Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8857272, 20 pages
https://doi.org/10.1155/2020/8857272

Research Article

A Lightweight Nature Heterogeneous Generalized Signcryption (HGSC) Scheme for Named Data Networking-Enabled Internet of Things

Manazara Rehman,[1] Hizbullah Khattak,[1] Ahmed Saeed Alzahrani,[2] Insaf Ullah,[3] Muhammad Adnan,[4] Syed Sajid Ullah,[1] Noor Ul Amin,[1] Saddam Hussain,[1] and Shah Jahan Khattak[5]

[1]*IT Department, Hazara University, Mansehra, 21120 KP, Pakistan*
[2]*Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia*
[3]*Department of Computing, HIET, Hamdard University, Islamabad Campus, Islamabad 44000, Pakistan*
[4]*Division of Computer and Information Sciences, Higher Colleges of Technology, 17155, Al Ain, UAE*
[5]*Pakistan Engineering Council (PEC), Attatruk Avenue (East) Sector G-5/2, Islamabad 44000, Pakistan*

Correspondence should be addressed to Insaf Ullah; insafktk@gmail.com

Received 27 April 2020; Revised 3 July 2020; Accepted 30 July 2020; Published 18 November 2020

Academic Editor: Fawad Zaman

Internet of Things (IoT) is the collection of different types of smart objects like mobile phones, sensors, cars, smart cities, smart buildings, and healthcare, which can provide a quality life to humans around the globe. These smart objects sense and produce a huge amount of data for distribution. The current hostcentric networking paradigm is not that scalable to provide a suitable solution to the idea of IoT. For scalable connectivity and efficient distribution, Named Data Networking (NDN) has been envisioned as a promising solution for future internet architecture. On the other hand, the significant issues regarding the adaptation of NDN with IoT possess security concerns such as authentication, confidentiality, integrity, and forward secrecy. As IoT is a heterogeneous environment, it demands a different type of security, according to the environmental situation such as public key infrastructure (PKI), identity-based cryptosystem (IBC), and certificateless cryptosystem (CLC). This paper presents a new concept of CLC to IBC heterogeneous generalized signcryption for the first time to fulfil the prime security requirements of NDN-based IoT. The proposed scheme provides the security properties according to situational needs without disturbing the structural policy of NDN. Considering the resource-constrained nature of IoT, we used a lightweight type of elliptic curve called the hyperelliptic curve cryptosystem which offers the same level of security as that of bilinear pairing and an elliptic curve cryptosystem using a minimum key size. Further, we compare the proposed scheme with recently proposed identity-based as well as certificateless generalized signcryption schemes, and the results give satisfactory outputs in terms of computational and communication resources. Furthermore, we simulate the proposed scheme with Automated Validation of Internet Security Protocols and Applications (AVISPA), and the results show that our scheme is valid and safe. Additionally, we provide a practical scenario of the proposed on NDN with an IoT-based smart city.

1. Introduction

Nowadays, Internet of Things (IoT) is deployed in almost every environmental domain, such as smart grid, smart homes, smart cities, smart building, healthcare, and smart agriculture, by connecting and controlling a large number of objects [1]. The increasing numbers of smart applications and their heterogeneity raise some challenges regarding their

connectivity, communication, scalability, mobility, and amount of generating data [2]. To address these challenges, Named Data Networking (NDN) has been projected as a future internet architecture [3]. In general, NDN deals with two packets: the interest packet and the data packet. The communication of the NDN is based on the altercation on interest packets that carry the request. Further, the NDN node maintains three types of data structures that are the Content Store (CS), which stores the copy of the contents with itself in the CS for future use; Pending Interest Table (PIT), which enlists all the requests of the incoming interfaces in the PIT table; and Forwarding Information Base (FIB), which forwards the requests from one node to another based on routing protocols [4]. Security is instigated on each packet, so the authenticity can be achieved at a time inside the network [5]. Whenever a consumer sends an interest packet for some specific contents, the NDN router performs CS lookup; if the requested contents are available, then the router simply forwards the contents directly from its CS to the requested consumer [6]. If the requested contents are not available in the CS, then the NDN router checks its PIT table for that requested content; if the contents have been requested before, then the PIT table updates with an entry of that specific interface in the PIT table. If the contents are being requested for the first time, then the PIT table marks up an entry of that interface and forwards the request to the next router based on FIB as shown in Figure 1. The monumental features of NDN like in-network caching, scalability, name-based routing, and mobility are a suitable option for fulfilling the demands of IoT applications.

