ISSN: 0974-5823 Vol. 7 (Special Issue, Jan.-Feb. 2022)

International Journal of Mechanical Engineering

An Approach for attack resistance nonce misuse Authenticated Encryption Cipher for Multi-Tenant Security

Suresh Prasad Kannojia[1], Jitendra Kurmi[2]

[1,2]Department of Computer Science

University of Lucknow, Lucknow

Abstract

In order to support the communication speed for high-speed communication, a security algorithm related to the transmission of information is required. An authenticated encryption (AE) algorithms are used to provide security for high-speed communications and security services such as confidentiality, integrity and authenticity for Transport Layer Security. Most of the AEAD algorithms are vulnerable to side-channel, forgery and salamander attacks. In this paper we proposed a state-of-the-art AEAD algorithm, which provides attack resistance nonce misuse authenticated encryption cipher for multi-tenant security based on AES-GCM and AES-GCM-SIV. It contains four keys to improve the security and performance of the proposed algorithm. We also compared the performance of AES-GCM and AES-GCM-SIV with the proposed algorithm on the basis of time taken to encrypt/decrypt a message block of size 128/1024/2048/4096 bit. We found that the performance of our proposed algorithm is better than AES-GCM-SIV and AES-GCM in terms of execution time. To avoid the use of same key for multiple purposes, improving security boundaries and for performance, overhead was significant to use multiple key.

Keywords: AES-GCM, SIV, Nonce, AEAD Cipher, Attack Resistant, Security, TLS

1. Introduction:

Authenticated encryption (AE) approaches to integrate the authentication and encryption procedures into a single, safe, and resource-efficient solution. Data secrecy is generally proven to be insufficient, and verifying the source and integrity of information is necessary. Consequently, AE algorithms become increasingly relevant as device contexts and platforms change. The results are known as AE, where information of packet headers (AEAD), and message numbers are incorporated in the plaintext. These extra data require authentication and are referred to as related data. The general AEAD equations are given below:

$$\text{EncK } (N_{\text{Nonce}}, A_{\text{Data}}, P_{\text{Message}}) = (\text{CText, AuthTag}) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

$$\text{DecK } (N_{\text{Nonce}}, A_{\text{Data}}, \text{CText, AuthTag}) = (P_{\text{Message}}, \perp) \ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

K stands for the secret key, A_{Data} for associated data, P_{Message} for plaintext, and N_{Nonce} for the nonce, which is a one-of-a-kind non-repeating integer. These data are sent through the Enc encryption algorithm, which produces two outputs: CText, which stands for plaintext encryption, and AuthTag, which stands for authentication tag. With the same secret key, the nonce delivers numerous message blocks. The decryption technique uses N_{Nonce}, A_{Data}, CText, AuthTag, and K as inputs to obtain plaintext such as P_{Message} and AuthTag is checked before generating plaintext or message. We proposed a state-of-the-art AEAD algorithm, which provides attack resistance nonce misuse authenticated encryption cipher for multi-tenant security called as AES-GCM-SIV-FK.

In this paper, we provide recent work has been done by various researcher and discuss the reasons behind the paper in Section II. Our contribution and important of nonce misuse resistance are discussed in Section III. Section IV consist of proposed state-of-the-art AEAD algorithm encryption/decryption mechanism for multi-tenant security. Section V discusses the experimental setup and result analysis and comparison has been done in Section VI. Finally, we give an overall conclusion of about this paper based on experimental result.

2. Literature Review

Gueron et.al [1-2] submitted the Nonce-based key derivation authentication encoding cipher variations. The plaintext attack, authentication encryption, and nonce-misuse resistance aspects of block cipher modes are all examined in this work. The key-derivation function is also used in this paper to improve hardware performance. Existing authentication encryption techniques provide a solution for TLS webservers by adding proper nonce. It also looks at how well nonce-based authentication encryption techniques like TLS and QUIC perform.

Mennink et.al suggested pseudorandom permutation function for GCM, GCM-SIV and AES by using Block cipher designs. The author also argues that by using various cipher suites message encryption efficiency and security can be improved [3]. Shoni Gilboa et al. proposed the uses and advantage of random permutation from random process to block cipher [4]. Ashur et al. present a simple way to improve authentication encryption robustness [5]. The performance of authentication encryption (AE) algorithms like AES-GCM, CHACHA20, and POLY1305 is ensured in this work. This study also improves performance by using AES NI and when the AES NI instructions are not used. Antonie Joux investigated the GCM security offered during selected cipher text attack (CCA) and found an authentication failures in AE algorithms similar to the GCM NIST version [6]. Bock et al. offer many realistic deception attacks against GCM in TLS [7]. The researcher focuses on the latest cipher suit and QUIC web servers for security violence of AES GCM [8].

