Classification of the CAESAR Candidates

October 4, 2014

Farzaneh Abed, Christian Forler, and Stefan Lucks

Bauhaus-Universität Weimar
`<firstname>.<lastname>@uni-weimar.de`

Abstract. In this work we give an overview of the candidates submitted to the CAESAR competition which are not withdrawn yet. Furthermore, we propose a classification with regard to their core primitives that includes several design characteristics.

Keywords: authenticated encryption, CAESAR competition, classification.

1 Introduction

Two important goals of cryptography are to guarantee the privacy and authenticity of messages that shall be submitted over an insecure channel. Authenticated encryption (AE) schemes are key-based transformations that aim at providing both goals [13,14] simultaneously. Many settings require to also authenticate associated public data, Rogaway introduced the notion of authenticated encryption with associated data (AEAD) [64].

Until around 2000, approaches to designing AE schemes regarded authentication and encryption as two independent cryptographic building blocks. *Generic composition* schemes simply combined an IND-CPA-secure (indistinguishable against chosen-plaintext attacks) cipher and a SUF-CMA-secure (unforgeable against chosen-message attacks) MAC. Bellare and Namprempre [13,14] showed that, among the available constructions Encrypt-and-MAC, MAC-then-Encrypt, and Encrypt-then-MAC, only the latter inherits sufficient security in general. However, while generic composition allows that both individual components can be analyzed and exchanged easily, it always suffered from being neither very efficient nor very robust to implementation errors (see, e.g., [27]).

The work by Bellare and Namprempre [13] renewed our way of concerning AE. The authors defined the notion for integrity of ciphertext (INT-CTXT) and showed that schemes which offer both INT-CTXT- and IND-CPA-security implicitly also provide security against chosen-ciphertext attacks. Most importantly, their work lead to the understanding that modern authenticated encryption should be seen as a cryptographic building block on its own rather than as the mere combination of two. Hence, the modern approach to design AE schemes is to construct dedicated schemes.

The CAESAR Competition. The cryptographic community tremendously increased its understanding of block ciphers and hash functions from the public competitions for the AES and SHA-3 standards. The CAESAR (Competition for Authenticated Encryption: Security, Applicability, and Robustness) contest aims at a somewhat similar goal for the field of AE. At March 2014, it called for submissions of novel AE designs that should offer advantages over the widely used modes AES-GCM [55] and AES-OCB3 [49].

Contribution. This paper tries to provide a comprehensible overview over the first-round CAESAR candidates, and is influenced by the summary that was compiled by Bart Preneel at the Dagstuhl seminar 14021 [9], and the Authenticated Encryption Zoo hosted by Stefan Kölbl et al. [47]. Note that we consider only recommended parameter sets for unbroken candidates. We propose an intuitive categorization according to the underlying major cryptographic primitive. We recall the desired functional, security, robustness, and performance criteria and provide easy-to-grasp comparisons of the submissions according to each of these. We consider design and security parameters, attacks, and existing cryptanalysis for each candidate.

Disclaimer. While we try best in correctly understanding and categorizing the CAESAR submissions, we may misinterpret or simply oversee design features of some scheme. Moreover, the submissions are subject to changes by their respective designers, within or beyond the scope of the CAESAR competition. We try to keep this document up-to-date during the competition. In case you spot mistakes, please write us an email and we will update this document if necessary. We include all schemes that have not been withdrawn from the competition at the time of writing this document, at the time, we exclude AES-COBRA [6], CBEAM [68], FASER [23], HKC [38], McMambo [51], PAES [79], and PANDA [80].

Outline. The remainder of this paper is organized as follows: Section 2 lists the functional characteristics of authenticated encryption schemes. Section 3 recalls the relevant security and robustness notions and criteria. The schemes are compared in a table at the end of each section.

2 Design Classification

Authenticated Encryption Scheme. Let $k, \nu, t \geq 1$, $K \in \{0,1\}^k$ denote a secret key, $N \in \{0,1\}^\nu$ a nonce, $H \in \{0,1\}^*$ a header (equivalent to associated data), $M \in \{0,1\}^*$ a message, $T \in \{0,1\}^t$ an authentication tag, and $C \in \{0,1\}^*$ a ciphertext. An authenticated encryption scheme with associated data is a triple $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, with a key-generation procedure $\mathcal{K}$ that returns a randomly chosen key K, a deterministic encryption algorithm $\mathcal{E}_\mathcal{K}(N, H, M)$, and its inverse decryption algorithm $\mathcal{D}_\mathcal{K}(N, H, C, T)$. $\mathcal{E}$ always outputs a ciphertext-tag pair (C, T), and $\mathcal{D}$ outputs either the plaintext M that corresponds to C, or the bot symbol $\perp$ if the tag is invalid:

$$\mathcal{E} : \{0,1\}^k \times \{0,1\}^\nu \times \{0,1\}^* \times \{0,1\}^* \rightarrow \times \{0,1\}^* \times \{0,1\}^t$$
$$\mathcal{D} : \{0,1\}^k \times \{0,1\}^\nu \times \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^t \rightarrow \times \{0,1\}^* \cup \{\perp\}.$$

We use these notions in the remainder of this paper. Note that the CAESAR call for submissions demanded a slightly different API, where the nonce is split into a public and a secret message number (PNM, SNM).

2.1 Underlying Constructions, Modes, Primitives

It is natural to classify AE schemes by their underlying construction. While the most intuitive way to construct an AE scheme is to design a block-cipher- based mode of operation, the heterogeneous CAESAR portfolio includes a variety of different designs, including stream-cipher-, hash-/compression- function- and permutation-based constructions, as well as combinations thereof. This section briefly recalls the primitives and kinds that appear as base of the CAESAR submissions.

