This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MDAT.2017.2682234, IEEE Design and Test

Comparative study of Authenticated Encryption targeting lightweight IoT applications

Sandhya Koteshwara, University of Minnesota, USA and Amitabh Das, Intel Corporation, USA

***Abstract*—Today's smart, connected devices pose a complex challenge to cryptographic designers in terms of better resource efficiency, area constraints, flexibility and robustness. In this paper, we look at some new Authenticated Encryption (AE) schemes from an ongoing Competition for Authenticated Encryption: Security, Applicability, and Robustness (CAESAR), and evaluate their benefits to lightweight applications such as sensor networks, RFID, IPSec, Internet-of-Things, implantable medical devices and wearables. We summarize candidates after extensive studies of their properties such as nonce misuse resistance, security measures, parallelizabilty and existing hardware and software implementations, and recommend a few candidates most suitable for different applications.**

***Index Terms*—Authenticated Encryption, CAESAR, IoT, Symmetric key cryptography**

I. Introduction

Authenticated encryption schemes are a class of symmetric-key cryptographic algorithms that simultaneously provide both confidentiality and authenticity of data. Confidentiality involves protecting data from being disclosed without authorization, while authenticity encompasses ensuring both integrity of data and verification of its source. Some applications require the handling of additional data such as packet headers, which demand authentication without the need for any encryption. Such schemes are broadly classified as Authenticated Encryption with Associated Data (AEAD) and will largely be the focus of this paper. A typical interface for AEAD, is shown in Fig. 1 and can be described using the equations 1 and 2.

$$Enc : E_k(N, AD, P) \Rightarrow \{C, T\} \quad (1)$$

$$Dec : D_k(N, AD, C, T) \Rightarrow \{P, \perp\} \quad (2)$$

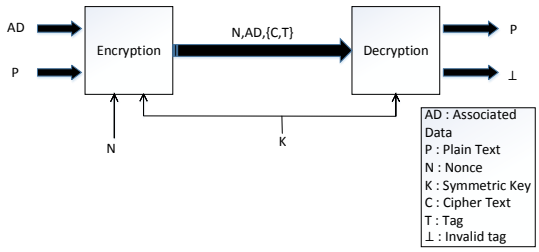


Fig. 1. General scheme of an Authenticated Encryption scheme with Associated Data

It is possible to build a generic composition for authenticated encryption schemes, combining an encryption scheme which is IND-CPA and MACs which are unforgeable [1]. However, these approaches are not efficient both with respect to performance and resource usage. Unless implemented correctly, these compositions tend to be error-prone as has been observed in practice. For example, padding oracle attacks have broken the security of Mac-then-Encrypt schemes, while using the message directly for tag calculation in Encrypt-and-MAC schemes makes the confidentiality of the message questionable. Moreover, authentication of associated data provides an additional challenge when trying to combine the two goals of security as outlined in [2]. Hence, the need for a secure and efficient authenticated scheme, continues to be a challenge for today's designers.

While many proposals for AE have been presented over the last decade, the encryption schemes that have gained popularity include the GCM(Galois/Counter mode) [3], CCM(Counter with CBC-MAC) [4], EAX [5] and OCB(Offset Codebook) [6] modes. The AES-GCM uses AES in the counter mode (CTR) and a Galois mode of authentication. Several advantages including ease of implementation have made AES-GCM to be popularly adopted in various applications.

Contributions. In this paper, we look at AEAD schemes from a perspective of usage in lightweight applications. In this regard, we first study the architecture of AES-GCM and then look into some new and efficiently constructed schemes from an ongoing competition CAESAR. This paper also serves as a comparison between the different candidates of CAESAR with AES-GCM as reference. An overview of this kind, considering functional and architectural aspects, performance measures, comparison with AES-GCM and an application oriented discussion does not exist in current literature and will be highly beneficial for designers of cryptographic protocols for embedded system platforms and SoCs.

