Hindawi
Mobile Information Systems
Volume 2021, Article ID 2427434, 10 pages
https://doi.org/10.1155/2021/2427434

Research Article

Cost-Effective Proxy Signcryption Scheme for Internet of Things

Insaf Ullah,[1] **Ali Alkhalifah**,[2] **Muhammad Asghar Khan**,[1] **and Samih M. Mostafa**[3]

[1]*HIET, Hamdard University Karachi, Islamabad Campus, Islamabad 44000, Pakistan*
[2]*Department of Information Technology, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia*
[3]*Faculty of Computers and Information, South Valley University, Qena 83523, Egypt*

Correspondence should be addressed to Muhammad Asghar Khan; khayyam2302@gmail.com

Received 8 July 2021; Revised 31 August 2021; Accepted 15 October 2021; Published 30 November 2021

Academic Editor: Zengpeng Li

The Internet of things (IoT) has emerged into a revolutionary technology that enables a wide range of features and applications given the proliferation of sensors and actuators embedded in everyday objects, as well as the ubiquitous availability of high-speed Internet. When nearly everything is connected to the Internet, security and privacy concerns will become more significant. Furthermore, owing to the resource-constrained nature of IoT devices, they are unable to perform standard cryptographic computations. As a result, there is a critical need for efficient and secure lightweight cryptographic scheme that can meet the demands of resource-constrained IoT devices. In this study, we propose a lightweight proxy in which a person/party can delegate its signing authority to a proxy agent. Existing proxy signcryption security approaches are computationally costly and rely on RSA, bilinear pairing, and elliptic curves cryptography (ECC). The hyperelliptic curve cryptosystem (HECC), on the other hand, employs a smaller key size while maintaining the same level of security. When assessed using the random oracle model (ROM), the proposed scheme provides resilience against indistinguishable under adaptive chosen ciphertext attacks (IND-CCA) and unforgeable under adaptive chosen message attacks (UU-ACMA). To demonstrate the viability of the proposed scheme, security analyses and comparisons with existing schemes are performed. The findings show that the proposed scheme provides high security while reducing computational and communication costs.

1. Introduction

Modern enterprises and business organizations require the delegation of signing rights due to a lack of processing capability or the temporal absence of an agent. Similarly, it attracted e-commerce applications like signing the business contract and online proxy auction. To provide the delegation of rights, Mambo et al. [1, 2] were the first who contributed a new method called a proxy signature. This approach includes three participants: original signer, proxy signer, and a verifier/receiver. The original signer can delegate its signing rights to the delegated agent/proxy signer. Later, the delegated agent uses this sign on the behalf of its delegator and delivers it to the respective verifier/receiver. Unfortunately, the schemes in [1, 2] do not provide any solution to prevent it from misuse. Another attempt in enhancing proxy signature was made by Kim et al. [3]. They claim that the partial delegation with a warrant is more impactful and secure than full delegation in terms of computations and more processing speed. But it gives unlimited delegation resulting misuse of delegation. Another scheme proposed in [4] gives the concept of nonrepudiation by devising the threshold proxy signature scheme (TPSS). The scheme successfully preserves the nonrepudiation between the original sender and proxy groups without involving the trusted third party.

Though, the proxy signature will fail when communication includes some commercial secrets. Thus, to resolve this problem, Gamage et al. [5] designed a proxy signcryption approach by combining proxy signature and the encryption in a single logical step. The proficiency and security strength of the given approach relies upon the discrete logarithm problem which causes making it more costly in terms of both computation and communication. In addition, the proposed approach does not provide some security services like forward secrecy and public verifiability. Zhang [6] contributed publicly verifiable and forward secure proxy

signcryption scheme. His proposed scheme is inefficient as it needs a secure channel between the sender and proxy. In addition, the proposed scheme creates more computational cost and requires more bandwidth for communication. Li and Chen [7] used pairing phenomena in the identity-based proxy signcryption (IDBPYS) scheme that necessarily requires a safe medium for transferring the secret key to the user. Wang et al. [8] proposed an IDBPYS scheme that satisfies the security parameters like forward secrecy and public verifiability. But their proposed approach faces the key escrow problem. Duan et al. [9] presented a secure delegation-by-warrant IDBPYS scheme which is secure under the random oracle model (ROM). In this approach, efficiency and hardiness of security are based on bilinear pairing. It requires more communication bandwidth and creates high computation cost. Elkamshouchy et al. [10] improved the proxy signcryption techniques and proposed a new publicly verifiable proxy signcryption scheme based on the discrete logarithm problem (DLP). The authors claim that the given approach achieves the security properties of confidentiality and authenticity through an unsecured channel. Since, it depends upon DLP, which consumes more computing power. Furthermore, the proxy signcryption idea was furnished by Elkamshouchy et al. [11]. They attempted to improve the security of this scheme, but the scheme is affected by high computing power and extra bandwidth due to utilizing hard problems, i.e., integer factorization problem (IF), DLP, Diffie–Hellman problem (DHP), and DSA problem. So, IF, DHP, and DLP require more machine cycles and more computational power. Elkamchouchi and Abouslseoud [12] successfully enabled the partial delegation rights in their scheme by utilizing bilinear pairings on EC. However, in the given approach, the proxy signcrypter utilizes the signcrypting right incorrectly in light of the fact that in partial delegation, there is no limitation on proxy signcrypter. Lin et al. [13] designed a new provable secure proxy signcryption approach utilizing bilinear pairing. Unluckily, their proposed approach does not ensure the security requirement of warrant unforgeability. For further improving, Elkamchouchi et al. [14] proposed the notion of warrants-based proxy signcryption which is good for low resource devices. The security hardness and efficiency of this scheme are completely based on the elliptic curve cryptography that leads to more power consumption of the machine. Yanfeng et al. [15] presented a secure certificateless proxy identity-based signcryption scheme. They proposed elliptic curve discrete logarithm problem (ECDLP) for the efficiency and security in their scheme. But the scheme needs a secured channel for the partial private key distribution to the users. Elkamchouchi et al. [16] introduced two proxy signcryption schemes: one relies on DLP and other on ECDLP, respectively. They claim that this approach has less computational and communication costs. The scheme is still affected by more machine power consumption and extra communication bandwidth. Furthermore, the proposed scheme was not provable secured. Lo and Tsai [17] coined a provable secure proxy signcryption scheme depending on the bilinear pairings. They demonstrate better performance and secrecy in terms of in-distinguishability and unforgeability. Furthermore, they proved the security requirements of the given approach under the ROM. Then, for improving security services, Ming and Wang [18] proposed a provable secured proxy signcryption on the standard model. Because of heavy computations due to bilinear pairing, the proposed approach can still be affected by more machine control usage and extra communication of information transmission. Insafullah et al. [19] presented a lightweight proxy signcryption approach based on HECC. They claim that their newly designed scheme ensures all the security services with low computational and communication costs. Unfortunately, the scheme is affected by using more major operations over the hyperelliptic curve. Abdelfatah [20] coined a novel proxy signcryption approach and its EC variant. Hui and Lunzhia [21] coined a new proxy signcryption with EC. Waheed et al. [22] coined a new proxy signcryption with EC. Hundera et al. [23] coined a novel proxy signcryption approach with bilinear pairing for cloud data sharing. However, the designed approaches in [20–23] have been affected by more computational cost and extra communication bandwidth due to EC and bilinear pairing.

