Efficient FPGA Implementation of an Integrated Bilateral Key Confirmation Scheme for Pair-Wise Key-Establishment and Authenticated Encryption

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

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

Abstract. The purpose of this paper is to propose a bilateral key confirmation scheme which provides a trustworthy key establishment between two communicating parties. There are various cryptographic schemes proposed based on unilateral key confirmation. But, such schemes do not confirm the equality of the common secret information computed independently by each communicating party, and do not consider whether the other end is the intended owner of the shared secret. However, exchanging of the secret information blindly without verifying that both of the ends have computed the same common secret information and without ensuring the identity of the other end with whom they are communicating, can create security risks since attackers can impersonate acting as a claimed sender or recipient. The proposed work provides bilateral key confirmation for pair-wise key-establishment based on FPGA by integrating a key agreement protocol and an authenticated encryption scheme. The implementation outcomes show the proposed scheme's reasonable hardware complexity and enhanced performance compared to existing similar works.

Keywords: Authenticated encryption · FPGA · Hybrid cryptography · Key agreement · Key confirmation

1 Introduction

In cryptography, establishment of secret keying material between communicating ends can be done electronically based on public key methods such as key-agreement protocols for key exchange [1, 2], or key transport for secure key distribution [3, 4]. When establishing a pair-wise key-agreement, the secret keying material will not directly be sent from one end to another. But, the two ends exchange only the required information from which both of them can compute a shared secret independently. Therefore, this method requires selection and exchange of valid domain parameters before performing the computation of the secret information for key establishment. In case of

F. A. Zimale et al. (Eds.): ICAST 2018, LNICST 274, pp. 429–438, 2019.
https://doi.org/10.1007/978-3-030-15357-1_36

key-transport, the secret keying material which is selected by the sender is wrapped with a key-wrapping algorithm being encrypted by the public key of the recipient and then transported to the other end. The recipient then unwraps the encrypted key using the same algorithm and the corresponding private key. Various hybrid cryptosystems have been proposed by different researchers to effectively utilize the advantages of symmetric and asymmetric key methods [4–6]. The well-known integrated encryption schemes such as Diffie-Hellman Integrated Encryption Scheme (DHIES) and Elliptic Curve Integrated Encryption Scheme (ECIES) are also hybrid cryptosystems which compose a public key key-agreement schemes, namely, Diffie-Hellman (DH) key exchange and Elliptic Curve Diffie-Hellman (ECDH) key exchange algorithms respectively, a hash function, a Key Derivation Function (KDF) or Hash-based Message Authentication Code (HMAC)-based KDF (HKDF), a symmetric key encryption algorithm, a Message Authentication Code (MAC) algorithm, and digital signature schemes [7–9]. Though they integrate different crypto mechanisms together, the objective is to provide better security by combining their advantages. The advantages can be described in terms of performance and security. Performance in this case means to utilize fast symmetric key algorithms for large data encryption and decryption instead of using public key algorithms for this purpose which are considered slower. On the other hand, security refers to utilizing the public key algorithms for secret key distribution to be used by symmetric key schemes, as well as for signature generation and verification. The key derivation function (KDF) in DHIES and ECIES is used for generation of one or more suitable secret keys from the exchanged shared secret for encryption (ENK key) and for Message Authentication Code (MAC) generation (MAC key) [7–9]. KDF or HKDF can be used to obtain keys of a required format from the result of a DH or ECDH key exchange suitable for the selected symmetric key algorithm such as AES. Keyed cryptographic hash functions are commonly used to construct Hash-based Message Authentication Code (HMAC) for key derivation in HKDF [10].

The integrated encryption schemes (DHIES and ECIES) have used unilateral key confirmation. However, bilateral key confirmation [10] is important since key agreement algorithms exchange secret information which are required for computation of common shared secret between two communicating ends. Without confirmation of the equality of the shared secrets created at both ends, and without verifying the identification of the entity communicating at the other end, exchanging secret information blindly can create security risks. In this paper, a pair-wise key-establishment method with bilateral key confirmation capability is presented by integrating a key agreement and an authenticated encryption schemes for authenticated encryption/decryption and authenticated key distribution.