Block Cipher. A block cipher is a keyed family of fixed-length permutations that allows encryption of fixed-length message blocks with the help of a secret key that is shared between sender and receiver. For block-cipher-based candidates, we state which encryption mode(s) they inherit from, if any. We see the following modes adopted by the CAESAR submissions, and use the following acronyms:

CFB	Ciphertext feedback mode [61].
CTR	Counter mode [61].
ECB	Electronic codebook mode [61].
EME	Encrypt-Mix-Encrypt mode [36,35].
LEX	Leakage extraction mode [19].
OFB	Output-feedback mode [61].
OTR	Two-branch two-round Feistel [57].
PFB	Plaintext feedback mode [83].
TAE	Tweakable authenticated encryption [52,53].
TC3	Tweakable-Cipher-3 [67].
XEX	XOR-encrypt-XOR (Evan-Mansour) [65].

Masking. Most modern block-cipher-based schemes mask the in- and outputs to the block cipher to prevent them from being under control of adversaries. We point out the following approaches:

AX	Addition and XOR.
Doubling	XOR with a key-dependent variable that is incremented by doubling it in Galois Field [65].
GFM	Multiplication with a key-dependent variable in Galois-Field.

Stream Cipher. A stream cipher is a symmetric pseudo-random bit generator (PRBG) that takes a fixed-length secret key and generates a keystream of variable length. Like block ciphers, stream ciphers can be used as a core primitive in authenticated encryption scheme to achieve confidentiality and integrity as long as the cipher is secure [28].

Key-Less Permutation. A key-less permutation is a bijective mapping on fixed-length strings. Permutations received a high level of attention during the SHA-3 competition[1] – last but not least due to its winner [16]. Quite a number of CAESAR submissions use a key-less permutation as their underlying primitive. The most famous keyless permutations is the sponge construction [15], which is an iterated function with variable-length in- and outputs from a permutation (or transfomration) that itself operates on a fixed-length state. Literally, the sponge is said to *absorb* its inputs block by block first before it processes and *squeezes* it out afterwards. Closely related to sponges are duplex constructions [17]. Unlike a sponge that is stateless between calls, the duplex accepts calls that take an input string and return an output string that will depend on all previous inputs.

Hash Function/Compression Function. A hash function maps strings of arbitrary length to fixed-length outputs. For cryptographic hash functions, it is not feasible to find a collision, preimage and second preimage. A compression function is defined similarly as a hash function, but compresses two fixed-length inputs to a single fixed-length output.

Feistel-State-Based. The structure of a number of CAESAR candidates is similar to that of Type-3 Feistel schemes [84]. Such schemes maintain a multi-block state $S_0, \ldots, S_n$, which is updated by feeding in one message block (e.g., $S_0 = S_0 \oplus M$) and updating each state with the result of its neighbor state block, processed by a round function: $S_i = S_i \oplus f(S_{i-1})$.

2.2 Functional Characteristics

Parallelizable. Various block-cipher modes for authenticated encryption are inherently sequential. Many designs chose their sequential nature to satisfy stricter notions of security, others to achieve lightweight implementations. We call an encryption operation parallelizable if the processing of the i-th input block does not depend on the output of processing the j-th block, for any $i \neq j$. We call an AE scheme *fully parallelizable* if encryption and decryption are both parallel operations. Note that we regard parallelizable encryption and decryption separately.

Online. A cipher is called online if the encryption of the i-th input block M_i depends only on the blocks $M_1, \ldots, M_{i-1}$. We call an AEAD scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ on-line if $\mathcal{E}$ is an on-line cipher and $\mathcal{D}$ its inverse operation. Schemes that are not online are called offline or two-pass.

Inverse-Free. AE schemes that employ only an encryption or decryption function save precious area space. Wlog., we call an AE scheme inverse-free if it does not require only either its underlying primitive's forward or inverse operation, e.g., as does require the block cipher's decryption function.

AES-Based. During the years, the AES has seen various cryptanalytic efforts that investigated its design in detail and lead to a high level of trust of its security. Moreover, starting with Intel's Westmere microarchitecture in 2011, current processors provide native AES instructions that allow fast constant-time encryption and decryption. Hence, AE schemes that build upon standardized primitives can benefit from the available instruction sets and existing cryptanalysis.

---

[1] http://competitions.cr.yp.to/sha3.html

Incremental Authenticated Encryption. AE schemes are frequently used to encrypt lots of data, wherein subsequent messages differ only by a fraction (e.g., a single block) from each other. An AE scheme is said to provide *incremental authenticated encryption*, if, given a previous authenticated ciphertext and tag (C, T) for a message M, encrypting and authenticating a message M' that differs only from M only in a fraction can be computed significantly faster. At Dagstuhl'14 [9], Yasuda described several classes of incremental AE; in this paper, we assume that recomputation requires only the costs for processing the changed blocks and the cost for deriving the tag.
Note that some schemes may provide this property under the requirement of reusing the nonce. We consider nonce misuse to be an *erroneous* usage which should not be encouraged to obtain a nice "feature". Hence, we denote a scheme to provide incremental authenticated encryption only if the nonce does not have to be repeated.

Incremental Associated Data. This property is similar to incremental AE. Say an intermediate result of a previous associated data processing was cached, and the current one changes only in a fraction. We say a scheme to provide Inc associated data, if in such a situation only the changed blocks and a finalization step need to be recomputed.

Fixed Associated Data Reuse. Some applications use the same or slightly modified associated data values for subsequent messages [70]. Schemes that can cache and reuse the result of processing the associated data of the previous encrypted message may allow for a considerable speed-ups. We say that such schemes provide associated-data reuse. Note that this implies that the nonce is not part or appended to the associated data.

Intermediate Tags. Intermediate tags [17] allow the receiver to detect early if parts of a decrypted message are invalid, which saves computations when authenticating large messages. Such information can be integrated easily into an OPRP-CCA-secure on-line cipher by adding well-formed redundancy, such as fixed constants or checksums [2]. Hence, we say that an AE scheme provides this property, if it is on-line and OPRP-CCA-secure.

