J Inf Process Syst, Vol.17, No.2, pp.253~270, April 2021
https://doi.org/10.3745/JIPS.03.0156

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

NIST Lightweight Cryptography Standardization Process: Classification of Second Round Candidates, Open Challenges, and Recommendations

Dennis Agyemanh Nana Gookyi*, Guard Kanda*, and Kwangki Ryoo*

Abstract

In January 2013, the National Institute of Standards and Technology (NIST) announced the CAESAR (Competition for Authenticated Encryption: Security, Applicability, and Robustness) contest to identify authenticated ciphers that are suitable for a wide range of applications. A total of 57 submissions made it into the first round of the competition out of which 6 were announced as winners in March 2019. In the process of the competition, NIST realized that most of the authenticated ciphers submitted were not suitable for resource-constrained devices used as end nodes in the Internet-of-Things (IoT) platform. For that matter, the NIST Lightweight Cryptography Standardization Process was set up to identify authenticated encryption and hashing algorithms for IoT devices. The call for submissions was initiated in 2018 and in April 2019, 56 submissions made it into the first round of the competition. In August 2019, 32 out of the 56 submissions were selected for the second round which is due to end in the year 2021. This work surveys the 32 authenticated encryption schemes that made it into the second round of the NIST lightweight cryptography standardization process. The paper presents an easy-to-understand comparative overview of the recommended parameters, primitives, mode of operation, features, security parameter, and hardware/software performance of the 32 candidate algorithms. The paper goes further by discussing the challenges of the Lightweight Cryptography Standardization Process and provides some suitable recommendations.

Keywords

Authenticated Encryption, CAESAR, IoT, Lightweight Cryptography, NIST

1. Introduction

The Fourth Industrial Revolution [1] commonly termed as Internet-of-Things (IoT) is changing the human living conditions like never before. IoT is built on the foundation laid by the first, second, and third industrial revolutions which include the invention of the steam engine, electricity, and the emergence of digital technology respectively. IoT is primarily made up of connected devices that gather, store, and transmit sensitive consumer information over mostly insecure channels [2]. A simple example of IoT is the automatic gathering of electricity usage information from smart meters at homes by utility companies. This is useful because utility companies no longer have to go through the long tedious process of sending workers to manually read electricity meters. Though this is beneficial, it also opens up a

Manuscript received March 24, 2020; first revision April 23, 2020; second revision May 30, 2020; accepted June 14, 2020.

Corresponding Author: Kwangki Ryoo (kkryoo@gmail.com)

* Dept. of Information and Communication Engineering, Hanbat National University, Daejeon, Korea (dennisgookyi@gmail.com, kandaguard@gmail.com, kkryoo@gmail.com)

number of threat scenarios. An adversary could monitor the power usage in a home to plan a heist during the absence of the occupants.

In this era of IoT, secured communication is crucial. There is a need to trust devices and communication channels with two main issues: confidentiality and authentication. Confidentiality protects the contents of messages using encryption protocols while authentication ensures that messages are not forged using protocols like hash algorithms. A cryptographic protocol that provides both confidentiality and authentication is known as authenticated encryption (AE) [3]. An AE consists of both an encryption and a decryption process. The encryption process produces a ciphertext and a tag using plaintext and a secret key. The decryption process uses the tag, the ciphertext, and a secret key to produce the plaintext if and only if the tag passes a verification process. Real-world applications such as network protocols consist of secret (the message) and non-secret (address) information. For that matter, it is essential to encrypt only the secret information and authenticate both the secret and non-secret information. This can only be achieved using the notion of authenticated encryption with associated data (AEAD) [4] where the non-secret information is termed as the associated data.

Due to the growing demand for AEAD schemes, the National Institute of Standards and Technology (NIST) in January 2013 announced the CAESAR (Competition for Authenticated Encryption: Security, Applicability, and Robustness) competition [5] seeking AE schemes that are convenient for widespread adaptation. The competition received 57 submissions and after a rigorous three-round process of reviewing, analyzing, and comparisons, 6 submissions were declared winners and standardized in March 2019 [6]. Midway through the CAESER completion, NIST realized that most of the submitted schemes were not suitable for resource-constrained devices like radio-frequency identification (RFID) tags, sensors, and smart cards. NIST, therefore, issued a call for lightweight AEAD schemes with optional hashing algorithms [7] in August 2018 as part of the Lightweight Cryptographic Standardization Process [8]. The call received a total of 57 submissions out of which 56 were selected as round-one candidates in April 2019 [9]. In August 2019, 32 out of the 57 submissions made it into the second round of the competition which is estimated to end in the year 2021 [10].

This work provides a comparative analysis of the 32 candidate algorithms that made it into the second round of the lightweight AEAD standardization process. The contributions of this paper are as follows:

- This work summarizes the NIST submission requirements for the lightweight AEAD schemes and the evaluation criteria.
- The 32 second-round candidates are classified according to their underlying design approaches which include block cipher-based, tweakable block cipher-based, stream cipher-based, and permutation-based.
- The 32 candidates are compared in terms of their security parameters, mode of operation, features, and hardware/software resource utilization.
- The work goes further by discussing some challenges with the lightweight cryptography standardization process and offers some useful recommendations.

The rest of this paper is organized as follows: Section 2 highlights some related works, Section 3 discusses the NIST lightweight AEAD submission requirements and some evaluation criteria, Section 4 gives the metrics used to compare the 32 candidate algorithms, Section 5 details the comparative analysis of the candidates, Section 6 discusses some open challenges with the lightweight standardization process and offers some recommendations, Section 7 ends the paper with the conclusion and future direction of the work.

2. Related Work

NIST's search for lightweight AEAD schemes was initiated in August 2018. A total of 56 out of 57 candidates made it into the first round of the competition of which 32 were selected as round-two candidates in August 2019. Though the competition has been ongoing for more than a year, survey works that give a comparative analysis of the candidate algorithms do not exist. To the best of our knowledge, the only paper that comes close to this work was published in November 2019 by Rezvani and Diehl [11]. The paper implemented three of the candidate algorithms in the Artiz-7 hardware platform. The three candidates include GIFT-COFB, Spook, and SpoC. The results from the hardware implementation indicate that SpoC consumed the least hardware resources while GIFT-COFB achieved the highest throughput.

The lightweight AEAD competition sprang from the CAESAR competition which ended in March 2019 with 6 submissions selected as winners. The CAESAR competition lasted for almost 5 years with only four research papers that gave a comparative analysis of the candidate algorithms. This section, therefore, examines the four research papers that surveyed the CAESAR competition and use them as a framework for the survey of the lightweight AEAD competition.

The first survey paper on the CAESAR competition was published in 2016 by Abed et al. [12]. The paper provided a very comprehensive and easy-to-grasp overview of CAESAR competition first-round candidates. The candidates were classified by their design approach which included stream cipher-based, block cipher-based, permutation/sponge-based, compression function-based, and dedicated structure-based. The paper also compared the candidates in terms of their functional features, security parameters, and their robustness.

