notions here. We mention *nonces* rather than (PMN, SMN) pairs when we describe the notions, as most of the schemes do not use the SMN.

Nonce-misuse resistant AE: The original security notion for AE schemes that are robust to reuse of nonces (as described in Section 2). MRAE schemes retain full authenticity if nonces are reused and privacy is only damaged by the possibility to detect repetition of messages (if nonce and AD repeat as well). This security goal is targeted by HS1-SIV and some modes of Deoxys, Joltik, Kiasu and Julius.

Nonce-misuse resistant online AE: Fleischmann et al. have formalized a security notion for online AE schemes (OAE) that retain full level of authenticity and *some* privacy if nonces are reused; such schemes would leak the length of longest common prefix in n-bit blocks for two plaintexts, n being the blocksize internally used by the scheme. Several candidates have followed this notion (COPA, POET, ElmD and some modes of Prøst). This notion is seen as controversial by some. In 2015, Hoang et al. [41] pointed out several shortcomings of the OAE notion and argued that the privacy assurance offered by OAE-secure schemes under nonce-reuse should not be labelled as nonce-misuse resistance. They also pointed out that several candidates have further weakened the notion of nonce-misuse resistance to suit their scheme's abilities (e.g., Minalpher).

Decryption-misuse resistance: Andreeva et al. have formalized security of AE schemes under release of unverified plaintext. The motivation

for this is the fact that when decrypting, many AE schemes first internally decrypt the ciphertext to a putative plaintext and release it once the authenticity check is successful, which leaves the possibility of the putative plaintext leaking. This type of security is achieved by POET, AEZ, Minapher and some modes of Primates and Prøst.

Robust AE: H o a n g et al. have formalized "best possible" AE security [40]. The notion captures security of an AE scheme with selectable ciphertext expansion by indistinguishability from a family of random injections. Security in the sense of Robust AE (RAE) implies security under both the nonce-reuse and the decryption reuse.

Additionally to the type of security guarantees, candidates are also required to commit to *quantitative* security levels in bits for privacy and authenticity separately, i.e., indicate the complexity of the attack on the scheme in terms of the complexity of an exhaustive search on indicated number of bits. The candidates have several sets of quantitative security levels, which are both instance-and-security goal-specific.

3.2. Second round candidates

We list the 29 second round candidates in Tables 1 and 2, specifying what primitive they are based on and what security they target.

Second round tweaks

Upon advancing to the second round, candidates were allowed to tweak their schemes in a way that would not substantially change them. This option was used by 18 candidates, two of which merged to form a single candidate. We list these and briefly discuss the introduced tweaks.

ACORN: The designers changed the number of times the internal state gets updated in various stages of processing of an encryption (and decryption) query. "The main reason for the change is to increase the steps in the initialization, so as to provide better protection of the secret key when nonce is reused."

AES-JAMBU: No change was made to the mode itself; the authors have added a new recommended instance and changed the precise claims of the security under nonce-reuse (these do not target any well-defined notion).

AES-OTR: A heuristic measure that encodes nonce and tag length into a "nonce" has been introduced to OTR following an extensive discussion about the security of parallel instances of AE schemes with the same key but different tag-lengths.

TABLE 1. Second round CAESAR candidates, their construction principles, functionalities and security goals. This Table draws heavily on the surveys published on AE Zoo [7] website and in a work of Abed et al. [6].

Candidate	Construction and features	Security
ACORN [73]	Stream cipher-based, uses LFSRs. Fully parallelizable, online inverse free.	Nonce-based AE security.
AEGIS [76]	Dedicated primitive, uses AES round function. Parallelizable encryption, online, inverse-free.	Nonce-based AE security.
AES-COPA [10]	Blockcipher-based (uses AES) , derived from the EME [38] mode. Online, parallelizable, fixed AD reuse, incremental AD processing.	Online misuse resistant AE security. Provably secure.
AES-JAMBU [74]	Blockcipher-based (uses AES), a variant of CFB mode. Online, inverse-free	A variant of OAE.
AES-OTR [53]	Blockcipher-based (uses AES). Fully parallelizable, online, inverse-free, fixed AD reuse.	Nonce-based AE security. Provably secure.
AEZ [39]	Blockcipher-based (uses round-reduced AES), derives from OTR and EME modes. Fully parallelizable, inverse-free, fixed AD reuse.	Robust AE security. Provably secure (prove-then prune).
Ascon [30]	Sponge-based. Online, inverse-free.	Nonce-based AE security.
CLOC and SILC [42]	Blockcipher-based (use AES and Twine), derive from CFB mode. Online, inverse-free, fixed AD reuse.	Nonce-based AE security. Provably secure.
Deoxys [44]	Tweakable blockcipher-based, proposes two modes (derive from TAE [50] and CTR). Both fully parallelizable, online.	Nonce-based AE security, MRAE. Provably secure.
ELmD [28]	Blockcipher-based (uses AES), derives from EME. Fully parallelizable, online.	Online misuse-resistant AE security. Provably secure.
HS1-SIV [47]	Stream cipher-based, uses SIV construction and ChaCha stream cipher. Inverse-free, static AD reuse.	MRAE security. Provably secure.
ICEPOLE [54]	Sponge-based. Parallelism supported by mode itself. Online, inverse-free.	Nonce-based AE security. Provably secure.
Joltik [45]	Tweakable blockcipher-based, proposes two modes (derive from TAE and CTR). Both fully parallelizable, online.	Nonce-based AE security, MRAE. Provably secure.
Ketje [20]	Sponge-based, uses Keccak-f. Online, inverse-free.	Nonce-based AE security.

The continuation of Table 1 from the previous page.