3 Security

Following the notions by Bellare et al. [13], we consider an authenticated encryption scheme secure (in the CCA3 sense) iff it provides data privacy (in the sense of indistinguishable from an ideal authenticated encryption against chosen-plaintext attacks, IND-CPA) and ciphertext integrity secure against forgery (INT-CTXT). More formally, we call an AE scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ secure iff the IND-CPA + INT-CTXT-advantage is negligible for any nonce-respecting adversary. We define an online authenticated encryption scheme Π to be secure (in the OCCA3 sense) iff it provides OPRP-CPA and INT-CTXT security. We recall the notions in brief in the following subsection.

3.1 Privacy and Integrity Notions.

Let $\mathcal{A}^{\mathcal{O}}$ be a computationally bounded adversary with access to an oracle $\mathcal{O}$, which responds either with real encryptions using $\mathcal{E}$ or a random permutation π, as given in Definition 1. In the beginning, the oracle tosses a fair coin to obtain a bit b. Thereupon, $\mathcal{A}$ can query messages to $\mathcal{O}$. Depending on b, $\mathcal{A}$ obtains either "real" encryptions for the messages it sends, or just the "random" outputs. Hence, the challenge for $\mathcal{A}$ is to guess b.

Definition 1 (IND-CPA-Security). *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an authenticated encryption scheme. Then, the IND-CPA-advantage of a computationally bounded adversary $\mathcal{A}$ for Π is defined as*

$$\mathbf{Adv}_{\Pi}^{\textsf{IND-CPA}}(\mathcal{A}) \leq \left| \Pr\left[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathcal{E}(\cdot,\cdot)} \Rightarrow 1\right] - \Pr\left[\mathcal{A}^{\$(\cdot,\cdot)} \Rightarrow 1\right] \right|.$$

We define $\mathbf{Adv}_{\Pi}^{\textsf{IND-CPA}}(q, \ell, t)$ as the maximum advantage over all IND-CPA-adversaries $\mathcal{A}$ on Π that run in time at most t, and make at most q queries of total length ℓ to the available oracles.

Definition 2 (INT-CTXT-Security). *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an authenticated encryption scheme. Then, the INT-CTXT-advantage of a computationally bounded adversary $\mathcal{A}$ for Π is defined as*

$$\mathbf{Adv}_{\Pi}^{\textsf{IND-CPA}}(\mathcal{A}) \leq \Pr\left[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathcal{E}(\cdot,\cdot),\mathcal{D}(\cdot,\cdot)} \Rightarrow \textit{ forges}\right]$$

We define $\mathbf{Adv}_{\Pi}^{\textsf{INT-CTXT}}(q, \ell, t)$ as the maximum advantage over all INT-CTXT-adversaries $\mathcal{A}$ on Π that run in time at most t, and make at most q queries of total length ℓ to the available oracles.

For the definitions and security notions for online ciphers, please see Bellare et al. [12].

Quantitative Security Statements. The CAESAR call demanded quantitative claims of security for every scheme in terms of query and time complexity for privacy and integrity. The query complexity q represents the logarithm base-2 of the number of blocks that an adversary has to query in order to violate the claimed security goals with probability of 1/2 or greater. The time complexity t reflects the log base-2 of the number of calls to the underlying primitive function that any adversary has to perform in order to break to goals with probability of 1/2, if it has only a small amount of plaintext-ciphertexts.

Provable Security. We indicate which schemes provide a security proof under common well- established assumptions, e.g., the abstraction of their underlying primitive with a random PRF/PRP.

3.2 Robustness

An AE scheme is called robust if it provides CCA3/OCCA3 and additional security against more general adversaries. Note that security proofs for AE schemes used to rely on two common assumptions: (1) nonce-respecting adversaries, and (2) secure underlying primitives. While both aspects are well-understood in theory, they are hard to guarantee in practice. Thus, security issues were overlooked or ignored in various cases and security applications were put at high risk. We consider two robustness notions in the established security definitions: resistance against nonce-ignoring adversaries and against leakage of would-be plaintexts. Like before, we distinguish between on-line and off-line (two-pass) schemes.

Security Against Nonce-Ignoring Adversaries. There is an ongoing discussion about the most appropriate definition of robust AE. Rogaway and Shrimpton [66] follow a strict interpretation of (nonce-)misuse-resistant AE (MRAE). According to their notion, an MRAE-secure lets adversaries gain no advantage when a nonce repeats, except for noticing when the same message was encrypted multiple times. Clearly, following this interpretation implies that MRAE-secure schemes can not be online.

In contrast, the notion of nonce-misuse resistance by Fleischmann et al. [29] exclusively targeted online ciphers; the authors considered a nonce repetition as an erroneous usage, against which resistant schemes should provide a second line of defense. Following their definition, an online AE scheme is called secure against nonce-ignoring adversaries if all an adversary can learn from repeating nonces is the longest common prefix of messages. Thus, the privacy protection degenerates from PRP-CPA security to OPRP-CPA-security in this case. Note that using a nonce-misuse resistance online AE scheme is like wearing a seatbelt: it can save your day, but it can never serve as an excuse for careless driving, i.e., nonce reuse. To respect both views, we opt for a two-way strategy: for two-pass schemes we indicate nonce-misuse resistance iff they provide MRAE (which is equivalent to PRP-CPA and INT-CTXT) security [66]; for online schemes, we indicate nonce-misuse resistance if they provide OPRP-CPA and INT-CTXT security.

Security Against Plaintext-Aware Adversaries. An unverified plaintext denotes the message that results from decrypting an unauthentic ciphertext. The security arguments for AE schemes usually require that adversaries never learn anything about such unverified plaintexts. However, for larger data streams or in real-time environments, it may be hard or even impossible to buffer the decryption until the tag is verified. Fouque et al. [30] proposed to blind decryption results by re-encrypting it with a secret that is released only after the verification succeeded; though, this solution is not practically relevant either.