The second survey work on the CAESAR competition candidates was published in 2017 by Kavun et al. [13]. The paper presented a comprehensive survey of hardware performance of various AE schemes which include the encrypt-then-MAC scheme, block cipher-based modes, and permutation-based schemes. The work reported the area, throughput, and power consumption of each scheme. The research revealed that the permutation-based candidates performed better as compared to the other schemes.

The third research paper that surveyed the CAESAR competition candidates was published in 2018 by Zhang et al. [14]. The paper summarized the requirements of the CAESAR competition and the progress of the candidates. The candidate schemes in the final round were grouped according to their design features and encryption modes. The paper ends with some trends of design together with future analysis of AE schemes.

The fourth survey paper on the CAESAR competition was published in 2019 by Agrawal et al. [15]. This work surveyed lightweight AE schemes that were proposed after the year 2010 which is, therefore, a new direction in the field of lightweight cryptography. The paper surveyed a total of 17 AE schemes out of which 9 schemes were part of the CAESAR competition. The paper reported the performance of the lightweight AE schemes which included hardware and software metrics.

The security features of the CAESAR candidates extracted from the four survey papers [12-15] include the following:

- All the CAESAR candidates accept as inputs a public number known as the nonce together with the secret key, the associated data, and the message. The output consists of the tag and the ciphertext. The nonces have bit lengths of not less than 64 while that of the secret keys is no less than 80. The message, ciphertext and the associated can have variable lengths divided into blocks of no less than

64 bits.

- The nonces are assumed to be unique with every encryption call.
- Most of the candidates claim to have security guarantees (both confidentiality and integrity) of up to the birthday-bound. The birthday-bound refers to the processing of 2^{64} bits of data. This means that one secret key can be used to process data less than 2^{64} blocks with the encryption calls limited to 2^{64}. Going beyond the birthday-bound, the security of the candidates could be compromised.

The four survey papers on the CAESAR competition were used as a framework for this work. This paper classifies the schemes in the second round of the NIST lightweight AEAD competition into block cipher-based, tweakable block cipher-based, stream cipher-based, and permutation-based. The schemes are then compared in terms of their security parameters, features, modes of operation, primitive cryptographic algorithms, and hardware/software performance metrics. The work discusses some open challenges and gives some useful recommendations. This paper provides useful information for researchers in the cryptographic community as well as the lightweight AEAD competition committee. This work assists cryptographers in the AEAD community to select their favorite candidate based on the security parameters, primitives, and implementation results.

3. NIST Lightweight AEAD Competition Requirements

The growth and evolution of the IoT inspired NIST to dedicate a competition for selecting lightweight AEAD algorithms for resource-constrained devices. A number of IoT applications are deployed in various industries which include health care, environment monitoring, supply chain, transportation, surveillance, and home support.

3.1 NIST Lightweight AEAD Competition Target Devices

For the past decade, cryptographic algorithms are classified into conventional cryptography and lightweight cryptography. The target devices for conventional cryptographic algorithms include servers, desktops, tablets, and smartphones. These devices are high end with a large amount of storage space and high processing power. Due to the high memory capacity, high processing speed, and high-power consumption of the high-end devices, lightweight cryptography is unnecessary in these devices. The era of IoT has introduced several low-cost devices into networks. These low-cost devices are usually constrained in terms of memory capacity, processing speed, and power consumption. The low-cost devices are the target for the NIST lightweight AEAD competition. Conventional cryptography cannot be implemented in low-cost devices such as embedded systems, RFID, and sensor networks. Embedded systems consist of a microcontroller that has extremely constrained random-access memory (RAM) and read-only memory (ROM). Some have as little as 16 bytes of RAM. RFID and sensor network devices are usually implemented as application specific integrated circuit (ASIC) to meet their ultra-constrained nature. Some RFID tags are not powered by batteries and draw a limited amount of power from their environment. For such a device, cryptographic algorithms should not only use fewer gate counts but also consume a small amount of power. Saarinen and Engels [16] listed the constraints of devices like the RFID tag for cryptographic applications.

3.2 NIST Lightweight AEAD Competition Profile

In April 2017, NIST published the profiles for the lightweight cryptography standardization process [17]. The profile is a group of engineering requirements that are made up of: the functionality that is provided by an algorithm; the physical characteristics of the environment that the algorithm is implemented; the performance characteristics (power consumption, latency, and throughput) of an algorithm; and the security characteristics of an algorithm (security strength and attack models). The objective of the profile is to state the requirements that submitted algorithms must meet before they are considered for evaluation and subsequently standardized. The profile for lightweight AEAD for constrained hardware devices is summarized in Table 1.

NIST clearly stated the evaluation criteria for the lightweight AEAD competition [18]. The first evaluation criteria are that the submissions must meet the minimum requirements stated in Table 1. The second evaluation criteria are regarding cost. The submission will be evaluated using various cost metrics which include energy consumption, memory, and area. The third evaluation criteria are regarding performance. Performance metrics such as power consumption, latency, and throughput will be used to evaluate the submission. The fourth evaluation criteria have to do with a third-party analysis of the submissions. Submissions that include several third-party analyses are highly favored. The last evaluation criteria are the suitability of the submissions in hardware and software platforms.

Table 1. Profile for NIST lightweight AEAD competition

Engineering requirement	Description
Functionality	Authenticated encryption with associated data (AEAD)
Design goals	Performs better in resource constrained environments Short messages of 8 bytes should be supported The length of the message should be an integer number of bytes
Physical characteristics	The target should be toward resource constrained hardware platforms Very compact hardware implementations should be achievable
Performance characteristics	A wide range of vendors and standard cells should be considered when reporting hardware resource utilizations Very high flexibility to enable different implementation strategies The initial computation of the key should be efficient
Security characteristics	Key, nonce, and tag lengths of not less than 128-bit should be supported The minimum length of the plaintext/associated data should be up to 2^{50}-1 bytes Attacks should require a minimum of 2^{112} computations

4. Lightweight AEAD Round-Two Classification Metrics

Out of the 56 candidates that were selected as round-one candidates for the NIST lightweight AEAD standardization process, 32 made it into the second round of the competition. The 32 candidates are listed in Table 2. It is important to note that a total of 111 researchers from 18 countries contributed to the development of the 32 algorithms. Only one researcher's country of origin is unknown at the time of writing this paper.

Table 2. Round-two candidates for the NIST lightweight AEAD competition [10]

Candidates
ACE, ASCON, COMET, DryGASCON, Elephant, ESTATE, ForkAE, GIFT-COFB, Gimli, Grain-128AEAD, HyENA, ISAP, KNOT, LOTUS-AEAD, mixFeed, ORANGE, Oribatida, PHOTON-Beetle, Pyjamask, Romulus, SAEAES, Saturnin, SKINNY-AEAD, SPARKLE, SPIX, SpoC, Spook, Subterranean 2.0, SUNDAE-GIFT, TinyJAMBU, WAGE, Xoodyak

This work overviews and classifies the 32 candidates. For the classification process, it is important to define the metrics used to classify the candidates. This section, therefore, defines six classification metrics used to group the 32 candidates that made it into the second round of the NIST lightweight AEAD standardization process. The 6-classification metrics include general parameters, underlying construction/ primitives, modes of operation/design structure, functional features, security parameters, and hardware/ software resource utilization.