1.1. Motivations and Contributions. Keeping in view all the above proxy signcryption approaches, we identified that there is still a need for improvement in computational cost and bandwidth utilization. Though the abovementioned techniques are based on some prominent security techniques, i.e., RSA, bilinear pairing, and EC, HECC provides an equal level of security with 80 bits key size as compared to the elliptic curve with 160 bits key size and RSA and bilinear pairing with 1024 bits key size, respectively. Therefore, in order to decrease computational costs and channel bandwidth consumption, we design a cost-effective proxy signcryption scheme based on HECC that perform three roles of proxy delegator/original signcrypter, proxy signcrypter, and proxy unsigncrypter. The following are the main contributions of this study:

(i) We make a new proxy signcryption approach with the help of the hyperelliptic curve cryptosystem

(ii) We prove that the proposed approach is resilient against indistinguishable under adaptive chosen ciphertext attacks (IND-CCA) and unforgeable under adaptive chosen message attacks (UU-ACMA), when it is tested through the random oracle model (ROM).

(iii) Our approach reduces the computational cost and communication costs as compared to its counterpart schemes

1.2. Organization of the Study. The organization of the study is as follows. Section 2 defines the basic preliminaries and threat model. The proposed model and the algorithm are defined in Section 3. Section 4 contains the security analysis of the proposed approach. Furthermore, in Section 5, we describe the computation and communication overheads analysis. Section 6 discusses the communication overhead, and Section 7 presents the conclusion.

2. Preliminaries

This section includes some formal definitions of the hyperelliptic curve discrete logarithm problem and hyperelliptic curve Diffie–Hellman problem; furthermore, the explanation of the threat model is provided.

Definition 1. Suppose a devisor $\mathcal{D}$ of order n and an instance $\xi = \delta \cdot \mathcal{D}$ is given, so, to extract δ from ξ is said to be hyperelliptic curve discrete logarithm problem (HDL).

Definition 2. Suppose a devisor $\mathcal{D}$ of order n and an instance $\xi = \ell \cdot \delta$ is given, so, to extract δ and ℓ from ξ is said to be hyperelliptic curve Diffie–Hellman problem (HDDH).

2.1. Threat Model. Here, we are trying to explain the threats against our proposed scheme regarding the security requirements of indistinguishable under adaptive chosen ciphertext attacks (IND-CCA) and unforgeable under adaptive chosen message attacks (UU-ACMA) by adversary $\mathcal{A}$. The following Definitions 3 and 4 can be better explaining the threats against our newly proposed scheme.

Definition 3. The newly proposed scheme can be IND-CCA secure, if $\mathcal{A}$ with the help of challenger $\mathcal{A}$ cannot win with nonnegligible benefit in the following game.

Setup: $\mathcal{A}$ executes the setup part to make the global parameter param and sends it to $\mathcal{A}$.

2.1.1. Phase 1

Hash queries: $\mathcal{A}$ submits these queries and Φ can check the value for the ask queries if the value is found in the list; then, it gives the value to $\mathcal{A}$; otherwise, Φ selects a random value for each ask query and sends them to $\mathcal{A}$.

Private key generation query: $\mathcal{A}$ can submit queries for private key of signer and Φ executes the key generation algorithm to produce the required private key and dispatch it to $\mathcal{A}$.

Proxy delegation query: when this query is submitted by $\mathcal{A}$, Φ responds as valid delegation for ask query to $\mathcal{A}$.

Proxy signcryption query: when this query is submitted with message and private key of proxy and delegation by $\mathcal{A}$, Φ responds as valid proxy signcryption tuple for asking query to $\mathcal{A}$.

Proxy unsigncryption query: when this query is submitted with proxy signcryption tuple by $\mathcal{A}$, Φ responds as valid plaintext which is generated through proxy unsigncryption for asking query to $\mathcal{A}$.

Challenge: two equal lengths plaintext $\mathfrak{m}_a$ and $\mathfrak{m}_b$ will send by $\mathcal{A}$, and Φ uniformly chooses a bit $b \in \{0, 1\}$ and computes a ununderstandable text ψ^* on $\mathcal{M}b$.

2.1.2. Phase 2. In this phase, $\mathcal{A}$ should make same queries as phase 1 with the following constraints:

(i) $\mathcal{A}$ will not send a request for any user private key

(ii) $\mathcal{A}$ never asks for proxy unsigncryption for ciphertext ψ^*

(iii) At the end of this phase, $\mathcal{A}$ generates a bit b^* and succeeds this game if $b^* = b$.

Definition 4. The newly proposed scheme can be UU-ACMA secure, if $\mathcal{A}$ with the help of challenger Φ cannot win with nonnegligible benefit in the following game.

Setup: same as above IND-CCA game.

Query: same as above IND-CCA game.

Forgery: finally, $\mathcal{A}$ outputs a proxy signcryption tuple and succeeds in this game if the following events happen successful.

(i) The generated proxy signcryption text is valid

(ii) The private key of proxy signcrypter never been asked

(iii) The proxy signcryption text is not generated using proxy signcryption query

3. Proposed Model

We present here our cost-effective proxy signcryption scheme for low constraint environment. Our proposed scheme is comprised of four phases such as public key verification, proxy delegation, proxy signcryption, and proxy unsigncryption, respectively. The block diagram of our cost-effective proxy signcryption scheme is shown in Figure 1 and the symbols used in algorithm in Table 1. Four types of roles used in our scheme are public key verification, proxy delegator/original signcrypter, proxy signcrypter, and proxy unsigncrypter. First of all, each user verifies the requested user public key from certificate authority (CA). A proxy delegator first sends a warrant message with the signature to delegate the signcryption privileges to proxy signcrypter. Later, proxy signcrypter verifies the received message and computes the signcryption on behalf of the proxy delegator and then delivers it to the proxy unsigncrypter. After receiving a proxy signcryption tuple, proxy unsigncrypter verifies the authentication and performs the steps of unsigncryption.

3.1. Setup. In this section, the certificate authority (CA) pick HEC with 80 bits parameter size, make a system parameter set as $\ell = \{\text{HEC}, \mathcal{D}, h_0, h_1, h_2, h_3, \zeta\}$, where ζ is the public key CA and made as by selecting π at random, and then compute $\zeta = \pi \cdot \mathcal{D}$. Finally, CA makes sure the availability of ℓ in a network publicly.