Again, (at least) two views on the same. The problem was concerned first by Abed et al. [2] in their notion of decryption-misuse resistance for online AE schemes. Their notion follows from OPRP-CCA-security, which is the strongest form of non-malleability and decryption-misuse resistance an online cipher can provide, i.e., an adversary that manipulates the i-th block will obtain garbled pseudorandom outputs starting from that block. Andreeva et al. [7] formalized and generalized this view. They introduced two notions of plaintext awareness (PA1, PA2) for privacy and the INT-RUP notion for integrity. Their definitions reflect that no adversary can gain any advantage of having a decryption oracle that always returns a plaintext from any ciphertext input.

As for nonce-misuse, we opt for a two-way strategy. For two-pass schemes we indicate decryption-misuse resistance iff they provide PRP-CCA security; for online schemes, we indicate decryption-misuse resistance if they offer OPRP-CCA security.

3.3 Overview

In the following, we give an overview over the functional and security properties of the remaining CAESAR submissions. Tables 1- 6 list the properties and parameters of block-cipher-based and non-block-cipher-based AE schemes.

Candidate	**Mode**	**Masking**	**Primitive**	**Features**						**Security**		
				Parallelizable Enc/Dec	*Online*	*Inverse-Free*	*Incremental AD/AE*	*Fixed AD reuse*	*Intermediate Tags*	*Security proof*	*Nonce-MR*	*Decryption-MR*
++AE [63]	ECB	AX	AES	●/●	●	–	–/–	–	–	–	●	–
AES-CMCC [72]	CBC	–	AES	–/●	–	–	–/–	–	–	●	–	–
AES-COPA [8]	EME	Doubling	AES	●/●	●	●	●/–	●	–	●	●	–
AES-CPFB [58]	CTR,PFB	–	AES	●/–	●	●	–/–	–	–	●	–	–
AES-JAMBU [76]	OFB	–	AES	–/–	●	●	–/–	–	–	–	●	–
AES-OTR [56]	OTR	Doubling	AES	●/●	●	●	●/–	●	–	●	–	–
AEZ [39]	OTR	–	AES-4	●/●	–	●	●/–	●	–	●	●	●
AVALANCHE [4]	ECB	–	AES	●/●	●	●	–/–	–	–	●	–	–
CBA [40]	ECB	Doubling	AES	●/●	●	●	●/–	●	–	–	–	–
CLOC [41]	CFB	–	AES^a	–/–	●	●	–/–	●	–	●	–	–
Deoxys$^{\neq}$ [43]	TAE	–	Deoxys-BC	●/●	●	–	–/–	–	–	–	–	–
Deoxys$^{=}$ [43]	EME	–	Deoxys-BC	●/●	●	–	–/–	–	–	–	●	–
ELmD [25]	EME	Doubling	AES	●/●	●	–	–/–	–	●	●	●	–
iFeed[AES] [83]	iFeed	Doubling	AES	●/–	●	●	●/–	●	●	●	●	–
iSCREAM [32]	TAE	–	iSCREAM	●/●	●	●	–/–	–	–	–	–	–
Joltik$^{\neq}$ [44]	TAE	–	Joltik-BC	●/●	●	–	–/–	–	–	–	–	–
Joltik$^{=}$ [44]	EME	–	Joltik-BC	●/●	●	–	–/–	–	–	–	●	–
Julius-CTR [11]	CTR	GFM	AES	●/●	–	●	–/–	–	–	●	–	–
Julius-ECB [11]	ECB	GFM	AES	●/●	–	–	–/–	–	–	●	●	–
KIASU$^{\neq}$ [45]	TAE	–	KIASU-BC	●/●	●	–	–/–	–	–	●	–	–
KIASU$^{=}$ [45]	EME	–	KIASU-BC	●/●	●	–	–/–	–	–	●	●	–
LAC [82]	LEX	–	L-Block	●/●	●	–	–/–	–	–	–	–	–
Marble [34]	ECB	Doubling,AES-4	AES-4	●/●	●	–	–/–	●	–	–	●	●
OCB [50]	XEX	Doubling	AES	●/●	●	–	–/–	–	–	●	–	–
POET [1]	ECB	AES-4/10	AES	○/○	●	●	●/–	●	●	●	●	●
SCREAM [32]	TAE	–	SCREAM	●/●	●	●	–/–	–	–	–	–	–
SHELL [74]	EME	CTR,Doubling	AES	●/●	●	–	–/–	–	–	●	●	–
SILC [42]	CFB	–	AES^a	–/●	●	●	–/–	–	–	●	–	–
Silver [62]	TAE	–	MAES	●/●	●	–	–/–	–	–	●	–	–
YAES [21]	CTR	–	AES	●/●	●	●	●/–	●	–	–	–	–

Table 1: Block-cipher-based candidates. ^{a} = Primary recommendation is AES-based, ● = Provides feature, – = Seems not to provide feature, ○ = Pipelineable.