- **General parameters**: The general parameters used by all the NIST lightweight AEAD submissions include the key, nonce, and the tag. A key in cryptography is a string made up of bits that is used by an algorithm to convert the plaintext into ciphertext and vice versa. In cryptography, to ensure secure communication between two entities, the key must remain private. The security of a cryptographic algorithm is directly dependent on the security and length of the secret key. A nonce in cryptography is a value that is unique. It can be generated using a counter (until it overflows) or using a random number generator (previous values need to be stored in other to detect repetitions). When generating a nonce, it is crucial to ensure that it never repeats itself. A tag in AEAD is used to ensure that the ciphertext or associated data has not been modified. If either the ciphertext or associated data is modified, the computation of the validation tag at the receiving end will result in a completely different value. If the tag from the sender and the tag generated by the receiver does not match, it indicates that either the ciphertext or associated data has been modified.
- **Underlying construction/primitives**: The underlying construction/primitive metric examines the cryptographic primitives used to construct the lightweight AEAD schemes. A primitive is simply a standard cryptographic algorithm employed by the submission. The round two lightweight AEAD schemes are classified by their underlying constructions which include block cipher-based, tweakable block cipher-based, stream cipher-based, and permutation-based. The block cipher-based candidates employ standard block ciphers in the construction of their AEAD schemes. A block cipher is a keyed permutation that transforms a fixed-length plaintext into a fixed-length ciphertext using a secret key. The tweakable block cipher-based candidates use standard tweakable block ciphers in the construction of their AEAD schemes. A tweakable block cipher is the same as a block cipher with additional inputs known as tweaks or tweakey. A tweak is used as an initialization vector while a tweakey is a combination of the secret key and the tweak. The stream cipher-based candidates employ standard stream ciphers in the construction of their AEAD schemes. A stream cipher is designed using a pseudo-random number generator that generates the keystream. To generate the ciphertext, the plaintext is XORed with the keystream one bit at a time. The permutation-based candidates employ standard permutation algorithms in the construction of their AEAD schemes. A permutation algorithm is a key-less bijective mapping on fixed-length inputs.

- **Modes of operation/design structure**: The mode of operation is defined as the encryption mode when the message is larger than the block size of a cipher. When the message or plaintext is larger than the block size of a cipher, different encryption modes are employed to ensure that the ciphertext is safe. This is usually used by the block cipher-based candidates. The design structure describes the primitives used by the stream cipher and permutation-based candidates. The round two submissions will be classified by their modes of operation and design structures.
- **Functional features**: Each of the round two candidates presents some special features that enhance the performance of the AEAD scheme. Here, the round two candidates are classified using features such as pass, parallelizable encryption/decryption, online, and inverse-free. The pass feature consists of a one-pass scheme and a two-pass scheme. In a one-pass scheme, the message together with the associated data is processed just once to generate the ciphertext and the tag. In a two-pass scheme, the message is processed twice to produce the ciphertext in the first process and the tag in the second process. An AEAD scheme is parallelizable if and only if the processing of the **i**-th input block is independent on the processing of the **j**-th input block where $\mathbf{i} \neq \mathbf{j}$. Encryption and decryption can be done in parallel. An AEAD scheme is considered online if the encryption of the **i**-th block of message $\mathbf{M_i}$ is directly dependent on the previous blocks block $\mathbf{M_1}\ldots\mathbf{M_{i-1}}$. An AEAD scheme that is not online is considered as an offline scheme. An AEAD scheme is said to be inverse-free if the standard primitive is in one direction only. For example, in a block cipher based AEAD scheme, only the encryption operation is needed while the decryption operation is not used. An inverse-free scheme is preferable for lightweight applications because it requires less memory and area.
- **Security parameters**: An AEAD scheme is secured if it satisfies two security notions which include privacy or confidentiality and authenticity [3]. The privacy or confidentiality is defined in terms of indistinguishability from a random oracle while the authenticity describes an adversary's ability to generate valid ciphertext and tag pairs that have not been generated by the encryption oracle.
- **Hardware/software results**: The resources used for software implementation of the AEAD schemes are measured using metrics such as the number of registers and memory usage. For low-cost devices, software implementation results are not as important as hardware implementation results. This is simply because the first step in designing lightweight algorithms is to make sure that the algorithm fits into constrained devices [19]. The hardware results are measured in terms of look-up tables (LUTs) for field programmable gate array (FPGA) designs and gate equivalents (GE) for ASIC designs. Not all the second round lightweight AEAD submissions reported their hardware results.

5. Overview of NIST Lightweight AEAD Candidates

This section discusses the 32 candidate algorithms that made it into the second round of the NIST lightweight AEAD standardization process. The candidates are categorized based on their underlying constructions which include block cipher-based, tweakable block cipher-based, stream cipher-based, and permutation-based. The comparison is made among the candidates in terms of their general parameters, mode of operation/design structure, features, security parameters, and hardware resource utilization. Table 3 gives a summary of the NIST lightweight AEAD candidates that made it into the second round of the competition.