The rest of the paper is organized as follows: Sect. 2 presents related works. The proposed work is explained in Sect. 3. Implementation approaches are discussed in Sect. 4. Section 5 summarizes the results. Finally, Sect. 6 concludes the paper.

2 Related Works

Various research works have been proposed based on the combination of public key and symmetric key algorithms to provide authenticated key agreement and encryption. Hybrid cryptosystems based on the combinations of public key and symmetric key algorithms can be found in [4–6]. Hybrid cryptosystems based on integrated encryption schemes such as DHIES and ECIES can also be found in [7–9]. In such systems only unilateral key confirmation is considered. Figure 1 depicts the DHIES presented in [7]. In this figure, ***M*** stands for plaintext, ***g*** is generator of cyclic group. Public keys of two communicating parties are represented by g^u and g^v respectively. Private keys of the two ends are represented as ***u*** and ***v*** respectively. Also, in the figure, ***E*** represents a symmetric key algorithm, ***H*** represents a hash function, and ***T*** stands for Message Authentication Code (MAC) generation function.

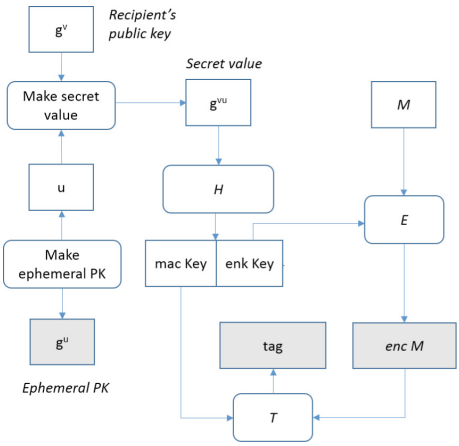

Fig. 1. DHIES functional diagram (Source: [7])

Similarly, Fig. 2 represents the ECIES functional diagram at the sender end [8, 9]. The working principle of the DHIES and ECIES are similar, but the former uses DH key exchange method, and the later uses ECDH key exchange method based on elliptic curve cryptography for key agreement. But, in both cases, a unilateral key confirmation approach has been followed. In DHIES and ECIES, the hash function and the KDF generate a MAC key and an ENC key which are the keys used for authentication and encryption respectively. The important issue here is that the MAC tag and the encrypted message are sent together to the other end without ensuring whether same shared secret is generated at the other end, and even without exactly knowing who the owner of the common secret information is at the other end.

The bilateral key confirmation scheme proposed in this work allows both of the communicating ends to ensure that they have generated equal secret key and also helps to confirm with whom the secret information sharing is done creating a trustworthy key establishment.

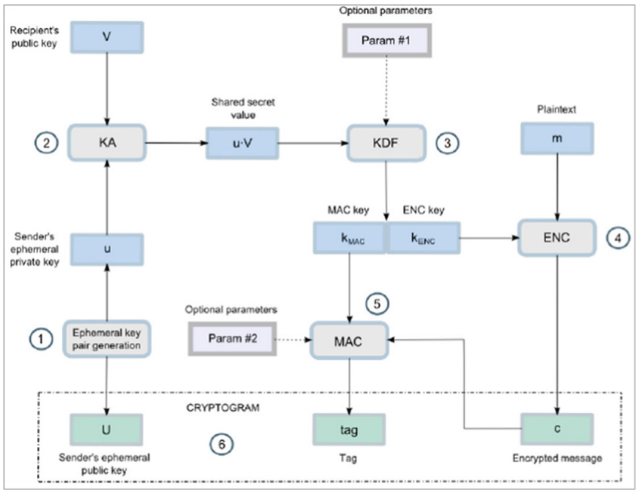

Fig. 2. ECIES encryption functional diagram (Source: [8, 9])

3 The Proposed Method

The proposed method integrates the Diffie-Hellman (DH) key agreement and Advanced Encryption Standard-Galois Counter Mode (AES-GCM) authenticated encryption [11] schemes for bilateral key confirmation and authenticated encryption as shown in Figs. 3, 4 and 5. Unlike the DHIES and ECIES, the output of the KDF is used for ENC key and ***IV***, instead of ENC key and MAC key. This is because the MAC is generated using the hash sub-key (***H***) which is computed using the ENC key itself as part of the AES-GCM process such that: $\boldsymbol{H = AES\ (ENC\ key,\ 0^{128})}$. This is computed at both ends to generate and exchange MAC tags which are related to the encrypted identification data of party A and party B (ID_A and ID_B) used for authentication and verifying that the same secret key is generated at both communicating ends. The proposed method is presented based on the following four major steps (see Sects. 3.1 to 3.4) and also depicted in Figs. 3, 4 and 5.