Construction	Candidate	Design	Primitive	Features						Security		
				Parallelizable Enc/Dec	*Online*	*Inverse-Free*	*Incremental AD/AE*	*Fixed AD reuse*	*Intermediate Tags*	*Security proof*	*Nonce-MR*	*Decryption-MR*
Feistel-state-based	AES-AEGIS [78]	–	AES-round	●/–	●	●	–/–	–	–	–	–	–
	MORUS [77]	–	MORUS	–/–	●	●	–/–	–	–	–	–	–
	Tiaoxin [60]	–	AES-round	●/●	●	●	–/–	–	–	–	–	–
Stream-cipher-based	ACORN [75]	–	ACORN	●/●	●	●	–/–	–	–	–	–	–
	Calico [71]	–	ChaCha, SipHash	–/–	●	●	–/–	–	–	–	–	–
	Enchilada [37]	–	ChaCha, Rijndael	●/●	●	●	●/–	●	–	●	–	–
	HS1-SIV [48]	SIV	ChaCha, Poly1305	–/–	–	●	–/–	–	–	●	●	–
	Raviyoyla [73]	–	MAGv2	–/–	●	●	–/–	–	–	–	–	–
	Sablier [81]	–	Sablier	●/●	●	●	●/–	●	–	–	–	–
	TriviA-ck [22]	–	Trivia-SC	●/●	–	●	–/–	–	●	●	–	–
	Wheesht [54]	–	Wheesht	–/–	●	●	–/–	–	–	–	–	–
CF-based	OMD [24]	–	SHA-256/512	–/–	●	●	●/–	●	–	●	–	–
Permutation-based	Minalpher [70]	XEX	Minalpher-P	●/●	●	–	–/–	–	–	●	●	●
	PAEQ [20]	PPAE	AESQ	●/●	●	●	●/–	●	–	●	●	–
	Prøst-COPA [46]	EME	Prøst	●/●	●	●	●/–	●	–	–	●	–
	Prøst-OTR [46]	OTR	Prøst	●/●	●	●	●/–	●	–	–	–	–
Sponge-based	Artemia [3]	JHAE	Artemia	–/–	●	●	–/–	–	–	●	●	–
	Ascon [26]	Duplex	Ascon	–/–	●	●	–/–	–	–	–	●	–
	ICEPOLE [59]	Duplex	n.n.	●/●	●	●	–/–	–	●	●	●	–
	Ketje [18]	Duplex	Keccak-f	–/–	●	●	–/–	–	●	●	–	–
	Keyak [33]	Duplex	Keccak-f	●/●	●	●	–/–	–	●	●	–	–
	NORX [10]	Duplex	n.n.	●/●	●	●	–/–	–	–	–	–	–
	π-cipher [31]	Duplex	n.n.	●/●	●	●	–/–	–	–	–	–	–
	PRIMATEs-GIBBON [5]	Duplex	PRIMATE	–/–	●	●	–/–	–	–	–	–	–
	PRIMATEs-HANUMAN [5]	Duplex	PRIMATE	–/–	●	●	–/–	–	–	–	–	–
	PRIMATEs-APE [5]	Duplex	PRIMATE	–/–	●	–	●/–	●	–	–	●	●
	Prøst-APE [46]	Duplex	Prøst	–/–	●	–	●/–	●	–	–	●	●
	STRIBOB [69]	Duplex	n.n.	–/–	●	●	–/–	–	–	●	–	–

Table 2: Candidates based on Feistel-state, stream ciphers, compression functions, (non-sponge) permutations, and sponges in particular. n.n. = Unnamed custom primitive, ● = Provides feature, – = Seems not to provide feature.

Candidate	**Parameters**			**Privacy**	**Integrity**
	k	ν	t	q/t	q/t
++AE	128	64	128	64/128	64/126.75
AES-CMCC-32-64	128	32^*	64	64/128	64/128
AES-CMCC-32-32	128	32^*	32	64/128	32/128
AES-CMCC-16-32	128	16^*	32	64/128	16/128
AES-CMCC-32-16	128	32^*	16	64/128	32/128
AES-CMCC-16-16	128	16^*	16	64/128	16/128
AES-COPA	128	128	128	64/128	64/128
AES-CPFB	128	96	128	64/128	64/128
AES-JAMBU	128	64	64	64/128	64/128
AES-OTR-128	128	96	128	64/128	128/128
AES-OTR-256	256	96	128	64/256	128/256
AEZ	128	96	128	61/128	128/128
AVALANCHE-512	512	160	128	103/256	127/256
AVALANCHE-448	448	128	128	71/192	127/192
AVALANCHE-384	384	80	128	55/128	127/128
CBA-128-32	128	96	32	47/128	47/128
CBA-128-64	128	96	64	63/128	63/128
CBA-128-96	128	96	96	63/128	63/128
CBA-192-64	192	96	64	47/192	47/192
CBA-256-96	256	96	96	63/256	63/256
CLOC-AES-12	128	96	64	64/128	64/128
CLOC-AES-8	128	64	64	64/128	64/128
CLOC-TWINE-6	80	48	32	32/ 80	32/ 80
Deoxys$^{\neq}$-128-128	128	64	128	64/128	128/128
Deoxys$^{\neq}$-256-128	256	64	128	128/256	128/256
Deoxys$^{=}$-128-128	128	64	128	64/128	64/128
Deoxys$^{=}$-256-128	256	64	128	64/256	64/256
ELmD-0-f	128	64	128	62.8/128	62.4/128
ELmD-127-f	128	64	128-255	62.8/128	62.3/128
iFeed[AES]-128-96	128	96	128	64/128	128/128
iFeed[AES]-128-104	128	104	128	64/128	128/128

Table 3: Parameter sets for block-cipher-based candidates. * = 128-bit SNM optional.

Candidate	**Parameters**			**Privacy**	**Integrity**
	k	ν	t	q/t	q/t
Joltik$^{\neq}$-64-64	64	32	64	32/ 64	64/ 64
Joltik$^{\neq}$-80-48	80	24	64	24/ 80	64/ 80
Joltik$^{\neq}$-96-96	96	48	64	48/ 96	64/ 96
Joltik$^{\neq}$-128-64	128	32	64	32/128	64/128
Joltik$^{=}$-64-64	64	32	64	32/ 64	32/ 32
Joltik$^{=}$-80-48	80	24	64	24/ 80	24/ 32
Joltik$^{=}$-96-96	96	48	64	48/ 96	48/ 32
Joltik$^{=}$-128-64	128	32	64	32/128	32/ 32
Julius-ECB-R.	128	96	128	64/128	128/128
Julius-ECB-C.	128	64	128	64/128	64/128
Julius-CTR-R.	128	96	128	64/128	64/128
Julius-CTR-C.	128	64	128	64/128	64/128
KIASU$^{\neq}$	128	32	128	32/128	64/128
KIASU$^{=}$	128	32	128	32/128	64/128
LAC	80	64	64	40/ 80	64/ 80
Marble	128	0	128	128/128	128/128
OCB-128-64	128	128	64	64/128	64/ 64
OCB-128-96	128	128	96	64/128	64/ 96
OCB-128-128	128	128	128	64/128	64/128
OCB-192-64	192	128	64	64/192	64/ 64
OCB-192-96	192	128	96	64/192	64/ 96
OCB-192-128	192	128	128	64/192	64/128
OCB-256-64	256	128	64	64/256	64/ 64
OCB-256-96	256	128	96	64/256	64/ 96
OCB-256-128	256	128	128	64/256	64/128
POET-4	128	128	128	64/128	55/128
POET-10	128	128	128	64/128	64/128
SCREAM	128	96	128	64/128	64/128
SHELL-128-64	128	64	128	55/ 80	55/ 80
SHELL-128-80	128	80	128	55/ 80	55/ 80
SILC/AES-8	128	64	64	64/128	64/128
SILC/AES-12	128	96	64	64/128	64/128
SILC/PRESENT	80	48	32	32/ 80	32/ 80
SILC/LED	80	48	32	32/ 80	32/ 80
Silver	128	128	128	64/128	128/128
YAES	128	127	128	48/ 64	55/128

