Structural Classification of Authenticated Encryption Schemes

Avik Chakraborti[1], Nilanjan Datta[2], Ashwin Jha[1] and Mridul Nandi[1]

[1] Indian Statistical Institute, Kolkata, India
{avikchkrbrti,ashwin.jha1991,mridul.nandi}@gmail.com
[2] Institute for Advancing Intelligence, TCG CREST, Kolkata, India
nilanjan_isi_jrf@yahoo.com

Abstract. In this short note, we aim to give a structural classification of modes of operations for authenticated encryption (AEAD). First, we briefly discuss various features that are desirable in an AEAD mode. Then, we classify AEAD modes according to their structure, understand their target area of applications, discuss their basic design goals and associated features. Finally, we give a brief description of each of the 32 second round candidates in NIST LwC project, distributing them in appropriate class based on their structure.

Keywords: Lightweight · AEAD · Feedback · Parallel · SIV · Sponge · Stream Cipher

1 Authenticated Encryption Schemes

Authenticated encryption (AE) schemes are symmetric-key primitives that achieve both confidentiality and authenticity. An authenticated encryption scheme $\mathcal{A}$ is a tuple (E, D) of two algorithms, the encryption algorithm E and the verified decryption algorithm D. The working principle of any AE scheme $\mathcal{A}$ is as follows: Suppose Alice and Bob are two parties sharing a common secret key K. Whenever Alice wants to send a message M to Bob, she sends (C, T), where the ciphertext-tag pair (C, T) is the output of encryption algorithm, i.e., $(C, T) = \mathcal{A}.\mathsf{E}(K, M)$. When Bob receives a ciphertext-tag pair (C', T'), he runs $\mathcal{A}.\mathsf{D}(K, C', T')$, and gets some message M' in case (C', T') is a valid ciphertext-tag pair, referred as (C', T') *authenticates successfully*, and an error symbol $\perp$ when (C', T') is invalid. Note that, for all K and M, $\mathcal{A}.\mathsf{D}(K, \mathcal{A}.\mathsf{E}(K, M)) = M$ always holds.

Additionally, Alice may want to send some associated data or header A, which is not private but requires authentication. In that case E and D are extended, say E' and D', respectively, to take the additional input A, and we must have $\mathcal{A}'.\mathsf{D}'(K, A, \mathcal{A}'.\mathsf{E}'(K, A, M)) = M$ for all (K, A, M). This modified tuple $\mathcal{A}' = (\mathsf{E}', \mathsf{D}')$ is called an AE with associated data functionality, or an AEAD scheme. Note that $\mathcal{A}'$ guarantees privacy or confidentiality of M and authenticity of (A, M). Some popular AEAD modes of operation include OCB [11], GCM [9], COLM [4], etc. In this note, we will consider AEAD mode of operation.

1.1 Desirable Features of an AEAD Scheme

We briefly discuss some desirable properties of an AEAD mode of operation:

1. Inverse-free: An authenticated encryption is called *inverse-free* if both the encryption and verified decryption algorithm do not invoke the inverse (if exists) of the underlying primitive. This property is particularly useful for area-efficient lightweight AEAD designs specially when both encryption and verified decryption are implemented in the same device.

2. Compact state: State size is the size of memory needed for storing internal values (like key) and temporary variables arising in the processing of an AEAD scheme. A compact state is important since the state size directly corresponds to the size of memory (i.e., RAM and register in software and hardware, respectively) required to implement the AEAD scheme.

3. Online: An authenticated encryption is called *online* if it allows encryption to produce ciphertext blocks before subsequent plaintext blocks (or the plaintext length) are known, and decryption to produce plaintext blocks before subsequent ciphertext blocks (or the ciphertext length) are known. Formally, the i^{th} ciphertext block depends only on the first i plaintext blocks:

$$\forall\ i = 1 \text{ to } \ell,\ C_i = f_i(M_1, M_2, \ldots, M_i).$$

   Online constructions are particularly useful in real-time streaming protocols (e.g. SRTP, SRTCP and SSH), Optical Transport Networks, Smart cards, where block-wise encryption/decryption is required and ciphertext/plaintext can be released on the fly in order to reduce the end-to-end latency and/or compensate for low memory.

