Reconfigurable Integrated Cryptosystem for Secure Data Exchanges Between Fog Computing and Cloud Computing Platforms

Abiy Tadesse Abebe[1(✉)], Yalemzewd Negash Shiferaw[1], and P. G. V. Suresh Kumar[2]

[1] Addis Ababa Institute of Technology, AAU, Addis Ababa, Ethiopia
abiytds@yahoo.com, yalemzewdn@yahoo.com
[2] Ambo University, Ambo, Ethiopia
pendemsuresh@gmail.com

Abstract. This study is aimed to propose a cryptosystem which integrates only two algorithms for secure data transmissions during fog nodes to cloud server communications. It provides a method of authenticated key distribution and authenticated encryption with robust and multiple crypto services. It is optimized for high throughput achievement based on FPGA to improve the efficiency of the existing hybrid cryptosystems and integrated encryption schemes which incorporated many independent algorithms for strong security. The separate keys which are needed for each component algorithm leading to extra key management and key storage requirements and the overall hardware complexity with increased computation cost are some of the limitations of the existing methods. The implementation outcomes show the efficiency, enhanced throughput, and reasonable resource utilization of the proposed method compared to the existing reported outcomes. It can be suitable for securing data exchanges among high performance computing environments including secure communications between fog computing layer and central cloud which require high speed cryptosystems with strong security and lower latency.

Keywords: AEGIS · Authenticated encryption · FPGA · Integrated cryptosystem · Key distribution · RSA

1 Introduction

For effective utilization of the advantages of the modern cryptographic mechanisms such as symmetric key and asymmetric key algorithms, hash functions, Message Authentication Code (MAC) generators, and digital signature algorithms [1], many researchers have proposed various methods. They have increased the capabilities of the existing algorithms in terms of performance and security. For instance, the authenticated encryption algorithms [2, 3] which can provide data confidentiality, data integrity, and data origin authentication services simultaneously using only one algorithm are improvements over symmetric key algorithms. Similarly, signcryption methods [4, 5] which can provide data confidentiality and digital signature crypto services simultaneously using only one

N. G. Habtu et al. (Eds.): ICAST 2019, LNICST 308, pp. 492–501, 2020.
https://doi.org/10.1007/978-3-030-43690-2_35

algorithm are improvements over the public key algorithms. Despite their advantages, the authenticated encryption and signcryption methods generally share the inherent limitations of symmetric and asymmetric key algorithms. Symmetric key algorithms are efficient, but are limited by the lack of key distribution capability requiring sharing of secret key before starting secret communications. The asymmetric key algorithms have circumvented the need of sharing of secret key by providing key pair, one for encryption and another for decryption. But, their performance is generally considered as slower because of the intensive mathematical operations needed for encryption and decryption processes.

To provide more crypto services for strong security, various researchers have proposed hybrid cryptosystems [6, 7] and integrated encryption schemes such as Diffie-Hellman Integrated Encryption Scheme (DHIES) [8] and Elliptic Curve Integrated Encryption Scheme (ECIES) [9] by integrating different crypto mechanisms. The main goal of these methods has been to effectively utilize the advantages of the different cryptographic mechanisms particularly the symmetric key and asymmetric key algorithms so that the combined cryptosystem could use the symmetric key algorithm for large amount of data encryption and decryption while using the asymmetric key algorithm for key distribution. As a result, cryptosystems as efficient as the symmetric key algorithms and as secured as asymmetric key algorithms have been developed in addition to the services of hashing, MAC generation, and digital signature.

Secure information exchanges between central cloud servers and the recently introduced intermediate layer called fog computing layer, which is sited between the cloud and the Internet of Things (IoT) devices for efficient communication, require high speed cryptosystems which can provide high throughput and low latency along with major cryptographic services for strong security, as the attack surface is wider [10]. Therefore, hardware based implementations of cryptosystems are useful to provide high speed performance as well as physical security. Depending on the application scenarios, FPGA based implementations are preferable as they can be reconfigured based on the contemporary attack risks and are also as flexible as software for implementations and as high speed as hardware in terms of performance [11]. In this work, our aim is to effectively utilize and optimize the existing crypto mechanisms and provide the major cryptographic security services along with better performance suitable for the intended application by integrating only few number of algorithms while saving extra costs of key management, key storage, and hardware complexity compared to existing similar FPGA based implementations.