Table 3. Classification of NIST lightweight AEAD candidates

Underlying construction	Candidates	Parameters			Primitives	Mode/ Structure	Features				Security		Hardware results
		Key (bits)	Nonce (bits)	Tag (bits)			Parallelizable Enc/Dec	Online	Pass	Inverse-free	Confidentiality	Integrity	
Block cipher based	Comet	128	64/128	64/128	AES	CTR/Beetle	√/√	√	1	√	2^{64}	2^{64}	
	GIFT-COFB	128	128	128	GIFT	COFB	√/√	√	1	√	2^{64}	2^{58}	√
	HYENA	128	96	128	GIFT	HYFB (CFB/PFB)	√/√	√	1	√	2^{64}	2^{58}	
	mixFeed	128	120	128	AES	mixFeed	√/√	√	1	√	2^{60}	2^{50}	
	Pyjamask	128	128	128	Pyjamask	OCB	√/√	√	1	√	2^{64}	2^{64}	
	SAEAES	128	64	64	AES	SAEB	√/√	√	1	√	2^{62}	2^{58}	√
	SUNDAE-GIFT	128	96	128	GIFT	SUNDAE	/		2		-	-	√
	TinyJambu	128	96	64	XOR	JAMBU	√/√	√	1	-	2^{64}	2^{128}	√
	Saturnin	128	128	128	SATURNIN	CTR/Cascade	√/√	√	2	√	2^{128}	2^{128}	
Tweakable block cipher based	ForkAE	128	104	128	SKINNY	PAEF	√/√	√	2		2^{112}	2^{116}	
	ESTATE	128	128	128	TweGIFT TweAES	FCBC/OFB	√/√	√	1	√	2^{64}	2^{64}	√
	LOTUS-AEAD	128	128	64	TweGIFT	OTR	√/√	√	1	√	2^{64}	2^{64}	√
	Romulus	128	96/128	128	SKINNY	COFB	/		1	√	2^{128}	2^{128}	√
	Spook	128	128	128	CLYDE	S1P	√/√	√	1		2^{121}	2^{121}	
	SKINNY-AEAD	128	128	128	SKINNY	OCB3	√/√	√	1	√	2^{128}	2^{128}	√
Stream cipher based	Grain-128AEAD	128	96	64	Grain-128a	LFSR/NFSR	/		1	√	2^{110}	2^{95}	√
Permutation based	ACE	128	128	128	Simeck	Duplex	√/√	√	1	√	2^{128}	2^{128}	√
	ASCON	128	128	128	ASCON-p	Duplex	√/√	√	1	√	2^{128}	2^{128}	
	DryGASCON	128	128	128	ASCON-p	Drysponge	√/√	√	1	√	2^{128}	2^{128}	√
	Elephant	128	200	128	Keccak-p	CTR	√/√	√	1	√	2^{127}	2^{127}	
	Gimli	128	128	128	GIMLI-24	Duplex	√/√	√	1	√	2^{128}	2^{128}	√
	ISAP	128	128	128	Keccap-p ASCON-p	Sponge	√/√	√	1	√	2^{128}	2^{128}	√
	KNOT	128	64	64	RECTANGLE	Duplex	√/√	√	1	√	2^{125}	2^{125}	√
	ORANGE	128	128	128	PHOTON	Sponge	√/√	√	1	√	2^{128}	2^{128}	
	Oribatida	128	64/128	96/128	SimP	Duplex	√/√	√	1	√	2^{121}	2^{121}	√
	PHOTON-beetle	128	128	128	PHOTON	Beetle	√/√	√	1	√	2^{128}	2^{121}	
	SPARKLE	128	256	128	SPARX	Beetle	√/√	√	1	√	2^{120}	2^{120}	
	SPIX	128	128	128	sLiSCP-light	Duplex	√/√	√	1	√	2^{128}	2^{128}	√
	SpoC	128	128	64/128	sLiSCP-light	Beetle	√/√	√	1	√	2^{112}	2^{112}	√
	Subterranean 2.0	128	128	128	Subterranean	Duplex	/	√	1		2^{92}	2^{92}	√
	WAGE	128	128	128	Welch-Gong	Duplex	√/√	√	1	√	2^{128}	2^{128}	√
	Xoodyak	128	128	128	XOODOO	Duplex	-	-	1	√	2^{160}	2^{160}	

5.1 Underlying Construction of NIST Lightweight AEAD Candidates

The 32 NIST lightweight AEAD candidates are categorized into four groups based on their underlying construction which include block cipher-based, tweakable block cipher-based, stream cipher-based, and permutation-based.

- **Block cipher-based candidates**: Out of the 32 candidates that made it into the second round of the NIST lightweight AEAD competition, 9 of them were designed using block ciphers. The block cipher-based candidates include Comet, GIFT-COFB, HYENA, mixFEED, Pyjamask, SAEAES, SUNDAE-GIFT, TinyJambu, and Saturnin. From Table 3, all the candidates used a key size of 128-bit with the nonce and tag sizes ranging from 64-bit to 128-bit. In terms of the parallelizability and online features, only SUNDAE-GIFT does not provide those features while the other candidates do. Again, only SUNDAE-GIFT and Saturnin use a two-pass scheme while the other candidates use a one-pass scheme. In terms of the inverse-free feature, SUNDAE-GIFT is not inverse free while TinyJambu did not state whether it was inverse-free or not. The rest of the candidates are all inverse-free. In terms of security (confidentiality and integrity), the required computation ranges from 2^{50} to 2^{128}. Only SUNDAE-GIFT did not provide its security parameters. Only 4 out of the 9 candidates provided some hardware resource utilization results.
- **Tweakable block cipher-based candidates:** A total of 6 tweakable block cipher-based candidates made it into the second round of the NIST lightweight AEAD competition. They include forkAE, ESTATE, LOTUS-AEAD, Romulus, Spook, and SKINNY-AEAD. The candidates consist of a key size of 128-bit, nonce sizes that range from 96-bit to 128-bit, and tag sizes that range from 64-bit to 128-bit. The primitives used by the candidates include SKINNY, TweGIFT, TweAES, and CLYDE. Only Romulus does not provide the parallelizable and online features while the rest of the candidates do. All the candidates are inverse-free except forkAE and Spook. Only forkAE operates on a two-pass scheme while the rest of the candidates operate on a one-pass scheme. SKINNY-AEAD achieves the highest security level of 2^{128} for confidentiality and integrity while ESTATE and LOTUS-AEAD achieve the lowest (2^{64} for both security parameters). Spook and forkAE do not provide hardware resource utilization while the rest of the candidates do.
- **Stream cipher-based candidates:** Only one stream cipher-based candidate made it into the second round of the NIST lightweight AEAD competition. The candidate is known as Grain-128AEAD with a 128-bit key, 96-bit nonce, and 64-bit tag. The candidate uses Grain-128a primitive and provides features such as online, parallelizable, and inverse-free with a one-pass scheme. The candidate achieves confidentiality and integrity security levels of 2^{110} and 2^{95} respectively and also provides hardware utilization results.
- **Permutation based candidates:** The permutation-based construction provided the highest number of candidates in the second round of the NIST lightweight AEAD competition. The candidates are 16 in total and include ACE, ASCON, DryGASCON, Elephant, Gimli, ISAP, KNOT, ORANGE, Oribitida, PHOTON-beetle, SPARKLE, SPIX, SpoC, Subterranean 2.0, WAGE, and Xoodyak. The candidates use a key size of 128-bit, nonce sizes that range from 64-bit to 256-bit, and tag sizes that range from 64-bit to 128-bit. The primitives used by the candidates include Simeck, ASCON-p, Keccak-p, GIMLI-24, RECTANGLE, PHOTON, SimP, SPARX, sLiSCP-light, Subterranean, Welch-Gong, and XOODOO. Xoodyak did not state whether it provided features such as online

and parallelizable. The rest of the candidates provide those features except Subterranean 2.0 which is not parallelizable. All the candidates use a one-pass scheme and are also inverse-free except for Subterranean 2.0. Xoodyak achieves the highest security level of 2^{160} for confidentiality and integrity while Subterranean 2.0 achieves the lowest security level of 2^{92} for both parameters. A total of 6 out of the 16 candidates do not provide hardware utilization results.