3.2. Key Generation. In this subsection, the participants $(i = \mathcal{U}o, \mathcal{PS}, \mathcal{PU})$ first compute their public and private keys in the following way. The participants (i) randomly selects a number $a_i \in \{1, 2, \ldots, q-1\}$ and calculates $f_i = a_i \cdot D$. So, a_i and f_i represent the participants (i) private and public keys.

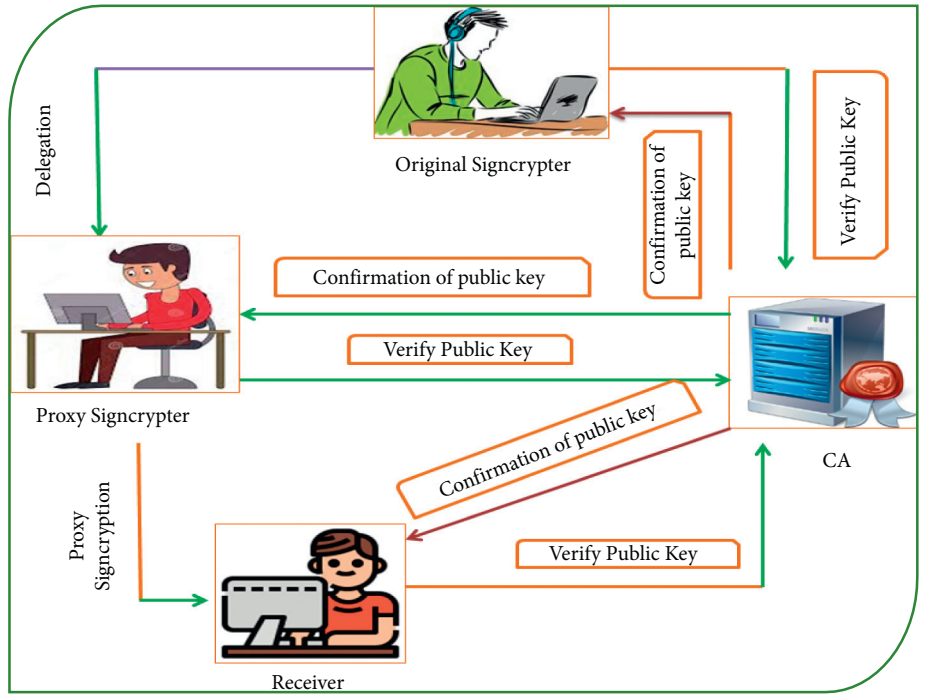

Figure 1: Framework model of the proposed proxy signcrypton scheme.

Table 1: Symbols used in the algorithm.

Notations of algorithm	Descriptions
$\mathcal{D}$	Divisors of the hyperelliptic curve
$\mathcal{U}o, \mathcal{PS}, \mathcal{PU}$	Represents the role of delegator, proxy signcrypter, and unsigncrypter
$a_{\mathcal{U}}$, $a_{\mathcal{P}}$, a_r	Private keys of delegator, proxy signcrypter, and unsigncrypter
$f_{\mathcal{U}}$, $f_{\mathcal{P}}$, f_r	Public keys of delegator, proxy signcrypter, and unsigncrypter
mw, m	Warrant message and message (plain text)
N_a, N_p	Nonce for delegator and proxy signcrypter
$E_{\mathcal{K}}$, $D_{\mathcal{K}}$	Encryption and decryption
h_0, h_1, h_2, h_3	Hash functions
$\mathcal{K}, \mathcal{K}_{sr}$	Preshared, computed shared key among proxy signcrypter and unsigncrypter
$\mathcal{X}_{\mathcal{K}}$, $\mathcal{Y}_{\mathcal{K}}$	Secret and public key for proxy signature generation and verifications

3.3. Proxy Delegation. In this subsection, the original signcrypter/proxy delegator $\mathcal{U}_o$ gives the right of the sign to proxy signcrypter $\mathcal{PS}$.

(i) The original signcrypter selects $\ell \in \{1, 2, \ldots, q-1\}$

(ii) Compute $\mathcal{W} = \mathcal{Z} \cdot \mathcal{D}$

(iii) Compute $\mathcal{T} = h_0(\text{mw}, \mathcal{W})$ and also compute $\mathcal{V} = (\mathcal{Z} - a_{\mathcal{U}} \cdot \mathcal{T}) \bmod q$

(iv) Sends $\delta = (\mathcal{V}, \mathcal{J}, \text{mw})$ to proxy signcrypter PS

After receiving $\delta = (\mathcal{V}, \mathcal{J}, \text{mw})$ for validation, $\mathcal{PS}$ performs the following equations:

$$\begin{aligned}\mathcal{W} &= \mathcal{V} \cdot \mathcal{D} + h1(\text{mw}, N_a, \mathcal{J}) \cdot f_{\mathcal{U}} \\ &= \mathcal{V} \cdot \mathcal{D} + h1(\text{mw}, N_a, \mathcal{J}).f_{\mathcal{U}} = (\mathcal{Z} - a_{\mathcal{U}}.h1(\text{mw}, N_a, \mathcal{J})).\mathcal{D} + h1(\text{mw}, N_a, \mathcal{J}) \cdot a_{\mathcal{U}} \cdot \mathcal{D} \\ &= \mathcal{D} \cdot (\mathcal{Z} - a_{\mathcal{U}}.h1(\text{mw}, N_a, \mathcal{J}) + a_{\mathcal{U}}.h1(\text{mw}, N_a, \mathcal{J})) \\ &= \mathcal{D} \cdot (\mathcal{Z}) = \mathcal{Z} \cdot \mathcal{D} = \mathcal{W}.\end{aligned} \tag{1}$$

After validation, the proxy signcrypter $\mathcal{PS}$ generates the secret key $\mathcal{X}_{\mathcal{K}} = (\mathcal{V} + a_{\mathcal{P}}) \bmod q$ and then calculates and publishes the public key $\mathcal{Y}_{\mathcal{K}} = \mathcal{X}_{\mathcal{K}} \cdot \mathcal{D}$.

3.4. Proxy Signcryption. In this subsection, proxy signcrypter $\mathcal{PS}$ performs the following steps to generate signcryption on the message (m).