The rest of the paper is organized as follows: Sect. 2 describes related works. Section 3 describes the background of the algorithms used in this work. Section 4 explains the proposed method. Implementation approaches are described in Sect. 5. Section 6 presents the implementation outcomes. Finally, Sect. 5 concludes the paper.

2 Related Works

Several researchers have proposed various techniques to increase the capabilities of the existing standard cryptographic algorithms in terms of performance and security. In addition to authenticated encryption algorithms [2, 3] and signcryption methods [4, 5],

hybrid cryptosystems [6, 7] and integrated encryption schemes have been proposed to enhance the efficiencies of such schemes, and at the same time, to remove their inherent limitations. Integrated Encryption Schemes such as Diffie Hellman Integrated Encryption Scheme (DHIES) [8] and Elliptic Curve Integrated Encryption Scheme (ECIES) [9] are standardized methods integrating different cryptographic mechanisms such as hash function, message authentication code algorithm, Key Derivation Function (KDF), in addition to the encryption algorithm and key exchange protocol [8, 9]. Moreover, some hybrid cryptosystems have also included digital signature algorithm [7]. Since these cryptosystems combined four or more different algorithms to provide more cryptographic security services such as data confidentiality, data integrity, authentication, non-repudiation, etc., where each component algorithm requiring a separate key for security purpose [8, 9], the key management, key storage, and compatibility issues as well as the overall storage space requirement with significant hardware complexity of the system need critical considerations.

The FPGA based integrated cryptosystem proposed in this work incorporates a public key scheme and an authenticated encryption algorithm to provide authenticated key distribution and authenticated information exchange for secure communications between fog computing and the cloud using only two algorithms for improved performance, lower latency, and reduced hardware complexity.

3 Background

3.1 AEGIS-128 Algorithm

AEGIS-128 algorithm [12] is an Advanced Encryption Standard (AES) [13] based algorithm which uses the AES round functions such as Sub Bytes, Shift Rows, Mix Columns, and Add Round Key, excluding the last round. AEGIS-128 algorithm uses 128 bits key and 128 bits Initialization Vector (IV) to perform authenticated encryption and authenticated decryption processes. It can process less than 2^{64} lengths of plaintext and the associated data. The recommended length of the tag to be used for authentication is 128 bits, though lesser lengths of tags can also be used.

The algorithm uses the round functions to update the 640 bits (80 bytes) of state, S_i, with 128 bits (16 bytes) of data blocks, m_i, using its state update function such that: $S_{i+1} = StateUpdate128(S_i, m_i)$ as shown in Fig. 1. In Fig. 1, **R** indicates the AES encryption round function not being XORed with the round key. w is a temporary 16-byte word. The process can also be expressed step by step as follows:

$$S_{i+1,0} = AESRound\big(S_{i,4}, S_{i,0} \oplus m_i\big);$$

$$S_{i+1,1} = AESRound\big(S_{i,0}, S_{i,1}\big);$$

$$S_{i+1,2} = AESRound\big(S_{i,1}, S_{i,2}\big);$$

$$S_{i+1,3} = AESRound\big(S_{i,2}, S_{i,3}\big);$$

$$S_{i+1,4} = AESRound\big(S_{i,3}, S_{i,4}\big);$$

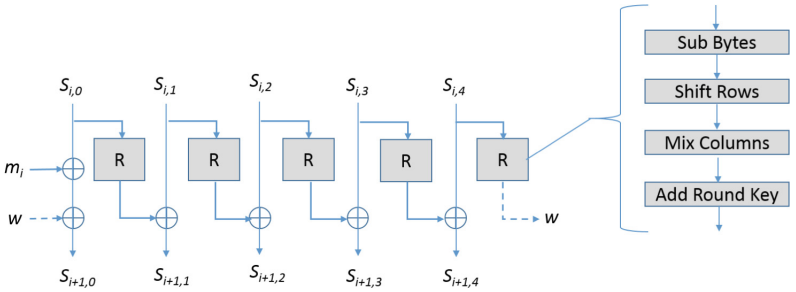

Fig. 1. The state update function of AEGIS-128

AEGIS-128 performs different execution steps including initialization, process of the authenticated data, encryption, and finalization. The decryption process follows the reverse order of the encryption, and requires the exact values of key size, *IV* size, and tag size for decryption and verification tasks. The details of AEGIS-128 algorithm can be found in [12]. It is designed for the security of high performance applications.