5.2 NIST Lightweight AEAD Candidates Primitives

The 32 candidate schemes that made it into the second round of the NIST lightweight AEAD competition make use of four main cryptographic primitives which include block ciphers, tweakable block ciphers, stream ciphers, and permutation algorithms. Most of the primitives are well known and cryptographically secured. Table 4 summarizes the primitive algorithms used by the NIST lightweight AEAD candidates.

- **Block cipher-based primitives:** The 9-block cipher-based candidates use 4 block ciphers in their schemes which include the Advanced Encryption Standard (AES), GIFT, Pyjamask, and Saturnin. All the block ciphers make use of the substitution permutation network (SPN) structure with key sizes that range from 128-bit to 256-bit. The plaintext and ciphertext block sizes range from 64-bit to 256-bit. The AES 128-bit key size uses the least number of rounds (10) while the number of rounds for GIFT can go as high as 40 depending on the key size. AES uses the most memory (2048 bits for encryption and decryption) since it consists of 16×8 input/output substitution boxes (S-Box) for encryption and decryption. GIFT uses the least memory of just 64 bits.
- **Tweakable block cipher-based primitives:** A total of 4 tweakable block cipher primitives are used by the 6 tweakable block cipher-based candidates. They include SKINNY, TweAES, TweGIFT, and CLYDE. TweGIFT, TweAES, and CLYDE make use of a key size of 128-bit while that for SKINNY is dependent on the block size. All the ciphers use the SPN structure. The tweak sizes range from 4-bit to 128-bit, while the plaintext and ciphertext block sizes range from 64-bit to 128-bit. TweAES make use of the least number of rounds (10) while the number of rounds for SKINNY can go as high as 56 depending on the cipher block size. SKINNY and TweAES use the most memory (2040 bits) while TweGIFT and CLYDE use the least memory of just 64 bits.
- **Stream cipher-based primitives.** The stream cipher-based candidate makes use of the well-known Grain-128a primitive. The Grain-128a stream cipher is made up of linear feedback shift register (LFSR) and non-linear feedback shift register (NLFSR) structure. The key size of the cipher is 128-bit with an initialization vector of 96-bit. The cipher can use either 256 or 320 rounds.
- **Permutation-based primitives:** The 16 permutation-based candidates make use of 12 permutation algorithms which include Simeck, XOODOO, Welch-Gong, Subterranean, sLiSCP-light, SPARX, PHOTON, SimP, RECTANGLE, Keccap-p, ASCON-p, and GIMLI-24. Simeck, Welch-Gong, Subterranean, SPARX, and RECTANGLE have key sizes that range from 64-bit to 256-bit while others do not make use of a key. The permutation sizes of the algorithms range from 25-bit to 1600-bit with input/output block sizes that range from 32-bit to 1600-bit. Only PHOTON and RECTANGLE make use of S-Box memory which takes up 64 bits and 1024 bits, respectively.

Table 4. NIST lightweight AEAD candidates' primitives

Construction	Primitive	Parameters (bits)						Structure	Rounds	S-Box memory (bits)	
		Key	Tweak	Initial vector	Permutations	Plaintext	Ciphertext			Encryption	Decryption
Block cipher	AES	128/192/256	-	-	-	128	128	SPN	10/12/14	2048	2048
	GIFT	128	-	-	-	64/128	64/128	SPN	28/40	64	64
	PYJAMASK	128	-	-	-	96/128	96/128	SPN	14	32/64	32/64
	SATURNIN	128	-	-	-	256	256	SPN	{0, …, 31}	128	128
Tweakable block cipher	SKINNY	n, 2n, 3n (n = block)	-	-	-	64/128	64/128	SPN	n = 32/40 2n = 36/48 3n = 40/56	128/2048	128/2048
	TweAES	128	4	-	-	128	128	SPN	10	2048	2048
	TweGIFT	128	4	-	-	64/128	64/128	SPN	28/40	64	64
	CLYDE	128	126	-	-	128	128	SPN	12	64	64
Stream cipher	Grain-128a	128	-	96	-	-	-	LFSR/NLFSR	256/320	-	-
Permutation	Simeck	64/96/128	-	-	-	32/48/64	32/48/64	Feistel	32/36/44	-	-
	XOODOO	-	-	-	384	128/352	-	Cyclist	12 (default)	-	-
	Welch-Gong	80/96/112/128	-	32/64	-	80/96/112/128	80/96/112/128	LFSR	11 stage LFSR	-	-
	Subterranean	128	-	128	-	128	128	XOR, NAND, NOT	5 steps	-	-
	sLiSCP-light	-	-	-	192/256	192/256	-	SPN	108/144	-	-
	SPARX	128/256	-	-	-	64/128	64/128	ARX	24/40	-	-
	PHOTON	-	-	-	256	256	-	SPN	12	64	64
	SimP	-	-	-	192/256	192/256	-	Feistel	52/68/104/136	-	-
	RECTANGLE	80/128	-	-	-	64	64	SPN	25	1024	1024
	Keccap-p	-	-	-	25/50/100/200/ 400/800/1600	25/50/100/200/ 400/800/1600	-	Sponge	Any positive integer	-	-
	ASCON-p	-	-	-	320	320	-	SPN	30/32	-	-
	GIMLI-24	-	-	-	384	384	-	SPN	24	-	-

5.3 NIST Lightweight AEAD Candidates Modes of Operation

The mode of operation of an AEAD scheme is very important because an adversary gains a lot of advantages if the mode of operation is weak. When designing an AEAD, different modes can be adopted. The straightforward approach is to use an existing mode approved by NIST. This is the safest method to design an AEAD. A possible approach is to modify and improve an existing mode. This method leads to a design tradeoff between security and performance. A more challenging approach is to design a completely new mode. This method takes a very long time before the mode is approved. Most of the candidates that made it into the second round of the NIST lightweight AEAD competition use existing modes recommended by NIST.

A total of 20 modes of operation were employed by the 32 NIST lightweight AEAD candidates. The modes of operation include Counter (CTR), Ciphertext Feedback (CFB), Plaintext Feedback (PFB), Hybrid Feedback (HYFB), Output Feedback (OFB), Combined Feedback (COFB), Beetle, Minimally XORed Feedback (mixFeed), Offset Codebook (OCB), Small (Simple, Slim, Sponge based) AEAD from Block cipher (SAEB), Small Universal Deterministic Authenticated Encryption (SUNDAE), JAMBU, Parallel AE from a Forkcipher (PAEF), Cipher Block Chaining (FCBC), Offset Two-Round (OTR), Sponge One-Pass (S1P), LFSR/NLFSR, Sponge, Duplex, and DrySponge. The frequently used mode is the duplex which is employed by a total of 9 out of the 32 candidates. Brief descriptions of the different modes of operations are given in Table 5.