Keyak [21]	Sponge-based, uses Keccak0f. Parallelism supported by the mode itself. Online, inverse-free.	Nonce-based AE security. Provably secure.
Minalpher [69]	Permutation-based, uses tweakable Even-Mansour construction. Online.	A weak variant of OAE security. Provably secure.
MORUS [75]	Dedicated primitive. Online, inverse-free.	Nonce-based AE security.
NORX [12]	Sponge-based. Parallelism supported by the mode itself. Online, inverse-free.	Nonce-based AE security. Provably secure.
OCB [48]	Blockcipher-based (uses AES), derives from ECB. Fully parallelizable, online, incremental AD, static AD reuse.	Nonce-based AE security. Provably secure.
OMD [27]	Compression-function based (uses SHA256 and SHA512 comp. functions), derives from Merkle-Damgård construction. Online, inverse-free, incremental AD, static AD reuse.	Nonce-based AE security. Provably secure.

AEZ: Minor modifications have been made to the mode, mainly to prevent an attack by Leurent [49].[3]

CLOC and SILC: Both CLOC and SILC, formerly two separate candidates by the same team, received a minor tweak, namely the used parameters (used blockcipher, nonce length and tag length) and nonce are encoded into the "nonce" as a heuristic measure against attacks with variable tag length.

COLM: COLM resulted from merging of the candidates AES-COPA and ElmD. It combines the features of the former submissions: it is a blockcipher-based, OAE secure AEAD scheme that is based on EME mode (COPA and ElmD) that uses a specific linear mixing function for message processing (ElmD) and xor-mixing for associated data (COPA), uses direct encryption in both layers of EME (COPA) and supports intermediate tags (ElmD).

HS1-SIV: Authors of HS1-SIV replaced the ϵ-AXU hash used for AD-processing by a faster one and introduced a minor tweak to allow static AD reuse.

ICEPOLE: The number of internal iterations of the used permutation was increased in calls made to generate the authentication tag to prevent easy forgeries [29].

[3]This attack did not invalidate the security claims but its seriousness lies in the fact that it is a key-recovery.

TABLE 2. Second round CAESAR candidates, their construction principles, functionalities and security goals. This table draws heavily on the surveys published on AE Zoo [7] website and in a work of Abed et al. [6].

Candidate	Construction and features	Security goals
PAEQ [22]	Permutation-based. Fully parallelizable. Online, inverse-free, incremental AD and message, static AD reuse.	Nonce-based AE security and MRAE. Provably secure.
π-Cipher [35]	Sponge-based. Parallelism supported by the mode itself. Online, inverse-free.	Nonce-based AE security.
POET [4]	Blockcipher-based (uses AES). Pipelineable, online, incremental AD, static AD reuse.	Online misuse-resistant AE security. Provably secure.
PRIMATEs [8]	Sponge-based, several modes using dedicated permutation. All online, two modes inverse-free, one mode incremental AD and static AD reuse.	Nonce-based AE security. Online misuse-resistant and decryption misuse-resistant AE security. Provably secure.
SCREAM [36]	Tweakable blockcipher-based, based on TAE mode. Fully parallelizable, online, inverse-free.	Nonce-based AE security. Provably secure.
SHELL [71]	Blockcipher-based (uses AES), based on EME mode. Online.	Online misuse-resistant security. Provably secure.
STRIBOB [67]	Sponge-based. Online, inverse-free.	Nonce-based AE security. Provably secure.
Tiaoxin [57]	Dedicated primitive, uses AES round function. Fully parallelizable, online, inverse-free.	Nonce-based AE security.
TriviA-ck [25]	Stream cipher-based, based on Trivium. Fully parallelizable, inverse-free.	Nonce-based AE security. Provably secure.

Keyak: The mode of operations proposed by the Keyak team underwent a noticeable change. These were 1) allowing to mix tag production and PT encryption in one call to the underlying permutation to decrease the computational cost of short message-encryptions, and 2) introduction of full-state absorbing to decrease the computational cost of processing long queries, following the result of M e n n i n k et al. [52].

NORX: Only minor changes to the integration of mode's parameters and the parameters of the proposed instances were made in case of NORX.

OMD: A simple change was done in the initialization of the OMD mode to incorporate the tag length in the processing of queries as a heuristic measure against attack with variable tag length.

π-Cipher: Designers of π-Cipher changed an internal padding rule and the number of internal iterations of the underlying permutation. The number of proposed instances was also reduced.

POET: The authors of POET have introduced a number of modifications, such as support of intermediate tags, simplifications of the proposed mode, or security analysis in the RUP model.

SCREAM: One mode (iScream) was removed from the submission, mistakes leading to easy forgeries were fixed and two modifications of the proposed blockcipher were done in case of Scream.

SHELL: The plaintext is now padded before encryption instead of using the method XLS [63] to deal with arbitrary-length plaintexts following an attack on this method [56].

STRIBOB: The designers of STRIBOB made changes to the lower-level primitive.

TriviA-ck: The internal state size of Trivia-ck was reduced and the authors also modified the initialization of the state and a bit ordering convention in a low-level subroutine.

4. CAESAR candidate OMD

Offset Merkle-Damgård (OMD) [26], [59] is one of the second-round CAESAR candidates; it is the only one that is based on a compression function. OMD is a mode of operations for a keyed compression function[4] that derives from Merkle-Damgård construction for hash functions and uses whitening offsets (similarly to many other candidates, e.g., OCB) to achieve nonce-based AE security. There are 9 instances of OMD proposed by its designers. Six instances use compression function of the SHA256 hash function and three instances use the compression function of the SHA512 [3].

OMD is parametrized by a keyed compression function $F : \mathcal{K} \times \left(\{0,1\}^n \times \{0,1\}^m\right) \to \{0,1\}^n$ and a tag length $\tau \leq n$; where the key space $\mathcal{K} = \{0,1\}^k$ and $m \leq n$. To process an encryption query, OMD internally splits the message (and AD) into m-bit (and $(m+n)$-bit) blocks and uses the compression function F and key-dependent Δ-offsets to process the inputs block by block.