However, security is considered to be the fundamental need for NDN-based IoT devices. Additionally, the NDN-based IoT environment requires different types of security properties such as authentication, confidentiality, and integrity, which can be achieved from a digital signature, encryption, or signcryption according to the environmental situation. Moreover, IoT is a heterogeneous environment where the sender and receivers may come from different types of environments. Here, the concept of heterogeneous signcryption is a suitable option that makes use of two different types of cryptosystems in a single algorithm [7]. On the other hand, the IoT devices may demand a digital signature, encryption, or signcryption, separately or in combination. For this type of situation, the heterogeneous signcryption becomes effortless due to its nongeneralized nature such as providing signcryption only. Here, the concept of generalized signcryption may be able to provide a digital signature, encryption, or signcryption using a single algorithm [8]. Likewise, the generalized signcryption cannot fulfill the requirement of IoT devices due to its homogeneity.

Generally, the security and efficiency of the aforementioned schemes are based on computationally hard problems like RSA, bilinear pairing, and elliptic curve cryptosystem. The RSA provides a solution using a 1024-bit large key which is firmly based on large factorization [9–11]. However, due to the limited processing capabilities of IoT devices, the 1024-bit key is not an efficient solution. On the other hand, bilinear pairing suffers from the issue of high pairing operations and is 12.93 times worse than RSA [12]. Hence, to tackle the weaknesses of both RSA and bilinear pairing, a new type of cryptosystem was introduced [13] called the elliptic curve cryptosystem. Unlike RSA and bilinear pairing, the security difficulty of the elliptic curve cryptosystem is based on a small key size of 160 bits. However, the 160-bit key is still not appropriate for resource-limited IoT devices [14]. Hence, in [15], a new type of cryptosystem was introduced, called the hyperelliptic curve cryptosystem, which suits the resource-limited nature of IoT devices by using a small key of 80 bits [16, 17].

The above discussion motivates us to contribute a new concept of heterogeneous generalized signcryption for NDN-based IoT which will combine the idea of heterogeneous signcryption with generalized signcryption to fulfill the conditional demands of IoT. The features of this new concept are mentioned as follows:

(1) First, we introduced a new concept of CLC to IBC heterogeneous generalized signcryption

(2) We provide the proper syntax of our proposed scheme

(3) We also provide a proper algorithm for the proposed scheme on the basis of the hyperelliptic curve cryptosystem which is suited for the IoT environment

(4) We prove the security properties such as authentication, confidentiality, unforgeability, forward secrecy, and integrity of the proposed scheme

(5) We also compared our proposed scheme with recently published CLC and IBC generalized signcryption schemes, and the results give satisfactory outputs in terms of computational and communication resources

(6) We also validate the security of our scheme through AVISPA, and the results show that our proposed scheme is valid and safe

(7) We practically deployed our scheme on the NDN-based smart city

1.1. Paper Organization. The organization of the paper is shown in Figure 2.

2. Related Work

Here, we divided the related work into three parts such as identity-based generalized signcryption, certificateless generalized signcryption, and heterogeneous generalized signcryption.