4. Single-pass: An authenticated encryption is called *single-pass* (or *one-pass*) if one needs to make only a single pass through the data, simultaneously doing what is needed to ensure both privacy and authenticity. Typically, the computational cost for single-pass schemes is about half as compared to two-pass schemes. As a result single-pass schemes are considered more efficient as compared to multi-pass schemes.

5. Rate: The *rate* of an AEAD is defined as the number of blocks of message (plaintext) processed per non-linear (block-cipher, field multiplication) operation. For example, rate of SIV [12] is 0.5, while rate of OCB [11] is 1. Constructions with higher rate have shorter latency and achieve high speed.

6. Optimal: An authenticated encryption scheme is called *optimal* if the number of non-linear operations it uses is the minimum possible. For nonce based AEAD, the minimum number of non-linear operations required to process a data with a block associated data and m block plaintext is $(a + m + 1)$ [7]. For deterministic AEAD, this number is $(a + 2m)$. This property makes a construction efficient for short messages and reduces the latency.

7. Nonce-misuse Resistant: An authenticated encryption is called *misuse resistant* if the scheme provides security even if nonce is repeated. SIV [12] modes are in general misuse resistant. Constructions such as COPA [5] and ELmD [8] provide some weaker notions of nonce-misuse resistance: they provide online PRF security if nonce repeats.

8. Integrity Security under RUP (or INT-RUP security): An authenticated encryption scheme is said to provide *INT-RUP* security if the mode provides integrity even in scenarios when unverified plaintexts are released. INT-RUP security is particularly significant in lightweight applications, where often the memory buffer is quite limited. Along with the usage for small devices (smart-cards, RFID tags) with low buffer size, RUP security may be necessary due to lack of memory or high performance requirements (e.g. high speed, low latency, and long messages). In practice, it is useful in real-time streaming protocols, Smart cards, where block-wise decryption is required and plaintext can be released on the fly in order to reduce the end-to-end latency and/or compensate for low memory. Constructions such as OCB [11] and COPA [5] are insecure in INT-RUP setting. Constructions such as OCBIC [13] can be shown to achieve INT-CTXT security even in RUP setting. ELmD [8] uses a completely different approach of generating intermediate tags, at the cost of ciphertext expansion, to stop any release of unverified plaintext.

1.2 Structural Classification of AEAD Modes

It is not possible for AEAD scheme to satisfy all of the properties discussed in the previous section, as some of the properties are incompatible with each other. Therefore, an optimal trade-off for choosing some good properties is essential in the design of an AEAD mode of operation. Now, we classify the AEAD schemes in the following five classes:

1. Parallel Modes
2. Feedback based Modes
3. SIV Modes
4. Sponge Modes
5. Stream Cipher Modes

In the remainder of this paper, we discuss these classes in more detail. For each class, we give a brief description of their area of applications, discuss the basic design goals and features, and identify the second round candidates falling under that class.

2 Parallel Modes

An authenticated encryption mode is *parallel* if all the ciphertext blocks can be computed in parallel, allowing both hardware and software acceleration proportional to the available computational units. In parallel authenticated encryption modes, the inputs to the block ciphers depend on the plaintext, and not on the previous block cipher outputs or ciphertexts. Hence, *parallelizability* is achieved in the computation between *distinct block cipher calls*. For example Intel's Haswell CPU allows up to seven AES computations to be executed at once, and a parallel AEAD mode can easily exploit these resources.

2.1 Target Applications

- A parallel authenticated encryption mode can have a fully pipelined implementation. It reduces the latency and provides better performance in terms of speed. They are typically used in both *high-speed* hardware and commodity processors.
- The parallel design allows to efficiently process subsequent message blocks exploiting the *CPU pipeline* and *multi-threading* techniques.
- Practical use of parallel authenticated encryption includes memory encryption applications and vehicular security applications, where latency is critical and affects overall performance directly.
- Parallelizability is a rather inherent feature of hardware implementations, both in ASIC and FPGA. Also in general-purpose software, parallelizable encryption algorithms have profited in terms of performance due to the bitslice approach for a long time already.
- Useful in real-time streaming protocols (e.g. SRTP, SRTCP and SSH), where ciphertext/plaintext are released on-the-fly in order to reduce the end-to-end latency.