[4]In practice, OMD is instantiated with compression functions of standard hash functions (e.g., SHA- or the SHA-2 family). These are keyless, so they must be keyed, e.g., by prepending the secret key to each processed data block.

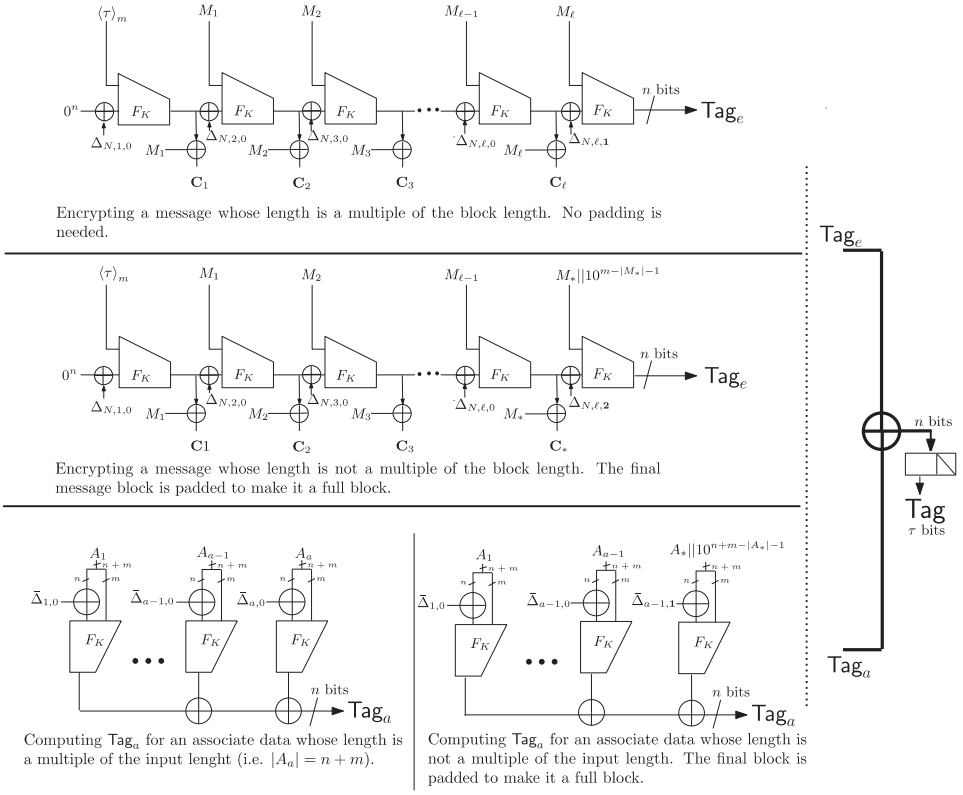

FIGURE 1. The encryption process of OMD using a compression function F and τ-bit tags.
(**TOP**) The encryption process when the message length is a multiple of the block length m and no padding is required. (**Middle**) The encryption process when the message length is not a multiple of the block length and the final block M_* is padded to make a full block $M_*||10^{m-|M_*|-1}$. (**Bottom, Left**) Computing the intermediate value T_a when the bit length of the associated data is a multiple of the **input** length $n + m$. (**Bottom, Right**) Computing T_a when the bit length of the associated data is not a multiple of $n + m$ and the final block is padded to make a full block $A_*||10^{n+m-|A_*|-1}$ is needed. The output ciphertext is $C||\mathsf{Tag}$. Here, xor of two strings of unequal length X, Y is defined as $\mathsf{left}_s(X) \oplus \mathsf{right}_s(Y)$ with $s = \min(|X|, |Y|)$.

The whitening offsets, denoted $\Delta_{N,i,j}$ and $\bar{\Delta}_{i,j}$ are computed as a function of the secret key, the nonce, the number of the current call to F made inside a query and a domain-separation constant. A visualization of the encryption algorithm is given in Figure 1.

Security

OMD is provably secure in the sense of nonce-based AEAD; the results concerning its security are summarized in Theorem 1. When we apply the security bounds from Theorem 1 to the proposed instances of OMD, we see that thanks to the wide output block of SHA256 and SHA512 compression functions, OMD offers higher quantitative security levels than most of the candidates. Especially the instances that use the compression function of SHA-512 benefit from this; their provable security guarantee for message confidentiality holds up to 2^{255} adversarial queries. As a purely nonce-based AEAD scheme, OMD does not offer any security guarantees in the case that nonces get reused.

Theorem 1. *Fix $n \geq 1$ and $\tau \in \{0, 1, \ldots, n\}$. Let $F\colon \mathcal{K} \times \left(\{0,1\}^n \times \{0,1\}^m\right) \to \{0,1\}^n$ be a PRF, where the key space $\mathcal{K} = \{0,1\}^k$ for $k \geq 1$ and $1 \leq m \leq n$. Then*

$$\begin{aligned}
\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{OMD}[F,\tau]}(t, q_e, \sigma_e, \ell_{\max}) &\leq \mathbf{Adv}^{\mathrm{prf}}_{F}(t', 2\sigma_e) + \frac{3\sigma_e^2}{2^n} \\
\mathbf{Adv}^{\mathrm{auth}}_{\mathrm{OMD}[F,\tau]}(t, q_e, q_v, \sigma, \ell_{\max}) &\leq \mathbf{Adv}^{\mathrm{prf}}_{F}(t', 2\sigma) + \frac{3\sigma^2}{2^n} + \frac{q_v \ell_{\max}}{2^n} + \frac{q_v}{2^\tau}
\end{aligned}$$

where q_e and q_v are, respectively, the number of encryption and decryption queries, $\ell_{\max}$ denotes the maximum number of m-bit blocks in an encryption or decryption query, $t' = t + cn\sigma$ for some constant c, and σ_e and σ are the total number of calls to the underlying compression function F in all queries asked by the CPA and CCA adversaries against the privacy and authenticity of the scheme, respectively.

