Secure Communication for 6TiSCH Wireless Networks Based on Hybrid ECC and AES Algorithms

Chengqi Hou, Wei Yang(✉), Zhiming Zhang, Qinghua Liu, and Jianmao Xiao

School of Software, Jiangxi Normal University, Nanchang, China
yw@jxnu.edu.cn

Abstract. The 6TiSCH protocol stack has been widely utilized in the industry to build highly reliable and energy efficiency wireless sensor networks (WSNs). The communication security between nodes mainly relies on the AES encryption algorithm, but the 6TiSCH protocol does not regulate the protection strategy for encryption keys in the network. Therefore, this paper proposes a hybrid ECC-AES data encryption scheme based on the 6TiSCH protocol stack. The Elliptic Curve Diffie Hellman (ECDH), Elliptic Curve Qu-Vanstone (ECQV) algorithms are used to negotiate the shared key for each two nodes in a WSN, and AES encryption with dynamic session keys are derived from the shared key during the nodes' communication phase. Meanwhile, since the ECC algorithm is used in the resource-constrained nodes, this paper considers the influence of the underlying elliptic curve operations on the computation speed of the shared key, and proposes a regular window algorithm to accelerate the scalar multiplication operation based on previous researches. To prove the scheme's viability, we conduct simulation experiments on the generation time of shared keys, and the experimental results prove that the encryption scheme under the regular window scalar multiplication has an impressive key generation speed.

Keywords: 6TiSCH Protocol · Security Communication · ECC Algorithm · AES Algorithm

1 Introduction

With the continuous development of communication technology and microelectronics, wireless sensor networks (WSNs) have been widely used in the fields of smart home, environmental monitoring and industrial control [1]. In order to achieve highly reliable, low-power data transmission between WSNs, the IEEE 802.15.4e [2] standard was released in 2012, which proposed a Time Slotted Channel Hopping (TSCH) technology. This technology coordinates the working state through precise time synchronization, and makes the communication channel for nodes change with the change in different communication time-slots, nodes that are non-working are in a dormant state. Only when there are data that need to be transmitted, the nodes turn on the RF module for

Y. Cao and X. Shao (Eds.): MONAMI 2022, LNICST 474, pp. 306–315, 2023.
https://doi.org/10.1007/978-3-031-32443-7_22

data transmission, as a way of reducing the energy consumption of sensor nodes and solving the idle listening problem [3] during the data communication phase. Based on this standard, in 2013, the Internet Engineering Task Force (IETF) started to develop a whole industrial IoT protocol stack – IETF 6TiSCH [4], aiming at low power consumption, high-reliability data and low latency transmission in industrial process control [5].

In the 6TiSCH protocol stack, the encryption and authentication operations for data security are implemented in the link layer using IEEE 802.15.4e protocol. The encryption algorithm in the protocol uses AES-CCM mode [6], which is a combination of Cipher Block Chaining Message Authentication Code (CBC-MAC) mode for authentication and Counter mode (CTR) mode for encryption. The hardware implementation of the AES algorithm [7] can finish encrypting and authenticating data in a short time. However, the keys in the AES algorithm generally come from the pre-set by the managers, and the whole network shares the same key, at this time, if there is a malicious node in the network that leaks the AES key of the whole network, it will expose the data transmission of the whole network nodes to danger.

Research on 6TiSCH WSNs has mainly focused on the resource scheduling of the network, while research on 6TiSCH network security is scarce. Sajjad [8] analyzed the security of the IEEE 802.15.4 protocol and pointed out that the protocol is susceptible to jamming-influenced DOS attacks at the MAC layer, and the IEEE 802.15.4 protocol itself does not specify the way for creating and exchanging keys during data encryption. To solve the DOS attacks caused by malicious nodes through jamming, the authors [9] proposed a dynamic scheme DISH with random replacement of time-slots and channels and proved that the scheme can effectively resist DOS attacks. However, the key negotiation and management problems in 6TiSCH networks are still a major threat that hinders secure transmission between nodes, and in [10], the authors hypothesize that when the keys are in the hands of malicious nodes, they will intercept and tamper with packets in inter-node data communication, and then attack the protocol stack through traffic dispersion attacks and overload attacks.