3.1 Key Exchange

In the first step, the following main tasks are performed by the communicating parties.

(i) sharing of authentic public parameters which are used for generation of public keys
(ii) selection of private key, and then, computation and exchange of public keys
(iii) computation of shared secret and generation of secret key and ***IV***
(iv) exchange of encrypted identities for key confirmation.

Before starting of the key exchange process, both communicating ends (the sender and recipient) first share authentic public parameters. These public parameters and randomly selected private keys at each end will be computed to produce the respective

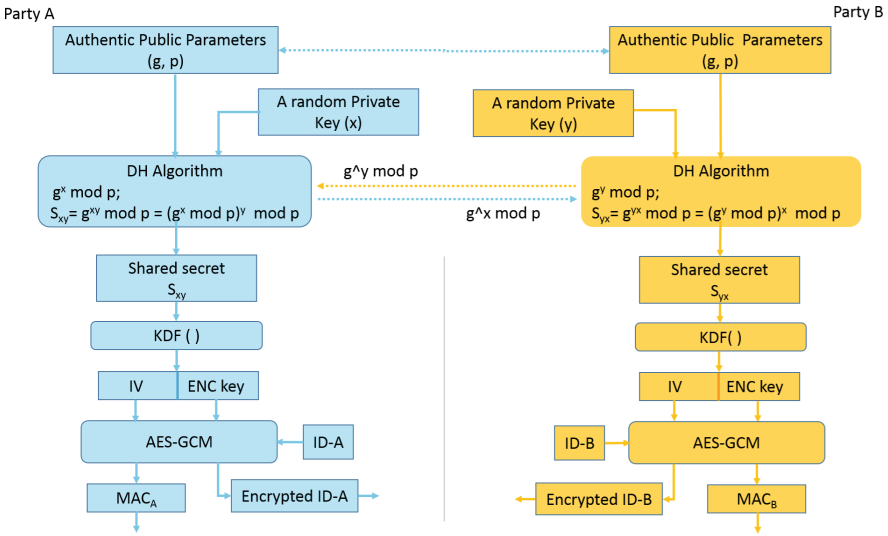

Fig. 3. Authenticated key agreement

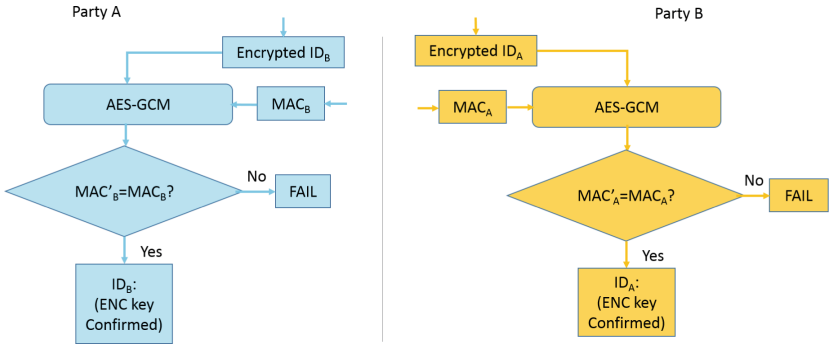

Fig. 4. Bilateral key confirmation

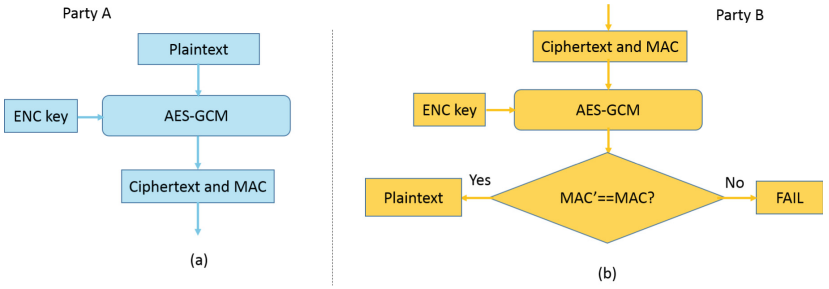

Fig. 5. Encryption and Decryption: (a) authenticated encryption; (b) authenticated decryption