Features and variants of OMD

One of the main motivations for the design of OMD was cryptographic diversity. Following the proverb "do not put all your eggs into one basket", the designers of OMD extended the spectrum of lower-level primitives that can be used to create secure AE schemes to keyed compression functions. Thanks to the structure inherited from Merkle-Damgård construction, OMD can process a message efficiently, in a single pass that integrates computation of the ciphertext and the authentication tag in an online fashion. The same however prevents any parallelization of the OMD mode. OMD also benefits from static AD reuse.

Two variants of OMD have been proposed, demonstrating the potential of compression-function based AE. These variants are not part of the CAESAR submission. The first variant is a nonce misuse-resistant variant of OMD, (for short MR-OMD) [60]. This combines the chained message encryption of OMD with a dedicated, highly efficient PRF in an SIV-like construction. MR-OMD requires a single secret key and achieves the MRAE security.

The second variant is the pure OMD (pOMD for short) [61]. In this variant, processing of AD was integrated into the main Merkle-Damgård-like processing chain, enabling one to process AD almost free of charge, processing the encryption queries up to 50 % faster than OMD. The core idea of pOMD is to xor the blocks of AD to secret state before each call to the compression function. This motivated the research that proved security of a similar measure for sponge--based AE schemes [52] and was applied to Keyak.

5. The state of authenticated encryption

In the previous sections, we have reviewed the CAESAR competition and its candidates, as a major scientific effort in the field of AE. We step back and consider authenticated encryption from a wider perspective. We make a few observations about the state and recent development of AE from different perspectives and finally try to foresee possible research directions in the future.

AE as a primitive

Authenticated encryption was identified as a primitive in its own right a mere decade and a half ago. Since then, the general understanding about AE has not changed; AE is a primitive that guarantees confidentiality for a plaintext and authenticity for the plaintext and some additional, unencrypted data. The understanding of what *exactly* should AE be as a primitive—what interface should it offer and what *exactly* should it do—evolved nevertheless. Apart from the most frequently followed syntax for nonce-based AEAD schemes (q.v. Section 2), other approaches to defining the syntax and functionality of AE schemes appeared.

In their paper from 2012, F l e i s c h m a n n et al. [34] proposed syntax for deterministic online AEAD schemes that process header-message pairs, but have no explicit IV-input. Instead they use an implicit IV by imposing non-empty associated data headers. Their approach was followed by several AE schemes (e.g., COPA [11], ElmD and POET) and paraphrased in several talks [23], [37]. H o a n g et al. [41] pointed out that this change in AE interface leads to an unfortunate misjudgement of security guarantees that can be expected when using an arbitrary, or no IV.

The syntax for AE schemes proposed in the CAESAR's call for submissions deviates from the AEAD syntax as well, as pointed out in Section 3. In this case, a new, nonce-like input was added to the interface. Since this interface was proposed only recently, it is not yet clear whether the SMN is really a relevant addition to the syntax of AE schemes. The decision not to support SMN made by the majority of the CAESAR candidates would suggest otherwise.

Most of the formal definitions of AE as a primitive assume a deterministic and stateless encryption algorithm, however stateful variants of AE with associated data were proposed recently. These can be seen as stateful extensions of the classical AEAD as this approach builds on an API-based syntax that allows a scheme to be used in the traditional stateless, nonce-based AEAD sense, but supports "sessions" for encrypting message-AD pairs in a stateful way. The candidates Keyak and Ketje implement this type of syntax and functionality. Similar approach was proposed by Hoang et al. [41] in their notion of online AE.

The prevailing approach to defining the syntax and functionality of AE schemes has been, for over a decade, the nonce-based AEAD as defined by Rogaway. Yet, the stateful extension of this approach is likely to gain more popularity, as it can be mapped very well to many real-world communication scenarios.

Security goals of AE

The general security goals of AE, i.e., to ensure confidentiality and authenticity, were quickly captured in the early security notions [16], [46], [64]. The most frequently followed formalization among these is the security definition for nonce-based AEAD by Rogaway (q.v. Section 2). The research of security guarantees that can be expected from AE schemes did not stop there. On the contrary, a number of works studied security of AE in various adversarial situations.

A big part of these works has focused on extending the security models for AE to cover non-standard situations that occur if an AE scheme is used in a way that it is normally not supposed to, or else when the scheme is *misused*. The reuse of nonces with the same key is such a misuse—it is not covered by the classical nonce-based AE notion, yet can occur in real life. The notion of MRAE was the first to define the desirable behaviour of AE schemes under nonce reuse, and the notions of OAE [34] and OAE2 [41] address the same for online AE schemes.

The release of unverified plaintext is another type of misuse that can occur in practice, as many schemes decrypt a ciphertext to a putative plaintext before the final authentication check. This type of misuse was treated with a security definition by Andreeva et al. [9]. The notion of Robust AE takes a different approach and formalizes a very strong security goals in a way that implies security in case of nonce-misuse, and decryption misuse, but also if different amounts of ciphertext expansion (or different tag lengths) are used with the same key. This new type of misuse has been identified by the community only recently. Some of the second-round candidates have already included heuristic measures against it and the first formal treatment for nonce-based AE schemes was proposed by Reyhanitabar et al. [62].

Another line of work focused on analysing and defining the AE security in the context of streaming channels [33] or in the presence of ciphertext fragmentation [24], a usage scenario and a communication protocols' artifact that frequently appears in practice.