Advanced Encryption Standard uses counters with authentication message chaining for cipher block mode cryptography. These algorithms complexities are not explored, and the reader is mentioned in previous literature [9-12]. Rather than, we concentrate on AES-GCM utilized for most actual applications such as SSL/TLS. The architectures of many new algorithms, suggested as part of the current competition for AE, will be presented with AES-GCM as a basic architecture: Security, enforcement, and robust (CAESAR) competition.

The two algorithms AES-GCM synthetic IV and Deoxy are addressed in this paper, were given at the FPGA-based implementation in [13] and [14], respectively (SIV). In addition, this research provides various improvements and comparisons of other alternatives such as PRIMATE-APE and authenticating authentication tag-encoding pipelines (POET). A co-design method for the hardware-software is also offered to get resources that are effective for modern IoT systems.

3. Our Contribution This section discusses the limitations of current AEAD algorithms and our contribution in this paper.

Various problems have been identified in existing Authenticated Encryption algorithms which are listed below.

1) Existing algorithms in terms of single-user security are still vulnerable. So, we use multi-user security in the implementation of AE systems.

2) Several security characteristics are needed and missing in current AEAD algorithms, such as resistance to misrepresentation, resilience against plaintext leak, forbidden attack, side-channel attacks, and invisible salamanders attacks.

3) In the most extensively used AES-GCM algorithm [15], cryptographic analysis has discovered clusters of weak keys.

4) It is preferable to have greater performance and protection while maintaining the same level of security. This is particularly true when the number of devices in modern applications increases and their size reduces.

In the CAESAR competition, new authenticated algorithms AES-GCM-SIV was made to ease some of these concerns. This competitive challenge aims to identify an AE algorithm portfolio that offers advantages over AES-GCM and can be widely used [16-18]. In the literature, both hardware and software approaches are implemented independently. ATHENaGMU and the SUPERCOP team for the benchmarking software work together to bring different implementations on one platform [19]. They are trying to get them together.

3.1 Nonce-Misuse Resistance: According to the AEAD systems, nonces are crucial for the protection of symmetrical key encryption methods. These methods are designed to encrypt several communications with the same key while using different nonces. The nonce cannot thus be repeated with the same key. Although, as a result of numerous genuine implementations, this looks to be an easy requirement, it is not simple to achieve. Many ways are available to prevent nonces recurrence using the same key. One way is to refresh the keys regularly. On the other side, it will be difficult to exchange periodically secret keys between two people, with many apps not being able to do so. The most practical way would be to program and utilize a private key over the whole life in systems with IoT devices. The nonces are not repeated in the second approach that are derived from counters. The nonce values for large enough counters are dependent on each counter increase and are not repeated. The nonces begin to repeat, jeopardizing the algorithm's security, when counter is allowed to overflow due to flaws or any other types of attacks. Finally, the unique nonce are designed using large and high quality random number generators. It's impossible to ensure that this requirement is satisfied again, given the reducing sizes of gadgets. This has led to the popularity of algorithms that offer nonce misuse protection since IoT devices have become increasingly widespread.

3.3 Algorithms with Nonce-Misuse Resistance: Although the nonce is repeated, procedures guarantee that the encrypted message only contains restricted information, and it is almost impossible to recover the full plaintext [20]. The two kinds of algorithms with nonce misuse-resistant are complete and partially. While it is ideal to have total nonce-misuse resistance, partly unnecessary resistance in some situations may be necessary.

Our research analyses the performance and security of our proposed nonce misuse resistance AES-GCM-SIV-FK algorithm and compare it with existing AES-GCM and AES-GCM-SIV, AEAD algorithms, which are further examined below.

4. Our Proposed Model: The proposed AEAD algorithm has two significant differences. Firstly, the pseudorandom function generates four different keys instead of two keys. Secondly, rather than using the key and a nonce to initialize the AES-CTR encryption, a synthetic IV is computed from the caller's nonce and an encrypted POLYVAL hash of the plaintext. That means that even if the caller uses the same nonce twice, which is terrible, the synthetic IV and Message encryption key will still be different as long as the plaintext messages are different or same and makes the nonce resistant. When encrypting the message, we have to pass over it twice to compute the hash and once to encrypt it.