On the other hand, many researchers have focused on secure communication in WSNs using modified AES algorithms, for the reason that hardware-accelerated AES algorithms are suitable for implementation on restricted nodes with fast computation speed and less energy consumption, but in application scenarios such as industrial scenarios where high-security data transmission is required, AES is difficult to guarantee a sufficient degree of security. Sciancalepore et al. [11] proposed a security framework in IEEE 802.15.4 protocol with Diffie-Hellman (DH) protocol for key negotiation and AES algorithm for encryption. The article [12] proposed an encryption scheme in WSNs, where the scheme divides the plaintext into three parts for hybrid encryption, using AES, DES, and RSA algorithms for encryption respectively. Both articles use the high-energy RSA algorithm on restricted nodes, it will accelerate the energy consumption of nodes in the network. The paper [13] proposes a key extension algorithm based on the AES algorithm to increase the degree of confusion caused by the encryption process and complete hardware implementation of the improved algorithm, but the reconfiguration of the encryption steps of the AES algorithm itself could hardly to solve the problem of key leakage fundamentally.

To solve the key establishment problem between 6TiSCH network nodes, this paper proposes a hybrid encryption method of AES and ECC algorithms in the 6TiSCH sensor network, using elliptic curve encryption to establish keys on two nodes and encrypting the data by AES-CCM mode. The ECC algorithm is chosen because it has a shorter key length for the same level of security, 160 bit key length ECC algorithm can achieve the same level of security as the 1024 bit key length RSA algorithm [14]. Meanwhile, to accelerate the key generation time, this paper considers the underlying optimization algorithm of the ECC algorithm, and combines the work of Rivain [15] to propose a scalar multiplication algorithm with a regular window method to accelerate our proposed hybrid encryption algorithm.

This paper is organized as follows: Sect. 2 proposes a secure communication scheme based on a hybrid ECC-AES algorithm in the 6TiSCH WSN. Section 3 considers the scalar multiplication operation of the ECC algorithm, and a regular window algorithm is proposed to accelerate key generation time, in Sect. 4, the proposed scheme is simulated under different elliptic curve parameters, and Sect. 5 concludes the scheme and presents the future work.

2 Hybrid ECC-AES Encryption Scheme

In this section, we present a secure communication scheme with a hybrid ECC-AES algorithm. The scheme establishes a shared key for every two nodes in the sensor network by using the Elliptic Curve Diffie-Hellman (ECDH) algorithm, but the ECDH algorithm has the risk of being subject to man-in-the-middle (MITM) attacks. To resist MITM attacks, we generate an implicit certificate for each node when it joins the network using the ECQV algorithm [16]. The scheme divides the process of node communication into three phases, node joining phase, shared key establishment phase and dynamic key encryption phase, which are described in detail below. The main notations used in this paper are shown in Table 1.

First, taking a brief of the ECC algorithm, the ECC algorithm is defined on a finite field F_q, the general form of an elliptic curve is Weiertass equation: $y^2 = x^3 + ax + b, (4a^3 + 27b^3 \neq 0)$. The algorithm chooses a basis point G on the elliptic curve, it is feasible to select a positive integer to compute $k \cdot G$. This operation is called scalar multiplication, while it is computationally infeasible to find the integer k by the result $k \cdot G$ and the point G. This is the Elliptic Curve Discrete Logarithm Problem (ECDLP), which the security of the ECC algorithm is built on this.