What is common to all the mentioned, advanced security notions for AE is that they strive to extend the understanding of AE security to practically relevant usage scenarios, motivated by practical issues that arise in real-world applications. Although the portfolio of AE security definitions is already rich in strong AE notions, especially after the definition of RAE security, there are still problems to be addressed. RAE security implies some structural and performance constraints which prevent the use of RAE schemes in some applications. Some applications may require robustness to certain types of misuse only, which in turn implies the need for specialized AE notions. New types of "misuse" may also be discovered, as demonstrated by recognition of the problems that appear with the tag-length misuse.

AE schemes

Thanks to the CAESAR competition, the portfolio of existing AE schemes has been considerably extended. We now have very efficient nonce-based AE schemes constructed from every kind of symmetric-key primitive: blockciphers, permutations, stream ciphers and compression functions. For some types of constructions, techniques that allow to greatly increase efficiency without compromising security were discovered and analysed, e.g., the full-state absorption for sponge-based schemes [52]. A new approach to constructing AE schemes also appeared; some CAESAR candidates are directly constructed as an efficient primitive from low-level building blocks rather than a mode of operations (e.g., AEGIS). A number of new results that show how to efficiently achieve strong security goals (under various types of misuse) have been presented as well (e.g., AEZ).

The research induced by the competition has identified new, desirable features of AE schemes (such as static AD reuse or incremental AD computation, refer to Section 3) as well as the ways to achieve them. CAESAR also stimulated research of lower-level primitives themselves. A number of candidates uses a dedicated lower-level building block, e.g., the PRIMATE permutation used in PRIMATEs' modes of operation, or the tweakable blockciphers of Deoxys and Joltik.

Future of AE

In the area of AE, the theoretical research has been in close contact with the practical world and its needs. The theoretical research problems are induced by real-world issues related to the use of AE schemes and the new AE schemes are being designed to be efficient and applicable in the practical protocols and systems.

It is therefore safe enough to assume, that the theoretical research in AE will remain active in the coming years. Beside the fact that some security notions are not applicable to all the AE schemes and all the use cases (e.g., RAE security and online AE schemes), new challenges, be it in form of new types of "misuse", or constraints imposed by concrete usage scenarios, are bound to emerge in practice

and new, specific security notions will be required to address them. For example, the security of stateful authenticated encryption schemes under various types of misuse may need to be investigated.

Challenges also remain in the direction of AE schemes construction. Constructing efficient AE schemes secure beyond-birthday-bound, constructing AE schemes that are highly optimized for very specific usage scenarios, designing efficient modes of operation of various primitives, or constructing AE schemes that retain security in presence of newly discovered types of misuse are a few examples of possible research directions.

REFERENCES

[1] *Crypto-competitions google group*, https://groups.google.com/forum/#!topic/crypto-competitions/upaRX2jdVCQ

[2] *Cryptographic competitions: CAESAR submissions*, http://competitions.cr.yp.to/caesar-submissions.html

[3] *Secure hash standard,* http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf

[4] ABED, F.—FLUHRER, S.—FOLEY, J.—FORLER, C.—LIST, E.—LUCKS, S.––MCGREW, D.—WENZEL, J.: *Poet,* https://competitions.cr.yp.to/round2/poetv20.pdf

[5] ABED, F.—FORLER, C.—LIST, E.—LUCKS, S.—WENZEL, J.: *Don't panic! The cryptographer's guide to robust (on-line) encryption: draft*, https://www.uni-weimar.de/fileadmin/user/fak/medien/professuren/Mediensicherheit/Research/Drafts/nonce-misuse-oae.pdf

[6] ABED, F.—FORLER, C.—LUCKS, S.: *General overview of the authenticated schemes for the first round of the CAESAR competition*, IACR Cryptology ePrint Archive 2014, http://eprint.iacr.org/2014/792

[7] ABED, F.—KÖLBL, S.—LAURIDSEN, M. M.—RECHBERGER, C.—TIESSEN, T.: *Authenticated encryption Zoo*, https://aezoo.compute.dtu.dk/

[8] ANDREEVA, E.—BILGIN, B.—BOGDANOV, A.—LUYKX, A.—MENDEL, F.––MENNINK, B.—MOUHA, N.—WANG, Q.—YASUDA, K.: *Primates,* https://competitions.cr.yp.to/round2/primatesv102.pdf

[9] ANDREEVA, E.—BOGDANOV, A.—LUYKX, A.—MENNINK, B.—MOUHA, N.––YASUDA, K.: *How to securely release unverified Plaintext in authenticated encryption*, in: Advances in Cryptology—ASIACRYPT '14 (P. Sarkar, T. Iwata, eds.), 20th Internat. Conf. on the Theory and Appl. of Cryptology and Inform. Security Kaoshiung, Taiwan, 2014, Lecture Notes in Comput. Sci., Vol. 8873, Springer, Berlin, 2014, pp. 105–125.

[10] ANDREEVA, E.–BOGDANOV, A.–LUYKX, A.–MENNINK, B.–TISCHHAUSER, E.––YASUDA, K.: *Aes-copa.* https://competitions.cr.yp.to/round2/aescopav2.pdf

[11] ANDREEVA, E.–BOGDANOV, A.–LUYKX, A.–MENNINK, B.–TISCHHAUSER, E.––YASUDA, K.: *Parallelizable and authenticated online ciphers.* in: Advances in Cryptology—ASIACRYPT '13, 19th Internat. Conf. on the Theory and Appl. of Cryptology and Inform. Security, Bengaluru, India, 2013, Lecture Notes in Comput. Sci., Vol. 8269, Springer, Berlin, 2013, pp. 424–443.

[12] AUMASSON, J. P.—JOVANOVIC, P.—NEVES, S.: *Norx,* https://competitions.cr.yp.to/round2/norxv20.pdf

[13] BELLARE, M.—DESAI, A.—JOKIPII, E.—ROGAWAY, P.: *A concrete security treatment of symmetric encryption*, in: 54th Annual Symp. on Found. of Comput. Sci.––FOCS '97, Miami Beach, FL, 1997, IEEE Comput. Soc., 1997, pp. 394–403.