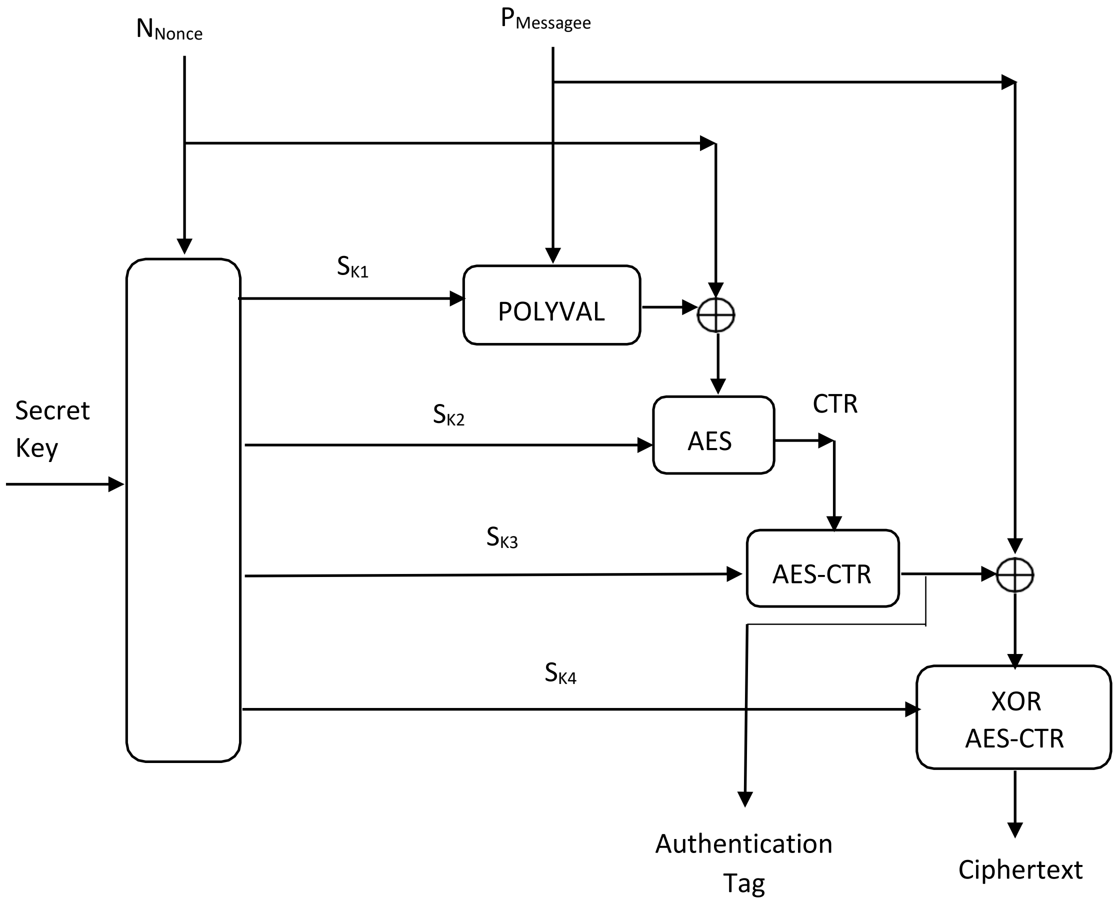


Fig. 1 Overall architecture Encryption/Decryption of AES-GCM-SIV-FK

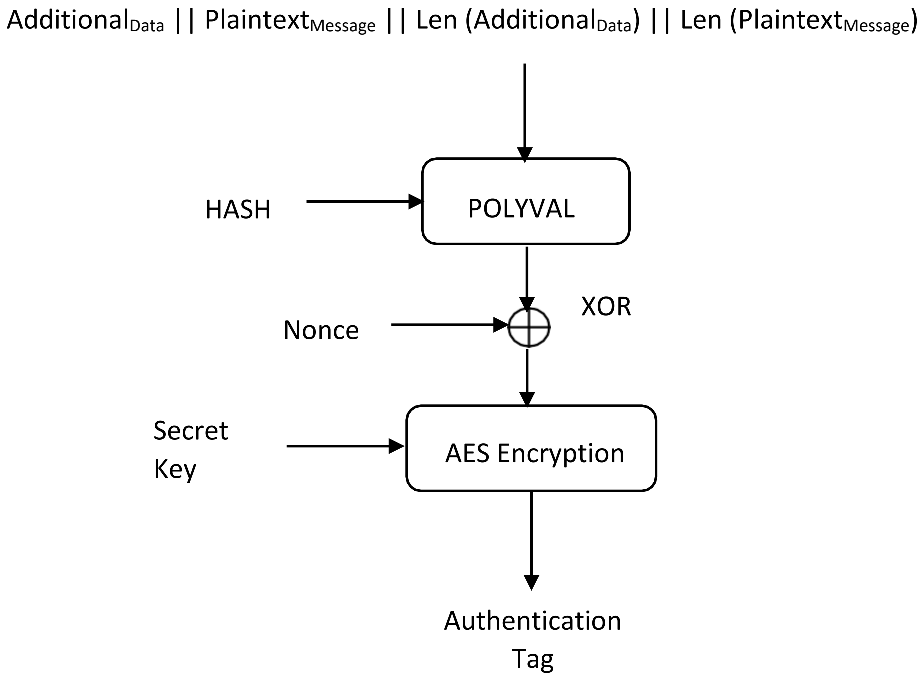

Additional_Data || Plaintext_Message || Len (Additional_Data) || Len (Plaintext_Message)
HASH
POLYVAL
Nonce
XOR
Secret Key
AES Encryption
Authentication Tag

Fig. 2 Polyval function to calculate the Hash

Table 1

Proposed Encryption Algorithm

```
Key: Secret Key($S_{Key}$)
Key size ($Key_{Size)}$ = 128-bit or 256-bit
If $S_{Key}$= 128, Apply then AES-128 else AES-256
Inputs: (Secret Key ($S_{Key}$), Nonce ($N_{Nonce}$), Additional Data ($A_{Data}$), Plaintext Message ($P_{Message}$))
Encryption:
($S_{K1}$, $S_{K2}$, $S_{K3}$, $S_{K4}$) = $Key_{Derivation}$($S_{Key}$, $N_{Nonce}$)
Ti = Turnc64 ($AESS_{K}$($N_{Nonce}$||[i]32)) then,
$S_{K1}$=$T_0$||$T_1$, $S_{K2}$=$T_1$||$T_2$, $S_{K3}$=$T_3$|$T_4$, $S_{K4}$=$T_4$||$T_5$
AuthTag = Hash-then-Encrypt [$H_{Hash}$, $E_{Encrypt}$] $S_{K1}$, $S_{K2}$, $S_{K3}$, $S_{K4}$ ($N_{Nonce}$, $A_{AuthTag}$ , $P_{Message}$)
$E_{Encrypt}S_{K2}$ ($H_{Hash}S_{K1}$ ($A_{Authtag}$, $P_{Message}$) + $N_{Nonce}$), then
$H_{Hash}S_{K1}$ ($A_{AuthTag}$, $P_{Message}$)= 0||$Trunc_{n-1}$ (POLYVAL($S_{K1}$, Encode($A_{AuthTag}$, $P_{Message}$))
CipherText = [[CTR[AES]$S_{K2}$]$S_{K3}$ ($Trunc_{n-1}$(T), $P_{Message}$)]$S_{K4}$
```

Table 2

Proposed Decryption Algorithm

```
Key: Secret Key($S_{Key}$)