2.2 Basic Design Principle

The basic design principle for parallel authenticated encryption modes follow the *ECB* structure. To ensure security, some *additional masking state* is defined using the nonce and input block number. The nonce ensures privacy among different messages and the block number ensures privacy within a message. The typical choices are:

- Xor-Encrypt-Xor (XEX) paradigm, where a plaintext block is masked, encrypted and again masked to generate the corresponding ciphertext block.
- Encrypt the plaintext blocks using a tweakable block cipher with tweak defined as a pair (nonce, block number).

As speed and latency are the major factors in these designs, the tag generation should be done cheaply, and typically checksum of the plaintexts is used for it. The desired features are single pass, high rate (ideally rate 1), online, optimal. In addition, INT-RUP security is also useful for parallel modes, since one of their target area of application is real-time streaming protocols which requires on-the-fly decryption without waiting for the verification.

2.3 NIST Round 2 Candidates

1. Pyjamask: It is block cipher based construction and uses OCB mode of authenticated encryption which is an XEX based design. The mode is single-pass, online, rate 1, uses optimal number of block cipher calls. The construction provides 64-bit security, when the mode works on 128-bit blocks. However, it doesn't achieve RUP security.
2. SKINNY-AEAD: It is a tweakable block cipher based construction and uses plain ΘCB mode of authenticated encryption. The mode is online, rate 1, uses optimal number of tweakable block cipher calls. This construction also provides birthday bound security and doesn't achieve RUP security.
3. LOTUS-AEAD and LOCUS-AEAD: These two schemes use nonce-derived keys, which gives full n-bit security, where n denotes the block size. Hence, the mode works perfectly on 64-bit blocks. Moreover, use of two block cipher calls for each message blocks, and use of the intermediate values as checksum give the constructions RUP security.
4. PAEF (ForkAE): Uses a forkcipher with 128 bit block and 288 bit tweakey.

3 Feedback based Modes

A feedback function can be viewed as a linear function that takes a block cipher output and a plaintext block to produce the corresponding ciphertext block and an updated state which is used as the next block cipher input. Feedback based approach is becoming a very popular method of constructing lightweight block cipher based authenticated encryption. It is indeed a nice method to reduce the state memory, at the cost of losing parallelizability.

3.1 Target Applications

- Feedback based modes are primarily targeted for resource constrained environments such as RFID tags, sensor networks.
- Useful in applications with very stringent memory size criteria such as the authenticated encryption should fit into small hardware area and/or small code for 8-bit CPUs.

3.2 Basic Design Principles

In feedback based authenticated encryption modes, a feedback function is applied on the input data block, previous block cipher output, and the auxiliary secret state to generate the output data (ciphertext) block, next block cipher input and an updated auxiliary secret state. The auxiliary secret state is typically a function on N. The generic structure is depicted in Fig. 1. Here we list down all the existing feedback functions used in the

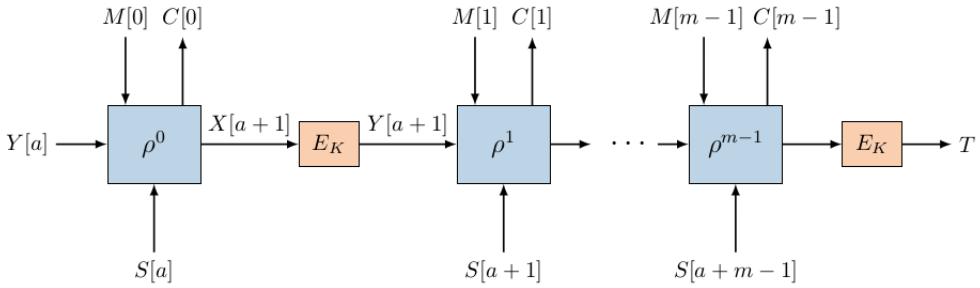

Figure 1: General Structure of Feedback-based Authenticated Encryption Modes.

literature:

- Plaintext Feedback (PFB). Here the plaintext itself is used as the next block cipher input.
- Ciphertext Feedback (CFB). Here the ciphertext is used as the next block cipher input.
- Output Feedback (OFB). Here the previous block cipher output is used as the next block cipher input.
- Combined Feedback (CoFB). This type of feedback uses a combination of previous block cipher output and the plaintext block to define the next block cipher input.
- Hybrid Feedback (HyFB). This type of feedback uses a hybrid combination of (PFB and CFB) or (PFB and OFB) or (CFB and OFB).