2.1 Node Joining Phase

In this phase, to generate certificates for each edge node to join the WSN, the scheme adopts Certificate Authority (CA) in the joining phase, and the certificates do not use traditional X.509 certificates, because such certificates are too large in byte length and cause high energy consumption and transmission delay in the edge nodes during transmission, so this paper utilizes the ECQV protocol to generate implicit certificates to reduce such costs. During the join phase, the edge node obtains synchronization with the network through the beacon frame, gets the certificate through CA, and computes its public and private keys, only four information exchanges with the CA are required in the process. Throughout the communication process, it can be assumed that the parameters

Table 1. Symbols appearing in this paper

Symbol	Explanation
a, b	The two coefficients of the elliptic curve
F_q	Selected finite fields in the ECC algorithm
G	The selected base point, where $G = (x, y)$
n	The order of the base point G on the curve, $n \cdot G = 0$
r_A	Random number generated by node A
$Cert_A$	Certificate of node A
ASN	Time slot values in the network
K_{AB}	The shared key established by nodes A, B
$H()$	A hash function that compresses the given input into a 128bit bit output
d_A	The private key of node A
Q_A	The public key of node A
$addr_A$	64-bit MAC address of node A
ID_A	Identity information of node A in the network
$nonce_A$	The nonce value generated by node A based on the number of network time slots and its own MAC address

$\{F_q, a, b, G, n\}$ of the elliptic curve have been predetermined and stored in the edge node A and the CA. See Fig. 1 for the communication diagram of this phase.

1. The CA randomly selects $d_{CA} \in \{2, 3, \ldots, n-1\}$ as the private key of the CA in the initial stage, and calculates $Q_{CA} = d_{CA} \cdot G$ as the public key of the CA
2. CA will broadcast a beacon frame every certain time-slots, the frame includes network related information, and the node A expects to join the WSN will continuously listen to message communication in the network, once the beacon frame is received, the node will enter the computation state.
3. The edge node will randomly generate $r_A \in \{2, 3, \ldots, n-1\}$ and compute $R_A = r_A \cdot G$. The identity information ID_A of the node is generated based on the computed result $R_A = (x_A, y_A)$, $ID_A = H(x_A||addr_A)$. After the calculation is completed, the node sends an association request frame to CA, and the payload information in the frame includes $\{R_A, ID_A\}$.
4. After receiving the relevant information, the CA will calculate the certificate and part of the key information for the node according to the ECQV protocol.

$$P_A = R_A + r_{CA} \cdot G \tag{1}$$

$$cert_A = code(P_A, ID_A) \tag{2}$$

$$w = H(cert_A) \cdot r_{CA} + d_{CA}(modn) \tag{3}$$

where r_{CA} is the random number generated by CA, and CA will send the result $\{P_A, cert_A, w\}$ as the payload in an association response frame to the edge node after the calculation step is completed.

5. The edge node receives the relevant information in the frame and calculates the public and private key of the node based on the relevant key information.

$$d_A = r_A \cdot H(cert_A) + w(mod n) \tag{4}$$

$$Q_A = d_A \cdot G \tag{5}$$

$$Q_A^{'} = P_A \cdot H(cert_A) + Q_{CA} \tag{6}$$

The edge node checks whether the calculation result of Q_A is equal to $Q_A^{'}$, if so, it accomplishes the authentication to CA, proves that the association response frame is indeed sent by CA, and the node takes $\{d_A, Q_A\}$ as the key pair of the edge node, $cert_A$ as the certificate of the edge node, and pledge node sends the association confirm frame to CA, means that node joins the network successfully.