II. AES-GCM

AES built in the Counter (CTR) mode of operation and hashing over the Galois field are combined to obtain AES-GCM authenticated encryption. Offering several advantages, such as high speeds at low cost and low latency, parallelizability and efficient software implementations with table driven operations, the AES-GCM has been widely researched architecturally. Several implementations exist in literature serving both efficiency and high performance [7], [8]. A simplified architectural representation of AES-GCM is provided in Fig. 2., to demonstrate its ease of implementation in hardware.

Even though AES-GCM is the most widely adopted algorithm, a few shortcomings have been observed in the algorithm in recent literature. Message forgery attacks have been demonstrated on the polynomial hashing used for authentication based on weak keys [9]. Also, the requirement

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MDAT.2017.2682234, IEEE Design and Test

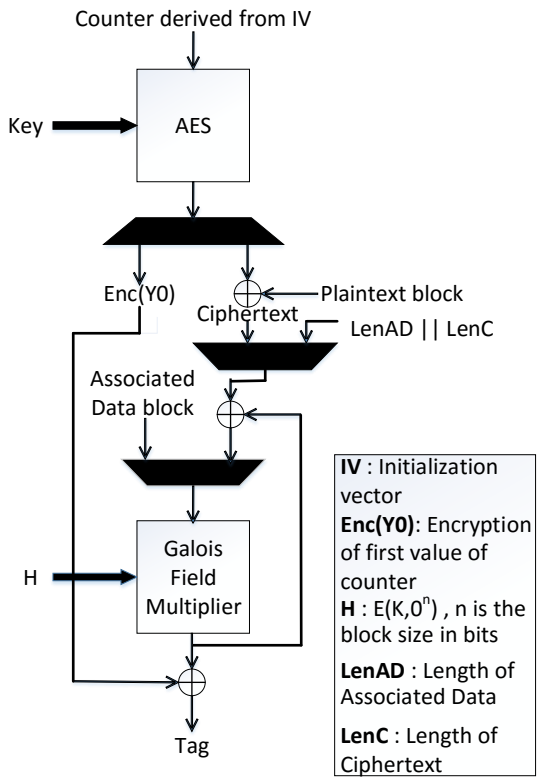


Fig. 2. Simplified block diagram of AES-GCM using minimal resources

of a unique nonce for each key is cumbersome and requires careful implementation and good practices from users and is found to be seldom followed in practice [10]. Moreover, the algorithm still suffers from having to use Galois field multipliers for authentication and there is a precomputation overhead for every new key input into the design.

Changing scenarios of applications of AEAD algorithms has necessitated the need to revisit AES-GCM and explore its advantages or shortcomings from this new paradigm. With security becoming indispensable in several areas of IoT, we need to look into implementations to consider the suitability of algorithms in reference to these devices. For example, could an extremely lightweight protocol be built for resource constrained devices? Could security be given a higher importance in comparison to performance? Would the energy efficiency and availability constraints be met? Are the algorithms flexible enough to provide for updation of parameter values when necessary? Hence, a need to broaden the scope of study of authenticated encryption schemes beyond AES-GCM has risen, paving the way for new areas of research.

III. CAESAR COMPETITION

CAESAR is a competition for Authenticated Encryption: Security, Applicability and Robustness, aimed at looking for new and improved versions of AEAD schemes, possibly overcoming the shortcomings of AES-GCM algorithm [11]. The competition began in 2014 and is currently in the third round, with 29 candidates selected for the 2nd round, from 54 initial submissions. The call for submissions puts forward the goals of a new AEAD algorithm presented in the general format of an AEAD encryption scheme as described in Fig.1. The nonce in this case however, has been termed public message number and/or secret message numbers.

A comprehensive and easy to grasp summary of the first round of CAESAR candidates has been provided in [12], highlighting important features and categorizing them based on the underlying primitives used. In this paper, we update this overview to consider only the second round candidates and include software and hardware implementations from various publications and tools existing in current literature. We also highlight some architectural details of each of the candidates and bring out important features. We intend the usage of this article as a reference for designers looking for new AEAD schemes more suitable to their applications without having to go through overwhelming number of papers. As a further step, we use these data summaries to select candidates for lightweight implementation scenarios discussed previously.

