Computer Systems Science & Engineering
DOI: 10.32604/csse.2023.032486
Article

An Improved Pairing-Free Ciphertext Policy Framework for IoT

M. Amirthavalli*, S. Chithra and R. Yugha

Department of Information Technology, Sri Sivasubramaniya Nadar College of Engineering, Rajiv Gandhi Salai (OMR), Kalavakkam, 603110, Tamil Nadu, India
*Corresponding Author: M. Amirthavalli. Email: amirthavalli0987@gmail.com
Received: 19 May 2022; Accepted: 27 June 2022

Abstract: Internet of Things (IoT) enables devices to get connected to the internet. Once they are connected, they behave as smart devices thereby releasing sensitive data periodically. There is a necessity to preserve the confidentiality and integrity of this data during transmission in public communication channels and also permitting only legitimate users to access their data A key challenge of smart networks is to establish a secure end-to-end data communication architecture by addressing the security vulnerabilities of data users and smart devices. The objective of this research work is to create a framework encompassing Ciphertext policy Attribute-based Encryption scheme using block encryption and BLAKE hashing technique. An improved Pairing-Free-Ciphertext policy Attribute-based encryption algorithm has been developed to overcome the aforementioned challenges. Further, a comparative study has been performed between the proposed scheme and the different encryption algorithms. It is found that the proposed scheme scores well over the already existing schemes. The scheme is evaluated in terms of execution time and communication overhead. The robustness of the proposed scheme is also analyzed from the perspective of several security goals.

Keywords: Internet of things; access control; cipher policy based-attribute based encryption (CP-ABE); access matrix; performance analysis

1 Introduction

IoT technology has penetrated in different sectors to provide smart services to users. For example, smart farming and precision agriculture is seen as an essential automation in agriculture sector that benefits farmers in multiple scenarios [1]. The operation of IoT system release sensitive and/or critical information that requires protection, making the security of their resources and services an imperative design. Attribute based Encryption (ABE) schemes can be seen as a viable and promising approach for IoT research community. It supports mixed encryption and provides access control measures for user privileges. Mixed encryption combines Symmetric Encryption (SE) methods for data and Asymmetric Encryption (AE) methods for access policies. Sahai and Waters pioneered the notion of ABE and it has been manifested itself into several versions today [2]. The concept of ABE is to authorize a data sender in-order to enforce fine-grained access control by specifying set of attributes to access original data. Further, it also provides confidentiality of data during secure transmission by using encryption algorithm.

ABE comes in two variants namely, cipher text-policy ABE (CP-ABE), and key-policy ABE (KP-ABE). In CP-ABE, data sender uses an access tree to encrypt the data and data receiver's secret key is generated over a set of attributes. Consequently, the data receiver should comply with access policy of DS to decrypt the cipher-text [3]. In KP-ABE technique, data receiver secret keys are generated based on an access policy, where cipher-texts are embedded with sets of descriptive attributes [4]. The appealing advantage of CP-ABE over KP-ABE is that the sender exerts more control over who has access to the data. Further, it enforces a fine-grained access control on the cipher-texts and provides highly expressive access policies.

A few lightweight CP-ABE schemes have recently been proposed using pairing-free concepts. Recently, large numbers of light weight symmetric block ciphers are emerging for low computational devices. Due to the simple operation and high efficiency in software and hardware, these ciphers usually have very good software and hardware performances. Light weight cipher methods can be applied to resource constraint IoT devices in preserve data security. ABE encryption can be appended with symmetric block encryption to impose attribute-based access control [5,6]. In this context, the aim of this work is to create a framework encompassing CP-ABE algorithm, block encryption and hashing technique which can preserve various security goals.

2 Related Works

Lewko and Waters developed a multi-authority Decentralized ABE (DeABE) algorithm [7]. DeABE converts any monotone Boolean access structures into an equivalent Linear Secret-Sharing Schemes (LSSS) matrix. The advantage of representing an access policy in LSSS matrix can be obscure for an adversary. In LSSS oriented CP-ABE schemes, the size of ciphertext is linear. Chandrasekaran and Balakrishnan, constructed an efficient and secure data communication in Wireless Body Area Network (WBAN) using the Twofish (TF) symmetric algorithm and CPABE-CSC (TF-CPABE) with policy updating [8].

Hu et al. [9], proposed a protocol for efficient and secure data communication architecture using Ciphertext-Policy Attribute-based Encryption (CP-ABE). Banerjee et al. [10], presented a multi-Authority CP-ABE with constant-size key and cipher text anonymous user access control scheme with three-factor authentication for IoT environment.

Odelu et al. [11] proposed a novel RSA-based CP-ABE scheme with constant size secret keys and cipher texts (CSKC) for battery-limited mobile devices and has O (1) time complexity for each decryption and encryption. Ambrosin et al. [12], has shown the feasibility of adopting ABE in diverse IoT platform namely, Intel Galileo Gen 2 and inferred that adopting ABE in the IoT environment would be feasible.

In recent years, pairing free-ABE (PF-ABE) schemes have gained extensive attention amongst the research community with the intention of meeting the resource constraint nature of IoT devices. Yao et al. [13] pioneered the concept of a lightweight no-pairing Elliptic Curve Cryptography (ECC)-based KP-ABE scheme for the resources-constraint Unit IoT based applications.

Hong et al. [14] have proposed key-insulated KP-ABE algorithm without pairing, but their scheme is secured under the computational Diffie–Hellman (ECCDH) assumption using the random oracle model which was also vulnerable to the collusion attack. Karati et al. [15] have proposed a threshold-based ABE scheme without bilinear pairing and without the linear secret sharing scheme. They used modular exponential operations, which are significantly more expensive.

Sowjanya et al. [16], have proposed a lightweight ECC-based without bilinear pairing KP-ABE scheme for unit IoT applications like the ambient-assisted living (AAL) system Ding et al. [17], have designed a novel PF-CP-ABE scheme that replaces complicated bilinear pairing with simple scalar multiplication on

elliptic curves, thereby reducing the overall computation overhead. Sowjanya et al. [18] presented a ECC-based CP-ABE scheme without a without a bilinear pairing operation for Wireless Body area Network (WBAN).