Key size ($Key_{Size}$) = 128-bit or 256-bit
If $S_{Key}$= 128, Apply then AES-128 else AES-256
Inputs: (Secret Key ($S_{Key}$), Nonce ($N_{Nonce}$), Additional Data ($A_{Data}$), Plaintext Message ($P_{Message}$))
Decryption:
($S_{K1}$, $S_{K2}$, $S_{K3}$, $S_{K4}$) = $Key_{Derivation}$ ($S_{Key}$, $N_{Nonce}$)
Ti = Turnc64 ($AESS_{K}$($N_{Nonce}$||[i]32)) then,
$S_{K1}$=$T_0$||$T_1$, $S_{K2}$=$T_1$||$T_2$, $S_{K3}$=$T_3$|$T_4$, $S_{K4}$=$T_4$||$T_5$
PlainText = [AES[CTR[Ciphertext]$S_{K4}$]$S_{K3}$]$S_{K2}$
```

We have also used POLYVAL instead of GHASH (shown in fig. 2) while creating AES-GCM-SIV-FK (In AES-GCM, this is the hash function used shown in fig. 1). Changes have been made to ensure compatibility with the bit order in the AES-GCM byte (it does not follow the AES byte as a cipher that operates by definition in the 16-byte state). After AES-GCM was explained algebraically, we showed that GHASH is defined in a finite field by a series of operations of the type a * b * x ^ {127}. We can express the operation as (a * x) * (b * x) ^ {128} for AES-GCM with a fixed key (that is the hatch key). Since a * x is H * x, more multiplication is needed. The proposed state-of-the-art AEAD algorithm for attack resistance nonce misuse authenticated encryption cipher for multi-tenant security (AES-GCM-SIV-FK) algorithms is presented in Table 1 and Table 2, respectively.