3.2 RSA Algorithm

The RSA algorithm is a public key crypto scheme which was proposed by Rivest, Shamir, and Adleman in 1978 [14]. Its security strength relies on the factorization of large integer numbers, and as the key size increases, the security strength of the RSA algorithm also increases [14].

RSA uses two keys, a public key and a private key, which are mathematically related, but are difficult to obtain one by knowing only the other. Generation of the RSA keys is done based on the following key steps:

- Selection of two large secret primes p and q.
- Calculation of a public modulus $n = p.q$.
- Computation of $\phi(n) = (p-1).(q-1)$, where, $\phi(n)$ is called Euler's totient function.
- Selecting the public exponent $e \in \{1, 2, \ldots, \phi(n)-1\}$, such that e is relatively prime to $\phi(n)$, i.e., $GCD(e, \phi(n)) = 1$.
- Computation of the private key, d, such that $d.e \equiv 1 \, mod \, \phi(n)$, that means $d = e^{-1} \, mod \, \phi(n)$.

The public key generated is: $Pub_{key} = (e, n)$ and the private key is: $Priv_{key} = (d, n)$.

Encryption and Decryption processes are performed by applying modular exponentiation operation as shown by Eqs. (1) and (2), where, plaintext block M is encrypted to a ciphertext block C by Eq. (1) and the plaintext block M is obtained by Eq. (2) [14]:

$$C = M^e mod \, n \tag{1}$$

$$M = C^d mod \, n \tag{2}$$

4 The Proposed Method

For the security of the data exchanges during the communication between fog layer to the cloud server, we integrated AEGIS-128, a high speed authenticated encryption algorithm, and RSA, a public key algorithm. The AEGIS-128 can provide data confidentiality, data integrity, and authentication crypto services simultaneously, whereas RSA can provide encryption and digital signature security services simultaneously. Using only these two algorithms, the required major cryptographic security services can be obtained. The AEGIS-128 is intended for authenticated encryption/decryption of the actual large amount of data with high speed, and RSA is used for encryption/decryption of the secret key (the small amount of data) for key distribution with crypto services including digital signature, validation of data integrity, and verification of authenticity.

Figure 2 shows the structure of the proposed FPGA based integrated cryptosystem. Before any secure communication is started, it is assumed that the two communicating parties are agreed on the public parameters and have generated their independent key pair (a private key and a public key), and have exchanged the public keys through a trusted certificate authority or any other agreed secure method, but keeping their individual private keys secret. As shown in Fig. 2(a), at the sending end, a randomly generated symmetric key is used for encryption of the large amount data (plaintext) using AEGIS-128 algorithm. This same symmetric key is encrypted and signed by the public key of the recipient and the private key of the sender respectively, using the RSA algorithm. The ciphertext and MAC outputs of AEGIS-128 and the encrypted and signed key of RSA are then sent to the receiving end. At the receiving end, as shown in Fig. 2(b), RSA first verifies the signature using the public key of the sender. It performs decryption of the encrypted key using the private key of the recipient if and only if the signature is true; otherwise, it discards the received data. When the signature is valid and the symmetric key is recovered, then, RSA supplies the symmetric key to AEGIS-128. When the symmetric key is ready, the AEGIS starts the decryption process after validating the authenticity of the data origin and checking data integrity by comparing the received authentication tag $\boldsymbol{T}$ and the calculated authentication tag $\boldsymbol{T'}$ to ensure the validity of the received data. The plaintext will be obtained, accepted and further processed if and only if the two tags $\boldsymbol{T}$ and $\boldsymbol{T'}$ are equal; otherwise, error signal will be generated indicating invalid data reception, and then, the data will be immediately discarded before being utilized.

In this system, the RSA signature doesn't require hashing since it encrypts and signs a symmetric key which has only 128 bit key length (small data). The signature is used to ensure that the symmetric key is sent by a claimed sender. The ciphertext is not signed by the RSA since it is considered to be large amount data and effects the performance of the system. Instead, the AEGIS-128 performs encryption/decryption tasks of the large data and validates the authenticity of the data and checks the data integrity. Any modification which might be made by man-in-the-middle on the transported encrypted key or the ciphertext would be detected by RSA and AEGIS-128 respectively, since both schemes can validate authenticity and check data integrity; therefore, the data will be discarded if it is invalid.