2.1 Research Contributions

According to the literature survey, the existing schemes do not use any hashing algorithm to generate encryption session keys. Additionally, the integrity of the data is not ensured while it is in transit. As a result, the unique aspects of this work are as follows:

- In order to generate session encryption keys, BLAKE variants hashing algorithm is utilized.
- To generate the hash, a cryptographically secure random number is used.
- An improved version of pairing-free CP-ABE(IPF-CP-ABE) is proposed.
- A framework incorporating BLAKE-IPF-CP-ABE is presented.

3 Preliminaries

The following sub-sections explores about the concepts required to implement the proposed work.

3.1 Pairing-Free ECC

Let $Ep\ (a,\ b)$ be a set of elliptic curve points over the prime field F_p, defined by the following non-singular elliptic curve equation:

The additive cyclic elliptic curve group defined as point O is known as "point at infinity" or "zero point". A short explanation about the elliptic curve group properties is given below:

Point addition: Let $P,\ Q$ be two points on the curve, then $P + Q = R$, where the line joining P and Q intersects the curve at, and the reflection of it with respect to the x-axis is the point R.

- Point subtraction: If $Q = -P$, then $P + Q = -P$ i.e., the line joining P and −P intersects the curve at the point O.
- Point doubling: Point doubling is the addition of a point P on the curve to itself to obtain another point Q on the same curve.

The scalar point multiplication on the cyclic group G_p is defined as $kp = p + p \ldots + p(k\ times)$. Let $Ep\ (a,\ b)$ be a set of elliptic curve points over the prime field F_p, defined by the following non-singular elliptic curve equation:

$$y^2 modp = \left(x^3 + ax + b\right)\ mod\ p \tag{1}$$

With $a,\ b \in\ Fp$ and $(4a^3 + 27b^2)\ modp \neq 0$

Further, the elementary operations like point multiplication, addition scalar multiplication in the elliptic curve group are significant faster than the modular exponentiation and bilinear pairings performed in the multiplicative group. The relative computation cost of the bilinear pairing is roughly two to three times more than an elliptic curve scalar point multiplication (ECPM) [19,20].

3.2 BLAKE Hash Function

BLAKE is a cryptographic hash function created by O'Connor et al. The variants of BLAKE are much faster than MD5, SHA-1, SHA-2, SHA-3 algorithms [21]. BLAKE is 128-bit algorithm which can resist preimage, collision, or differentiability attack [22]. However, there are currently no attacks that are effective against these hashing algorithms. Tab. 1 illustrates versions of BLAKE algorithm that can save

space by limiting the maximum input size and number of rounds possible [23]. This feature makes it appropriate for IoT devices. The software implementation of BLAKE is faster compared to Keccak [24]. For the aforementioned reasons, digestive Message Authentication Code (MAC) of BLAKE has been considered for session key generation. Tab. 1 illustrates the features of BLAKE algorithm.

Table 1: BLAKE hash algorithms

BLAKE version	Input size	Digest size (Bytes)	Number of rounds	Word size (bits)
BLAKE2b	$0 \leq l < 64$	$0 \leq l < 64$	12	64
BLAKE2s	$0 \leq l < 32$	$0 \leq l < 32$	10	32
BLAKE3	$0 \leq l < 64$	$0 \leq l < 64$	7	32

3.3 Procedure for Conversion of Monotone Boolean Access Policy (ĂP) into LSSS Matrix

Attributes in the access policy ($\check{A}P$) used in CP-ABE is obfuscated by converting into access matrix. Regardless of the operator in the root node in access policy, it is given a vector value of $\vec{v} = 1$ at the start.

i) If the parent node type is either an AND/OR gate, it has a vector value of $\vec{v} = (1, \ldots, 1)$ during tree traversal.

ii) The algorithm identifies its pair of child nodes as follows if the parent node type is an AND gate: The right child's node, for example, has a vector value of $(1, \vec{v})$. Second, the vector value of the left child node is $(0,\ldots,0, -1)$.

iii) The access matrix stores the leaf node vector value. To make the dimension of the matrix identical, the vector value $\vec{v}$ is prefixed with '0' at the leaf node.

iv) The parameter ρ maps each row of the access matrix to attributes respectively. The access policy and its corresponding BET and LSSS is displayed in Fig. 1.

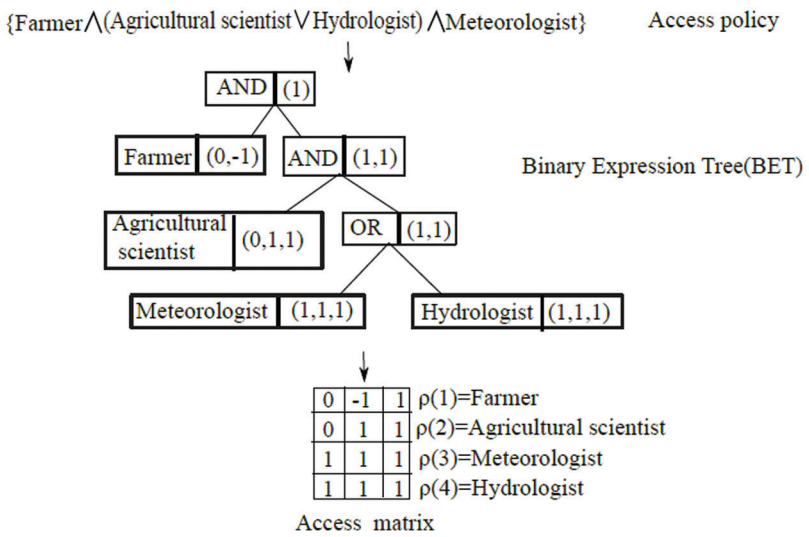


Figure 1: Access matrix representation

4 A Use-Case Model

Farming Area Network (FAN) communication model can be considered as a use-case for the proposed scheme. FAN is deployed in agro-climatic zones in hostile location. The physical and functional components of the architecture are discussed in detail as shown in Fig. 2.