5. Experimental Setup:

In this paper, we perform two experiments on the basics of execution time to measure the performance of our proposed algorithm AES-GCM-SIV-FK with AES-GCM and AES-GCM-SIV in terms of encryption time, decryption time, and speed (megabytes per second) on the Processor Intel(R) Core ™ i3-3217U CPU with 1.80 GHz clock rate and 8GB RAM. To compare the above-said algorithms, we implement the algorithm in python language with the same hardware support.

6. Experimental Result and Analysis:

Comparison among AEAD Algorithms: The comparison of AEAD algorithms is made based on execution time which is as follows:-

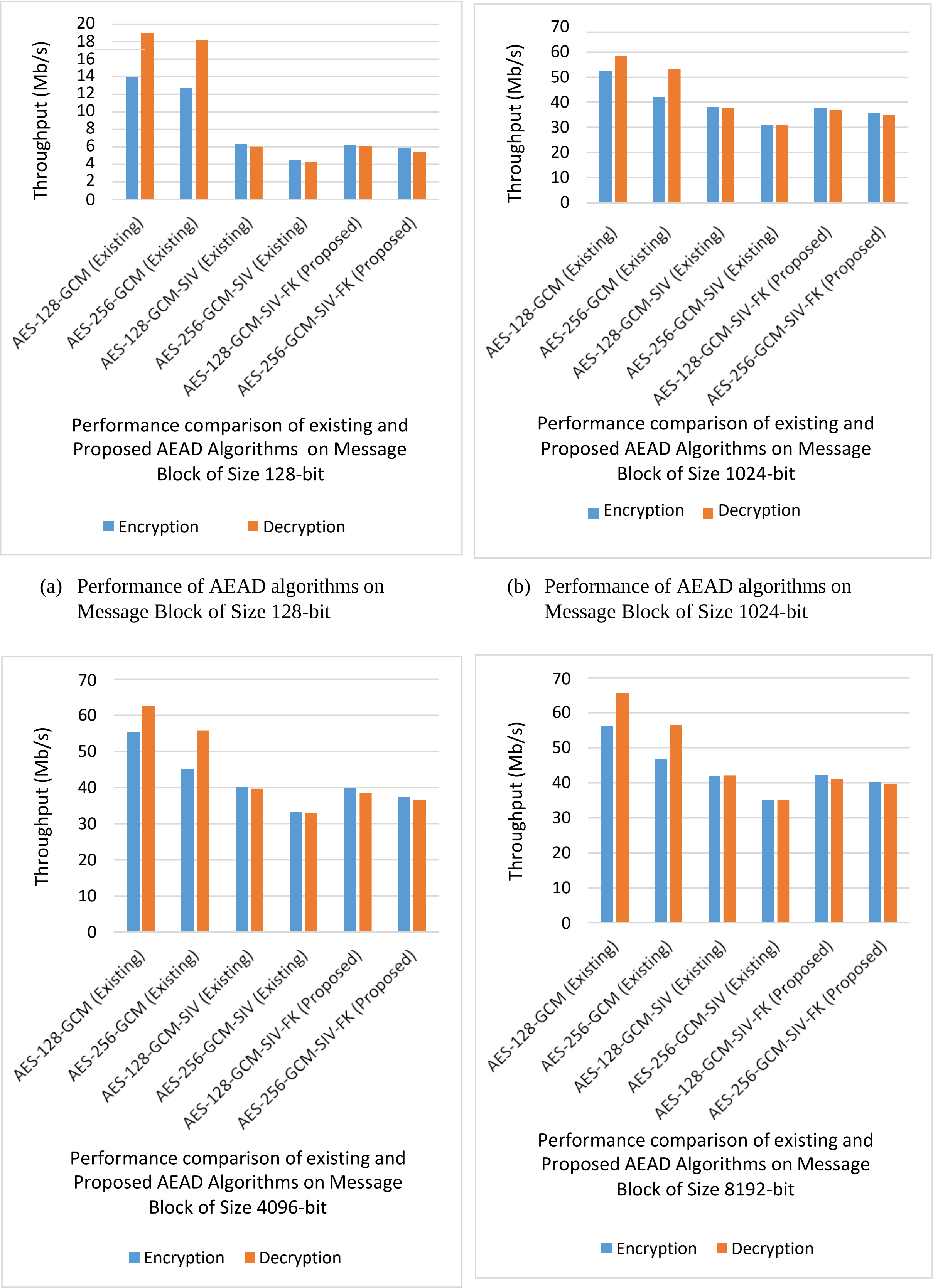
Execution Time: Three measurements of runtime during the AES-GCM-SIV-FK (Key size 128 bit and 256 bit) tests were made as a symmetrical algorithm: the time taken to run the entire cryptographical algorithm, the time taken solely for encryption, and only for decryption. Table 3 depicts the results of the complete algorithm execution times for the encryption and decryption procedures, respectively. The AES-GCM-SIV-FK performs well compared to the speed of AES-GCM and AES-GCM-SIV shown in Table 3. We compared the AES-GCM-SIV-FK with AES-GCM-SIV and AES-GCM shown in Fig 3. Which does not include the AES-NI and CLMUL instructions (officially termed "PCLMULQDQ"). All algorithms performance is measured in megabytes per second (Mb/s).