(i) First choose a random number $j \in \{1, 2, \ldots, q-1\}$

(ii) Compute $\mu = j \cdot \mathcal{D}$, where $\mathcal{D}$ is the divisor over the hyper elliptic curve

(iii) Compute $\mathcal{K} = h_1(\mathcal{K}_{sr} + \mu)$, where $\mathcal{K}_{sr}$ is the shared secret key between proxy and recipient

(iv) Compute the ciphertext $\mathcal{C} = E_{\mathcal{K}}(m)$, where m is the plain text

(v) Compute the hash function $\Omega = h_2(\mathcal{C}, \mu)$

(vi) Compute the signature $\mathcal{S} = ((j/\mathcal{X}_{\mathcal{K}}) - \Omega) \bmod q$, where $\mathcal{X}_{\mathcal{K}}$ is the proxy signcrypter secret key

(vii) Then, send $\psi = (\mathcal{C}, \mathcal{S}, \Omega)$ to the proxy unsigncrypter

3.5. Proxy Verification and Unsigncryption. In this subsection, receiving the tuple $\psi = (\mathcal{C}, \mathcal{S}, \Omega)$ proxy unsigncrypter carry out the subsequent steps for verification and decryption of the proxy signcrypted text.

(i) First recover $\mu = \mathcal{Y}_{\mathcal{K}} \cdot (\mathcal{S} + \Omega)$ and $\mu = \mathcal{X}_{\mathcal{K}} \cdot \mathcal{D} \cdot ((j/\mathcal{X}_{\mathcal{K}}) - \Omega + \Omega) = j \cdot \mathcal{D}$

(ii) After this, verify the signature $\Omega^* = h_2(\mathcal{C}, \omega)$ and accept if $\Omega = \Omega^*$

(iii) Compute $\mathcal{K} = h_1(\mathcal{K}_{sr} + \mu)$ and decrypt $(m) = D_{\mathcal{K}}(\mathcal{C})$

4. Security Analysis

Our scheme meets the security requirements of indistinguishable under adaptive chosen ciphertext attacks (IND-CCA) and unforgeable under adaptive chosen message attacks (UU-ACMA) by adversary $\mathcal{A}$. The following Theorems 1 and 2 can be better explaining the threats against our newly proposed scheme.

Theorem 1. *The newly proposed scheme can be IND-CCA secure, if $\mathcal{A}$ with the help of challenger Φ cannot win with nonnegligible benefit in the following steps.*

Proof. The instance of the hyperelliptic curve $(\mathcal{Q}, \mathcal{D}, \mathcal{V} \cdot \mathcal{D})$ is given to Φ and the task of Φ to compute $\mathcal{Q} = \mathcal{O} \cdot \mathcal{V} \cdot \mathcal{D}$.

Setup: Φ executes the setup part for to make the global parameter param and sends it to $\mathcal{A}$. □

4.1. Phase 1

Hash (h_0) queries: if $\mathcal{A}$ submits $(mw, \mathcal{J})$ query and Φ can check the value for a query if the value found in the list is (LH0), then it gives the values $(\mathcal{T}_i)$ to $\mathcal{A}$; otherwise, Φ selects $\mathcal{T}_i$ randomly and send them to $\mathcal{A}$.

Hash (h_1) queries: if $\mathcal{A}$ submits a query and Φ can check the value for a query if the value found in the list is (LH1), then it gives the values $(\mathcal{K}_i)$ to $\mathcal{A}$; otherwise, Φ selects $\mathcal{K}_i$ randomly and send them to $\mathcal{A}$.

Hash (h_2) queries: if $\mathcal{A}$ submits a query and Φ can check the value for a query if the value found in the list is (LH2), then it gives the values (Ω_i) to $\mathcal{A}$; otherwise, Φ selects Ω_i randomly and send them to $\mathcal{A}$.

Private key generation query: if $\mathcal{A}$ submits query for private key and public key of signer and Φ randomly select $a_i \in \{1, 2, \ldots, q-1\}$, calculate $f_i = a_i \cdot D$, and dispatch (a_i, f_i) to $\mathcal{A}$.

Proxy delegation query: when this query is submitted by $\mathcal{A}$, Φ responds as valid delegation δ to $\mathcal{A}$ in the following way.

(i) Φ randomly selects ℓ and $\mathcal{T}$ form $\{1, 2, \ldots, q-1\}$ and compute $\mathcal{W} = \mathcal{Z} \cdot \mathcal{D}$

(ii) Compute $\mathcal{V} = (\mathcal{Z} - a_{\mathcal{U}} \cdot \mathcal{T})$, set $\delta = (\mathcal{V}, \mathcal{J}, mw)$, and respond δ to $\mathcal{A}$ as a delegation

Proxy signcryption query: when this query is submitted with message (m) and private key of proxy (a_p) and delegation (δ) by $\mathcal{A}$, Φ responds as valid proxy signcryption ψ to $\mathcal{A}$ in the following way.

(i) Φ chooses random numbers $j, \mathcal{K}, \Omega, \mathcal{X}_{\mathcal{K}} \in \{1, 2, \ldots, q-1\}$

(ii) Computes the ciphertext $\mathcal{C} = E_{\mathcal{K}}(m)$

(iii) Computes the signature $\mathcal{S} = ((j/\mathcal{X}_{\mathcal{K}}) - \Omega)$

(iv) Set $\psi = (\mathcal{C}, \mathcal{S}, \Omega)$ and respond ψ to $\mathcal{A}$ as a proxy signcryption

Proxy unsigncryption query: when this query is submitted to ψ by $\mathcal{A}$, if this query is not for target participant, Φ responds as valid plaintext which is generated through proxy unsigncryption to $\mathcal{A}$. Otherwise, Φ outputs ψ as an invalid proxy signcryption tuple.

Challenge: two equal lengths plaintext $\mathfrak{m}_a$ and $\mathfrak{m}_b$ will send by $\mathcal{A}$, Φ uniformly chooses a bit $b \in \{0, 1\}$, and computes an un-understandable text ψ^* on $\mathcal{M}b$ as follows.

(i) Φ choose random numbers $\mathcal{X}_{\mathcal{K}}, \mu, \mathcal{K}_{sr} \in \{1, 2, \ldots, q-1\}$

(ii) Compute $\mathcal{K} = (\mu + \mathcal{K}_{sr})$, $\mathcal{C}^* = E_{\mathcal{K}}(m)$, and $\Omega^* = h_3(\mathcal{C}^*, \mu)$

(iii) Compute the signature $\mathcal{S}^* = ((j/\mathcal{X}_{\mathcal{K}}) - \Omega)$, set $\psi^* = (\mathcal{C}^*, \mathcal{S}^*, \Omega^*)$, and respond ψ^* to $\mathcal{A}$ as a proxy signcryption on $\mathcal{M}b$ to $\mathcal{A}$.

4.2. Phase 2. Just like phase 1, $\mathcal{A}$ can submit the identical queries, but it does not make a query for receiver private key and a massage corresponding to the ψ^*.