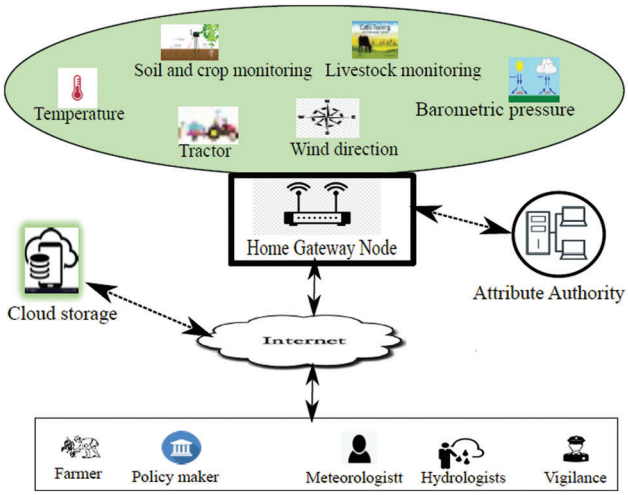


Figure 2: Farming area network model

4.1 FAN Devices

The FAN system comprises of diverse sensors. Further, on the basis of functionality, the sensors can be categorized into different FAN devices like meteorological devices, fire detection devices, livestock monitoring, autonomous tractor, crop and soil monitoring devices. These devices consist of network of sensors that captures and collects real time data like weather conditions, inferno values, health parameters and location-tracking of livestock, soil and crop related information. Basically, they act like data source (DS) generating data at subsequent interval of time.

4.2 Home Gateway Node (HGWN)

The FAN devices are connected to the internet through the gateway node. Initially, the FAN device authenticates itself by submitting the credentials to the HGWN. Similarly, authenticated users can access the services of the pertinent FAN devices through the gateway node. This acts as an interface between raw data generator (FAN devices) and data receivers. Besides, it stores ciphered data generated by FAN devices.

4.3 Attribute Authority (AA)

The vital and reliable entity in the entire architecture is AA. It offers role-based access control (RBAC) mechanisms to users. It can be intended that AA acts like a key generation center. It oversees access policy generation and updating, issuing and revoking users' attributes in accordance with their roles.

4.4 Data Receivers (DRs)

Data receiver refers to the end user of the services. The attributes that a DR is authorized is defined during the deployment of the system and can be updated later. A DR having set of attributes defined in access policy can decrypt the data sent by DS.

5 Proposed Framework

The proposed framework, shown in Fig. 3, uses five algorithms to provide secure end-to-end communication between DS and DR in an IoT environment. These include BLAKE Hashing Code, Block Encryption Algorithm, Improved PF-CP-ABE Scheme, and Block Decryption. The following paragraph describes the steps in the framework.

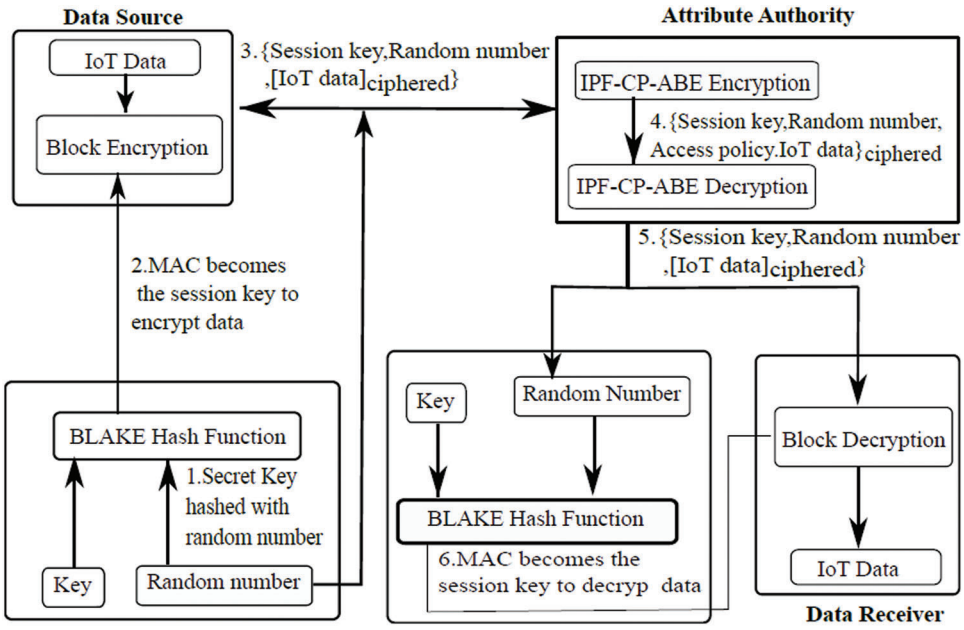


Figure 3: BLAKE-IPF-CP-ABE framework

Step 1: Hashing the secret key with a random number on the sender side: In the first step, the secret key is hashed with a random number. Appropriate and high-quality random number generators (RNG) help in the generation of effective encryption session keys. To create random session keys, cryptographically secure pseudo random number generators (CSPRNGs) are used effectively.

Step 2: BLAKE Digest turn into Encryption Session Key: The result of BLAKE hashing operation generates a 32-byte and 64-byte MAC. This MAC becomes the session key for block encryption of IoT data.

Step 3: A cryptographic session key, encrypted data, and random number are transmitted for storage at an attribute authority that functions as a trusted data center. The relevant DR can decrypt the data in accordance with the access policy.

Step 4: Session key and random number encryption using Improved PF-CP-ABE: Session keys and random numbers are encrypted in accordance with access policies using the IPF-CP-ABE algorithm, which produces a ciphertext CT. In Section 5.2, IPF-CP-ABE describes this step in more detail.

Step 5: Session key and Random number Decryption using IPF-CP-ABE: By DR leveraging the IPF-CP-ABE decryption procedure, the ciphered session key and random number are decrypted based on the access policy. This procedure recovers original session key and random number for further decryption of IoT data.

Step 6: Hashing the secret key with a random number on the receiver side: In DR, BLAKE key is hashed with random number to generate a 32-byte MAC that becomes the session decryption key. Both the generated session decryption key and transmitted session key are checked for equality. If verification succeeds, the DR

applies the decryption algorithm using the session key. Otherwise, the decryption fails, indicating that an adversary tampered with the transmitted data.

Step 7: DR Uses Session Key to Decrypt Data: The DR retrieves the encrypted data from the DS by executing the block decryption algorithm with the session key.