Proposed By	**AEAD Algorithms**	**Message Size (128-bit)**	**Message Size (1024-bit)**	**Message Size (4096-bit)**	**Message Size (8192-bit)**
Salowey, J. et.al [9]	AES-128-GCM Encryption [9]	14	52.3	55.4	56.2
	AES-128-GCM Decryption [9]	19	58.2	62.6	65.7
	AES-256-GCM Encryption [9]	12.7	42.2	45	46.8
	AES-256-GCM Decryption [9]	18.2	53.4	55.8	56.5
Gueron, S. et.al [2]	AES-128-GCM-SIV Encryption [2]	6.3	38.1	40.2	41.9
	AES-128-GCM-SIV Decryption [2]	6.0	37.7	39.7	42.1
	AES-256-GCM-SIV Encryption [2]	4.4	30.9	33.3	35.1
	AES-256-GCM-SIV Decryption [2]	4.3	30.9	33.1	35.2
Proposed	AES-128-GCM-SIV-FK Encryption (Proposed)	6.2	37.6	39.8	42.1
	AES-128-GCM-SIV-FK Decryption (Proposed)	6.1	36.9	38.5	41.1
	AES-256-GCM-SIV-FK Encryption (Proposed)	5.8	35.9	37.3	40.2
	AES-256-GCM-SIV-FK Decryption (Proposed)	5.4	34.8	36.7	39.6

Table 3. Performance Comparison of existing and proposed AEAD Algorithms with key size 128/256 bit (Mb/s)

The encryption speed of AES-GCM-SIV-FK is 2.25 times faster than AES-GCM-128-bit (x2.18 times faster for AES-GCM-256-bit) and 1.01 times faster than AES-GCM-SIV-128-bit (x1.31 times slower for AES-GCM-SIV-256-bit) for extremely short messages (a single block of 128 bit) shown in Fig 3. (a), whereas encryption of AES-GCM-SIV-FK is 1.39 times faster than AES-GCM-128-bit (x1.17 between AES-GCM-256-bit) and 1.01 times faster than AES-GCM-SIV-128-bit (x1.16 times slower for AES-GCM-SIV-256-bit) for messages (a single block of 1024 bit) shown in Fig 3. (b), whereas AES-GCM-SIV-FK is 1.39 times faster than AES-GCM-128-bit (x1.20 times faster for AES-GCM-256-bit) and 1.01 time faster than AES-GCM-SIV-128-bit (x1.12 times slower for AES-GCM-SIV-256-bit) for messages (a single block of 4096 bit) shown in Fig 3. (c), whereas AES-GCM-SIV-FK is 1.33 times faster than AES-GCM-128-bit (x1.16 times faster for AES-GCM-256-bit) and 1.0 times slower than AES-GCM-SIV-128-bit (x1.14 times slower for AES-GCM-SIV-256-bit) for messages (a single block of 8192 bit) shown in Fig 3. (d).