A. Primitives

For simplicity, we categorize the different submissions into three groups : block-cipher based, sponge/duplex based and others. These categories are based on the primitives that have been used to construct the candidates. Since, traditionally block ciphers have been used to construct AEAD, we see many submissions from this category. Many of them use AES or AES-like constructions while some of them build their own block cipher constructions. The next category is based on duplex constructions mostly using MonkeyDuplex construction derived from the permutation based sponge function from the new SHA3 standard, Keccak [13]. In the last category, we have grouped other types such as stream cipher based, permutation based and fiestel network based schemes.

B. Features

We now discuss some of the features of these candidates mostly from an architectural and application oriented perspective.

1) Nonce Misuse Resistance: Though the issue of nonce misuse has been debated over in several forums, with many accepting and others denying the issue, it is favorable for an algorithm to provide a nonce misuse resistant feature. Specifically in the case of IoT and like devices, storing and managing fresh nonces for each new message and key combination is going to become a challenging task. This will lead to error prone and incorrect implementation scenarios leading to security breaches. The nonce for the candidates has been split into a public message number(PMN) and secret message number(SMN). While the PMN behaves just like a typical nonce, the SMN is embedded inside the ciphertext and must be recoverable from it [14].

A few of the algorithms provide complete nonce misuse resistance in terms of both confidentiality and integrity, while a few provide for integrity with no confidentiality assured. Some of the algorithms use the SMN to guarantee security assuming different SMNs are used even when PMNs are repeated. While it is unclear how practical these categories

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MDAT.2017.2682234, IEEE Design and Test

maybe, providing for partial nonce misuse resistance can be considered a significant improvement over having none at all. We categorize the candidates based on their ability to provide nonce misuse resistance in a complete sense, partially or none at all. The summary for the same has been provided in Table. I.

2) Security: Security constraints for algorithms need to be evaluated in terms of both strong mathematical proofs and thorough cryptanalysis by external groups. Such a deep level of security analysis is beyond the scope of this paper. However, we summarize some of the security numbers as presented in [12], picking out values specific to a key size of 128 bits (as a more common case scenario) when the option is available. If not, other closer numbers such as 120 bits or 256 bits have been picked. The numbers are in terms of time complexity t and query complexity q, quantifying confidentiality and integrity. For our own analysis we consider that security of atleast 128 bits as acceptable for most lightweight industrial applications.

3) Parallelizability: Algorithms which are parallel can help both software and hardware designers in implementing better designs through the use of various optimization techniques known in literature. Hence, the feature of whether an algorithm is parallelizable or not, is useful in determining if a resource optimized implementation could be obtained. We also categorize the candidates based on whether they are inherently parallel or not and if they provide some form of partial parallelizability. This means that even though the algorithms are not fully parallel with respect to a single message, when multiple messages are handled, the algorithm can be constructed in a parallel fashion. This is presented in Table. I.

IV. Performance Analysis

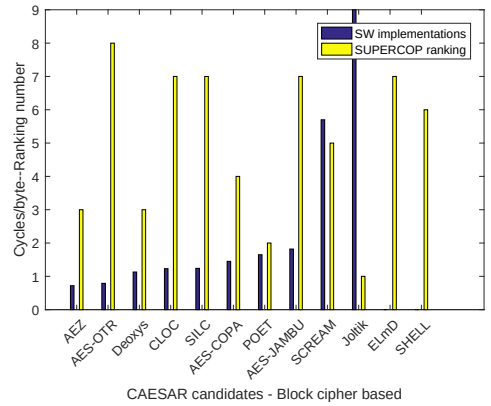
A. Software implementations

The measurement of performance with respect to a candidates software implementation, is annotated in terms of cycles per byte(cpb). As part of Round 1 and Round 2 submissions, software implementations have been submitted by each of the candidates. Most of these have been implemented on the latest processors including Intel Haswell or Sandy Bridge, employing well known AES NI instruction sets and SIMD instructions [15], [16], [17], [18], [19], [20], [21], [22], [23]. Apart from these implementations, we have also considered the implementations from SUPERCOP benchmarking tool [24], well presented in [25], for Intel Core i7-4770 and Intel Core i5-3210M. These resources have helped us rank the candidates in terms of software performance as represented in Fig.3. For custom implementations, cpb is reported and for benchmark implementations, a ranking number which provides relative speed is shown.