5.1 Notations and Abbreviations

The notation and abbreviation used in the proposed scheme are listed in Tab. 2

Table 2: Notations

Notations	Description
mpk	Master public key
P_{ui}	Public key
Attr_i	Attribute identifier
r_i	Random number
msk	Master secret key
RN_B	Random number of BLAKE variant
Sk_i	Secret key
UUID	Unique user identity
M	Sensor data
K	Session key
ĂP	Access policy
CT_i	Cipher text
μ, μ_1	Random number
M	LSSS matrix
Enc_t	Encryption time
Dec_t	Decryption time
$Exec_t$	Execution time
AA	Attribute authority
LSSS	Linear secret sharing scheme
SPECS	Public parameters
DS	Data sender
DR	Data receiver

5.2 Improved PF-CP-ABE Scheme

As illustrated in the Fig. 4, the scheme is broken down into sub-algorithms as follows:

5.2.1 Authority Setup/Initialization

An implicit variable (ECC domain parameters) is selected in this phase, which is carried out by the attribute authority. The AA constructs a universal set of attributes as $U = \{Attr_1, Attr_2, \ldots, Attr_n\}$. For

each attribute, $Attr_i \epsilon U$, the AA chooses a random number $r_i \epsilon Z_p$. The public specifications, $SPECS = \{U, Pu_i, mpk\}$ are disclosed and generated as follows.

$$Pu_i = r_i G \tag{2}$$

$$mpk = \omega G, \tag{3}$$

where ω is a random number $\omega \epsilon Z_r$

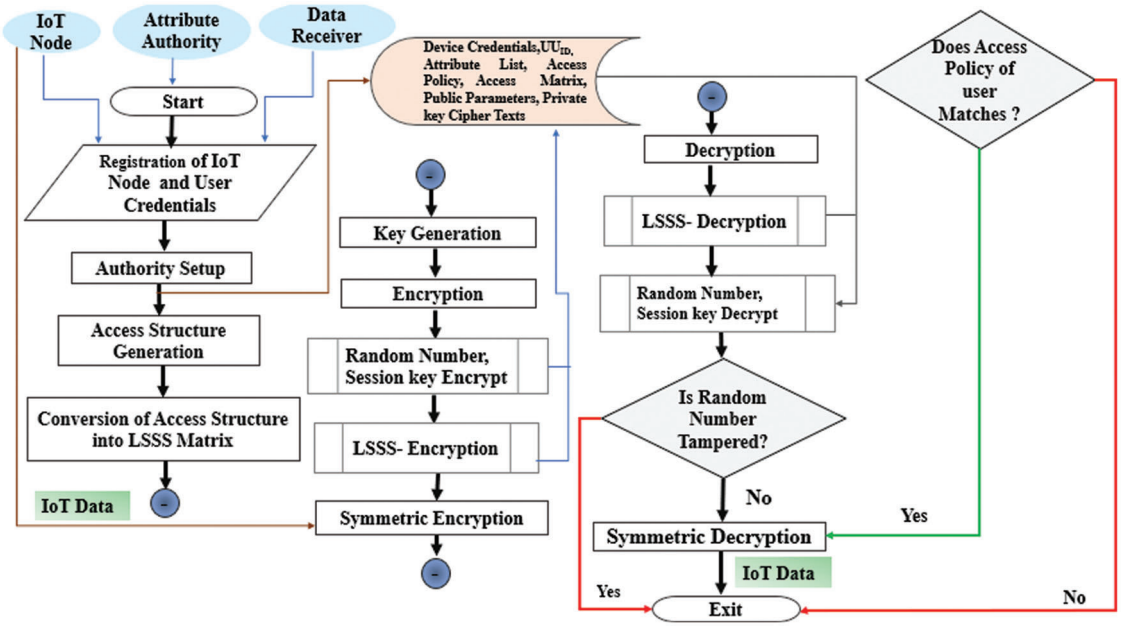


Figure 4: Processing steps of IPF-CP-ABE

5.2.2 Key Generation$(SPECS;\ Attr_i;\ UUID;\ msk)$

This algorithm is executed by AA. The key generation algorithm takes the public parameters, an attribute $Attr_i$, a unique user identity $UUID$, and the master secret key msk of the attribute authority as input. AA produces an attribute secret key Sk_i with an associated $UUID$ and distributes it to eligible users.

$$SK_{i,UUID} = r_i + H(UUID)\omega \tag{4}$$

5.2.3 Encryption $(Pu,\ M,\ \breve{A}P,\ K,\ RN_B)$

The input given to this algorithm is public key Pu, sensor data M, session key K, access policy $\breve{A}P$, a random number RN_B of BLAKE. This step consists of five sub-algorithms which are described as follows:

- SYM-ENCRYPT: Block encryption algorithms can be used to encrypt plaintext message M (sensor data), using the session key K. This module is executed by IoT node. The initial ciphertext is denoted by Eq. (5).

$$CT_0 = E\ (K,\ M) \tag{5}$$

- RN_B-ENCRYPT: RN_B is encrypted to a point P on the elliptic curve E. Choose a random number $\mu \epsilon \mathbb{Z}_p$. The second ciphertext component is represented by Eq. (6) as

$$CT_1 = RN_B + \mu G \tag{6}$$

- SK-ENCRYPT: The third cipher text is computed by encrypting the session key K to a point P on the elliptic curve E. Choose a random number $\mu_1 \epsilon \mathbb{Z}_p$.

$$CT_2 = K + \mu_1 G \tag{7}$$

- LSSS-Encrypt: A Column vector $\vec{v} = [s, q_1, q_2, \ldots q_3]$ is generated, whose elements $[s, q_1, q_2, \ldots q_3] \ \epsilon \ Zp$. The size of v is equal to the number of columns of the access matrix $\mathcal{M}_x$. The initial element s of vector v is the secret to be shared among the communicating parties. Calculate the first component λ_x. *The* term λ_x denotes a share and is computed as shown in Eq. (8).

$$\lambda_x = \mathcal{M}_x \vec{v} \tag{8}$$

Compute the second term $\beta_x . Pu_{\rho(x)}$. A column vector $\vec{u} = [0, u_1, \ldots, u_n]$ is generated with 0 as its first element.

$$\beta_x = \mathcal{M}_x . \vec{u} \tag{9}$$

$Pu_{\rho(x)}$ denotes the public key of each attribute in access matrix $\mathcal{M}_x$. Using step (1) and (2) the fourth cipher text component is computed for each attribute is denoted by Eq. (10):

$$CT_{3,x} = \lambda_x + \beta_x . Pu_{\rho(x)} \tag{10}$$

The fifth cipher text component is computed as:

$$CT_{4,x} = \beta_x . G \tag{11}$$

The final cipher text is given as

$$CT = \left(CT_0, CT_1, CT_2, CT_{3,x}, CT_{4x}\right) \tag{12}$$

5.2.4 Decryption (CT, Pu, SK)

The decryption algorithms are executed by the *AA* and *DR* respectively. Eventually, DR submits its *UUID* to *AA* to obtain the original data. AA verifies attribute list corresponding to a *UUID* to check whether it aligns with the access policy $\left(\breve{A}P\right)$. On successful match, the order of decryption is decoded as follows:

- LSSS-decrypt: The authority and DR decrypts the ciphertext component to map the matrix $\mathcal{M}_x$ to fetch the attributes in order to obtain the decryption key.

$$\sum CT_{4,x} \, SK_{i,UUID} = \lambda_x G + H(UUID) nG \tag{13}$$

DR then selects constants $c_x \epsilon \mathbb{Z}_r$ to obtain the secret key μG and $\mu_1 G$

$$\sum c_x (\lambda_x . G - \beta_x H(UUID) . nG) = \mu . G \tag{14}$$

$$\sum c_x \left(\lambda_x . G - \beta_x H(UUID) . nG \right) = \mu_1 G \tag{15}$$

- SK-DECRYPT: The DR computes the session key

$$CT_2 - \mu_1 G = K \tag{16}$$

- RN_B-DECRYPT: The DR computes the random number RN_B

$$CT_1 - \mu G = RN_B \tag{17}$$

- SYM-DECRYPT: Finally, the message is decrypted using the session key

$$CT_0 = D(K, M) \tag{18}$$

5.3 Attribute/User Revocation

The proposed scheme incorporates direct revocation of DR without affecting other DR's secret key. *AA* maintains the list of DR's attributes set and corresponding *UUID* for each recipient. Consider an example with the following details {DR = Meteorologist, attribute list = (temperature, wind speed, barometric pressure)}. To revoke a specified DR, the AA deletes its attribute list and its *UUID*. To illustrate this, suppose the role of vigilant personnel is inactive for a very long time, the *AA* can delete its *UUID* and the corresponding attribute list. Also, to revoke an attribute of the system, simply deletes the corresponding attribute's public key. Further, to revoke an attribute from the attribute list, delete the attribute from attribute list corresponding to its *UUID*. Based on the nature of application, the revocation time period might vary from weeks to months.

A session key elapses once the communication between the DS and DR terminates. There exists a trade-off if the life span of the session key is short, the security of an IoT system is enhanced, but at the cost of high computation and communication overheads. Normally, any FAN device routes the sensed data through HGN and is monitored by an *AA*.

Therefore, a DC can access the data through HGN. In this case, the expiry time of the session key may vary from weeks to months. Consider a scenario of direct communication between meteorological sensors and meteorologist during an adverse weather condition. The meteorologist adjusts the sensor parameters directly, where in the validity period of the session key lasts only few minutes. Further, setting the expiry time of session key relies on the type of critical and massive real-time IoT applications. There exist few methods to implicitly invalidate the session key at the end of its session, irrespective of session duration. Encryption session keys are updated periodically for key freshness and to maintain the trustworthiness of the IoT systems.

6 Security Analysis

The security of the TE-BLAKE3-PF-CP-ABE system is defined using an indistinguishability game between the adversary *Adv* and the challenger *Ch*.

6.1 Security Model

Init. Adv specifies an access structure $(M *, \rho)$ for challenge.

Setup. The challenger Ch impersonates as TA and executes the setup algorithm.

Phase 1: For decryption, Adv does numerous queries to collect the sets of secret key component $\{SK_i, \eta, \eta_1\}$ and access structure (M;, ρ) associating to the sets of attributes $\{Attr1, Attr2, \ldots, Attrn\}$.

Challenge: Now, *Adv* transmits two equal length messages $M0$ and $M1$ to *Ch* for encryption under the challenge access structure (M, ρ). Next, *Ch* flips a fair binary coin $B \in \{0, 1\}$ and generates the challenge ciphertext CT which is sent to *Adv*.

Phase 2: Both *Adv* and *Ch* repeats to query as in phase 1.

Guess: *Adv* sends a guess a bit μ to *Ch*. The adversary *Adv* wins the game if $B = B_0$. Adv wins the game with an advantage defined as $Pr[\mu 0 = \mu] - 1/2$.

6.2 Resistance to Attacks

This subsection discusses about the attacks and goals that the proposed framework can withstand in an adverse circumstance.

Data Confidentiality: The proposed framework uses PF-CP-ABE algorithm to encrypt IoT data, session key and random number. IoT data is kept confidential by DS through encryption using SYM-ENCRYPT algorithm. The session key K and RNB are kept secret by the DS using the two modules RN_B-ENCRYPT and SK-ENCRYPT respectively. Both modules use PF-ECC to encrypt the two parameters. The attacker cannot expose RNB due to problem and hence cannot derive the session key. Besides, LSSS-matrix oriented access policy provides fine-grained access control to DR to obtain RN_B.

Key and Data Integrity: The authenticated users (DR's) are assigned a cryptographically secure random $UUID$ with timestamp. During transit RN_B can be manipulated by an attacker during transit. This is detected at the recipient side, iff RN_B matches, a MAC generated session key can decrypt the message. Otherwise, decryption fails irrespective of valid UUID possessed by a user. DS and DR should establish a completely random session key, which is only valid for one transaction and cannot be reused therefore, preventing replication attacks. Moreover, the security goals of key and data integrity are preserved.

Non-repudiation of Origin (NRO): The NRO in the proposed scheme is guaranteed by secret random number (RNB) transmitted exclusively in a secure channel by DS. This ensures data origin authentication. Further, DR cannot claim denial of submission by DS. Similarly, DR cannot claim denial of delivery of data by DS.

Man in the Middle Attack (MITM): The FAN device must authenticate themselves to AA in FAN network to obtain the session key K. An energy efficient authentication method, a customized BLAKE2b (c-BLAKE2b) hashing algorithm is used. The hash value of the FAN node is generated by AA as H (ID_{DS}). AA can identify non-registered users termed as intruders. Therefore, the scheme is secure against MITM attack.