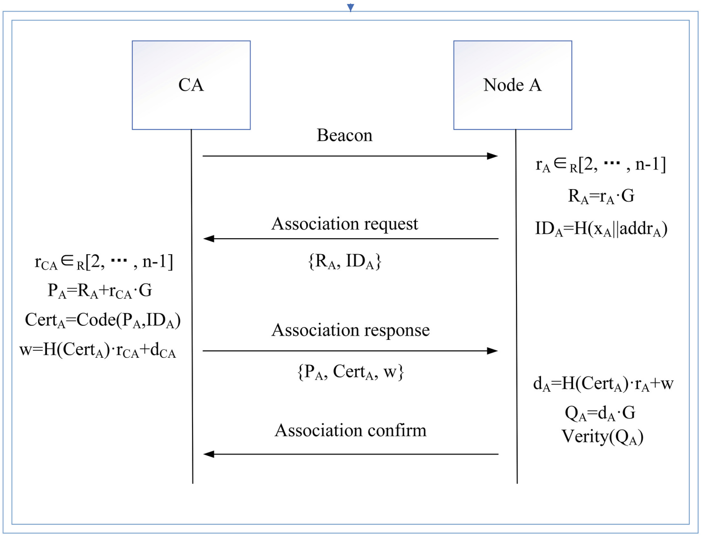
CA
Node A
Beacon
$r_A \in_R [2, \cdots, n-1]$
$R_A = r_A \cdot G$
$ID_A = H(x_A||addr_A)$
Association request
$\{R_A, ID_A\}$
$r_{CA} \in_R [2, \cdots, n-1]$
$P_A = R_A + r_{CA} \cdot G$
$Cert_A = Code(P_A, ID_A)$
$w = H(Cert_A) \cdot r_{CA} + d_{CA}$
Association response
$\{P_A, Cert_A, w\}$
$d_A = H(Cert_A) \cdot r_A + w$
$Q_A = d_A \cdot G$
Verity(Q_A)
Association confirm

Fig. 1. Diagram of frames exchange between edge node A and CA during the join phase

2.2 Shared Key Establishment Phase

When a node A in the WSN wants to establish a connection with another node B, it enters the shared key generation phase, and the key establishment exploits the ECDH algorithm. To avoid MITM attacks, the scheme adds the authentication operation of the node before calculating the shared key, and the authentication operation utilizes the elliptic curve implicit certificate generated by CA. Also, in order to resist replay attacks, the scheme also adds the nonce value field in the authentication operation as a check against replay attacks. The communication diagram for this phase is shown in Fig. 2.

1. Node A in the network expects to establish communication connection with B, it will send an association request frame to node B. The payload of the frame contains $\{P_A, cert_A, Q_A, nonce_A, MIC_A\}$, where MIC_A is an authentication message to prevent replay attack, $MIC_A = auth(P_A, cert_A, Q_A, nonce_A)$.
2. After receiving the information in the frame, node B will first authenticate the MIC value, node B calculates $MIC'_A = auth(P_A, cert_A, Q_A, nonce_A)$, check whether MIC'_A and MIC_A are equal, if not, node B abort the session, otherwise, it calculates $Q'_A = P_A \cdot H(cert_A) + Q_{CA}$, check whether Q'_A is equal to Q_A, if so, then node B stores the relevant information of node A and replies the association response frame to node A with $\{P_B, cert_B, Q_B, nonce_B, MIC_B\}$ as the payload.
3. Node A receives the reply frame, calculates $MIC'_B = auth(P_B, cert_B, Q_B, nonce_B)$ and verifies whether MIC'_B is equal to MIC_B, if not, the session is aborted, otherwise, it continues to calculate $Q'_B = P_B \cdot H(cert_B) + Q_{CA}$ and checks whether Q'_B is equal to Q_B, if equal, then node B stores the relevant information of node A and sends association confirm frame, representing that node A,B authentication is completed and the calculation of shared key can be carried out.
4. Nodes A, B calculate the shared key K_{AB} starting from the time slots of association confirm frame transmission and reception respectively.

$$K_{AB} = r_A \cdot Q_B = r_B \cdot Q_A \tag{7}$$

2.3 Dynamic Key Encryption Phase

After two nodes complete the establishment of the shared key, the encryption and authentication of the data communication should be updated at a certain time to avoid the probability of key leakage during the node communication process, but re-computing the second phase when the key needs to be replaced will highly increase the energy consumption of the restricted nodes and the time delay of the communication process. Therefore, the shared key K_{AB} which established during the second phase can be stored in the node's memory space, and the session key can be updated by K_{AB} in each time-slot of the node's communication. The key update formula is as follows:

$$K_{AES-CBC} = H(x_k||addr_A||addr_B||ASN) \tag{8}$$

$$K_{AES-CTR} = H(y_k||addr_A||addr_B||ASN) \tag{9}$$

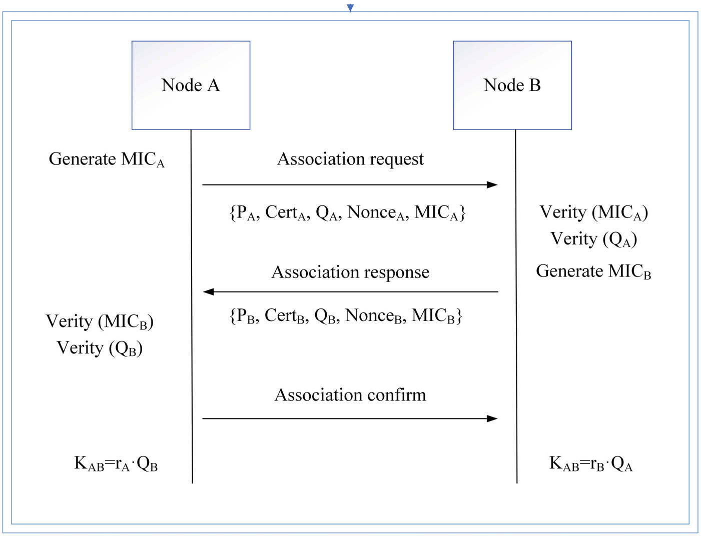

Fig. 2. Diagram of frames exchange between two 6TiSCH WSN nodes during the shared key establishment phase

It can be seen that only one hash operation is needed to update the session key, and the two communication nodes do not need to exchange any information, which achieves a fast key update with very little overhead, and the encryption process uses AES-CCM mode.

3 Consideration of ECC Optimization Algorithm

In the proposed dynamic shared key encryption scheme, since the usage of public key encryption algorithm, it is necessary to consider the overhead of the algorithm. The core operation in the ECC algorithm is the scalar multiplication $k \cdot G$, where k is a randomly large integer, and G is a base point on the elliptic curve. A large amount of research has been focused on acceleration of the scalar multiplication operation, such as the NAF window algorithm, Fixed-base comb algorithm, and other fast algorithms [17], but these algorithms usually have difficulty in achieving a regular computational flow during the scanning computation of k and are vulnerable to side-channel attacks such as Simple Power Attack (SPA). However, secure data transmission is extremely important in 6TiSCH industrial environments, so inspired by the work of Rivain [15], a regularized window method is proposed in this paper to defend against SPA attacks, and the pseudo-code implementation of the algorithm is shown below.

```
Regular window algorithm
INPUT: $G \in E(\mathbb{F}_P)\ , k =\ (l_{\mathrm{d}-1}, l_{\mathrm{d}-2}, \ldots, l_0)_{2^\omega}$
OUTPUT: $Q = [k] \cdot G$
Compute $Q_i = i \cdot Q\ , i \epsilon \{1,2, \ldots, 2^\omega - 1\}$
$R_0 = Q_0\ ,R_1 = Q_{l_{n-1}}$
for $i = l - 2$ to 0 do:
    $R_1 =\ 2^\omega \cdot\ R_1$
    $s = !!\,(l_i)$
    $R_s = R_S + Q_{l_i}$
end for
Return $R_1$
```

The algorithm uses unsigned bits representation,$k = \sum_{i=0}^{d-1} l_i \cdot 2^{i \cdot \omega}, l_i \epsilon \{1, 2, \ldots, 2^\omega - 1\}, \mathrm{d} = \lfloor \frac{n}{w} \rfloor$, this method avoids judging the positive and negative of l_i in the bit scanning. For the 0-bit scanning through k, which is the part of the Window NAF algorithm that does not need to be computed, but to avoid SPA attacks, the algorithm uses a virtual addition method to regular the computational flow. Virtual addition defines a virtual accumulator R_0. A virtual point addition operation with a similar amount of computation is performed on R_0 when bit 0 in k is scanned, but the virtual addition does not affect the final computation result.