After that, $\mathcal{A}$ results $b^* \in \{0, 1\}$, and if $b^* = b$, then Φ results 1. Otherwise, Φ results 0. If $\mathcal{Q} = \mathcal{O} \cdot \mathcal{V} \cdot \mathcal{D}$, ψ^* is valid signcrypted text, and for this reason can extricate b by utilized advantage π. Accordingly, $\Pr[\Phi \longrightarrow 1 | \mathcal{Q} = \mathcal{O} \cdot \mathcal{V} \cdot \mathcal{D}] = \Pr[b/= b | \mathcal{Q} = \mathcal{O} \cdot \mathcal{V} \cdot \mathcal{D}] = 1/2 + \pi$.

If $\mathcal{Q} \neq \mathcal{O} \cdot \mathcal{V} \cdot \mathcal{D}$, $\mathcal{A}$ cannot extricate b without advantages. Accordingly, $\Pr[\Phi \longrightarrow 1 | \mathcal{Q} \neq \mathcal{O} \cdot \mathcal{V} \cdot \mathcal{D}] = \Pr[b^* = b | \mathcal{Q} \neq \mathcal{O} \cdot \mathcal{V} \cdot \mathcal{D}] = 1/2$.

Probability analysis: suppose the queries (h_0, h_1, h_2), qpk, $qp\ d$, and $qpsn$ represent hash queries, private key

queries, proxy delegation queries, and proxy signcryption queries, separately.

Thus, we signify some measures ($\mathcal{MER}$) as follows:

(i) $\mathcal{MER}1$: Φ output is positive in private key queries, and the probability is $1 - qpk/2^k$.
(ii) $\mathcal{MER}2$: Φ output is positive in proxy unsigncryption queries, and the probability is $1 - 1/2^k$.
(iii) $\mathcal{MER}3$: Φ output is positive in challenge part, and the probability is $1/qpk - 2^k$.

So, the total probability will be as follows:

$$\begin{aligned}\Pr[\Phi \longrightarrow \pi] &= \Pr[\mathcal{MER}1 \wedge \mathcal{MER}2 \wedge \mathcal{MER}3] \\ &= \Pr\left[1 - \frac{qpk}{2^k} \wedge 1 - \frac{1}{2^k} \wedge \frac{1}{qpk} - 2^k\right] \\ &= \left(1 - \frac{qpk}{2^k}\right)\left(1 - \frac{1}{2^k}\right)\left(\frac{1}{qpk} - 2^k\right) \cdot \pi.\end{aligned} \tag{2}$$

Theorem 2. *The newly proposed scheme can be UU-ACMA secure, if $\mathcal{A}$ with the help of challenger Φ cannot win with nonnegligible benefit in the following steps.*

Proof. The instance of the hyperelliptic curve $(\mathcal{Q}, \mathcal{V} \cdot \mathcal{D})$ is given to Φ and the task of Φ to compute $\mathcal{Q} = \mathcal{V} \cdot \mathcal{D} = \mathcal{Y}_{\mathcal{K}}$.

Setup: Φ execute the setup part for to make the global parameter param ℓ and sends it to $\mathcal{A}$.

Phase 1

Queries: same like Theorem 1.

Forgery: according to forking lemma [24], Φ can get two valid proxy signcryption text that are $\psi = (\mathcal{C}, \mathcal{S}, \Omega)$ and $\psi^* = (\mathcal{C}, \mathcal{S}, \Omega^*)$. Then, for the verification, we get two equations that are $\mu = \mathcal{Y}_{\mathcal{K}} \cdot (\mathcal{S} + \Omega)$ and $\mu^* = \mathcal{Y}_{\mathcal{K}} \cdot (\mathcal{S} + \Omega^*)$. So, after subtraction, we can get the following results.

$$\begin{aligned}\mu - \mu^* &= \mathcal{Y}_{\mathcal{K}} \cdot (\mathcal{S} + \Omega) - \left(\mathcal{Y}_{\mathcal{K}} \cdot \left(\mathcal{S} + \Omega^*\right)\right) \\ &= \mathcal{Y}_{\mathcal{K}} \cdot \mathcal{S} + \mathcal{Y}_{\mathcal{K}} \cdot \Omega - \mathcal{Y}_{\mathcal{K}} \cdot \mathcal{S} + \mathcal{Y}_{\mathcal{K}} \cdot \Omega^* \\ &= \mathcal{Y}_{\mathcal{K}} \cdot \Omega - \mathcal{Y}_{\mathcal{K}} \cdot \Omega^* = \mu - \mu^* = \mathrm{j} \cdot \mathcal{D} - \mathrm{j}^* \cdot \mathcal{D} \\ &= \mathcal{V} \cdot \mathcal{D} \cdot \Omega - \mathcal{V} \cdot \mathcal{D} \cdot \Omega^* \\ &= \left(\mathrm{j} - \mathrm{j}^*\right) \cdot \mathcal{D} = \mathcal{V} \cdot \mathcal{D} \cdot \left(\Omega - \Omega^*\right) = \left(\mathrm{j} - \mathrm{j}^*\right) \\ &= \mathcal{V}\left(\left(\Omega - \Omega^*\right).\right.\end{aligned} \tag{3}$$

$\mathcal{V} = (\mathrm{j} - \mathrm{j}^*)/(\Omega - \Omega^*)$; hence, this is the solution for solving the hyperelliptic curve discrete logarithm problem.

Probability analysis: suppose the queries $(\mathrm{h}_0, \mathrm{h}_1, \mathrm{h}_2)$, $qpk, qp\ d$, and $qpsn$ represent the hash queries, private key queries, proxy delegation queries, and proxy signcryption queries, separately.

Thus, we signify some measures ($\mathcal{MER}$) as follows:

(i) $\mathcal{MER}1$: Φ output is positive in private key queries, and the probability is $1 - qpk/2^k$.
(ii) $\mathcal{MER}2$: Φ output is positive in proxy unsigncryption queries, and the probability is $1 - 1/2^k$.
(iii) $\mathcal{MER}3$: Φ output is positive in challenge part, and the probability is $1/qpk - 2^k$.

So, the total probability will be as follows:

$$\begin{aligned}\Pr[\Phi \longrightarrow \pi] &= \Pr[\mathcal{MER}1 \wedge \mathcal{MER}2 \wedge \mathcal{MER}3] \\ &= \Pr\left[1 - \frac{qpk}{2^k} \wedge 1 - \frac{1}{2^k} \wedge \frac{1}{qpk} - 2^k\right] \\ &= \left(1 - \frac{qpk}{2^k}\right)\left(1 - \frac{1}{2^k}\right)\left(\frac{1}{qpk} - 2^k\right) \cdot \pi.\end{aligned} \tag{4}$$

□

5. Computational Cost

The comparisons of the proposed and existing proxy signcryption schemes in terms of major operations are offered in table. In Table 2, computational cost in ms is provided. The symbols $\mathcal{EML}$, $\mathcal{P}r$, and $\mathcal{HEML}$ represent the exponential computations, elliptic curve multiplications, pairing operations, and hyperelliptic curve devisor multiplication, respectively. The other operations such as addition, subtraction, hash, and division are ignored because they require fewer computations time.