public keys for each end, using the key agreement algorithm. Then, the generated public keys will be exchanged as shown in Fig. 3. The key agreement algorithm at each end then computes a common shared secret using the owned private key and the received public key. A KDF is used to produce a secret key (ENK key) with appropriate key length and ***IV***, from the common shared secret, which then are used by AES-GCM algorithm for bilateral key confirmation and authenticated encryption.

3.2 Bilateral Key Confirmation

As a second step, bilateral key confirmation is performed to prove that the produced secret key is equal at each end, and also to assure that the other end is exactly the claimed owner of the shared secret. To do this, both communicating ends use the generated secret key to encrypt their respective identity (ID_A, and ID_B) respectively using the AES-GCM algorithm, and send the encrypted data with the corresponding Message Authenticated Codes (MAC_A and MAC_B) to the other end as shown in Fig. 3. The AES-GCM algorithm at each end then compares the received MAC and the calculated MAC′ for authentication of the IDs as shown in Fig. 4. For example, if $MAC_A = MAC'_A$, party B ensures that the message originator is party A; and similarly, if $MAC_B = MAC'_B$, then, party A confirms that the message originator is party B. It is only after the MACs are verified true that decryption of the encrypted message will follow. By decrypting the encrypted ID of the other end, then both ends confirm that the secret key is equal at both ends, and the other end is the claimed owner of the secret key, meeting the requirement of bilateral key confirmation. The computed ENC key is not directly used for encryption of the sensitive data without knowing first that the other end has exactly generated the same secret key and is the intended owner of that key.

Even if the two ends trust each other, it is important to confirm first that both of them have generated the same secret key and authenticate each other before encrypting and sending sensitive data blindly to the other end. This protects man-in-the-middle (MITM) not to establish a shared secret between the sender and recipient. Using the bilateral key confirmation scheme, the sender will not trust and accept the attacker's shared secret as a true recipient's secret information, and also the recipient will not trust and accept the attacker's shared key as a trusted sender's shared key.

3.3 Authenticated Encryption

In the third step, authenticated encryption is performed by the sender. After bilateral key confirmation, the sender uses the secret key and the AES-GCM algorithm to encrypt the message. The sender then sends the ciphertext and the corresponding MAC to the recipient as shown in Fig. 5(a). The AES-GCM algorithm is used here since it is fast compared to public key algorithms and can provide confidentiality, data integrity check, and authentication crypto services simultaneously.

3.4 Authenticated Decryption

The last step is authenticated decryption process performed at the receiving end. At the receiving end, the AES-GCM algorithm, first, validates the authenticity of the message by comparing the received MAC and the calculated MAC′ as shown in Fig. 5(b). If the MAC values are equal, then the message will be decrypted and utilized. But, if the MAC values are not equal, then the message will be discarded. Therefore, using the proposed method, authenticated key exchange along with bilateral key confirmation and authenticated encryption/decryption are possible.