Similarly the decryption speed of AES-GCM-SIV-FK is 3.11 times faster than AES-GCM-128-bit (x3.37 times faster for AES-GCM-256-bit) and 1.01 times slower than AES-GCM-SIV-128-bit(x1.25 times slower for AES-GCM-SIV-256-bit) for extremely short messages (a single block of 128 bit), whereas AES-GCM-SIV-FK is 1.57 times faster than AES-GCM-128-bit (x1.53 times faster for AES-GCM-256-bit) and 1.02 times faster than AES-GCM-SIV-128-bit (x1.12 times slower for AES-GCM-SIV-256-bit) for messages (a single block of 1024 bit) whereas AES-GCM-SIV-FK is 1.62 times faster than AES-GCM-128-bit (x1.52 times faster



(a) Performance of AEAD algorithms on Message Block of Size 128-bit

(b) Performance of AEAD algorithms on Message Block of Size 1024-bit

(d) Performance of AEAD algorithms on Message Block of Size 4096-bit

(c) Performance of AEAD algorithms on Message Block of Size 8192-bit

Fig. 3 Performance Comparison of different AEAD algorithms on different Message Block size

whereas AES-GCM-SIV-FK is 1.59 times faster than AES-GCM-128-bit (x1.42 times faster for AES-GCM-256-bit) and 1.02 times faster than AES-GCM-SIV-128-bit (x1.12 times slower for AES-GCM-SIV-256-bit) for messages (a single block of 8192 bit).

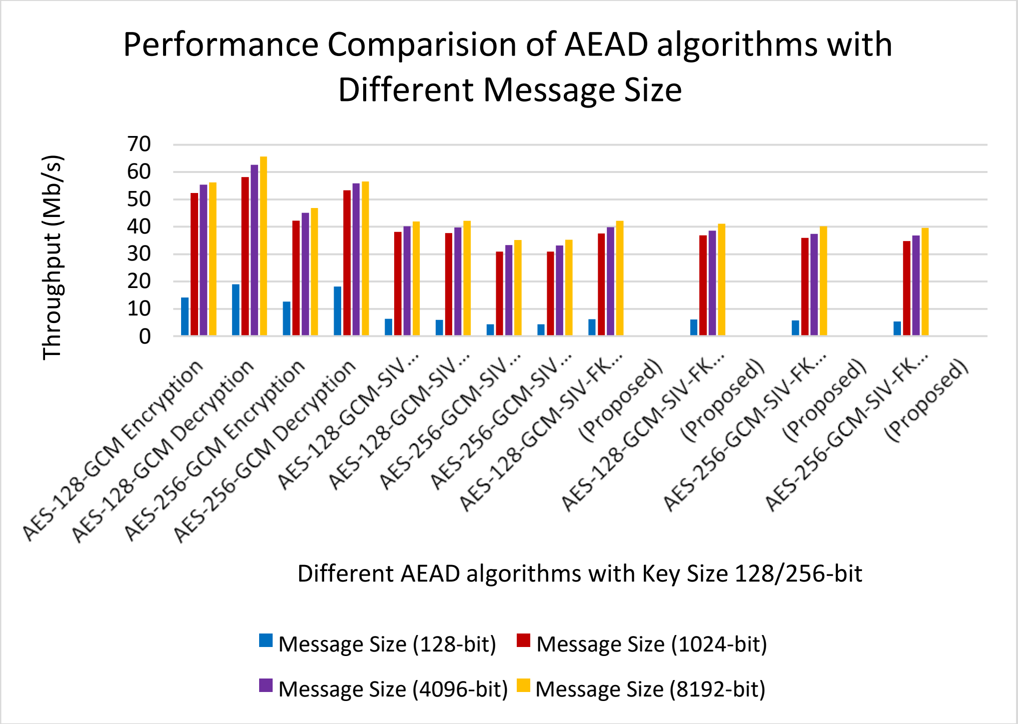


Fig 4. Performance comparison of AEAD algorithms on different message block size

The overall performance comparison of different AEAD algorithms on different message block of size 128/1024/4096/8192-bit shown in Fig 4. In the lack of AES-NI, the implementation does not utilize the authenticator hash computation, interleaving optimization with the AES operations.