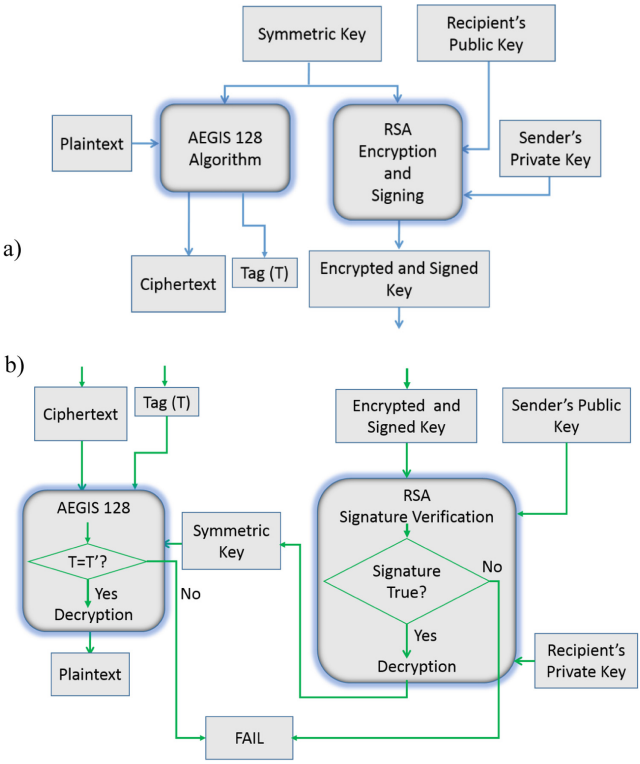

Fig. 2. Schematic structure of the proposed integrated cryptosystem: (a) Sending end (b) Receiving end

5 Implementation Approaches

The integrated cryptosystem composing AEGIS-128 authenticated encryption algorithm and, RSA public key scheme, has been implemented on Xilinx Virtex 5, Virtex 7, and Virtex II FPGA devices and synthesized using Xilinx ISE 14.5, Vivado Design Suite 2017.2, and Xilinx ISE 10.1, respectively. VHDL was used as a hardware description language.

As AEGIS-128 is an Advance Encryption Standard (AES) based authenticated encryption algorithm, for improving its throughput performance, we applied pipelined AES on the round operations for concurrent processing of the state in a clock cycle. In pipelining optimization technique [15], construction of the pipeline is performed by inserting registers at each round as shown in Fig. 3 [15]. The number of the pipeline steps, denoted by K, decides how many rounds should be executed in parallel to speed up the process. It is called a fully pipelined architecture if K is made to be equal to the total number of rounds in the algorithm. The required area and the target latency of the pipelining architecture are related to the applied number of K. By processing multiple blocks of date at the same time, the technique can increase the encryption and decryption

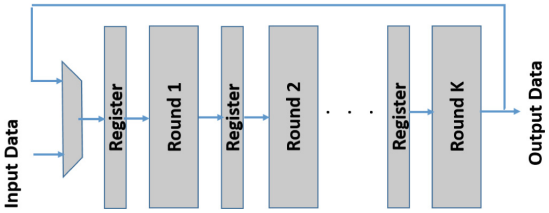

Fig. 3. Pipeline architecture

processes. In AEGIS-128 algorithm, five AES rounds are used. In this work, AEGIS-128 with 128 bits key and 128 bits blocks of data are implemented.

As the pipeline method processes encryption, decryption, or authentication in one clock cycle, the general throughput can be calculated as shown by Eq. (3):

$$Throughput\ (TP) = Max.Frequency\ x\ 128 \tag{3}$$

For RSA, as it involves modular exponentiation, Montgomery algorithm [16] is used to perform the modular multiplication. In this work, RSA is used only for encrypt and sign at the sending end, and then, verify and decrypt at the receiving end. This is performed as follows: let the sender wants to send an encrypted and signed message to a recipient. Let M denotes the message, d_s the sender's private key, and e_r the recipient's public key. Then, the sender encrypts the message M using the recipient's public key, e_r, as: $C = M^{e_r} \bmod n$, where, C is the ciphertext. The sender now signs a signature S by computing: $S = C^{d_s} \bmod n$. The sender then sends C and S to the receiving end. It is also possible to encrypt the signed result again by the public key of the recipient for more security (encrypt-sign-encrypt) as: $S_c = S^{e_r} \bmod n$. At the receiving end, verification of the signature and then decryption of the ciphertext will be performed as follows: let d_r be a private key of the recipient, and e_s, the public key of the sender. Verification of the signature is performed by computing: $S' = S^{e_s} \bmod n$, and if $C = S'$, the data is authentic, and then, decryption of the ciphertext will be performed to get the original data as: $M = C^{d_r} \bmod n$. Otherwise (if $C \neq S'$), all the received data will be discarded. When encrypt-sign-encrypt method is used, first, decryption of S_c is performed using private key of the recipient as: $S = S_c^{d_r} \bmod n$, keeping the other steps similar. For this work, RSA with 1024 bits modulus is used and integrated with AEGIS-128.