[14] BELLARE, M.—KILIAN, J.—ROGAWAY, P.: *The security of the cipher block chaining message authentication code,* J. Comput. Syst. Sci. **61** (2000), 362–399.

[15] BELLARE, M.—NAMPREMPRE, C.: *Authenticated encryption: relations among notions and analysis of the generic composition paradigm*, in: Advances in Cryptology––ASIACRYPT '00 (T. Okamoto, ed.), 6th Internat. Conf. on the Theory and Appl. of Cryptology and Inform. Security, Kyoto, Japan, Lecture Notes in Comput. Sci., Vol. 1976, Springer, Berlin, 2000, pp. 531–545.

[16] BELLARE, M.—ROGAWAY, P.: *Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient cryptography*, in: Advances in Cryptology—ASIACRYPT '00 (T. Okamoto, ed.), 6th Internat. Conf. on the Theory and Appl. of Cryptology and Inform. Security, Kyoto, Japan, Lecture Notes in Comput. Sci., Vol. 1976, Springer, Berlin, 2000, pp. 317–330.

[17] BERNSTEIN, D. J.: *Cryptographic competitions: CAESAR*, http://competitions.cr.yp.to

[18] BERNSTEIN, D. J.: *Cryptographic competitions: Disasters*, https://competitions.cr.yp.to/disasters.html

[19] BERNSTEIN, D. J.: *Cryptographic competitions: Features of various secret-key primitives,* https://competitions.cr.yp.to/features.html

[20] BERTONI, G.—DAEMEN, J.—PEETERS, M.—ASSCHE, G. V.—KEER, R. V.: *Ketje,* https://competitions.cr.yp.to/round1/ketjev11.pdf

[21] BERTONI, G.—DAEMEN, J.—PEETERS, M.—ASSCHE, G. V.—KEER, R. V.: *Keyak,* https://competitions.cr.yp.to/round2/keyakv2.pdf

[22] BIRYUKOV, A.—KHOVRATOVICH, D.: *Paeq,* https://competitions.cr.yp.to/round1/paeqv1.pdf

[23] BOGDANOV, A.—LAURIDSEN, M. M.—TISCHHAUSER, E.: *Aes-based authenticated encryption modes in parallel high-performance software,* DIAC presentation, 2014.

[24] BOLDYREVA, A.—DEGABRIELE, J. P.—PATERSON, K. G.—STAM, M.: *Security of symmetric encryption in the presence of ciphertext fragmentation,* in: Advances in Cryptology—EUROCRYPT '12, 31st Annual Internat. Conf. on the Theory and Appl. of Cryptographic Techniques, Cambridge, UK, 2012, Lecture Notes in Comput. Sci., Vol. 7237, Springer, Berlin, 2012, pp. 682–699.

[25] CHAKRABORTI, A.—NANDI, M.: *Trivia-ck,* https://competitions.cr.yp.to/round2/triviackv2.pdf

[26] COGLIANI, S.—MAIMUT, D.—NACCACHE, D.—DO CANTO, R. P.—REYHANITABAR, R.—VAUDENAY, S.—VIZÁR, D.: *OMD: a compression function mode of operation for authenticated encryption*, in: Selected Areas in Cryptography—SAC '14, 21st Internat. Conf., Montreal, QC, Canada, 2014 (A. Joux, A. Youssef, eds.), Lecture Notes in Comput. Sci., Vol. 8781, Springer, Berlin, 2014, pp. 112–128.

[27] COGLIANI, S.—ŞTEFANIA MAIMUŢ, D.—NACCACHE, D.—DO CANTO, R. P.––REYHANITABAR, R.—VAUDENAY, S.—VIZÁR, D.: *Offset Merkle-Damgård,* https://competitions.cr.yp.to/round2/omdv20.pdf

[28] DATTA, N.—NANDI, M.: *Elmd,* https://competitions.cr.yp.to/round2/elmdv20.pdf

[29] DOBRAUNIG, C.—EICHLSEDER, M.—MENDEL, F.: *Forgery attacks on round-reduced icepole-128,* Cryptology ePrint Archive, Report 2015/392, http://eprint.iacr.org/

[30] DOBRAUNIG, C.—EICHLSEDER, M.—MENDEL, F.—SCHLAFFER, M.: *Ascon,* https://competitions.cr.yp.to/round2/asconv11.pdf

[31] DWORKIN, M.: *Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality*. NIST Special Publication 800-38C, Gaithersburg, 2004.

[32] FERGUSON, N.: *Authentication weaknesses in gcm,* Comments submitted to NIST Modes of Operation Process, 2005.

[33] FISCHLIN, M.—GÜNTHER, F.—MARSON, G. A.—PATERSON, K. G.: *Data is a stream: Security of stream-based channels,* in: Advances in Cryptology—CRYPTO '15 (R. Gennaro, M. Robshaw, eds.), 35th Annual Cryptology Conf., Santa Barbara, CA, 2015, Lecture Notes in Comput. Sci., Vol. 9216, Springer, Berlin, 2015, pp. 545–564.

[34] FLEISCHMANN, E.—FORLER, C.—LUCKS, S.: *McOE: a family of almost foolproof on-line authenticated encryption schemes*, in: Fast Software Encryption—FSE '12, 19th Internat. Workshop, Washington, DC, USA (A. Canteaut, ed.), Lecture Notes in Comput. Sci., Vol. 7549, Springer, Berlin, 2012, pp. 196–215.