Perfect forward secrecy: An adversary cannot obtain the previous session key, even when a current session key is compromised, any attacker cannot derive any previous session key. A unique session key for each session is generated from CSRNG generator and BLAKE hash function. This property prohibits any revoked user from accessing future data. To revoke a specified DR, the AA deletes its attribute list and its $UUID$. Further, to revoke an attribute from the attribute list, delete the attribute from attribute list corresponding to its $UUID$. Revoked DR cannot decrypt the ciphertext, since their UUID is deleted. Hence, forward security is preserved by this scheme.

Resist to Collusion Attack: The AA assigns each DR a UUID and relevant set of attributes. Additionally, each attribute has an arbitrary secret key SK_i and computed Pu_i. Also, each DR has an access policy with the equivalence access matrix M, tagged with encrypted sensor values. Consumers with adequate attributes matching the $\breve{A}$ can decrypt data. Similarly, revoked users whose UUID is removed cannot access the data. An unauthorized user without proper credentials cannot collude with other consumers. Thus, the scheme can resist a collusion attack accomplished by unauthorized and revoked users.

7 Results and Discussions

The implementation of proposed framework BLAKE-IPF-CP-ABE scheme is done with AMD Ryzen at 3.60 GHz and 4 GB RAM. The scheme runs on Ubuntu 20.04.1 LTS using Python 3.7 in PyCharm framework. The evaluation of proposed scheme uses Rainfall data set which is freely available in public domain.

7.1 Performance Analysis

The proposed scheme employs Brainpool P256r1 elliptic curve group based on the short Weierstrass curve of the form $y^2 = (x^3 + a.x + b)$ to achieve a security level of 128 bits. The performance of the BLAKE-IPF-CPABE security algorithm have been compared with Lewko and Waters CPABE (LW-CP-ABE) and PF-CP-ABE schemes.

7.1.1 Encryption Time (Enc_t)

This indicates the time taken by the DS and AA to encrypt the IoT data embedded with access policy. This is calculated as MAC generation time using BLAKE algorithms to generate session key, starting and ending time of SYM-ENCRYPT adopted for IoT data encryption, initial time for and finishing time for RNB-ENCRYPT, SK-ENCRYPT and LSSS-ENCRYPT.

7.1.2 Decryption Time (Dec_t)

This indicates the time taken by the AA and DR to retrieve the original data by adhering to $\breve{A}P$. This is calculated as starting and ending time of RN_B-DECRYPT to generate session key, starting and ending time of SYM-DECRYPT, LSSS-DECRYPT.

7.1.3 Execution Time ($Exec_t$)

This indicates the total time taken for encryption and decryption.

$$\mathrm{Exec_t} = \mathrm{Enc_t} + \mathrm{Dec_t} \tag{19}$$

7.1.4 Comparative Analysis

The proposed BLAKE-PF-CP-ABE is compared with LW-CP-ABE for three versions of BLAKE algorithm and nine symmetric encryption algorithms. The scheme has been evaluated in terms of execution time performed on 1, 5 and 10 KB dataset with 7, 14, and 21 attributes respectively. The graph is plotted as shown in Fig. 4. It can be inferred that 128-bit ciphers such as LEA, AES, Twofish, Camelia and TEA performed subtle for three different attribute values of different data size. Similarly, Blowfish and DES variant have higher $Exec_t$ due to series of complicated operations and block size of 64 bits. In addition, Blowfish has 448-bit key length.

Tabs. 3–5 shows the values of the two schemes. However, the execution time of BLAKE2b oriented ABE is quite higher compared to BLAKE3 and BLAKE2s. Further, when the number of attributes and file size increases, $Exec_t$ of BLAKE2b is slightly higher. The reason can be stated that the computation done by the parameter block of BLAKE2b is high and the number of rounds is more compared to Blake2s and BLAKE3. The efficiency of the proposed scheme is compared with LW-CP-ABE. The operational cost of LW-CPABE is expensive due to bilinear pairing compared to PF-CPABE which uses scalar multiplication. Hence, the $Exec_t$ of PF-CPABE is slightly low compared to LW-CPABE. The lightweight block ciphers like LEA, ChaCha20 performed reasonably well for different data size when integrated with BLAKE and PF-CPABE algorithms. The performance analysis plotted in graph is shown in Fig. 5.

Table 3: 1 kb with 7 attributes

Algorithm	LW-PF-CP-ABE (ms)	PF-CP-ABE (ms)	IPF-CP-ABE (ms)		
			BLAKE3	BLAKE2b	BLAKE2s
LEA	175.23	165.11	322.37	374.97	371.94
AES	198.3	172.56	310.68	357.79	350.62
Twofish	178.45	169.89	305.39	352.91	357.88

Table 3 (continued)

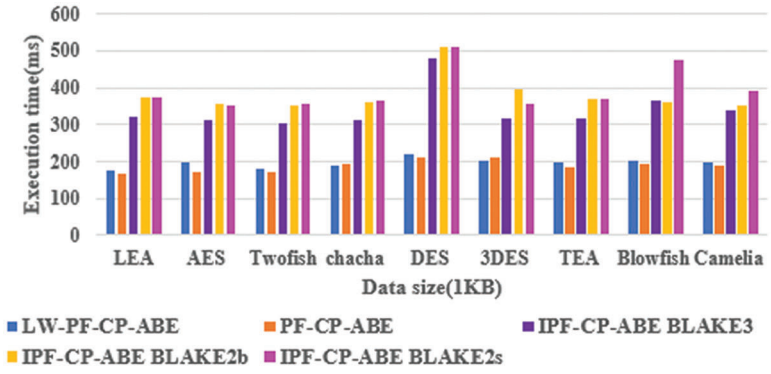
Algorithm	LW-PF-CP-ABE (ms)	PF-CP-ABE (ms)	IPF-CP-ABE (ms)		
			BLAKE3	BLAKE2b	BLAKE2s
Chacha	187.67	192.12	311.99	361.49	364.17
DES	219.56	212.23	478.9	508.95	511.51
3DES	200.2	211.45	315.88	395.27	356.79
TEA	199.81	185.79	316.82	367.93	368.69
Blowfish	202.34	195.17	365.06	362.35	474.57
Camelia	198.73	189.41	339.75	352.99	392.17