At the sending end, an '*Encrypt*' signal is asserted high so that both algorithms could read the symmetric key as input. A simple FSM control mechanism is used to synchronize the work of the two algorithms as an integrated system during decryption as shown in Fig. 4. A '*Start*' signal, which is in low state, setting the system in idle condition, will be asserted high when data is received. Then, RSA signature verification process will be started. When signature is true, the encrypted key will be decrypted. Then, the '*Sign*' signal will be high and the symmetric key will be supplied to AEGIS-128 decryption process making the AEGIS-128 '*Busy*' signal high. Otherwise, the key will be discarded and the system will return to the idle state again. When AEGIS-128 is busy with decryption process, the '*Busy*' signal is asserted high and waits until decryption process finishes. When the decryption process ends, the '*Busy*' signal becomes low and the system goes to idle state.

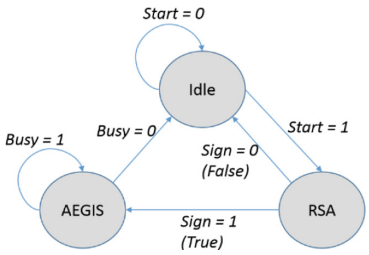

Fig. 4. A simple FSM control mechanism

6 Results

The implementation of the proposed integrated cryptosystem is performed on Xilinx Virtex 5, Virtex 7, and Virtex II FPGA devices for the purpose of comparison with the existing works. Tables 1, 2, and 3 show the performance comparisons of the proposed work with the existing reported outcomes. The FPGA resources consumed and the maximum frequency as well as the throughput achieved are presented in Table 1 compared to the results in [17]. In this case, the optimization method used by [17] is LUT based, whereas the present work used pipelining method. The implementation target of the present work is high throughput as mathematically expressed by Eq. (3). Therefore, for Virtex 5 device, our work achieved a throughput of 41.55 Gbps with area cost of 5478 slices and 4 BRAMs. For Virtex 7 device, this work achieved lesser area and better throughput compared to the work of [18] for the same pipelined optimization approach as shown in Table 2. Comparison of consumed FPGA resources on Virtex II device for the whole integrated cryptosystem of the present work with the ECIES implementation results reported by [19] is shown in Table 3. Smaller space utilization (slices and BRAMs) are shown in Table 3 for the present work compared to the reported outcomes of [19].

Table 1. Performance Comparison

Author	Target Device	Design	Slices	BRAM	Freq. (MHz)	Thrpt. (Gbps)
This work	Virtex 5	Pipelined	5478	4	324.6	41.55
[17]	Virtex 5	LUT based	1391	0	156.5	20.03

Contributions. The proposed method extends the capability of the authenticated encryption algorithm by integrating it with authenticated key distribution mechanism for strong security. It also increases the effectiveness of hybrid cryptosystems or integrated encryption schemes by reducing the extra key management and key storage requirements as well as hardware complexity. Integration of only two algorithms and providing

Table 2. Performance Comparison

Author	Target Device	Design	Slices	Thrpt. (Mbps)
This work	Virtex 7	Pipelined	9306	89354
[18]	Virtex 7	Pipelined	10610	88564

Table 3. Performance Comparison

Author	Target Device	Design	Slices	BRAM
This work	Virtex II	Hybrid of AEGIS + RSA	14572	4
	Virtex II	ECIES	21194	20

multiple crypto services for strong security as well as FPGA implementation and optimization of it for high throughput suitable for fog – cloud secure communications is another contribution of this study.