4 Performance Analysis

In order to investigate the performance of exploiting the ECC algorithm to generate shared keys in our scheme, this section utilizes the open-source platform OpenWSN to simulate the implementation of our proposed hybrid ECC-AES encryption scheme on PC and conducts comparative experiments on the speed of nodes computing the key K_{AB} using the ECDH algorithm in the shared key generation phase. For the elliptic curve parameters of the scheme, three different elliptic curves secp160r1, secp192k1, and secp256k1 are experimentally selected, and the modular reduction operation adopts the Pseudo-Mersenne primes reduction algorithm [18]. The experiments select two regular scalar multiplication algorithms, the regular window method (window size $\omega = 4$) proposed in this paper and the common regular scalar multiplication algorithm Montgomery ladder [15], which under two elliptic curve coordinates (affine and projective) [19]. Simulation experiments are conducted to test the speed of shared key computation operation ($K_{AB} = r_A \cdot Q_B = r_B \cdot Q_A$) with different parameters mentioned above. Fifty repetitions of the experiments are performed for each parameter setting, and the results of the running times are recorded and plotted in Fig. 3.

The experimental results show the specific time required to compute the shared key K_{AB} for the scheme with different parameter settings after the authentication message is completed by two WSN nodes. It can be found that an appropriate increase of a part of RAM and ROM for the computation process (15 point-pairs need to be stored in the case of window size $w = 4$), through the underlying optimization algorithm, i.e. projective coordinate transformation and sliding window algorithm, can significantly

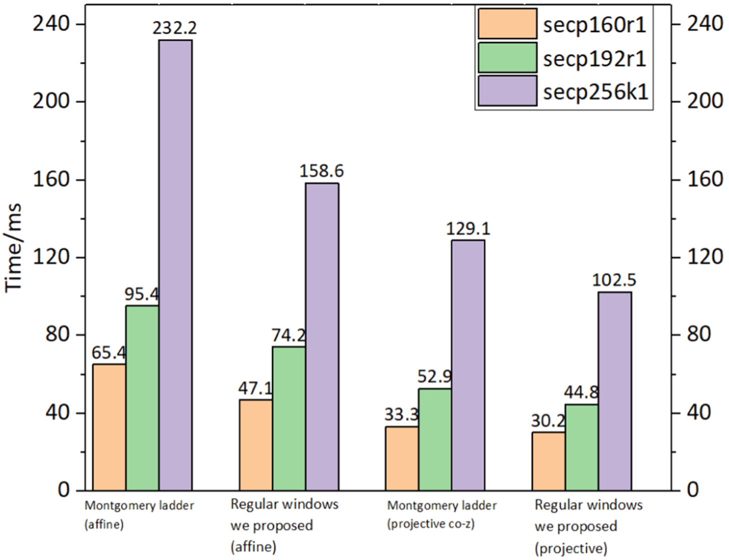

Fig. 3. Shared key calculation time ($K_{AB} = r_A \cdot Q_B = r_B \cdot Q_A$) under different elliptic curve parameters.

accelerate the key computation time of the nodes in the scheme. Since the window method requires pre-computation, this will bring a part of initialization time, but the percentage of this overhead will become smaller as the length of the elliptic curve key increases. Meanwhile, the window method is regularized in this paper, so it can resist most of the measured channel attacks.

5 Conclusion and Future Work

A hybrid ECC-AES encryption scheme is presented in this paper, and the scheme has proven to be highly robust in data communication between 6TiSCH WSN nodes. The scheme exploits the ECDH protocol to negotiate a shared key for two nodes in the network, the ECQV protocol to generate implicit certificates to resist man-in-the-middle attacks in the key generation phase, and add nonce values to the authentication process to prevent replay attacks, uses the underlying regular window method to avoid a certain degree of side-channel attacks. On the other hand, Simulation experiments are carried out on PC to calculate the shared key generate time in the scheme, it is proved that the scheme proposed in this paper is feasible in terms of calculation time.

Future work will focus on implementing the scheme proposed in this paper in a real wireless sensor network and proposing algorithms for generating group shared keys in the network.