Conclusion:

In high-speed communication, confidentiality and authenticity of data is achieved by AEAD ciphers. A state-of-the-art AEAD algorithm is proposed, which provides attack resistance nonce misuse authenticated encryption cipher for multi-tenant security. This paper properly investigates the performance and security analysis of existing AES-GCM and AES-GCM-SIV AEAD algorithms based on execution time for encryption/decryption for TLS web servers. Our proposed algorithm doesn't suffers from nonce misuse similar to AES-GCM-SIV nonce can be reused across an unbounded number of users in verification queries which degrade the multi-tenant security. So message encryption key will still be different as long as the plaintext messages are different or same and makes the nonce resistant. The encryption speed of our scheme is faster than AES-GCM and AES-GCM-SIV while the decryption speed is faster than AES-GCM and slower than AES-GCM-SIV. The performance is observed on Intel processor which doesn't support hardware acceleration. So in future our scheme can be fine-tuned to get better results with hardware acceleration for FPGA.

References

1. Gueron, S. and Lindell, Y., 2015, October. GCM-SIV: full nonce misuse-resistant authenticated encryption at under one cycle per byte. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (pp. 109-119).

2. Gueron, S., Langley, A., & Lindell, Y. (2019). AES-GCM-SIV: Nonce misuse-resistant authenticated encryption. Internet Engineering Task Force-IETF, Request for Comments, 8452.

3. Mennink, B. and Neves, S., 2017. Optimal PRFs from block cipher designs. IACR Transactions on Symmetric Cryptology, pp.228-252.

4. Gilboa, S. and Gueron, S., 2015. Distinguishing a truncated random permutation from a random function. arXiv preprint arXiv:1508.00462.

5. Ashur, T., Dunkelman, O. and Luykx, A., 2017, August. Boosting authenticated encryption robustness with minimal modifications. In Annual International Cryptology Conference (pp. 3-33). Springer, Cham.

6. Joux, A., 2006. Authentication failures in the NIST version of GCM. NIST Comment, p.3.

7. Böck, H., Zauner, A., Devlin, S., Somorovsky, J. and Jovanovic, P., 2016. Nonce-Disrespecting Adversaries: Practical Forgery Attacks on {GCM} in {TLS}. In 10th {USENIX} Workshop on Offensive Technologies ({WOOT} 16).

8. Arunkumar, B. and Kousalya, G., 2020. Nonce reuse/misuse resistance authentication encryption schemes for modern TLS cipher suites and QUIC based web servers. Journal of Intelligent & Fuzzy Systems, 38(5), pp.6483-6493.

9. Salowey, J., Choudhury, A. and McGrew, D., 2008. AES Galois Counter Mode (GCM) cipher suites for TLS. Request for Comments, 5288.

10. McGrew, D. and Bailey, D., 2012. Aes-ccm cipher suites for transport layer security (tls) (pp. 1-6). RFC 6655, July.

11. Bellare, M., Rogaway, P., and Wagner, D., 2004, February. The EAX mode of operation. In International Workshop on Fast Software Encryption (pp. 389-407). Springer, Berlin, Heidelberg.

12. Rogaway, P., Bellare, M. and Black, J., 2003. OCB: A block-cipher mode of operation for efficient authenticated encryption. ACM Transactions on Information and System Security (TISSEC), 6(3), pp.365-403.

13. Koteshwara, S., Das, A. and Parhi, K.K., 2017, May. FPGA implementation and comparison of AES-GCM and Deoxys authenticated encryption schemes. In 2017 IEEE International symposium on circuits and systems (ISCAS) (pp. 1-4). IEEE.

14. Koteshwara, S., Das, A. and Parhi, K.K., 2017, October. Performance comparison of AES-GCM-SIV and AES-GCM algorithms for authenticated encryption on FPGA platforms. In 2017 51st Asilomar Conference on Signals, Systems, and Computers (pp. 1331-1336). IEEE.

15. Handschuh, H. and Preneel, B., 2008, August. Key-recovery attacks on universal hash function-based MAC algorithms. In Annual International Cryptology Conference (pp. 144-161). Springer, Berlin, Heidelberg.

16. Bernstein, D.J., 2014. Caesar: Competition for authenticated encryption: Security, applicability, and robustness.

17. Abed, F., Forler, C. and Lucks, S., 2014. General overview of the first-round CAESAR candidates for authenticated encryption. IACR Cryptol. ePrint, Tech. Rep, 792, p.2014.

18. Koteshwara, S. and Das, A., 2017. Comparative study of authenticated encryption targeting lightweight IoT applications. IEEE Design & Test, 34(4), pp.26-33.

19. Gaj, K. and Team, A., 2016. Athena: Automated Tool for Hardware Evaluation.

20. Rogaway, P. and Shrimpton, T., 2007, August. Deterministic authenticated-encryption. In Advances in Cryptology– EUROCRYPT (Vol. 6).