Following the work done in [6], the minimum size of the auxiliary secret state required for rate-1 authenticated encryption with different feedback functions are given below:

Encryption	Decryption	Additional states to achieve Security
PFB	CFB	n-bits
CFB	PFB	-
OFB	OFB	-
CoFB	CoFB	$n/2$-bits
HyFB (CFB+PFB)	HyFB (PFB+CFB)	$n/2$-bits
HyFB (CFB+OFB)	HyFB (PFB+OFB)	-
HyFB (PFB+OFB)	HyFB (CFB+OFB)	-

However, the requirement of the auxiliary state can be completely removed at the cost of rate, by keeping portion of each block cipher output secret and using it in the subsequent block cipher input.

Since area-efficiency is the primary goal, having low state size, inverse-free, similarity in encryption and AD processing, and similarity in encryption and decryption circuit are considered as the major features that a feedback based authenticated encryption should achieve.

3.3 NIST Round 2 Candidates

1. GIFT-COFB: Uses rate-1 feedback based mode and combined feedback (CoFB) as the underlying feedback. It uses 192 bit state and 256-bit xor operations per block.

2. HyENA: Uses rate-1 feedback based mode and PFB-CFB type hybrid feedback as the underlying feedback. As already pointed out, due to this hybrid type feedback, such construction uses the optimal 192 bit state and only 128-bit xor operations per block.

3. COMET: Uses rate-1 feedback based mode with combined feedback. However, the mode uses a nonce derived key that ensures higher security. COMET achieves minimal state size, in the sense that the only state it requires (apart from a constant number of bits) is used for the block cipher. This is the smallest possible state size for nonce-based AEAD schemes with security level comparable with COMET.

4. mixFeed: Uses hybrid feedback and nonce derived key to keep it single state.

5. Romulus-N: The overall structure of Romulus-N shares similarity in part with a (TBC-based variant of) block cipher mode COFB.

6. SAEF (ForkAE): Uses simple feedback based mode with fork cipher.

7. SAEAES: Reduces the additional masking state size at the cost of decreasing the rate. It uses a type of block-cipher based Sponge like modes where the most significant 64-bits of the block cipher output is used with the 64-bits message blocks to generate the ciphertext block and the most significant 64-bits of the next block cipher input. The lower part is kept secret and used in the next block cipher input. Thus, it reduces the state size to only 128-bits, while making the rate 1/2.

8. TinyJAMBU: Uses 128-bit state and a 128-bit keyed permutation with 32-bit message absorption in the state and squeezing the output from a different 32-bits of state. The remaining 64-bit is kept secret and used in the next block cipher input. Thus, it reduces the state size to only 128-bits, while making the rate 1/4. Use of keyed permutation ensures no additional cost due to key scheduling.

4 SIV Modes

SIV mode is a deterministic authenticated encryption mode, where the message and the associated data is processed to generate the tag, which is used as the IV for an IV-based encryption mode to generate the ciphertext. On the positive side, these modes do not require the nonce to generate the additional source of randomness, however they come at the cost of efficiency, as these modes process the plaintext blocks twice (once for authentication, then for encryption), and hence they are two-pass.

4.1 Target Applications

- Provides defense in depth, i.e., maximal robustness even in undesired environment such as when nonces are repeated.

- This mode supports an essential requirement in lightweight applications, i.e., efficient short input data processing, while minimizing the memory consumption and pre-computation. In use cases with tight requirements on delay and latency, the typical packet sizes are small (way less than 1 KB) as large packets occupy a link for longer duration, causing more delays to subsequent packets and increasing latency. For example, Zigbee [3], Bluetooth low energy [1], sensor network security protocols, e.g., TinySec [10], EPC tag [2] limits the maximum packet length. These protocols can benefit from SIV based modes.