5.4 Hardware Evaluation of the NIST Lightweight AEAD Candidates

The hardware resource utilization of an algorithm is probably the most important metric when it comes to resource-constrained devices like RFID tags. This is because a secured algorithm that consumes a lot of hardware resources cannot fit into low-cost devices. The ongoing NIST AEAD competition is specifically intended for extremely low-cost devices and for that matter, it is important to examine the hardware resources reported by the candidates that made it into the second round of the competition. The hardware implementation result of a design is usually reported in terms of its gate area. The area of a design is usually dependent on the type of technology, device, and standard cell libraries used in the implementation. When it comes to hardware design, two main implementation approaches are usually considered. They include FPGA and ASIC implementations. ASIC implementation has an advantage when it comes to the cost per device in mass production. The disadvantage of ASIC implementation is that it takes a long time to get a product to market and it is also not reconfigurable. The major advantage of FPGA implementation is its re-programmability which affords its unlimited flexibility. ASIC implementation estimates resource utilization using GE. The GE is defined as the area used by the smallest two-input NAND gate. FPGA results are given in slices which is a basic configurable cell. A slice consists of flip-flops and a LUT. Table 6 illustrates the hardware results of some of the lightweight AEAD candidates.

A total of 19 out of the 32 second-round candidates reported the hardware resource utilization of their schemes. The candidates include GIFT-COFB, SAEAES, SUNDAE-GIFT, TinyJambu, ESTATE, LOTUS-AEAD, Romulus, SKINNY-AEAD, Grain-128AEAD, ACE, DryGASCON, Gimli, ISAP, KNOT, Oribitida, SPIX, SpoC, Subterranean 2.0, and WAGE. Out of the 19 candidates that submitted

their hardware results, only 4 reported both ASIC and FPGA implementation results. Considering the FPGA results, SAEAES consumes the least Look-up Tables (LUTs) of just 348 while the ASIC results show that TinyJAMBU consumes the least Gate Equivalents (GE) of just 1674.

Table 5. NIST lightweight AEAD candidates' modes of operation

Mode	Description
CTR	CTR is a very simple counter-based implementation. A counter based initialization value is XORed with the plaintext to generate the ciphertext. The mode is independent of the use of feedback and can therefore be implemented in parallel.
CFB	In CFB, the ciphertext is given as feedback to the next block of encryption
PFB	PFB is a rarely used mode of encryption. It works similar to CFB but the plaintext instead of the ciphertext is used as the feedback to the encryption block
HYFB	In HYPF the block cipher input partially CFB and partially PFB
OFB	OFB mode first uses the initialization vector (IV) and key stream for encryption. The output is then feedback as the IV.
COFB	The COFB mode is based on both CTR and OFB mode.
Beetle	In Beetle mode, a COFB is employed to create a difference between the ciphertext and the next feedback block.
mixFeed	mixFeed uses a mixture of the plaintext and the ciphertext as feedback to the block cipher.
OCB	OCB is a scheme that basically integrates a Message Authenticating Code (MAC) into the design of a block cipher. This avoids the use of two algorithms for authentication and encryption.
SAEB	SAEB uses the sponge-based design structure. The permutations technique in sponge is applied to a block cipher in SAEB. SAEB can be viewed as a cascaded n-bit block cipher.
SUNDAE	SUNDAE uses only one key and operate by making several cascaded block cipher calls.
JAMBU	JAMBU is a variant of the CFB mode. In JAMBU, the message block and an additional state are XORed with the internal state
PAEF	In PAEF mode, associated data and message are grouped into block of n bits. Each block is the processed with a call to a Forkcipher using a tweak.
FCBC	FCBC is an upgrade to the Encrypted Message Authentication Codes (EMAC). It can handle arbitrary length of messages without extra block cipher calls
OTR	The OTR mode uses a two-round Feistel permutation where the rounds are obtained from a tweakable block cipher
S1P	S1P make two calls to a tweakable block cipher. The first call is to generate the key while the second call is to generate the tag
LFSR/ NLFSR	This mode is a combination of a LFSR and a NLFSR. LFSR and NLFSR are registers that have inputs which are functions of their own states.
Sponge	The sponge operation is an iterated construction for building a function which has variable length input/output that is based on permutations
Duplex	The duplex mode of operation similar to the sponge construction. The duplex mode permits the alternation of the input and output like a full duplex communication.
DrySponge	DrySponge is directly derived from the duplex construction. It differs from the duplex mode in the way it concatenates the input with the internal state and also in the way it extracts the output from the internal state.

Table 6. Hardware implementation results of the NIST lightweight AEAD candidates

Underlying construction	Candidates	FPGA resource utilization			ASIC resource utilization		
		Device	LUTs	Frequency (MHz)	Technology (nm)	Gate count	Frequency (MHz)
Block cipher based	GIFT-COFB	-	-	-	90 (STM)	3927	10
	SAEAES	Virtex-7	348	145.9	45 (NANGATE)	3502	122
	SUNDAE-GIFT	-	-	-	90 (STM)	3494	10
	TinyJambu	-	-	-	90 (UMC)	1674	22.8
Tweakable block cipher based	ESTATE (TweAES)	Virtex-7	2235	314.71	-	-	-
	ESTATE (TweGIFT)	Virtex-7	1413	580.11	-	-	-
	LOTUS-AEAD	Virtex-7	865	424.45	-	-	-
	Romulus	-	-	-	65 (TSMC)	4498	1250
	SKINNY-AEAD	-	-	-	90 (UMC)	10239	227
Stream cipher based	Grain-128AEAD	-	-	-	65 (STM)	5258	1120
Permutation based	ACE	Spartan-6	1272	123	65 (STM)	4600	705
	DryGASCON	Zynq-7000	2054	150	-	-	-
	Gimli	Spartan-6	4398	36.24	28 (STM)	35452	441.1
	ISAP	-	-	-	130 (UMC)	14000	169
	KNOT	-	-	-	45 (NANGATE)	3628	1123
	Oribatida	Virtex-7	940	554.16	-	-	-
	SPIX	-	-	-	65 (STM)	2396	-
	SpoC	-	-	-	65 (STM)	1820	-
	Subterranean 2.0	Zynq SoC	763	200	45 (FreePDK)	4880	2597.4
	WAGE	Spartan-6	431	134	65 (STM)	3290	1120

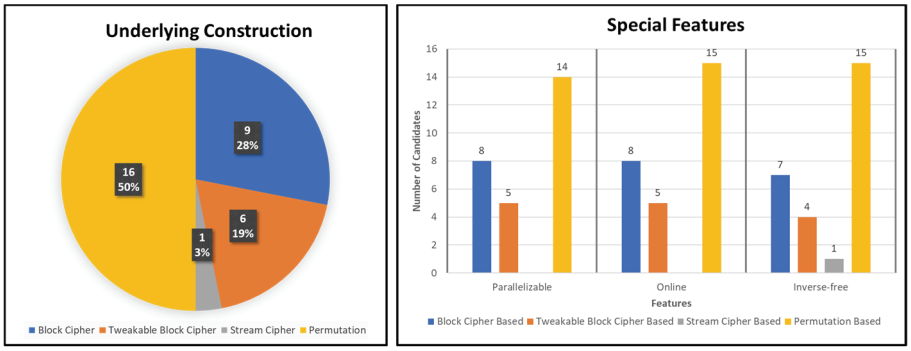
5.5 Interpretation of Lightweight AEAD Candidates Properties

This section gives graphical representations of the properties of the lightweight AEAD candidates. This is done to give an interpretation of all the properties discussed in Section 5. Here the important properties selected include the underlying construction, special features, general parameters, and security parameters. This section serves as a form of summary and explains the reason for some important properties possessed by the candidates.