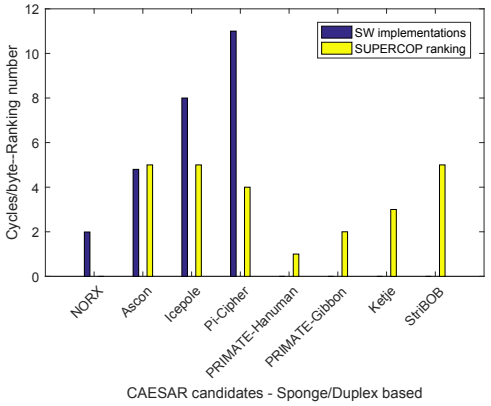
B. Hardware implementations

The difficulty of benchmarking hardware arises due to the existence of a varying range of hardware platforms. This includes families of FPGA's such as Xilinx which use LUTs for measurement and Altera which uses LEs, different CAD tools performing different type of optimizations giving varying synthesis results, ASIC libraries etc. In this regard, the Cryptographic Engineering Research Group(CERG), GMU is playing a major role in proposing a hardware API and releasing a benchmarking tool ATHENa [26]. Existing implementations from this tool and other hardware implementations from various existing publications have been considered for this analysis.

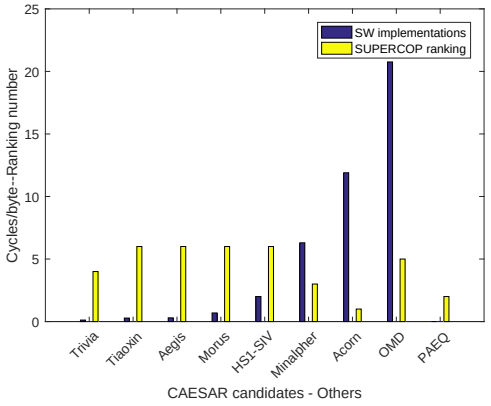
Even though we have performance numbers in terms of area and throughput for both FPGA and ASIC implemen-



(a) Block-cipher



(b) Sponge/Duplex



(c) Others

Fig. 3. Software performance numbers based on candidate implementations and SUPERCOP benchmarks

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MDAT.2017.2682234, IEEE Design and Test