- SIV is an excellent choice for energy efficient designs, another important design aspect of lightweight cryptography. Since low energy ciphers drain less battery, they are invaluable components of devices that operate on a tight energy budget such as handheld devices, medical implants or RFID tags.

4.2 Basic Design Principle

The basic design principle follows MAC-then-Encrypt paradigm. Typical choice is to use a single state message authentication followed by an IV-based encryption, where the tag is used as the IV. The basic structure of the mode is depicted in Fig. 2.

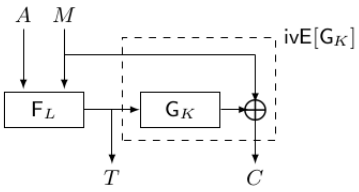

Figure 2: General Structure of SIV Authenticated Encryption Modes.

Keeping the efficiency in mind, priorities are given on making the authenticated encryption mode single state and single keyed. As efficient short data processing and energy efficiency are of prime importance, the optimal feature of deterministic authenticated encryption is very important.

4.3 NIST Round 2 Candidates

1. SUNDAE-GIFT: Single state (128-bit) block cipher based construction where CBC-MAC is followed by OFB mode of encryption. The construction achieves security even without nonce, however, it does not provide any security in RUP setting.

2. ESTATE: This can be viewed as a short-tweak tweakable block cipher (128-bit blocks, with 4-bit tweak) based variant of SUNDAE. The effective use of 4-bit tweaks for the purpose of domain separation ensures security in RUP setting, as well as makes the construction optimal in terms of the number of non-linear primitives invocation. This essentially makes it extremely efficient for short message processing and low buffer environment.

3. Romulus-M: This MRAE mode of operation that follows the structure of SIV and SCT. It is a tweakable block cipher based construction that uses 192-bit tweaks for 128 bit blocks and achieves full 128-bit security.

5 Sponge based Modes

Sponge is a symmetric-key modes of operations based on a fixed permutation as underlying primitive. Sponge based authenticated encryption modes typically uses the duplex construction that uses a fixed permutation to allow the absorption of the plaintext and squeeze the ciphertext at the same rate as the sponge construction. It is a simple and lightweight mode where neither key scheduling nor decryption algorithm implementation is required.

5.1 Target Applications

- Sponge constructions are well suited for achieving a balance between hardware cost and software efficiency.
- These modes are versatile in nature and they can be tuned to achieve good performance in any domain, including high speed implementation, memory-restricted environment and usual desktop computers.

5.2 Basic Design Principle

The basic design principle follows the duplex mode operation where at each step the upper layer (called the rate part of the sponge construction) of the previous permutation output is combined with the current plaintext block and generate the corresponding ciphertext block and the upper layer of the next permutation input. The lower layer of the previous permutation output is directly feeded into the lower layer of the next permutation input. The initial state is composed of the nonce and the key, and the key is injected directly into the state, without requiring any key scheduling. Typically the initialization and finalization overheads of sponge based constructions are much smaller as compared to the block cipher based modes. Additionally, use of keyed initialization as well as finalization ensures limited damage even in case of state recovery.

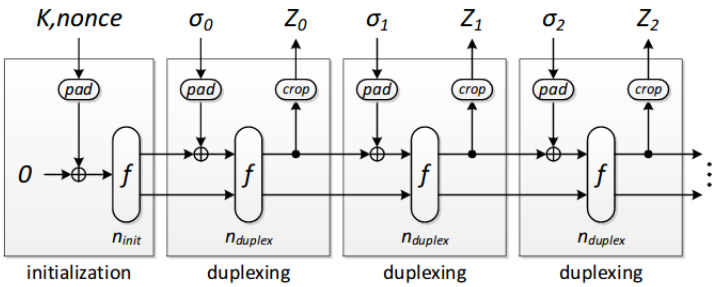

Figure 3: General Structure of Sponge Duplex Mode.

5.3 NIST Round 2 Candidates

1. ACE: Uses unified sponge duplex mode with keyed initialization and finalization phases. The mode uses sponge rate of 64-bits and sponge capacity of 256-bits, and it achieves security of 128-bit. It is designed to achieve a balance between hardware cost and software efficiency.