[35] GLIGOROSKI, D. – MIHAJLOSKA, H. – SAMARDJISKA, S. – JACOBSEN, H. – EL-HADEDY, M. – JENSEN, R. E. – OTTE, D.: *π-cipher,* https://competitions.cr.yp.to/round2/picipherv20.pdf

[36] GROSSO, V. – LEURENT, G. – STANDAERT, F. X. – VARICI, K. – JOURNAULT, A. – – DURVAUX, F. – GASPAR, L. – KERCKHOF, S.: *Scream,* https://competitions.cr.yp.to/round2/screamv3.pdf

[37] GUO, J.: *Marble specification version 1.0.,* DIAC presentation, 2014.

[38] HALEVI, S.—ROGAWAY, P.: *A parallelizable enciphering mode,* in: Topics in Cryptology—CT-RSA '04 (T. Okamoto, ed.), The Cryptographers' Track at the RSA Conf., San Francisco, CA, USA, 2004, Lecture Notes in Comput. Sci., Vol. 2964, Springer, Berlin, 2004, pp. 292–304.

[39] HOANG, V. T.—KROVETZ, T.—ROGAWAY, P.: *Aez,* https://competitions.cr.yp.to/round2/aezv4.pdf

[40] HOANG, V. T.—KROVETZ, T.—ROGAWAY, P.: *Robust authenticated-encryption AEZ and the problem that it solves*, in: Advances in Cryptology—EUROCRYPT '15 (E. Oswald et al., eds.), 34th Ann. Internat. Conf. on the Theory and Appl. of Cryptographic Tech., Sofia, Bulgaria, 2015, Lecture Notes in Comput. Sci., Vol. 9056, Springer, Berlin, 2015, pp. 15–44.

[41] HOANG, V. T.—REYHANITABAR, R.—ROGAWAY, P.—VIZÁR, D.: *Online authenticated-encryption and its nonce-reuse misuse-resistance*, in: Advances in Cryptology– –CRYPTO '15 (R. Gennaro, M. Robshaw, eds.), 35th Ann. Cryptology Conf., Santa Barbara, CA, USA, 2015, Lecture Notes in Comput. Sci., Vol. 9215, Springer, Berlin, 2015, pp. 493–517.

[42] IWATA, T.—MINEMATSU, K.—GUO, J.—MORIOKA, S.—KOBAYASHI, E.: *Cloc and silc,* https://competitions.cr.yp.to/round2/silcv2.pdf

[43] JEAN, J.—NIKOLIC, I.—PEYRIN, T.: *Tweaks and keys for block ciphers: The TWEAKEY framework,* in: Advances in Cryptology—ASIACRYPT '14 (P. Sarkar et al., eds.), 20th Internat. Conf. on the Theory and Appl. of Cryptology and Inform. Security, Kaoshiung, Taiwan, R.O.C., 2014, Lecture Notes in Comput. Sci., Vol. 8874, Springer, Berlin, 2014, pp. 274–288.

[44] JEAN, J.—NIKOLIĆ, I.—PEYRIN, T.: *Deoxys,* https://competitions.cr.yp.to/round2/deoxysv13.pdf

[45] JEAN, J.—NIKOLIĆ, I.—PEYRIN, T.: *Joltik,* https://competitions.cr.yp.to/round2/joltikv13.pdf

[46] KATZ, J.—YUNG, M.: *Unforgeable encryption and chosen ciphertext secure modes of operation*, in: Fast Software Encryption—FSE '00 (Schneier, B. ed.), 7th Internat. Workshop—FSE '00, New York, NY, USA, 2000, Lecture Notes in Comput. Sci., Vol. 1978, Springer, Berlin, 2001, pp. 284–299.

[47] KROVETZ, T.: *Hs1-siv,* `https://competitions.cr.yp.to/round2/hs1sivv2.pdf`

[48] KROVETZ, T.—ROGAWAY, P.: *Ocb,* `https://competitions.cr.yp.to/round1/ocbv1.pdf`

[49] LEURENT, G.: *Aez bbb,* Rump session talk at Eurocrypt '15.

[50] LISKOV, M.—RIVEST, R. L.—WAGNER, D.: *Tweakable block ciphers,* in: Advances in Cryptology—CRYPTO '02 (M. Yung, ed.), 22nd Ann. Internat. Cryptology Conf., Santa Barbara, CA, USA, 2002, Lecture Notes in Comput. Sci., Vol. 2442, Springer, Berlin, 2002, pp. 31–46.

[51] MCGREW, D. A.—VIEGA, J.: *The security and performance of the galois/counter mode (GCM) of operation,* in: Progress in Cryptology—INDOCRYPT '04 (A. Canteaut et al., eds.), 5th Internat. Conf. on Cryptology in India, Chennai, India, 2004, Lecture Notes in Comput. Sci., Vol. 3348, Springer, Berlin, 2004, pp. 343–355.

[52] MENNINK, B.—REYHANITABAR, R.—VIZÁR, D.: *Security of full-state keyed Sponge and Duplex: applications to authenticated encryption,* in: Adv. in Cryptology—ASIACRYPT '15 (T. Iwata et al., eds.), 21st Internat. Conf. on the Theory and Appl. of Cryptology and Inform. Security, Auckland, New Zealand, 2015, Lecture Notes in Comput. Sci., Vol. 9453, Springer, Berlin, 2015, pp. 465–489.

[53] MINEMATSU, K.: *Aes-otr,* `https://competitions.cr.yp.to/round2/aesotrv2.pdf`

[54] MORAWIECKI, P.—GAJ, K.—HOMSIRIKAMOL, E.—MATUSIEWICZ, K.—PIEPRZYK, J.—ROGAWSKI, M.—SREBRNY, M.—WÓJCIK, M.: *Icepole,* `https://competitions.cr.yp.to/round2/icepolev2.pdf`

[55] NAMPREMPRE, C.—ROGAWAY, P.—SHRIMPTON, T.: *AE5 security notions: definitions implicit in the CAESAR call,* IACR Cryptology ePrint Archive, 2013, 242.