TABLE I
COMPARISON OF CANDIDATES BASED ON NONCE MISUSE RESISTANCE, SECURITY AND PARALLELIZABILITY

Based on primitives	**Nonce-Misuse resistance**	**Candidates**	**Security in terms of time complexity (t) and query complexity (q)** **Key = 128 bits**	**Candidates**	**Parallelizability**	**Candidates**
Block cipher	Complete	Joltik, **Deoxys**, POET, AES-COPA, ELmD, **AEZ**, SHELL	t <128 bits q <64 bits	Joltik, SHELL	Yes	AES-COPA, **AES-OTR**, **AEZ**, **Deoxys**, ELmD, Joltik, Scream, Shell
	Partial	**CLOC**, **SILC**, **AES-JAMBU**	t = 128 bits q = 64 bits	ELmD, Scream, **CLOC**, **Deoxys**, **SILC**, POET, AES-COPA, **AES-JAMBU**	Partial	**CLOC**, **SILC**, POET
	None	Scream, **AES-OTR**	t =128 bits q >64 bits	**AES-OTR**, **AEZ**	No	**AES-JAMBU**
Sponge/Duplex	Complete	Primate-APE	t<128 bits q<64 bits		Yes	Icepole, **Keyak**, **NORX**, StriBOB
	Partial	Pi-Cipher(SMN based), Icepole(SMN based), **Keyak**, **NORX**	t=128 bits q=64 bits	**Ascon**, **NORX**	Partial	
	None	**Ascon**, Primate-HANUMAN, Primate-GIBBON, StriBOB, **Ketje**	t=128 bits q>64 bits	PRIMATEs, Pi-Cipher, StriBOB, Icepole, **Keyak**, **Ketje**	No	**Ascon**, **Ketje**, PRIMATEs
Others (Stream, Permutation, Feistel)	Complete	Paeq, HS1-SIV, Minalpher	t<128 bits q<64 bits		Yes	**Acorn**,**Aegis**, Paeq, Pi-Cipher, **Tiaoxin**, Trivia, Minalpher
	Partial	**Acorn**	t=128 bits q=64 bits	**Acorn**, **Aegis**	Partial	
	None	Trivia, **MORUS**, **Aegis**, OMD, **Tiaoxin**	t=128 bits q>64 bits	Trivia, **MORUS**, Paeq, HS1-SIV, OMD, Minalpher, **Tiaoxin**	No	HS1-SIV, **MORUS**, OMD

tations, we consider FPGA implementations to be more reliable. ASIC implementations provide the area performance in terms of gate counts which are completely dependent on process, technology and libraries used. The existing implementations [27], [28], [29], [30] for FPGA are available for different generations of Xilinx and Altera FPGAs, necessitating the need for normalization which has been done by taking reference implementations of AES-GCM on these same or similar platforms from the ATHENa website. Similar to the rankings we prepared for software performance, we have prepared charts to rank hardware performance in terms of area, throughput and throughput/area. These easy to grasp graphs give a good idea of algorithms which are lightweight or speed oriented and were used by us for our further analysis as well.

C. Memory footprint

Most candidates focus on software performance with respect to cpb and hardware performance with respect to area and throughput. However, it is of paramount importance for lightweight applications to look into their memory footprint with respect to embedded system implementations. We researched a few protocols in this regard and found performance numbers for the same. We summarize this in Table. II.

TABLE II
CANDIDATES WITH IMPLEMENTATIONS ON EMBEDDED PLATFORMS AND ASSOCIATED MEMORY FOOTPRINTS

Candidate	**Device**	**ROM (bytes)**	**RAM (bytes)**
CLOC	ATmega128	2980	362
SCREAM	Atmel AVR	6442	160
WhirlBOB	Cortex-A8	608	-
Minalpher	RL78 microcontroller	510	214

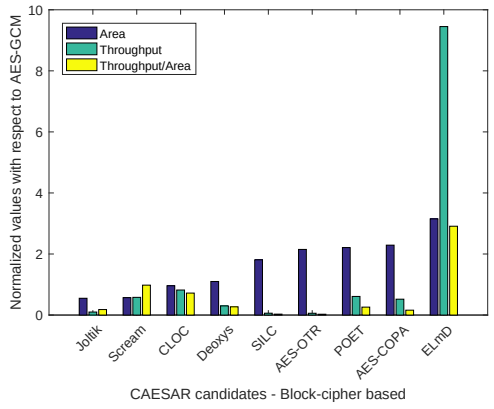
V. SUMMARY WITH RECOMMENDATIONS OF CANDIDATES

A. *Block-cipher based*

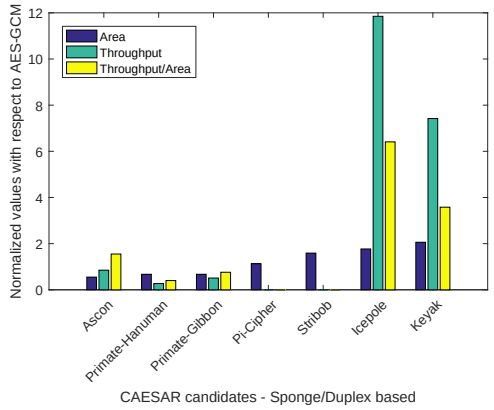
Joltik and **Deoxys** [11] are AE schemes built from custom made tweakable block ciphers (TBC) and use linear and lightweight transformations for their top module. They show excellent hardware and software performances and are available in both nonce misuse resistant and non-resistant modes. While Deoxys provides full security beyond birthday bound, Joltik has lower security at lower implementation cost. These algorithms have no precomputation overhead or long initialization.