Table 4: 5 kb with 14 attributes

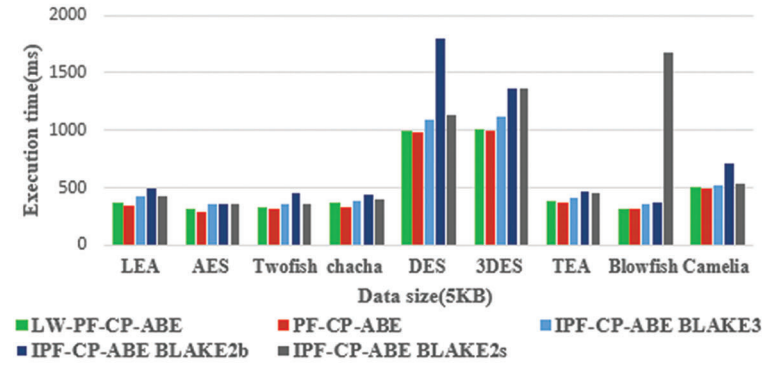
Algorithm	LW-PF-CP-ABE(ms)	PF-CP-ABE(ms)	IPF-CP-ABE(ms)		
			BLAKE3	BLAKE2b	BLAKE2s
LEA	361.1	340.23	423.487	490.697	425.137
AES	311.2	285.67	356.97	359.82	351.337
Two fish	325.8	312.12	357.097	455.776	359.747
Cha-cha	360.9	323.52	379.347	434.687	393.887
DES	995.22	981.6	1092.737	1803.717	1138.847
3DES	1000.4	994.49	1123	1368.627	1366.157
TEA	373	358.9	416.497	470.987	458.837
Blowfish	311.6	303.8	364.177	372.077	1673.21
Camelia	501.2	487.35	520.127	709.567	537.677

Table 5: 10 kb with 21 attributes

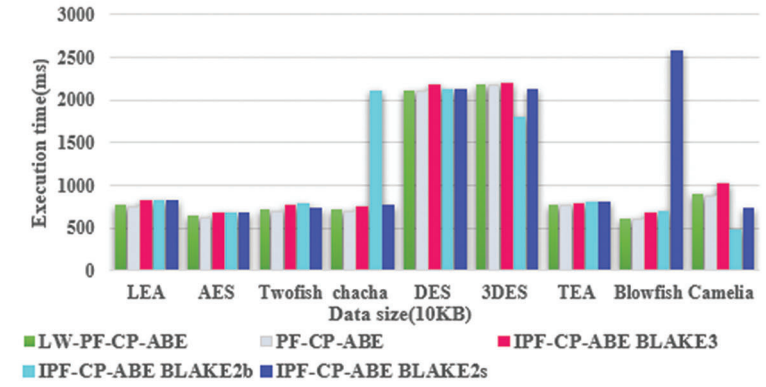
Algorithm	LW-PF-CP-ABE (ms)	PF-CP-ABE (ms)	IPF-CP-ABE (ms)		
			BLAKE3	BLAKE2b	BLAKE2s
LEA	780.1	755.41	815.556	810.776	822.006
AES	654.77	625	682.046	679.906	673.446
Two fish	720.22	695.26	759.086	775.855	737.476
Cha-cha	721	700.87	744.246	2100	761.726
DES	2119	2101.2	2170.156	2121.796	2126.076
3DES	2189.72	2169.98	2200	1791	2111.89
TEA	765.48	756	787.626	797.386	797.386
Blowfish	604.51	597.51	680.846	692.156	2577.046
Camelia	895.8	865.43	1018.366	480.246	735.606



(a) 1kb with 7 attributes



(b) 5kb with 14 attributes attributes



(c) 10kb with 21 attributes

Figure 5: Comparison plot

7.2 Computation Overhead (CO)

The processing effort of the BLAKE-IPF-CP-ABE framework between the DS and DR consists of operations such as encryption, decryption, hash, and number of attributes for access matrix generation. The cost of symmetric encryption as in Eq. (20) depends on the execution time of symmetric algorithms C_{se} which relatively varies based on the type of structure used. Similarly, the cost of decryption is calculated as in Eq. (21). The notations used in the analysis for calculating the computation overhead is tabulated in Tab. 6.

$$C_{oe} = (4C_{SM} + 3C_{SA})n_r + C_{MAC} + C_{se} \quad (20)$$

$$C_{od} = (4C_{SM} + 3C_{SA})n_r + C_{MAC} + C_{sd} \quad (21)$$

Table 6: Notations

Notation	Description
C_{SM}	Scalar multiplication
C_{SA}	Scalar addition
C_H	Cost of hash function
C_{MAC}	Cost of MAC function
C_{se}	Cost of the symmetric encryption function
C_{sd}	Cost of the symmetric decryption function
n_r	Number of rows in the access matrix

The elliptic curve BrainpoolP256r, i.e., 256-bit indicated by *S*. Due to session key and random number RN_B encryption, the suggested approach has a larger ciphertext bit size than the other two systems. The computational overhead is determined by the number of rows in the access matrix, the number of attributes in the global attribute set (n_{attr}), the number of attributes in the access policy (n_{aap}), and the minimum number of attributes required for decryption in the access policy (d_{attr}). The computation overhead for encryption is $(5n_r + 1)$ *S* and for decryption *is* $d_{attr} + 1$.

7.3 Communication Cost (CC)

The raw IoT data generated from various FAN devices at periodic intervals of time denoted as $\{t_1, t_2, \ldots, t_n\}$ is encrypted by lightweight cipher algorithms. The encrypted data is transmitted by DS in a secure channel and it can be either stored in the AA or HGN acting as a data sink. The data consumers can access the data from the data sink on a demand basis by submitting the relevant credentials. The communication cost at the sender side (CC_{DS}) is affected by the size of the encrypted data of variable length *l*, propagation delay (T_p), and transmission delay (T_D).

$$CC_{DS} = (CT_0 + CT_1 + CT_2)l + T_p + T_D \quad (22)$$

The communication overhead of IPF-CP-ABE varies based on the lengths of the ciphertext, public key, and private key. The ciphertext in the proposed system is $CT = \{CT_1, CT_2, CT_3, CT_4\}$. Thus, the size of the ECC-based public key is $2|S|$, while the size of the private key is supplied by $|S|$. Due to the session key and random number RN_B encryption, the suggested approach has a larger ciphertext bit size than the other two systems. The size of ciphertext is $(4n_r + 1)$ *S*. The public key length is $2(n_{attr} + 2)$ S and the private key size is $(n_{aap})S$.