Table 4: Parameter sets for block-cipher-based candidates. ECB-R. = ECB-Regular, ECB-C. = ECB-Compact, CTR-R. = CTR-Regular, CTR-C. = CTR-Compact.

Candidate	Parameters			Privacy	Integrity
	k	ν	t	q/t	q/t
AES-AEGIS-128	128	128	128	64/128	64/128
AES-AEGIS-256	256	256	128	128/256	128/128
MORUS-640	128	128	128	128/128	128/128
MORUS-1280	256	128	128	256/256	128/256
Tiaoxin	128	128	128	128/128	128/128
ACORN-128	128	128	128	64/128	64/128
Calico-512	512	64	64	63/128	64/128
Calico-256	256	64	64	63/128	64/256
Enchilada-128	256	64	128	128/128	128/128
Enchilada-256	256	64	128	128/255	128/255
HS1-SIV-Lo	256	96	64	56/256	56/256
HS1-SIV	256	96	128	112/256	112/256
HS1-SIV-Hi	256	96	256	168/256	168/256
Raviyoyla	256	128	128	128/256	128/256
Sablier	80	80	32	40/80	32/128
TriviA-ck	128	64	128	64/128	128/128
Wheesht	512	128^*	256	128/256	128/256
OMD	256	256	32-256	127/256	127/256
Minalpher	256	104	128	64/128	128/256
PAEQ-64	64	64	64	32/64	80/64
PAEQ-80	80	80	80	40/80	40/80
PAEQ-128	128	96	128	64/128	64/128
PAEQ-160	160	128	160	80/160	80/160
PAEQ-t-128	128	128	512	64/128	64/128
PAEQ-tnm-118	128	256	512	40/128	80/128
Prøst-COPA-128	128	128	256	64/128	64/128
Prøst-COPA-256	256	256	256	128/256	128/256
Prøst-OTR-128	128	64	128	64/128	64/128
Prøst/OTR-256	256	256	256	128/256	128/256

Table 5: Parameter sets for (from top to bottom) Feistel-state, stream-cipher-based, compression-function-based, and permutation-based candidates. * = 128-bit SNM.

Candidate	Parameters			Privacy	Integrity
	k	ν	t	q/t	q/t
Artemia-128	128	128	128	64/128	64/128
Artemia-256	256	256	256	64/128	128/128
Ascon-128	128	128	128	64/128	64/128
Ascon-96	96	96	96	96/96	96/96
ICEPOLE-128	128	128^*	128	126/128	128/128
ICEPOLE-128a	128	128	128	126/128	128/128
ICEPOLE-256a	256	96	128	62/128	128/128
Ketje/JR	128	128	128	128/128	128/128
Ketje/SR	96	80	96	96/128	96/128
Keyak	128	128	128	123/128	128/128
NORX/32-4-1	128	64	128	64/128	64/128
NORX/64-4-1	256	128	256	128/256	256/256
NORX/32-6-1	128	64	128	64/128	64/128
NORX/64-6-1	256	128	256	128/256	256/256
NORX/64-4-4	256	128	256	64/256	128/256
π-cipher/16-96	96	32^*	128	48/96	96/96
π-cipher/16-128	128	32^*	128	64/128	128/128
π-cipher/32-128	128	$128^\dagger$	256	64/128	128/128
π-cipher/32-256	256	$128^\dagger$	256	128/256	256/256
π-cipher/64-128	128	$128^\ddagger$	512	64/128	128/128
π-cipher/64-256	256	$128^\ddagger$	512	128/256	256/512
Pr.-HANUMAN-10	80	80	80	80/80	80/80
Pr.-HANUMAN-15	120	120	120	120/120	120/120
Pr.-GIBBON-10	80	80	80	80/80	80/80
Pr.-GIBBON-15	120	120	120	120/120	120/120
Pr.-APE-10	160	80	160	80/80	80/80
Pr.-APE-15	240	120	240	120/120	120/240
Prøst/APE-128	128	64	128	64/128	64/128
Prøst/APE-256	256	128	256	128/256	128/256
STRIBOB	192	128	128	64/191	127/128

Table 6: Parameter sets for sponge-based candidates. Pr. = PRIMATEs, $^*/^\dagger/^\ddagger$ = 128/256/512-bit SNM.

4 Acknowledgments

The authors would like to thank Bart Mennink for valuable comments and the fruitful discussion during the visit of Farzaneh Abed at ESAT KU Leuven. Furthermore, we thank Elena Andreeva, and Jakob Wenzel for their helpful comments.