2. Ascon: Uses duplex modes of operation such as Monkey Duplex with a stronger keyed initialization and keyed finalization function. It also maintains a 320-bit state with sponge rate 64-bits and sponge capacity of 256-bits, and achieves security of 128-bit. The small state and simple round function are well-suited for small implementations, without compromising on the full security.

3. DryGASCON: Follows a variant of Duplex Sponge mode, it differs from the Duplex Sponge construction in the way it merge the input with the state and in the way it extract the output from the state, separating the concerns of cryptographic strength and robustness against physical attacks. The construction uses a 578-bit state with sponge rate 128-bits, and achieves security of 160-bit.

4. Gimli: Uses duplex modes of operation such as MonkeyDuplex with a stronger keyed initialization and keyed finalization function. It maintains a 384-bit state with sponge rate 128-bits and sponge capacity of 256-bits, and achieves security of 128-bit. Gimli is designed for energy-efficient and side-channel-protected hardware and for microcontrollers and for efficient short message processing with a high security level.

5. ISAP: Follows Encrypt-then-MAC mode operation and sponge based fresh re-keying concept to provide robustness against certain types of implementation attacks while allowing to add additional defense mechanisms at low cost. It maintains a 400-bit state with sponge rate 144-bits and sponge capacity of 256-bits, and achieves security of 128-bit.

6. KNOT: Uses Monkey Duplex sponge mode with a 256-bit state with sponge rate 64-bits and sponge capacity of 192-bits, and achieves security of 128-bit. Due to the bit-slice style, it allows for very efficient and flexible hardware and software implementations.

7. Orange-Zest: Uses a 256-bit permutation with full state absorption. The full state absorption is achieved using a part (128 bit) of the output of previous execution of the underlying permutation. This dynamic secret state is used to mask a part of the ciphertext. The overall state size remains 384 bits. This is a lightweight cipher that has optimum throughput, which can absorb message at the rate of the state of the permutation.

8. Oribatida: Uses a variant of the Monkey-wrap design that extends previous designs by a ciphertext-block masking that boosts the security and ensures INT-RUP security. The construction uses a 320-bit state with a 256-bit permutation that has 128-bit rate and guarantees security of 128-bits.

9. PHOTON-Beetle: Uses duplex-sponge mode with a feedback function ρ that boosts the security without any additional masking state. Beetle uses 256-bit state with sponge rate of 128-bit and achieves security of 121 bits.

10. Sparkle: Uses 384-bit state size with 192-bit rate, and a high security of 184 bits. The aim is to use as little CPU cycles as possible while maintaining strong security guarantees and a small implementation size.

11. Spix: Employs Monkey duplexed Sponge-based mode of operation which provides better throughput. It maintains a 256-bit state with sponge rate 64-bits and sponge capacity of 192-bits, and achieves security of 128-bit. The use of partial SPN 256-bit permutation that ensures small hardware footprint.

12. Spoc: Uses duplex-sponge mode with capacity being masked with the data blocks instead of rate which improves the security and allows larger rate value per permutation call. It uses 192-bit state with sponge rate of 64-bit and achieves security of 118 bits.

13. Spook: Uses duplex-sponge mode along with two additional tweakable block cipher (TBC) calls with 128-bit TBC state. Spook maintains a 512-bit state with both the rate and the capacity are same and that is 256-bits. Spook is primarily designed to provide security against side channel attacks but with high energy efficiency. The additional TBC calls ensure leakage resilience of Spook.

14. Subterranean: Uses duplex-sponge mode. It uses 257-bit state with sponge rate of 33-bit and achieves security of 128 bits. Subterranean is efficient for low-area and low-energy implementations in dedicated hardware.

15. Wage: Uses duplex-sponge mode. It uses 259-bit state with sponge rate of 64-bit and achieves security of 128 bits. The underlying permutation is based n the initialization phase of the Welch Gong stream cipher.

16. Xoodyak: Uses duplex-sponge mode extended with an interface that allows absorbing strings of arbitrary length, their encryption and squeezing output of arbitrary length. It uses 384-bit state with three different rates of 96-bit, 352-bit and 192-bits.