Exactly half of the candidates have underlying construction made up of permutation primitives as illustrated in Fig 1(a). The reason for the high number of permutation-based candidates is that the winner of the Secure Hash Algorithm 3 (SHA-3) competition [20] is a permutation-based function known as Keccak [21]. This has motivated a lot of researchers to focus on developing permutation-based functions. A reason for the low number of block/tweakable block cipher-based candidates is that they require primitives that may consume a lot of hardware resources and therefore not suitable for low-cost devices. There is only one stream cipher-based candidate because most of their computations take a long time to complete.

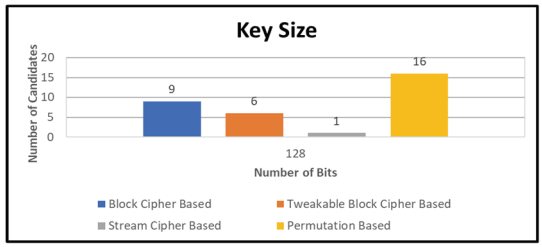
A majority of the submissions share some special features which include parallelism, online, and inverse-free. The parallelizable and online features enable fast computations while the inverse-free

feature minimizes the hardware area. It is therefore vital for the submissions to exhibit these features. Only four submissions do not provide the parallelizable and online features as shown in Fig. 1(b). Only three submissions do not provide the inverse-free feature. This is encouraging because it proves that most of the submissions were designed with low-cost features as an objective.

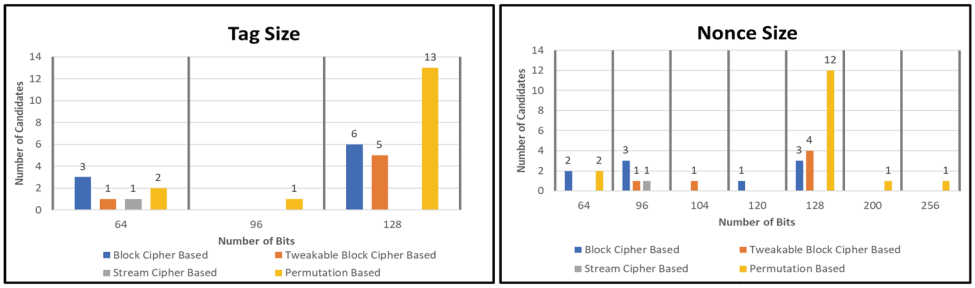


(a) (b)

Fig. 1. (a) Submissions per underlying construction and (b) submissions per special feature.



(a)



(b) (c)

Fig. 2. (a) Submissions per key size, (b) submissions per tag size, and (c) submissions per Nonce size.

The general parameters common to all the candidates include key size, tag size, and nonce size. In terms of key size, all the submissions use a 128-bit key as shown in Fig. 2(a). This is mainly because low-cost devices require medium security levels and a 128-bit key is enough to safeguard such devices. The tag sizes used by the submission include 64-bit, 96-bit, and 128-bit as shown in Fig. 2(b). The tag size also indicates the block size of each submission. Most of the submissions use a 128-bit tag size because that is the limit set for low-cost devices. The nonce sizes range from 64-bit to 256-bit as shown in Fig. 2(c). The 128-bit nonce size is used by most of the submissions for the same reason as the tag size.

In terms of security requirements, NIST requested a minimum of 2^{112} computations for both confidentiality and integrity. A total of eleven submissions did not meet this security requirement as shown in Fig. 3. A total of eighteen submissions recorded security requirements that are higher than the NIST request. Only one permutation-based submission did not meet the NIST security requirement. This is because most of the permutation-based candidates were designed after the NIST call for submissions. A majority of the block cipher-based candidates did not meet the NIST security requirements because most of them were designed for general purpose use before the NIST lightweight AEAD call for submissions. The submission with the lowest security recorded 2^{60} computations for confidentiality and integrity. This is however accepted in low-cost devices because the data in these devices are not static but continuously changing.

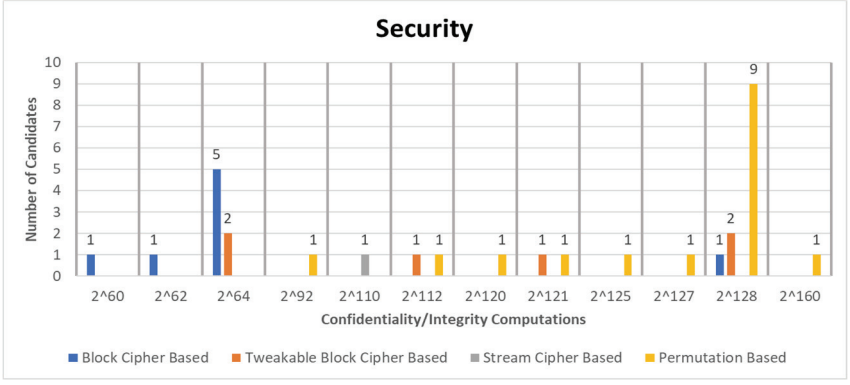


Fig. 3. Submissions per security features.

6. Challenges and Recommendation

The NIST lightweight AEAD standardization process is much needed in this era of the IoT. The standardization process serves as the first step in selecting AEAD algorithms which will be suitable for very low-cost devices. Low-cost devices refer to devices that are constrained in terms of hardware area, memory, and power consumption. These constraints mean that lightweight AEAD algorithms have to be designed to use as little hardware resources as possible.

The major challenge about the lightweight AEAD standardization process is that there is no minimum requirement on the hardware resources that the algorithms should meet. For that matter, a total of 13 submissions out of the 32 round-two candidates did not provide any hardware resource utilization of their algorithms. This makes it extremely difficult to judge the suitability of the algorithms for a low-cost device. The submissions that reported their hardware results did so using completely different FPGA devices, technology, and standard cell libraries. It is well established in the hardware design community that comparing algorithms that were designed using different hardware platforms is not recommended.

The recommendation is that a hardware framework should be made available for all the submissions. The hardware framework should consist of a specific FPGA device, a specific ASIC technology, and a specific standard cell library. All the submission should be designed using the hardware framework and the results recorded. This will enable the selection committee to easily compare and contrast among the different submissions.