Acknowledgments. This work is supported by the National Natural Science Foundation of China under Grant No.62002143 and the Natural Science Foundation of Jiangxi Province under Grant No. 20224BAB202011.

References

1. Gardašević, G., Katzis, K., Bajić, D., et al.: Emerging wireless sensor networks and Internet of Things technologies—foundations of smart healthcare. Sensors **20**(13), 3619 (2020)
2. 15.4e-2012: IEEE Standard for Local and Metropolitan Area Networks – Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC Sublayer, Institute of Electrical and Electronics Engineers Std. (2012)
3. Vilajosana, X., et al.: Ietf 6tisch: a tutorial. IEEE Commun. Surv. Tutor. **22**(1), 595–615 (2019)
4. 6TiSCH Homepage: https://datatracker.ietf.org/wg/6tisch/charter/. Last accessed Oct 2018
5. Scanzio, S., et al.: Wireless sensor networks and TSCH: a compromise between reliability, power consumption, and latency. IEEE Access **8**, 167042–167058 (2020)
6. Vilajosana, X., Watteyne, T., Vučinić, M., et al.: 6TiSCH: industrial performance for IPv6 Internet-of-Things networks. Proc. IEEE **107**(6), 1153–1165 (2019)
7. Shahbazi, K., Ko, S.B.: Area-efficient nano-AES implementation for Internet-of-Things devices. IEEE Trans. Very Large Scale Integ. Syst. **29**(1), 136–148 (2020)
8. Sajjad, S.M., Yousaf, M.: Security analysis of IEEE 802.15. 4 MAC in the context of Internet of Things (IoT). In: 2014 Conference on Information Assurance and Cyber Security (CIACS), pp. 9–14. IEEE (2014)
9. Tiloca, M,. Guglielmo, D.D., Dini, G., et al.: Dish: Distributed shuffling against selective jamming attack in ieee 802.15. 4e tsch networks. ACM Trans. Sensor Netw. **15**(1), 1–28 (2018)
10. Carignani, G., Righetti, F., Vallati, C., et al.: Evaluation of feasibility and impact of attacks against the 6top protocol in 6tisch networks. In: 2020 IEEE 21st International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), pp. 68–77. IEEE (2020)
11. Sciancalepore, S., Piro, G., Vogli, E., et al.: On securing IEEE 802.15. 4 networks through a standard compliant framework. In: 2014 Euro Med Telco Conference (EMTC), pp. 1–6. IEEE (2014)
12. Pooja, C.R.K.: Triple phase hybrid cryptography technique in a wireless sensor network. Int. J. Comput. Appl. **44**(2), 148–153 (2022)
13. Lavanya, R., Karpagam, M.: Enhancing the security of AES through small scale confusion operations for data communication. Microprocess. Microsyst. **75**, 103041 (2020)
14. Bafandehkar, M., Yasin, S.M., Mahmod, R., et al.: Comparison of ECC and RSA algorithm in resource constrained devices. In: 2013 International Conference on IT Convergence and Security (ICITCS), pp. 1–3. IEEE (2013)
15. Rivain, M.: Fast and regular algorithms for scalar multiplication over elliptic curves. Cryptology ePrint Archive (2011)
16. Campagna, M.: SEC 4: Elliptic curve Qu-Vanstone implicit certificate scheme (ECQV). Standards for Efficient Cryptography, Version 1 (2013)
17. Brown, M., Hankerson, D., López, J., et al.: Software implementation of the NIST elliptic curves over prime fields. In: Cryptographers' Track at the RSA Conference, pp. 250–265. Springer, Berlin, Heidelberg (2001)
18. Gura, N., Patel, A., Wander, A., et al.: Comparing elliptic curve cryptography and RSA on 8-bit CPUs. In: International workshop on cryptographic hardware and embedded systems, pp. 119–132. Springer, Berlin, Heidelberg (2004)
19. Omondi, A.R.: Cryptography arithmetic. Springer International Publishing (2020)