4 Implementation

In this work, the integrated scheme using Diffie-Hellman key-agreement protocol and AES-GCM algorithm is implemented on FPGA. VHDL is used as a hardware description language. Xilinx ISE 14.5 is used for synthesis, and the integrated simulator, ISIM, is used for simulation. Xilinx Virtex 5 FPGA device is used as a target implementation platform. It is assumed that the public parameters are authentic and shared between the two communicating ends before starting secure communication. Also, it is assumed that the two ends have exchanged the generated public keys at each end, using a chosen trusted method. The computed shared secret at both ends was made to produce 224 bits length value using SHA-224 [12] to produce 96 bits $\boldsymbol{IV}$ and 128 bits secret key which are suitable for AES-GCM algorithm. For bilateral key confirmation, the IDs of the respective ends were encrypted using the generated secret key so that both ends could assure that the secret key was equal, and the other end was the claimed owner of that key whose ID was sent with the corresponding MAC. After creating a trusted communication based on bilateral key confirmation, a message was encrypted using AES-GCM, and the ciphertext and the corresponding MAC were sent to the other end. The AES-GCM at the receiving end first compared the calculated MAC and the received MAC to verify the authentication and data integrity, and decrypted the message after verification. A Fully Pipelined AES [13] algorithm and bit-parallel Galois Hash (GHASH) have been implemented for AES-GCM [14]. In pipelining architecture, registers have been placed at each step/round to construct the pipeline as shown in Fig. 6. The depth of the pipeline, $\boldsymbol{K}$, determines how many data blocks can be processed concurrently. The architecture is fully pipelined when $\boldsymbol{K}$ equals the total number of rounds. The area and the latency of the pipelined architecture are proportional to $\boldsymbol{K}$. Pipelining can increase the encryption speed by processing multiple blocks of data simultaneously. The GHASH core has been implemented using the Mastrovito bit parallel multiplier [15]. The Mastrovito multiplier performs the finite field multiplication with parallel inputs and outputs with no clock cycle latency. The Mastrovito multiplier uses a Matrix Vector Product (MVP) which can compute a modulo reduced result in a single step [15]. Figure 7 shows the structure of the GHASH core.

For implementation of Diffie-Hellman key agreement algorithm, Montgomery multiplier [16] has been used to perform the modular multiplication which is used to

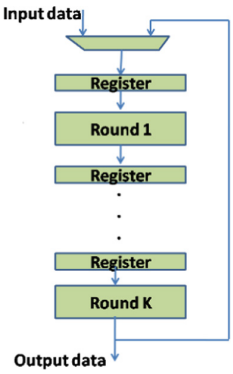

Fig. 6. Pipeline architecture

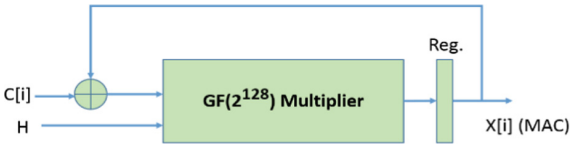

Fig. 7. Bit-parallel GHASH multiplier

speed up the process. The public parameters selected for implementation of the DH protocol are: p = 991 bits and q = 503 bits.

5 Results

The proposed cryptosystem has been implemented on Virtex 5 FPGA device. The implementation results and the performance comparisons with existing works are shown in Table 1, in terms of utilization of FPGA resources, maximum frequency, and the achieved throughput. As shown in Table 1, less number of Virtex 5 FPGA slices: 3533, 2478, and 3836 were utilized in the works presented by [17], [18] and [19] respectively, as they implemented only AES-GCM algorithm with no hybrid technique. However, 40, 41, and 50 BRAMs were utilized by [17], [18] and [19] respectively. Also, in [20], totally, 21194 slices and 20 BRAMs were used on Virtex II demonstrating the ECIES implementation. The hybrid scheme proposed in this work utilized 7886 FPGA slices, and 18 BRAMs as it is expected from the nature of the hybrid schemes. But concerning the throughput, the present work achieved 39.4 Gbps, which is an enhanced performance compared to the existing works with the same BRAM based optimization and single core AES-GCM implementation on the same implementation platform. Unlike the existing works where the implementations were restricted to performance enhancement and utilization of reasonable FPGA resources, but shared secret issue of AES-GCM has remained unresolved, the contributions of this work include authenticated encryption and authenticated key distribution with bilateral

key confirmation offering strong security with more crypto security services in addition to performance enhancement and reasonable FPGA resources utilization. The simulation wave form of the proposed scheme is shown by Fig. 8.

Table 1. Performance comparison

Author	Target device	Design	Slices	BRAM	Freq. (MHz)	Thrpt. (Gbps)
This work	Virtex 5	Hybrid	7886	18	308.2	39.4
[17]	Virtex 5	AES-GCM	3533	41	314	16.9
[18]	Virtex 5	AES-GCM	2478	40	242	30.9
[19]	Virtex 5	AES-GCM	3836	50	273.4	32.46
[20]	Virtex II	ECIES	21194	20	-	-