To show more clearly the comparisons between the proposed scheme and existing schemes, it has been observed from [25], by using the Multiprecision Integer and Rational Arithmetic C Library (MIRACL) and test the run time of the basic cryptographic operations. The running time for basic cryptographic operations is given in Table 3 (tested it 100 of times), an experiment donned through:

(i) Raspberry PI 3 B + Rev 1.3
(ii) OS: Ubuntu 20.04 LTS, 64-bit
(iii) with CPU: 64-bit, processor: 1.4 GHz Quad-Core
(iv) With 1 GB of memory

Also, we assume the half-time elliptic curve for the hyperelliptic curve because it is the generalized form of elliptic curve [26–30]. So, Table 3 provides details about the average time. Table 4 and Figure 2 show that our proposed scheme is computationally efficient from existing schemes. In Table 2, we provide the computational cost comparisons in milliseconds.

6. Communication Overhead

To design a cryptographic protocol for wireless communication, media is an important element because wireless protocols need lower communication overhead. Selecting a larger size of parameters can greatly affect the efficiency. In this section, we compare our newly designed scheme with

Table 2: Computational cost comparisons.

Schemes	Proxy delegation	Proxy signcryption	Proxy verification and unsigncryption	Total
Insafullah et al. [19]	3.42	1.14	3.42	7.98
Abdelfatah [20]	6.84	2.28	2.28	11.40
Guo and Deng [21]	9.12	11.40	11.40	31.92
Waheed et al. [22]	2.28	6.84	11.40	20.52
Hundera et al. [23]	64.16	0	128.32	192.48
Proposed	3.42	1.14	1.14	5.70

Table 3: Running time in milliseconds.

Primitive	Average time (in milliseconds)
$\mathcal{EML}$	2.28
$\mathcal{P}r$	32.08
$\mathcal{HEML}$	1.14

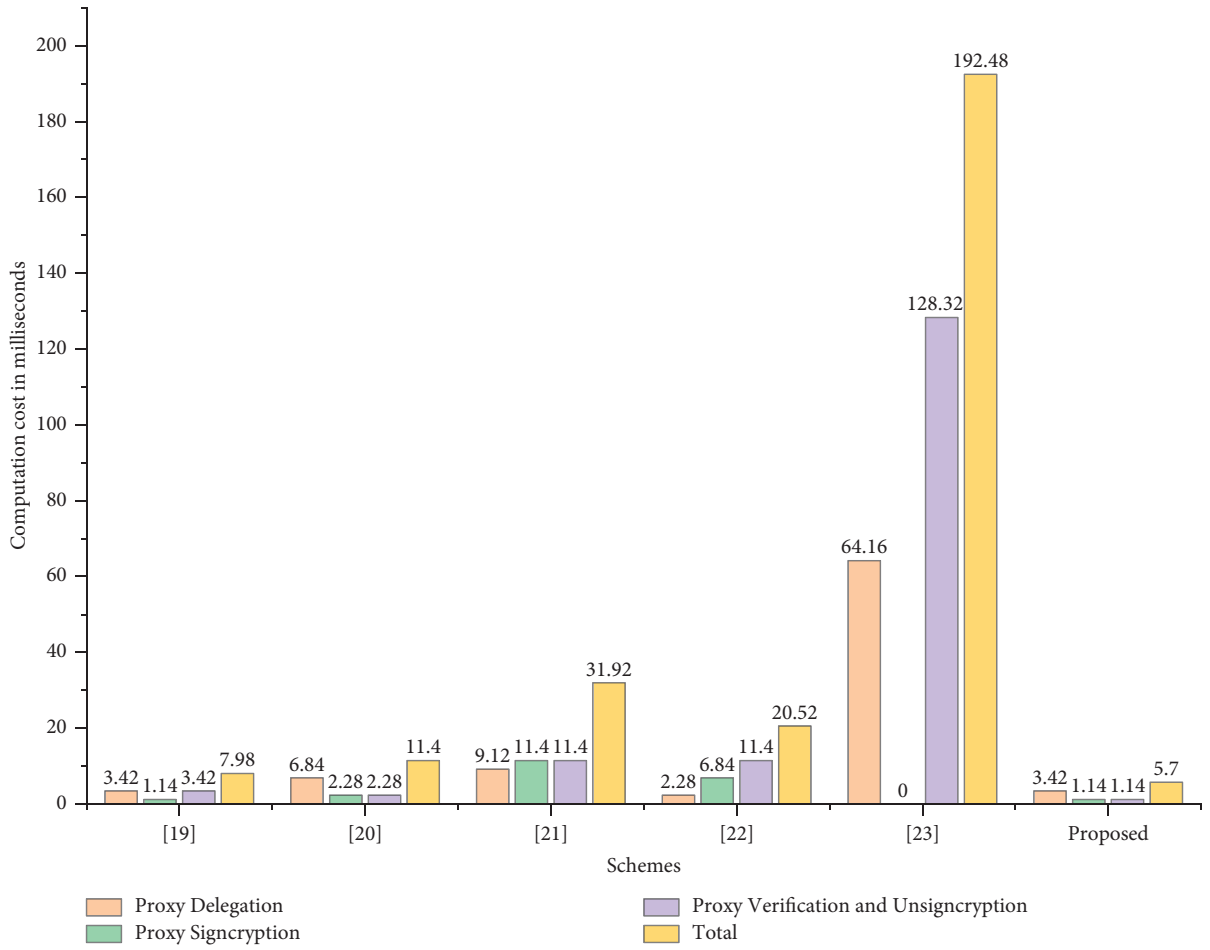

Figure 2: Computational cost comparisons.

Table 4: Major operation comparisons.

Schemes	Proxy delegation	Proxy signcryption	Proxy verification and unsigncryption	Total
Insafullah et al. [19]	$3\mathcal{HEML}$	$1\mathcal{HEML}$	$3\mathcal{HEML}$	$7\mathcal{HEML}$
Abdelfatah [20]	$3\mathcal{EML}$	$1\mathcal{EML}$	$1\mathcal{EML}$	$5\mathcal{EML}$
Guo and Deng [21]	$4\mathcal{EML}$	$5\mathcal{EML}$	$5\mathcal{EML}$	$14\mathcal{EML}$
Waheed et al. [22]	$1\mathcal{EML}$	$3\mathcal{EML}$	$5\mathcal{EML}$	$9\mathcal{EML}$
Hundera et al. [23]	$2\mathcal{P}r$	_	$4\mathcal{P}r$	$6\mathcal{P}r$
Proposed	$3\mathcal{HEML}$	$1\mathcal{HEML}$	$1\mathcal{HEML}$	$5\mathcal{HEML}$

Table 5: Communication overhead comparisons in terms of extra parameters.