[56] NANDI, M.: *On the minimum number of multiplications necessary for universal hash functions,* in: Fast Software Encryption—FSE '14, 21st Internat. Workshop, London, UK, 2014, Lecture Notes in Comput. Sci., Vol. 8540, Springer, Berlin, 2015, pp. 489–508.

[57] NIKOLIĆ, I.: *Tiaoxin,* `https://competitions.cr.yp.to/round2/tiaoxinv2.pdf`

[58] NIWA, Y.—OHASHI, K.—MINEMATSU, K.—IWATA, T.: *GCM security bounds reconsidered.* in: Fast Software Encryption—FSE '15 (G. Leander, G. ed.), 22nd Internat. Workshop, Istanbul, Turkey, 2015, Lecture Notes in Comput. Sci., Vol. 9054, Springer, Berlin, 2015, pp. 385–407.

[59] REYHANITABAR, R.: *OMD version 2: a tweak for the 2nd round*, `crypto-competitions` mailing list, August 27, 2015.

[60] REYHANITABAR, R.—VAUDENAY, S.—VIZÁR, D.: *Misuse-resistant variants of the OMD authenticated encryption mode*, in: Provable Security—ProvSec '14 (S.S.M. Chow et al., eds.), 8th Internat. Conf., Hong Kong, China, 2014, Lecture Notes in Comput. Sci., Vol. 8782, Springer, Berlin, 2014, pp. 55–70.

[61] REYHANITABAR, R.—VAUDENAY, S.—VIZÁR, D.: *Boosting OMD for almost free authentication of associated data*, in: Fast Software Encryption—FSE '15 (G. Leander, ed.), 22nd Internat. Workshop, Istanbul, Turkey, 2015, Lecture Notes in Comput. Sci., Vol. 9054, Springer, Berlin, 2015, pp. 411–427.

[62] REYHANITABAR, R.—VAUDENAY, S.—VIZÁR, D.: *Authenticated encryption with variable stretch,* Cryptology ePrint Archive, Report 2016/463, `http://eprint.iacr.org/`

[63] RISTENPART, T.—ROGAWAY, P.: *How to enrich the message space of a cipher,* in: Fast Software Encryption—FSE '07, 14th Internat. Workshop, Luxembourg, 2007, Lecture Notes in Comput. Sci., Vol. 4593, Springer, Berlin, 2007, pp. 101–118.

[64] ROGAWAY, P.: *Authenticated-encryption with associated-data*, in: Proc. of the 9th ACM Conf. on Computer and Comm. Security ACM—CCS '02, Washington, DC, USA, 2002, ACM New York, NY, USA, 2002, pp. 98–107.

[65] ROGAWAY, P.—BELLARE, M.—BLACK, J.—KROVETZ, T.: *OCB: A block-cipher mode of operation for efficient authenticated encryption*, in: Proc. of the 8th ACM Conf. on Computer and Comm. Security ACM—CCS '01, ACM New York, NY, USA, 2001, pp. 196–205.

[66] ROGAWAY, P.—SHRIMPTON, T.: *A provable-security treatment of the key-wrap problem*, in: Advances in Cryptology—EUROCRYPT '06 (S. Vaudenay, ed.), 25th Ann. Internat. Conf. on the Theory and Appl. of Cryptographic Tech., St. Petersburg, Russia, 2006, Lecture Notes in Comput. Sci., Vol. 4004, Springer, Berlin, 2006, pp. 373–390.

[67] SAARINEN, M. J. O.—BRUMLEY, B. B.: *Stribob,* `https://competitions.cr.yp.to/round2/stribobr2.pdf`

[68] SAARINEN, M. O.: *Cycling attacks on gcm, GHASH and other polynomial macs and hashes,* in: Fast Software Encryption—FSE '12 (A. Canteau, ed.), 19th Internat. Workshop, Washington, DC, USA, 2012, Lecture Notes in Comput. Sci., Vol. 7549, Springer, Berlin, 2012, pp. 216–225.

[69] SASAKI, Y. – TODO, Y. – AOKI, K. – NAITO, Y. – SUGAWARA, T. – MURAKAMI, Y.– –MATSUI, M. – HIROSE, S.: *Minalpher,* `https://competitions.cr.yp.to/round2/minalpherv11.pdf`

[70] VAUDENAY, S.: *Security flaws induced by CBC padding – applications to SSL, IPSEC, WTLS ...* in: Advances in Cryptology—EUROCRYPT '02 (L. R. Knudsen, ed.), 21st Internat. Conf. on the Theory and Appl. of Cryptographic Tech., Amsterdam, Netherlands, 2002, Lecture Notes in Comput. Sci., Vol. 2332, Springer, Berlin, 2002, pp. 534–546.

[71] WANG, L.: *Shell,* `https://competitions.cr.yp.to/round2/shellv20.pdf`

[72] WHITING, D.—HOUSLEY, R.—FERGUSON, N.: *Counter with CBC-MAC (CCM).* IETF RFC 3610 (Inform.), Sep. 2003, `http://www.ietf.org/rfc/rfc3610.txt`

[73] WU, H.: *Acorn,* `https://competitions.cr.yp.to/round2/acornv2.pdf`

[74] WU, H.—HUANG, T.: *Aes-jambu,* `https://competitions.cr.yp.to/round2/aesjambuv2.pdf`

[75] WU, H.—HUANG, T.: *Morus,* `https://competitions.cr.yp.to/round2/morusv11.pdf`

[76] WU, H.—PRENEEL, B.: *Aegis,* `https://competitions.cr.yp.to/round1/aegisv1.pdf`

Received September 19, 2016

EPFL-IC – LASEC
Station 14 – INF 240
CH–1015 Lausanne
Lausanne
SWITZERLAND
E-mail: damian.vizar@epfl.ch