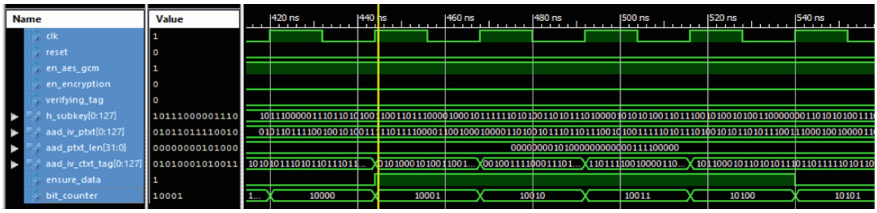

Fig. 8. Simulation waveform of the hybrid cryptosystem

6 Conclusions

An integrated scheme using DH algorithm for key agreement and AES-GCM for authenticated encryption of sensitive data has been implemented on Xilinx Virtex 5 FPGA platform offering bilateral key confirmation and authenticated encryption. Compared to the traditional hybrid cryptosystems which provided only one sided key confirmation, bilateral key confirmation allows both communicating ends to create a trustworthy communication. The proposed method saves extra resource requirement and key management by reducing the use of separate MAC algorithm compared to DHIES/ECIES schemes, and provides authenticated key distribution which is not addressed by existing FPGA based AES-GCM implementations. The implementation outcomes show that the proposed hybrid system has consumed reasonable amounts of FPGA resources with better throughput achievement which we can further improve by applying better optimization techniques.

References

1. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Trans. Inf. Theory **22**(6), 644–654 (1976)
2. Boneh, D., Shparlinski, I.E.: On the unpredictability of bits of the elliptic curve Diffie-Hellman scheme. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 201–212. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_12

3. Rivest, R., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. ACM Trans. Commun **21**, 120–126 (1978)
4. 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)
5. Nadjia, A., Mohamed, A.: AES IP for hybrid cryptosystem RSA-AES. In: IEEE 12th International Multi-Conference on Systems, Signals & Devices (SSD 2015), pp. 1–6 (2015)
6. Kapur, R.K., Khatri, S.K.: Secure data transfer in MANET using symmetric and asymmetric cryptography. In: IEEE International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions), pp. 1–5 (2015)
7. 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
8. Martínez, V.G., Alvarez, F.H., Encinas, L.H., Ávila, C.S..: A comparison of the standardized versions of ECIES. In: IEEE Sixth International Conference on Information Assurance and Security (2010)
9. Martínez, V.G., Álvarez, F.H., Encinas, L. H.: Analysis of ECIES and other cryptosystems based on elliptic curves. CSIC Digital (2013)
10. Barker, E., Chen, L., Roginsky, A., Vassilev, A., Davis, R.: Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography. NIST Special Publication 800-56A Revision 3, April 2018
11. Dworkin, M.: NIST Special Publication 800-38D: Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC (2007)
12. Federal Information Processing Standards (FIPS) Publication 180-4,: Secure Hash Standard (SHS), vol. 4 (2015)
13. Satoh, A., Sugawara, T., Aoki, T.: High-speed pipelined hardware architecture for Galois counter mode. In: Garay, J.A., Lenstra, A.K., Mambo, M., Peralta, R. (eds.) ISC 2007. LNCS, vol. 4779, pp. 118–129. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75496-1_8
14. Wang, J., Shou, G., Hu, Y., Guo, Z.: High-speed architectures for GHASH based on efficient bit-parallel multipliers. In: IEEE International Conference on Wireless Communications, Networking and Information Security (WCNIS), pp. 582–586 (2010)
15. Mastrovito, E.D.: VLSI architectures for computations in Galois fields. Ph.D. thesis, Linköping University, Department of Electrical Engineering, Linköping, Sweden (1991)
16. Montgomery, P.: Modular multiplication without trial division. Math. Comput. **44**, 519–521 (1985)
17. Zhou, G., Michalik, H., Hinsenkamp, L.: Improving throughput of AES-GCM with pipelined karatsuba multipliers on FPGAs. In: Becker, J., Woods, R., Athanas, P., Morgan, F. (eds.) ARC 2009. LNCS, vol. 5453, pp. 193–203. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00641-8_20
18. Abdellatif, K.M., Chotin-Avot, R., Mehrez, H.: Authenticated encryption on FPGAs from the static part to the reconfigurable part. Microprocess. Microsyst. **38**(6), 526–538 (2014)
19. 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 (2017)
20. 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)