References

1. Farzaneh Abed, Scott Fluhrer, John Foley, Christian Forler, Eik List, Stefan Lucks, David McGrew, and Jakob Wenzel. The poet family of on-line authenticated encryption schemes. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
2. Farzaneh Abed, Scott R. Fluhrer, Christian Forler, Eik List, Stefan Lucks, David A. McGrew, and Jakob Wenzel. Pipelineable On-Line Encryption. In Carlos Cid and Christian Rechberger, editors, *Fast Software Encryption*, Lecture Notes in Computer Science (to appear). Springer, 2014.
3. Javad Alizadeh, Mohammad Reza Aref, and Nasour Bagheri. Artemia. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
4. Basel Alomair. Avalanche. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
5. Elena Andreeva, Begül Bilgin, Andrey Bogdanov, Atul Luykx, Florian Mendel, Bart Mennink, Nicky Mouha, Qingju Wang, and Kan Yasuda. Primates. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
6. Elena Andreeva, Andrey Bogdanov, Martin M. Lauridsen, Atul Luykx, Bart Mennink, Elmar Tischhauser, and Kan Yasuda. Aes-cobra. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
7. Elena Andreeva, Andrey Bogdanov, Atul Luykx, Bart Mennink, Nicky Mouha, and Kan Yasuda. How to Securely Release Unverified Plaintext in Authenticated Encryption. pages 1–31, 2014. To appear.
8. Elena Andreeva, Andrey Bogdanov, Atul Luykx, Bart Mennink, Elmar Tischhauser, and Kan Yasuda. Aes-copa. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
9. Frederik Armknecht, Helena Handschuh, Tetsu Iwata, and Bart Preneel. Symmetric Cryptography (Dagstuhl Seminar 14021). *Dagstuhl Reports*, 4(1):1–16, 2014.
10. Jean-Philippe Aumasson, Philipp Jovanovic, and Samuel Neves. Norx. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
11. Lear Bahack. Julius. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
12. Mihir Bellare, Alexandra Boldyreva, Lars R. Knudsen, and Chanathip Namprempre. Online Ciphers and the Hash-CBC Construction. In Joe Kilian, editor, *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 292–309. Springer, 2001.
13. Mihir Bellare and Chanathip Namprempre. Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. In Tatsuaki Okamoto, editor, *ASIACRYPT*, volume 1976 of *Lecture Notes in Computer Science*, pages 531–545. Springer, 2000.
14. Mihir Bellare and Chanathip Namprempre. Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. *Journal of Cryptology*, 21(4):469–491, 2008.
15. G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche. Sponge Functions. Ecrypt Hash Workshop 2007, May 2007.
16. G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche. The Keccak SHA-3 submission. Submission to NIST (Round 3), 2011.
17. Guido Bertoni, Joan Daemen, Michael Peeters, and Gilles Van Assche. Duplexing the Sponge: Single-Pass Authenticated Encryption and Other Applications. In Ali Miri and Serge Vaudenay, editors, *Selected Areas in Cryptography*, volume 7118 of *Lecture Notes in Computer Science*, pages 320–337. Springer, 2011.
18. Guido Bertoni, Joan Daemen, Michaël Peeters, and Ronny Van Keer Gilles Van Assche. Ketje. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
19. Alex Biryukov. Design of a New Stream Cipher-LEX. In Matthew J. B. Robshaw and Olivier Billet, editors, *New Stream Cipher Designs - The eSTREAM Finalists*, volume 4986 of *Lecture Notes in Computer Science*, pages 48–56. Springer, 2008.
20. Alex Biryukov and Dmitry Khovratovich. Paeq. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
21. Antoon Bosselaers and Fre Vercauteren. Yaes. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
22. Avik Chakraborti and Mridul Nandi. Trivia-ck. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
23. Faith Chaza, Cameron McDonald, and Roberto Avanzi. Faser: Authenticated encryption in a feedback shift register. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
24. Simon Cogliani, Diana Ştefania Maimuţ, David Naccache, Rodrigo Portella do Canto, Reza Reyhanitabar, Serge Vaudenay, and Damian Vizár. Offset merkle-damgård (omd), a caesar proposal. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
25. Nilanjan Datta and Mridul Nandi. Elmd. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
26. Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon: A family of authenticated encryption algorithms. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
27. Niels Ferguson and Bruce Schneier. A cryptographic evaluation of IPsec. *Counterpane Internet Security, Inc*, 3031, 2000.
28. Niels Ferguson, Doug Whiting, Bruce Schneier, John Kelsey, Stefan Lucks, and Tadayoshi Kohno. Helix - Fast Encryption and Authentication in a Single Cryptographic Primitive. In *Proc. Fast Software Encryption 2003, volume 2887 of LNCS*, pages 330–346. Springer-Verlag, 2003.