6 Stream Cipher Modes

6.1 Basic Design Principle

Stream Cipher modes of authenticated encryption use the principle of generating two key streams from a short key, and use keystream one for encryption and the other for authentication. Precisely, the mode uses an IV based stream cipher that takes as input a key and an IV and generates two keystreams: Encryption keystream and Authentication keystream. The encryption function `enc` simply adds the encryption keystream to the message stream to generate a ciphertext stream. The authentication module `auth` takes the authentication key stream, message stream and ciphertext stream to generate a tag. Popular choices for `auth` can be universal hash, shift register, permutation, block cipher modes and others. A well known example of stream cipher based authenticated encryption is the combination of ChaCha stream cipher and Poly1305 MAC used in the TLS protocol suite.

6.2 Target Applications

- Stream cipher based designs are specifically used to speed up the data process as well as to have low circuit complexity. More precisely, these designs target to achieve high area efficiency maintaining a high speed.

- These designs are the best choice for applications where plaintext comes with unknown length like a secure wireless connection. In addition, stream cipher based designs are excellent choice to process long messages.

- Another application can be military cryptography such that the keystream can be generated separately and put under strict security measure and fed to other devices to perform one time pad.

6.3 NIST Round 2 Candidates

- Elephant: The keystreams are generated in a Xor-Permute-Xor mode on the IV, where the master key is used to generate the masking state. The auth function uses a permutation module to process a message, an associated data and the master key.
- Grain-128 AEAD: Grain-128 AEAD adopts the design of Grain-128 and Grain v1 and extends it for authentication. It processes an authentication keystream with a shift register and the shift register outputs are processed with the associated data and message.
- Saturnin: Saturnin uses a tweakable block cipher cascade construction to process the associated data and the message.

References

[1] Bluetooth low energy. http://www.bluetooth.com/Pages/Low-Energy.aspx/.

[2] Electronic product code (epc) tag data standard (tds). http://www.epcglobalinc.org/standards/tds/.

[3] Zigbee alliance. http://www.zigbee.org.

[4] Elena Andreeva, Andrey Bogdanov, Nilanjan Datta, Atul Luykx, Bart Mennink, Mridul Nandi, Elmar Tischhauser, and Kan Yasuda. COLM v1. CAESAR Competition.

[5] Elena Andreeva, Andrey Bogdanov, Atul Luykx, Bart Mennink, Elmar Tischhauser, and Kan Yasuda. AES-COPA v.2. Submission to CAESAR. 2015. https://competitions.cr.yp.to/round2/aescopav2.pdf.

[6] Avik Chakraborti, Nilanjan Datta, Ashwin Jha, Snehal Mitragotri, and Mridul Nandi. From combined to hybrid: Making feedback-based AE even smaller. *IACR Trans. Symmetric Cryptol.*, 2020(S1):417–445, 2020.

[7] Avik Chakraborti, Nilanjan Datta, and Mridul Nandi. On the optimality of non-linear computations for symmetric key primitives. *J. Mathematical Cryptology*, 12(4):241–259, 2018.

[8] Nilanjan Datta and Mridul Nandi. Proposal of ELmD v2.1. Submission to CAESAR. 2015. https://competitions.cr.yp.to/round2/elmdv21.pdf.

[9] Morris Dworkin. Recommendation for block cipher modes of operation: Galois/counter mode (GCM) and GMAC. NIST Special Publication 800-38D, 2011. csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf.

[10] Chris Karlof, Naveen Sastry, and David Wagner. Tinysec: A link layer security architecture for wireless sensor networks. In *Proceedings of Embedded Networked Sensor Systems*, SenSys '04, pages 162–175. ACM, 2004.

[11] Ted Krovetz and Phillip Rogaway. The Software Performance of Authenticated-Encryption Modes. In *FSE*, pages 306–327, 2011.

[12] Phillip Rogaway and Thomas Shrimpton. A Provable-Security Treatment of the Key-Wrap Problem. In *EUROCRYPT*, pages 373–390, 2006.

[13] Ping Zhang, Peng Wang, Honggang Hu, Changsong Cheng, and Wenke Kuai. INT-RUP security of checksum-based authenticated encryption. In *Provable Security - 11th International Conference, ProvSec 2017, Xi'an, China, October 23-25, 2017, Proceedings*, pages 147–166, 2017.