Schemes	Proxy delegation	Proxy signcryption	Total
Insafullah et al. [19]	$2\|q\| + \|mw\|$	$3\|q\| + \|mw\| + \|C\| + \|H\|$	$5\|q\| + \|mw\| + \|C\| + \|H\|$
Abdelfatah [20]	$2\|p\| + \|mw\|$	$1\|p\| + \|C\| + \|H\|$	$3\|p\| + \|mw\| + \|C\| + \|H\|$
Guo and Deng [21]	$2\|p\| + \|mw\|$	$5\|p\| + \|C\| + \|mw\|$	$7\|p\| + \|C\| + 2\|mw\|$
Waheed et al. [22]	$\|mw\| + \|H\|$	$3\|p\| + \|C\| + \|H\|$	$3\|p\| + \|mw\| + \|C\| + \|H\|$
Hundera et al. [23]	$2\|\mathcal{G}\| + \|mw\|$	$1\|\mathcal{G}\| + \|C\| + \|H\| + \|mw\|$	$3\|\mathcal{G}\| + \|C\| + \|H\| + 2\|mw\|$
Proposed	$2\|q\| + \|mw\|$	$1\|q\| + \|C\| + \|H\|$	$3\|q\| + \|mw\| + \|C\| + \|H\|$

Table 6: Communication overhead comparisons for 1 kb ciphertext and warrant size.

Schemes	Proxy delegation	Proxy signcryption	Total
Insafullah et al. [19]	1184	2800	3984
Abdelfatah [20]	1344	1696	3040
Guo and Deng [21]	1344	2848	4192
Waheed et al. [22]	1536	2016	3552
Hundera et al. [23]	2048	3072	5120
Proposed	1184	1616	2800

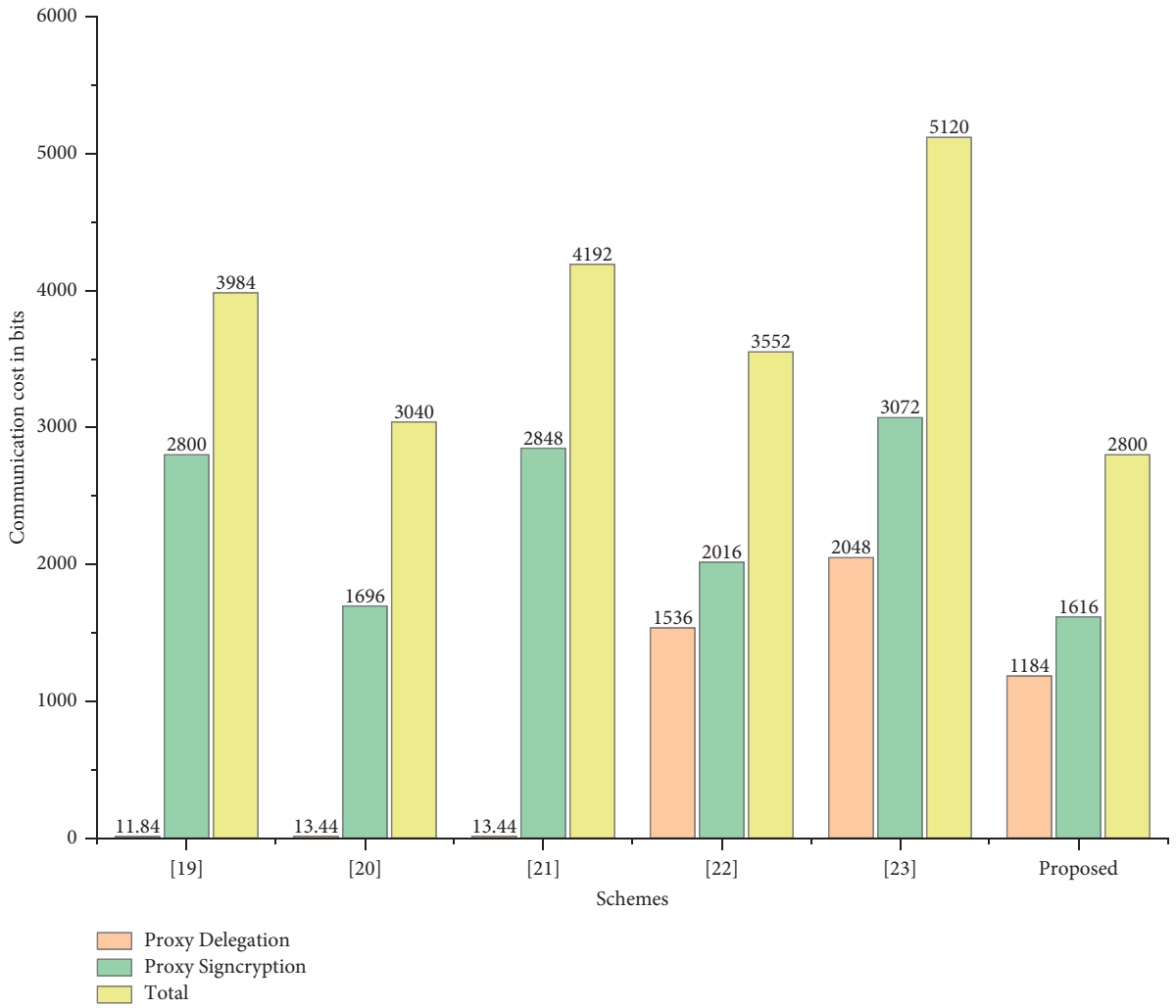

Figure 3: Communication cost comparisons.

previous schemes in terms of communication overhead. For generalization, we suppose that

(i) $|p|$ is a prime number $\geq 2^{160}$

(ii) $|q|$ is a prime number $\geq 2^{80}$

(iii) $|\mathcal{G}|$ where $\mathcal{G}$ be a group $\geq 2^{512}$

(iv) $|H|$ is a hash with 512 bits

Table 5 represents the communication cost of the designed and previous schemes; furthermore, Table 6 and Figure 3 show that when we consider 1 kb message or warrant, then our scheme is best from existing schemes.

7. Conclusion

In this article, we proposed a cost-effective proxy signcryption scheme for IoT devices. The proposed approach ensures the security properties such as unforgeability and confidentiality when it is tested through the ROM. The proposed scheme is lightweight due to the usage of HECC, which provides the same level of security with a lower-key size. A detailed security as well as performance analysis is conducted with the relevant existing schemes. The results demonstrate that the proposed scheme improves the overall computational cost and communication overhead, these being 5.7 ms and 2800 bits, respectively, which authenticates the superiority of our scheme from the existing schemes. Finally, we concluded that the proposed scheme could be of prime importance for the Internet of things devices.