SCREAM [11] also makes use of a custom built Tweakable authenticated encryption block(TAE) providing security beyond the birthday bound and masking to achieve security

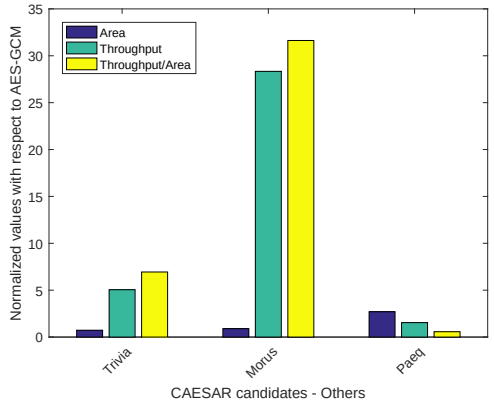
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MDAT.2017.2682234, IEEE Design and Test



(a) Block-cipher



(b) Sponge/Duplex



(c) Others

Fig. 4. Hardware performance numbers based on candidate implementations and ATHENa tools

against side channel attacks such as Differential Power analysis and Electro-Magnetic analysis. Being a highly parallel architecture, allows SCREAM to exploit SIMD instructions providing for good software implementations and efficient hardware architectures. However, Scream does not provide any kind of nonce misuse resistance.

CLOC and **SILC** [11] use AES as the underlying block cipher and provide partial nonce misuse resistance with respect to integrity, while providing no guarantees on confidentiality. They handle short data very well by having minimal precomputation and low memory requirements. They fare decently well in terms of both software and hardware performances and provide acceptable levels of security.

POET [11] is another well rounded candidate based on AES and providing features such as complete nonce misuse resistance and parallelizability. These algorithms are targeted at low latency, high throughput applications with large data. While the scheme is well pipelineable and offers a birthday bound on security, the requirement of inverse operation of decryption may serve as a caveat.

AEZ [11] implements an encode-then-encipher mode of operation providing for strong security and complete nonce misuse resistance. The architecture automatically exploits novelty and redundancy in messages, giving good software implementation performance. However, this architecture suffers from being extremely difficult to implement in hardware.

B. Sponge/Duplex based

Based on the Spongewrap/MonkeyDuplex mode of operation and eliminating inverse operations, **ASCON** [11] presents a low hardware scheme with extremely low memory requirements. While the scheme provides full security of 128 bits, it offers no nonce misuse resistance.

The **PRIMATEs** [11] come in 3 flavors and have varying goals of lightweight (HANUMAN), High speed (GIBBON) and additional security (APE). While these schemes have shown good hardware and software performances, and provide high security numbers, only APE provides complete nonce misuse resistance.

This **Pi-Cipher** [11] architecture uses an ARX based function and a parallel, incremental architecture most suitable for long messages. Its main advantage is that it does not use the inverse operation. While providing for intermediate tags, these schemes also provide partial nonce misuse resistance in terms of SMN as described before. Simple operations of its underlying function, results in good hardware performances while providing for decent software speeds and high security.

C. Others

While there are many schemes in this category, **MORUS** [11] which uses a dedicated structure that has both hardware and software performance benefits by making use of simple operations such as shift, and, xor and SIMD instructions, respectively. Excellent security is provided in terms of both confidentiality and integrity. However, no nonce misuse resistance is claimed.