8 Conclusions

The proposed BLAKE-IPF-CPABE framework establishes a secure data transmission between IoT node and data consumers deployed in remote location. Eventually, the scheme enforces a fine-grained access control by constructing an access matrix generated from an access policy using attributes. This allows only authenticated users to access the IoT sensor data. Further, integrity and message authentication of the sender is achieved by using RNB and MAC digest obtained using BLAKE variants. The execution time of BLAKE-PF-CPABE for various encryption algorithm is slightly less when compared with

LW-CP-ABE. It was found that BLAKE3 and BLAKE2s combined with LEA, Twofish oriented PF-CPABE had less execution time and memory consumption of 30%–40%. The scheme can be claimed to be secure against collusion attack and replication attacks. Therefore, it can be inferred that the proposed framework provides a secure and efficient data communication in IoT networks. The future work will focus on power consumption of the proposed scheme in real time setup and Break the Glass mode policy (BTG-P) for IoT environment.

Acknowledgement: The authors would like to appreciate the effort of the editors and reviewers. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] P. P. Ray, "Internet of things for smart agriculture: Technologies, practices and future direction," *Journal of Ambient Intelligence and Smart Environments*, vol. 9, no. 4, pp. 395–420, 2017.

[2] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. Annual Int. Conf. on the Theory and Applications of Cryptographic Techniques*, Berlin, Heidelberg, pp. 457–473, 2005.

[3] J. Bethencourt, A. Sahai and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. 2007 IEEE Symp. on Security and Privacy (SP '07)*, Berkeley, CA, USA, pp. 321–334, 2007.

[4] V. Goyal, O. Pandey, A. Sahai and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. on Computer and Communications Security*, New York, NY, United States, pp. 89–98, 2006.

[5] S. S. Dhanda, B. Singh and P. Jindal, "Lightweight cryptography: A solution to secure IoT," *Wireless Personal Communications*, vol. 112, no. 3, pp. 1947–1980, 2020.

[6] V. A. Thakor, M. A. Razzaque and M. R. A. Khandaker, "Lightweight cryptography algorithms for resource-constrained iot devices: A review, comparison and research opportunities," *IEEE Access*, vol. 9, pp. 28177–28193, 2021.

[7] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Proc. Annual Int. Conf. on The Theory and Applications of Cryptographic Techniques*, Berlin, Heidelberg, pp. 568–588, 2011.

[8] B. Chandrasekaran, R. Balakrishnan and Y. Nogami, "TF-CPABE: An efficient and secure data communication with policy updating in wireless body area networks," *ETRI Journal*, vol. 41, no. 4, pp. 465–472, 2019.

[9] C. Hu, H. Li, Y. Huo, T. Xiang and X. Liao, "Secure and efficient data communication protocol for wireless body area networks," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 2, no. 2, pp. 94–107, 2016.

[10] S. Banerjee, S. Roy, V. Odelu, A. K. Das, S. Chattopadhyay *et al.,* "Multi-authority CP-ABE-Based user access control scheme with constant-size key and ciphertext for IoT deployment," *Journal of Information Security and Applications*, vol. 53, no. 15, pp. 102503, 2020.

[11] V. Odelu, A. K. Das, M. Khurram Khan, K. R. Choo and M. Jo, "Expressive CP-ABE scheme for mobile devices in IoT satisfying constant-size keys and ciphertexts," *IEEE Access*, vol. 5, pp. 3273–3283, 2017.

[12] M. Ambrosin, A. Anzanpour, M. Conti, T. Dargahi, S. R. Moosavi *et al.,* "On the feasibility of attribute-based encryption on internet of things devices," *IEEE Micro*, vol. 36, no. 6, pp. 25–35, 2016.

[13] X. Yao, Z. Chen and Y. Tian, "A lightweight attribute-based encryption scheme for the internet of things," *Future Generation Computer Systems*, vol. 49, no. 1, pp. 104–112, 2015.

[14] H. Hong and Z. Sun, "A key-insulated ciphertext policy attribute based signcryption for mobile networks," *Wireless Personal Communications*, vol. 95, no. 2, pp. 1215–1228, 2017.

[15] A. Karati, R. Amin and G. P. Biswas, "Provably secure threshold-based abe scheme without bilinear map," *Arabian Journal for Science and Engineering*, vol. 41, no. 8, pp. 3201–3213, 2016.

[16] K. Sowjanya, M. Dasgupta, S. Ray and M. S. Obaidat, "An efficient elliptic curve cryptography-based without pairing KPABE for internet of things," *IEEE Systems Journal*, vol. 14, no. 2, pp. 2154–2163, 2020.

[17] S. Ding, C. Li and H. Li, "A novel efficient pairing-free CP-ABE based on elliptic curve cryptography for IoT," *IEEE Access*, vol. 6, pp. 27336–27345, 2018.

[18] K. Sowjanya and M. Dasgupta, "A ciphertext-policy attribute based encryption scheme for wireless body area networks based on ECC," *Journal of Information Security and Applications*, vol. 54, no. C, pp. 102559, 2020.

[19] C. A. Lara-Nino, A. Diaz-Perez and M. Morales-Sandoval, "Elliptic curve lightweight cryptography: A survey," *IEEE Access*, vol. 6, pp. 72514–72550, 2018.

[20] D. He, J. Chen and R. Zhang, "An efficient and provably-secure certificateless signature scheme without bilinear pairings," *International Journal of Communication Systems*, vol. 25, no. 11, pp. 1432–1442, 2012.

[21] J. P. Aumasson, S. Neves, Z. Wilcox-O'Hearn and C. Winnerlein, "BLAKE2: Simpler, smaller, fast as MD5," in *Proc. Int. Conf. on Applied Cryptography and Network Security*, Berlin, Heidelberg, pp. 119–135, 2013.

[22] G. Bertoni, J. Daemen, M. Peeters and G. V. Assche, "Keccak," in *Proc. Annual Int. Conf. on the Theory and Applications of Cryptographic Techniques*, Berlin, Heidelberg, pp. 313–314, 2013.

[23] V. Rao and K. V. Prema, "Light-weight hashing method for user authentication in Internet-of-Things," *Ad Hoc Networks*, vol. 89, no. 67, pp. 97–106, 2019.

[24] J. W. Bos, C. Costello, P. Longa and M. Naehrig, "Selecting elliptic curves for cryptography: An efficiency and security analysis," *Journal of Cryptographic Engineering*, vol. 6, no. 4, pp. 259–286, 2016.