In the future, we are intended to implement the same scheme on multimessage multireceiver environment using genus 3 of HECC.

Data Availability

The data generated or analyzed during this study are included within this article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signature: delegation of the power to sign messages," *IEICE Transactions on Fundamentals*, vol. E79-A, no. 9, pp. 1338–1353, 1996.

[2] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures for delegating signing operation," in *Proceedings of the 3rd ACM Conference on Computer and Communications Security*, pp. 48–57, New Delhi, India, March 1996.

[3] S. Kim, S. Park, D. Won, S. Park, and D. Won, "Proxy signatures, revisited," *Information and Communications Security*, vol. 1334, pp. 223–232, 1997.

[4] K. Zhang, "Threshold proxy signature schemes," in *Proceedings of the ISW'97, Information Security Workshop*, pp. 191–197, Ishikawa Japan, September 1997.

[5] C. Gamage, J. Leiwo, and Y. Zheng, "An efficient scheme for secure message transmission using proxy signcryption," Technical report 9801, Monash University, Melbourne, Australia, 1998.

[6] Z. A. Zhang, "New publicly verifiable proxy signcryption scheme," *Progress on Cryptography*, 2004.

[7] X. Li and K. Chen, "Identity based proxy-signcryption scheme from pairings," in *Proceedings of the 2004 IEEE International Conference on Services Computing (SCC'04)*, pp. 494–497, Washington, DC, USA, September 2004.

[8] M. Wang, H. Li, and Z. Liu, "Efficient identity based proxy-signcryption schemes with forward security and public verifiability," in *Proceedings of the 3rd International Conference on Networking and Mobile Computing (ICCNMC)*, pp. 982–991, Zhangjiajie, China, August 2005.

[9] S. Duan, Z. Cao, and Y. Zhou, "Secure delegation-bywarrant ID-based proxy signcryption scheme," in *Proceedings of the Computational Intelligence and Security Conference (CIS '05)*, pp. 445–450, Springer, Xian, China, December 2005.

[10] D. H. Elkamshoushy, A. K. AbouAlsou, and M. Madkour, "New proxy signcryption scheme with DSA verifier," in *Proceedings of the Twenty Third National Radio Science Conference (NRSC'2006)*, pp. 1–8, Al Minufiyah, Egypt, March 2006.

[11] H. Elkamshouchy, M. Nasr, and R. Ismail, "A new efficient strong proxy signcryption scheme based on a combination of hard problems," in *Proceedings of the International Conference on Systems, Man, and Cybernetics San Antonio (ICSMC'09)*, pp. 5123–5127, San Antonia, TX, USA, October 2009.

[12] H. Elkamchouchi and Y. A. Abouslseoud, "New proxy identity-based signcryption scheme for partial delegation of signing rights," *IACR Cryptology Eprint Archive*, vol. 41, 2008.

[13] H.-Y. Lin, T.-S. Wu, S.-K. Huang, and Y.-S. Yeh, "Efficient proxy signcryption scheme with provable CCA and CMA security," *Computers & Mathematics with Applications*, vol. 60, no. 7, pp. 1850–1858, 2010.

[14] H. M. Elkamchouchi, Y. Abouelseoud, and W. S. Shouaib, "A new proxy signcryption scheme using warrants," *International Journal of Intelligent Engineering Informatics*, vol. 1, no. 3/4, pp. 309–327, 2011.

[15] Q. Yanfeng, T. Chunming, L. Yu, X. Maozhi, and G. Baoan, "Certificateless proxy identity-based signcryption scheme without bilinear pairings," *China Communications*, vol. 10, no. 11, pp. 37–41, 2013.

[16] H. Elkamchouchi, E. Abu Elkhair, and Y. Abouelseoud, "An efficient proxy signcryption scheme based on the discrete logarithm problem," *International Journal of Information Technology, Modeling and Computing*, vol. 1, no. 2, pp. 7–19, 2013.

[17] N.-W. Lo and J.-L. Tsai, "A provably secure proxy signcryption scheme using bilinear pairings," *Journal of Applied Mathematics*, vol. 2014, Article ID 454393, 10 pages, 2014.

[18] Y. Ming and Y. Wang, "Proxy signcryption scheme in the standard model," *Security and Communication Networks*, vol. 8, no. 8, pp. 1431–1446, 2015.

[19] A. Insafullah, I. Haq, A. Amin, A. I. Umar, and H. Khattak, "Proxy signcrypion scheme based on hyper elliptic curves," *International Journal of Computer*, vol. 20, no. 1, pp. 157–166, 2016.

[20] R. I. A. Abdelfatah, "Novel proxy signcryption scheme and its elliptic curve variant," *International Journal of Computer Applications*, vol. 165, no. 2, pp. 36–43, 2017.

[21] H. Guo and L. Deng, "An identity based proxy signcryption scheme without pairings," *International Journal of Network Security*, vol. 22, no. 4, pp. 561–568, 2020.

[22] A. Waheed, A. I. Umar, M. Zareei et al., "Cryptanalysis and improvement of a proxy signcryption scheme in the standard computational model," *IEEE Access*, vol. 8, pp. 131188–131201, 2020.

[23] N. W. Hundera, Q. Mei, H. Xiong, and D. M. Geressu, "A secure and efficient identity-based proxy signcryption in cloud data sharing," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 1, pp. 455–472, 2020.

[24] M. Bellare and G. Neven, "Multi-signatures in the plain public key model and a general forking lemma," in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS'06)*, pp. 390–399, October 2006.

[25] B. Bera, S. Saha, A. K. Das, N. Kumar, P. Lorenz, and M. Alazab, "Blockchain-envisioned secure data delivery and collection scheme for 5G-based IoT-enabled internet of drones environment," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 9097–9111, 2020.

[26] M. A. Khan, I. Ullah, S. Nisar et al., "Multiaccess edge computing empowered flying ad hoc networks with secure deployment using identity-based generalized signcryption," *Mobile Information Systems*, vol. 2020, Article ID 8861947, 15 pages, 2020.

[27] M. A. Khan, "Efficient and provably secure certificateless key-encapsulated signcryption scheme for flying ad-hoc network," *IEEE Access*, vol. 8, pp. 36807–36828, 2020.

[28] M. A. Khan, "An efficient and secure certificate-based access control and key agreement scheme for flying ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 99, 2021.

[29] M. A. Khan, I. Ullah, S. Nisar et al., "Multiaccess edge computing empowered flying ad hoc networks with secure deployment using identity-based generalized signcryption," *Mobile Information Systems*, vol. 2020, Article ID 8861947, 15 pages, 2020.

[30] M. A. Khan, "Securing internet of drones with identity-based proxy signcryption," *IEEE Access*, vol. 9, pp. 89133–89142, 2021.