Lightweight applications such as smart card, RFID, etc. demand low area and low memory footprint. AEAD schemes suitable for implementation in wearables additionally require that the power consumed and energy per bit of processing be minimal. On the other hand, the security protocols of industrial wireless sensor nodes demand high performance in software or hardware depending on the implementation used. Parallelizability allows for efficient implementations which can be beneficial to protocols implemented in IPsec. Security critical applications such as automotive IoTs demand higher levels of security in terms complexity of attack and complete

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MDAT.2017.2682234, IEEE Design and Test

TABLE III
SUMMARY OF CANDIDATES HIGHLIGHTING PROPERTIES AND POSSIBLE APPLICATION SCENARIOS

Algorithms	Summary of properties	Application scenario
DEOXYS, MORUS, ASCON	Excellent performance characteristics both in hardware (throughput) and software (cpb)	Wireless sensor nodes, IPsec
POET, AEZ, DEOXYS	Provide high security (complexity of attack) and complete nonce misuse resistance	Automotive IoTs, Defense related
JOLTIK, CLOC/SILC, ASCON, PRIMATEs, PI-CIPHER, SCREAM	Implementations with low area, power and memory footprint requirements	RFID, smart cards, wearables, Implantable medical devices
DEOXYS, AEZ, POET, JOLTIK, PRIMATE-APE, CLOC/SILC	Complete or partial nonce misuse resistance of ensuring guaranteed authentication	All applications

nonce-misuse resistance. Finally for all lightweight applications, some level of nonce-misuse resistance is highly desirable since not having to generate and maintain unique nonces will result in lower resource requirements. The candidates discussed are suitable for one or more applications and the specific application to which they can be associated is listed in Table III. An attempt to identify and categorize candidates according to use cases is also being undertaken as part of Round 3 of CAESAR competition.

VI. Conclusions and Future Work

We have provided a comprehensive review of the competition and parameters to be evaluated with respect to considering new AE schemes for applications. In Table III, we tabulate the discussion of our selected candidates and provide a good reference for future users of the algorithms. Similar analysis can be carried out for other candidates, following the discussions in this paper. Also, we understand that no analysis is complete without a thorough power and energy efficiency analysis. This is the future scope of our research while we try to closely look into the architectural aspects of these candidates and provide more insights.

References

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

[2] P. Rogaway, "Authenticated-encryption with associated-data," in *Proceedings of the 9th ACM conference on Computer and communications security*. ACM, 2002, pp. 98–107.

[3] D. McGrew and J. Viega, "The Galois/counter mode of operation (GCM)," *Submission to NIST. http://csrc. nist. gov/CryptoToolkit/modes/proposedmodes/gcm/gcm-spec. pdf*, 2004.

[4] M. J. Dworkin, *Recommendation for block cipher modes of operation: The CCM mode for authentication and confidentiality*, 2007.

[5] M. Bellare, P. Rogaway, and D. Wagner, "The EAX mode of operation," in *Fast Software Encryption*. Springer, 2004, pp. 389–407.

[6] P. Rogaway, M. Bellare, and J. Black, "OCB: A block-cipher mode of operation for efficient authenticated encryption," *ACM Transactions on Information and System Security (TISSEC)*, vol. 6, no. 3, pp. 365–403, 2003.

[7] G. Zhou, H. Michalik, and L. Hinsenkamp, "Efficient and high-throughput implementations of AES-GCM on FPGAs," in *International Conference on Field-Programmable Technology, 2007. ICFPT 2007.* IEEE, 2007, pp. 185–192.

[8] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "Efficient and high-performance parallel hardware architectures for the AES-GCM," *IEEE Transactions on Computers,*, vol. 61, no. 8, pp. 1165–1178, 2012.

[9] M.-J. O. Saarinen, "Cycling attacks on GCM, GHASH and other polynomial MACs and hashes," in *Fast Software Encryption*. Springer, 2012, pp. 216–225.

[10] H. Böck, A. Zauner, S. Devlin, J. Somorovsky, and P. Jovanovic, "Nonce-disrespecting adversaries: Practical forgery attacks on gcm in tls," 2016.