7. Conclusions and Future Work

This work surveyed the 32 AEAD algorithms that made it into the second round of the ongoing NIST lightweight AEAD standardization process. The NIST lightweight AEAD standardization process was initiated in August 2018 to select lightweight AEAD algorithms that are suitable for resource-constrained devices like RFID tags, sensors, and smart cards. In August 2019, 32 submissions made it into the second round of the process which is due to end in the year 2021. This paper gives a comprehensive survey of the 32 NIST lightweight AEAD round-two candidates. The work starts by summarizing the NIST submission requirements for lightweight AEAD schemes and the evaluation criteria. The 32 candidates are then classified in terms of their underlying construction which include block cipher-based, tweakable block cipher-based, stream cipher-based, and permutation-based. The candidates are then evaluated based on their security parameters, mode of operation, special features, and hardware resource utilization. The paper also discusses some challenges with the standardization process and offers some useful recommendations.

This paper provides useful information for researchers in the cryptographic community as well as the lightweight AEAD competition committee. This work can assist cryptographers in the AEAD community to select their favorite lightweight AEAD candidates based on their security parameters, primitives, and implementation results.

Future research will concentrate on proposing and designing a general hardware architecture framework for implementing the NIST lightweight AEAD algorithms.

References

[1] K. Schwab, *The Fourth Industrial Revolution*. Geneva, Switzerland: World Economic Forum, 2016.

[2] K. Gafurov and T. M. Chung, "Comprehensive survey on internet of things, architecture, security aspects, applications, related technologies, economic perspective, and future directions," *Journal of Information Processing Systems*, vol. 15, no. 4, pp. 797-819, 2019.

[3] M. Bellare and C. Namprempre, "Authenticated encryption: Relations among notions and analysis of the generic composition paradigm," in *Advances in Cryptology – ASIACRYPT 2000.* Heidelberg, Germany: Springer, 2000, pp. 531-545.

[4] P. Rogaway, "Authenticated-encryption with associated-data," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, Washington, DC, 2002, pp. 98-107.

[5] CAESAR call for submissions [Online]. Available: https://competitions.cr.yp.to/caesar-call.html.

[6] CAESAR submissions [Online]. Available: https://competitions.cr.yp.to/caesar-submissions.html.

[7] National Institute of Standards and Technology, "Announcing request for nomination for lightweight cryptographic algorithms," 2018 [Online]. Available: https://csrc.nist.gov/News/2018/requesting-nominations-for-lightweight-crypto-algs.

[8] K. McKay, L. Bassham, M. Sonmez Turan, and N. Mouha, "Report on lightweight cryptography," National Institute of Standards and Technology, Gaithersburg, MD, Report No. IR-8114, 2017.

[9] National Institute of Standards and Technology, "Lightweight cryptography: round 1 candidates," 2021 [Online]. Available: https://csrc.nist.gov/Projects/lightweight-cryptography/round-1-candidates.

[10] National Institute of Standards and Technology, "Lightweight cryptography: round 2 candidates," 2021 [Online]. Available: https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[11] B. Rezvani and W. Diehl, "Hardware implementation of NIST lightweight cryptographic candidates: a first look," *IACR Cryptology ePrint Archive*, vol. 2019, article no. 824, 2019.

[12] F. Abed, C. Forler, and S. Lucks, "General classification of the authenticated encryption schemes for the CAESAR competition," *Computer Science Review*, vol. 22, pp. 13-26, 2016.

[13] E. B. Kavun, H. Mihajloska, and T. Yalcin, "A survey on authenticated encryption: ASIC designer's perspective," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, article no. 88, 2017. https://doi.org/10.1145/3131276

[14] F. Zhang, Z. Y. Liang, B. L. Yang, X. J. Zhao, S. Z. Guo, and K. Ren, "Survey of design and security evaluation of authenticated encryption algorithms in the CAESAR competition," *Frontiers of Information Technology & Electronic Engineering*, vol. 19, no. 12, pp. 1475-1499, 2018.

[15] M. Agrawal, J. Zhou, and D. Chang, "A survey on lightweight authenticated encryption and challenges for securing industrial IoT," in *Security and Privacy Trends in the Industrial Internet of Things*. Cham, Switzerland: Springer, 2019, pp. 71-94.

[16] M. J. O. Saarinen and D. W. Engels, "A do-it-all-cipher for RFID: design requirements," *IACR Cryptology ePrint Archive*, vol. 2012, article no. 317, 2012.

[17] L. Bassham, C. Calik, K. McKay, N. Mouha, and M. S. Turan, "Profiles for the lightweight cryptography standardization process," 2017 [Online]. Available: https://csrc.nist.gov/publications/detail/white-paper/2017/04/26/profiles-for-lightweight-cryptography-standardization-process/archive.

[18] National Institute of Standards and Technology, "Submission requirement and evaluation criteria for the lightweight cryptography submission process," 2018 [Online]. Available: https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/final-lwc-submission-requirements-august2018.pdf.

[19] D. A. N. Gookyi and K. Ryoo, "Selecting a synthesizable RISC-V processor core for low-cost hardware devices," *Journal of Information Processing Systems*, vol. 15, no. 6, pp. 1406-1421, 2019.

[20] National Institute of Standards and Technology, "Cryptographic hash algorithm competition," 2007 [Online]. Available: https://www.nist.gov/programs-projects/cryptographic-hash-algorithm-competition.

[21] G. Bertoni, J. Daemen, M. Peeters, and G. V. Assche, "Keccak specification," 2008 [Online]. Available: https://keccak.team/obsolete/Keccak-specifications.pdf.

Dennis Agyemanh Nana Gookyi **https://orcid.org/0000-0002-6622-0538**

He received a B.Sc. degree in Computer Engineering from Kwame Nkrumah University of Science and Technology, Ghana, in 2013 and M.Eng. degree in Information and Communication Engineering from Hanbat National University, South Korea in 2017 where he is currently a Ph.D. candidate. His research interests include SoC design and verification platforms, lightweight cryptography, and SoC design for security.

Guard Kanda **https://orcid.org/0000-0002-3009-9860**

He received a B.Sc. degree in Computer Science from Kwame Nkrumah University of Science and Technology, Ghana, in 2012 and M.Eng. degree in Information and Communication Engineering from Hanbat National University, South Korea in 2016 where he is currently a Ph.D. candidate. His research interests include soc design and verification platforms, lightweight cryptography, and SoC design for security.

Kwangki Ryoo **https://orcid.org/0000-0001-8574-7418**

He received B.S., M.S., and Ph.D. degrees in Electronic Engineering from Hanyang University, Korea in 1986, 1988, and 2000, respectively. From 1991 to 1994, he was an Assistant Professor at the Korea Military Academy (KMA) in South Korea. From 2000 to 2002, he worked as a Senior Researcher at the Electronics and Telecommunications Research Institute (ETRI), Korea. From 2010 to 2011, he was a Visiting Scholar at the University of Texas in Dallas. Since 2003, he has been a Professor at Hanbat National University, Daejeon Korea. His research interests include engineering education, SoC platform design and verification, image signal processing and multimedia codec design, and SoC design for security.