7 Conclusions and Future Work

An integrated cryptosystem using AEGIS-128 for authenticated encryption of actual data and RSA for authenticated key distribution by encrypting and signing of the symmetric key, has been implemented on different FPGA devices for fog-cloud security. The proposed cryptosystem has used only two algorithms providing multiple crypto services while saving extra key management and key storage requirements, as well as reducing computation costs. The implementation results show enhanced throughput achievement and reasonable FPGA resource utilization as compared to the existing reported outcomes of similar works. By applying additional optimization methods, we will further improve it to get smaller area and higher throughput.

References

1. Stinson, D.R., Paterson, M.B.: Cryptography Theory and Practice, 4th edn, pp. 1–9. CRC Press, Boca Raton (2019)
2. McGrew, D., Viega, J.: The Galois/Counter Mode of operation (GCM). Submission to NIST, May 2005
3. Koteshwara, S., Das, A.: Comparative study of authenticated encryption targeting lightweight IoT applications. IEEE Des. Test **34**(4), 26–33 (2017)

4. Pang, L., Kou, M., Wei, M., Li, H.: Efficient anonymous certificateless multi-receiver signcryption scheme without bilinear pairings. IEEE Access **6**, 78123–78135 (2018). https://doi.org/10.1109/access.2018.2884798
5. Zia, M., Ali, R.: Cryptanalysis and improvement of blind signcryption scheme based on elliptic curve. Electron. Lett. (2019). https://doi.org/10.1049/el.2019.0032
6. Gutub, A.A, Khan, F.A.: Hybrid crypto hardware utilizing symmetric-key & public-key cryptosystems. In: IEEE International Conference on Advanced Computer Science Applications and Technologies (ACSAT), pp. 116–121 (2013)
7. Alkady, Y., Habib, M.I., Rizk, R.Y.: A new security protocol using hybrid cryptography algorithms. In: IEEE International Computer Engineering Conference (ICENCO), pp. 109–115 (2013)
8. Abdalla, M., Bellare, M., Rogaway, P.: The oracle Diffie-Hellman assumptions and an analysis of DHIES. In: Naccache, D. (ed.) CT-RSA 2001. LNCS, vol. 2020, pp. 143–158. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45353-9_12
9. Martínez, V.G., Encinas, L.H., Dios, A.Q.: Security and practical considerations when ımplementing the elliptic curve ıntegrated encryption scheme. Cryptologia **39**(3), 244–269 (2015). https://doi.org/10.1080/01611194.2014.988363
10. Martin, et al.: OpenFog security requirements and approaches. In: IEEE Communications Society Invited Paper, November 2017
11. Rodríguez-Andina, J., Torre-Arnanz, E., Valdés-Peña, M.: FPGAs Fundamentals, Advanced Features, and Applications in Industrial Electronics. CRC Press, Boca Raton (2017)
12. Wu, H., Preneel, B.: AEGIS: a fast authenticated encryption algorithm. In: Lange, T., Lauter, K., Lisoněk, P. (eds.) SAC 2013. LNCS, vol. 8282, pp. 185–201. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-43414-7_10
13. FIPS Publication 197, the Advanced Encryption Standard (AES), U.S. DoC/NIST, November 2001
14. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signature and public-keycryptosystems. Commun. ACM **21**, 120–126 (1978)
15. Tadesse Abebe, A., Negash Shiferaw, Y., Gebeye Abera, W., Kumar, P.G.V.S.: Efficient FPGA ımplementation of an ıntegrated bilateral key confirmation scheme for pair-wise key-establishment and authenticated encryption. In: Zimale, F.A., Enku Nigussie, T., Fanta, S.W. (eds.) ICAST 2018. LNICST, vol. 274, pp. 429–438. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-15357-1_36
16. Montgomery, P.: Modular multiplication without trial division. Math. Comput. **44**, 519–521 (1985)
17. Abdellatif, K.M., Chotin-Avot, R., Mehrez, H.: AES-GCM and AEGIS: efficient and high speed hardware ımplementations. J. Signal Process. Syst. **88**(1), 1–12 (2016). https://doi.org/10.1007/s11265-016-1104-y
18. Katsaiti, M., Sklavos, N.: Implementation Efficiency and Alternations, on CAESAR Finalists: AEGIS Approach, pp. 661–665. https://doi.org/10.1109/dasc/picom/datacom/cyberscitec.2018.00117
19. Sandoval, M.M., Uribe, C.F.: A hardware architecture for elliptic curve cryptography and lossless data compression. In: IEEE International Conference on Electronics, Communications and Computers, pp. 113–118 (2005)