[11] 2014. [Online]. Available: http://competitions.cr.yp.to/caesar-call.html

[12] F. Abed, C. Forler, and S. Lucks, "General overview of the authenticated schemes for the first round of the caesar competition," Cryptology ePrint Archive: Report 2014/792.[2] CAESAR submissions, second-round candidates. Available: http://competitions. cr. yp. to/caesar-submissions. html, Tech. Rep.

[13] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche, "Permutation-based encryption, authentication and authenticated encryption," *Directions in Authenticated Ciphers*, 2012.

[14] C. Namprempre, P. Rogaway, and T. Shrimpton, "AE5 security notions," 2013.

[15] A. Bogdanov, M. M. Lauridsen, and E. Tischhauser, "AES-Based Authenticated Encryption Modes in Parallel High-Performance Software." *IACR Cryptology ePrint Archive*, vol. 2014, p. 186, 2014.

[16] [Online]. Available: http://ascon.iaik.tugraz.at/implementation.html

[17] T. Iwata, K. Minematsu, J. Guo, and S. Morioka, "CLOC: Authenticated encryption for short input," in *Fast Software Encryption*. Springer, 2014, pp. 149–167.

[18] T. Iwata, K. Minematsu, J. Guo, S. Morioka, and E. Kobayashi, "SILC: SImple Lightweight CFB," *CAESAR submission*, 2014.

[19] [Online]. Available: https://github.com/gvanas/KeccakCodePackage

[20] [Online]. Available: http://pi-cipher.org/

[21] H. Mihajloska, M. El Hadedy, and K. Skadron, "Lightweight version of π-cipher," 2015.

[22] [Online]. Available: http://primates.ae/implementation/

[23] A. Chakraborti, A. Chattopadhyay, M. Hassan, and M. Nandi, "TriviA: A Fast and Secure Authenticated Encryption Scheme," in *Cryptographic Hardware and Embedded Systems–CHES 2015*. Springer, 2015, pp. 330–353.

[24] [Online]. Available: https://bench.cr.yp.to/supercop.html

[25] [Online]. Available: http://www1.spms.ntu.edu.sg/ syllab/speed/

[26] K. Gaj, J.-P. Kaps, V. Amirineni, M. Rogawski, E. Homsirikamol, and B. Y. Brewster, "ATHENa-automated tool for hardware evaluation: toward fair and comprehensive benchmarking of cryptographic hardware using FPGAs," in *2010 International Conference on Field Programmable Logic and Applications (FPL),*. IEEE, 2010, pp. 414–421.

[27] K. M. Abdellatif, R. Chotin-Avot, and H. Mehrez, "AEGIS-Based Efficient Solution for Secure Reconfiguration of FPGAs," in *Proceedings of the Third Workshop on Cryptography and Security in Computing Systems*. ACM, 2016, pp. 37–40.

[28] M. Kotegawa, K. Iwai, H. Tanaka, and T. Kurokawa, "Optimization of Hardware Implementations with High-Level Synthesis of Authenticated Encryption," *Bulletin of Networking, Computing, Systems, and Software*, vol. 5, no. 1, pp. 26–33, 2016.

[29] M.-J. O. Saarinen, "Simple AEAD hardware interface (SÆHI) in a SoC: Implementing an on-chip Keyak/WhirlBob coprocessor," in *Proceedings of the 4th International Workshop on Trustworthy Embedded Devices*. ACM, 2014, pp. 51–56.

[30] [Online]. Available: http://www1.spms.ntu.edu.sg/ diac2015/program.shtml

Sandhya Koteshwara is a PhD student at the University of Minnesota, Twin Cities. She received her Masters degree in 2014 from the Department of Electrical Engineering, UMN. Her current research interests include hardware security, low power architectures for cryptographic algorithms and approximate computing. She is a student member of IEEE.

Amitabh Das is a security researcher in the Security Center of Excellence group at Intel Corporation, Hillsboro, Oregon, USA. He has a Ph.D. in Electrical Engineering from KU Leuven, Belgium. Some of his current research interests include hardware cryptography, embedded security, and hardware/software co-design. He is a member of IEEE.