29. Ewan Fleischmann, Christian Forler, and Stefan Lucks. McOE: A Family of Almost Foolproof On-Line Authenticated Encryption Schemes. In *FSE*, pages 196–215, 2012.
30. Pierre-Alain Fouque, Antoine Joux, Gwenaëlle Martinet, and Frédéric Valette. Authenticated On-Line Encryption. In *Selected Areas in Cryptography*, pages 145–159, 2003.
31. Danilo Gligoroski, Hristina Mihajloska, Simona Samardjiska, Håkon Jacobsen, Mohamed El-Hadedy, and Rune Erlend Jensen. π-cipher. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
32. Vincent Grosso, Gaëtan Leurent, François-Xavier Standaert, Kerem Varici, François Durvaux, Lubos Gaspar, and Stéphanie Kerckhof. Scream and iscream side channel resistant authenticated encryption with masking. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
33. Michaël Peeters Guido Bertoni, Joan Daemen, Gilles Van Assche, and Ronny Van Keer. Keyak. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
34. Jian Guo. Marble. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
35. Shai Halevi. EME*: Extending EME to Handle Arbitrary-Length Messages with Associated Data. In Anne Canteaut and Kapalee Viswanathan, editors, *Progress in Cryptology - INDOCRYPT 2004, 5th International Conference on Cryptology in India, Chennai, India, December 20-22, 2004, Proceedings*, volume 3348 of *Lecture Notes in Computer Science*, pages 315–327. Springer, 2004.
36. Shai Halevi and Phillip Rogaway. A Parallelizable Enciphering Mode. In Tatsuaki Okamoto, editor, *CT-RSA*, volume 2964 of *Lecture Notes in Computer Science*, pages 292–304. Springer, 2004.
37. Sandy Harris. Enchilada. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
38. Matt Henricksen, Shinsaku Kiyomoto, and Jiqiang Lu. The hkc authenticated stream cipher. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
39. Viet Tung Hoang, Ted Krovetz, and Phillip Rogaway. Aez. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
40. Hossein Hosseini and Shahram Khazaei. Cba mode. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
41. Tetsu Iwata, Kazuhiko Minematsu, Jian Guo, and Sumio Morioka. Cloc: Compact low-overhead cfb. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
42. Tetsu Iwata, Kazuhiko Minematsu, Jian Guo, Sumio Morioka, and Eita Kobayashi. Silc: Simple lightweight cfb. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
43. Jérémy Jean, Ivica Nikolić, and Thomas Peyrin. Deoxys. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
44. Jérémy Jean, Ivica Nikolić, and Thomas Peyrin. Joltik. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
45. Jérémy Jean, Ivica Nikolić, and Thomas Peyrin. Kiasu. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
46. Elif Bilge Kavun, Martin M. Lauridsen, Gregor Leander, Christian Rechberger, Peter Schwabe, and Tolga Yalç I_n . Prøst. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
47. Stefan Kölbl, Martin M. Lauridsen, Christian Rechberger, and Tyge Tiessen. Authenticated Encryption Zoo, 2014. `https://aezoo.compute.dtu.dk/`.
48. Ted Krovetz. Hs1-siv. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
49. Ted Krovetz and Phillip Rogaway. The Software Performance of Authenticated-Encryption Modes. In *FSE*, pages 306–327, 2011.
50. Ted Krovetz and Phillip Rogaway. Ocb. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
51. Watson Ladd. Mcmambo v1: A new kind of latin dance. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
52. Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable Block Ciphers. In Moti Yung, editor, *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 2002.
53. Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable Block Ciphers. *Journal of Cryptology*, 24(3):588–613, 2011.
54. Peter Maxwell. wheesht. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
55. David McGrew and John Viega. The Galois/Counter Mode of Operation (GCM). *Submission to NIST. http://csrc.nist.gov/CryptoToolkit/modes/proposedmodes/gcm/gcm-spec.pdf*, 2004.
56. Kazuhiko Minematsu. Aes-otr. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
57. Kazuhiko Minematsu. Parallelizable Rate-1 Authenticated Encryption from Pseudorandom Functions. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*, pages 275–292. Springer, 2014.
58. Miguel Montes and Daniel Penazzi. Aes-cpfb. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
59. Paweł Morawiecki, Kris Gaj, Ekawat Homsirikamol, Krystian Matusiewicz, Josef Pieprzyk, Marcin Rogawski, Marian Srebrny, and Marcin Wójcik. Icepole. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
60. Ivica Nikolić. Tiaoxin-346. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
61. US Department of Commerce. DES Modes of Operation. Technical Report FIPS PUB 81, US Department of Commerce / National Bureau of Standards, December 1998.
62. Daniel Penazzi and Miguel Montesg. Silver. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
63. Francisco Recacha. ++ae. http://competitions.cr.yp.to/caesar-submissions.html, 2014.

64. Phillip Rogaway. Authenticated-Encryption with Associated-Data. In *ACM Conference on Computer and Communications Security*, pages 98–107, 2002.
65. Phillip Rogaway. Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC. In Pil Joong Lee, editor, *Advances in Cryptology - ASIACRYPT 2004, 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, December 5-9, 2004, Proceedings*, volume 3329 of *Lecture Notes in Computer Science*, pages 16–31. Springer, 2004.
66. Phillip Rogaway and Thomas Shrimpton. A Provable-Security Treatment of the Key-Wrap Problem. In Serge Vaudenay, editor, *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 373–390. Springer, 2006.
67. Phillip Rogaway and Haibin Zhang. Online Ciphers from Tweakable Blockciphers. In Aggelos Kiayias, editor, *Topics in Cryptology - CT-RSA 2011 - The Cryptographers' Track at the RSA Conference 2011, San Francisco, CA, USA, February 14-18, 2011. Proceedings*, volume 6558 of *Lecture Notes in Computer Science*, pages 237–249. Springer, 2011.
68. Markku-Juhani O. Saarinen. The cbeamr1 authenticated encryption algorithm. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
69. Markku-Juhani O. Saarinen. The stribobr1 authenticated encryption algorithm. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
70. Yu Sasaki, Yosuke Todo, Kazumaro Aoki, Yusuke Naito, Takeshi Sugawara, Yumiko Murakami, Mitsuru Matsui, and Shoichi Hirose. Minalpher. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
71. Christopher Taylor. The calico family of authenticated ciphers. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
72. Jonathan Trostle. Aes-cmcc. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
73. Rade Vuckovac. Raviyoyla. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
74. Lei Wang. Shell. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
75. Hongjun Wu. Acorn: Alightweight authenticated cipher. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
76. Hongjun Wu and Tao Huang. Aes-jambu. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
77. Hongjun Wu and Tao Huang. The authenticated cipher morus. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
78. Hongjun Wu and Bart Preneel. Aegis:a fast authenticated encryption algorithm. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
79. Dingfeng Ye, Peng Wang, Lei Hu, Liping Wang, Yonghong Xie, Siwei Sun, and Ping Wang. Paes: Parallelizable authenticated encryption schemes based on aes round function. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
80. Dingfeng Ye, Peng Wang, Lei Hu, Liping Wang, Yonghong Xie, Siwei Sun, and Ping Wang. Panda. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
81. Bin Zhang, Zhenqing Shi, Chao Xu, Yuan Yao, and Zhenqi Li. Sablier. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
82. Lei Zhang, Wenling Wu, Yanfeng Wang, Shengbao Wu, and Jian Zhang. Lac: A lightweight authenticated encryption cipher. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
83. Liting Zhang, Wenling Wu, Han Sui, and Peng Wang. iFeed[AES]. http://competitions.cr.yp.to/caesar-submissions.html, 2014.
84. Yuliang Zheng, Tsutomu Matsumoto, and Hideki Imai. On the Construction of Block Ciphers Provably Secure and Not Relying on Any Unproved Hypotheses. In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 461–480. Springer, 1989.