2.1. Identity-Based Generalized Signcryption Schemes. Lal and Kushwah in 2008 [18] introduced the concept of an identity-based generalized signcryption (ID-BGS) scheme for the first time to solve the certificate management issues of PKI-based generalized signcryption. In 2010, the concept was used by Liang et al. [19] for key management issues in mobile ad hoc networks (MANET). The proposed scheme saves memory storage of users and minimizes computational and communication resources. Kushwah and Lal in 2011 [20]

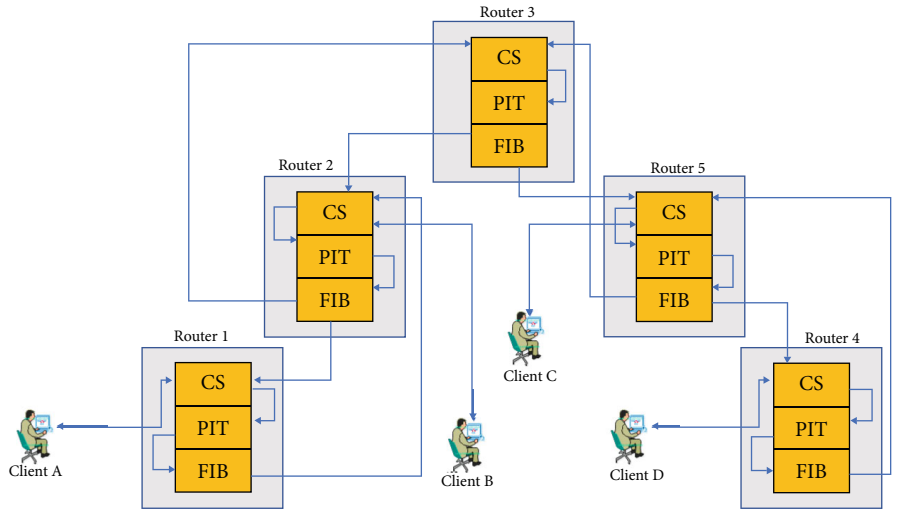

Figure 1: The basic architecture of NDN.

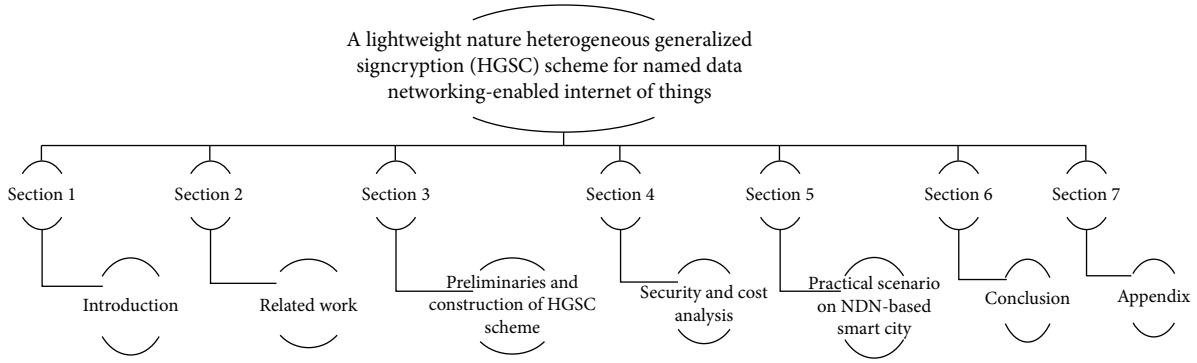

Figure 2: Paper organization.

proposed an efficient ID-BGS scheme for wireless sensor networks (WSN). The authors used bilinear pairing and proved the security of the proposed scheme under the random oracle model (ROM). Wei et al. [21] proposed an efficient ID-BGS for obtaining the confidentiality and authenticity of big data. Mishra and Singh in 2014 [22] surveyed the existing identity and certificateless generalized signcryption schemes. Based on security limitations in the existing schemes, the authors proposed two schemes to improve the limitations. Shen et al. in 2017 [23] improve the security of existing IBGS schemes which is suitable for low storage devices. Waheed et al. in 2019 [24] proved that the security of the Wei et al. [21] scheme is susceptible to attack and insecure. In the proposed cryptanalysis, the authors launched a security attack on the Wei et al. [21] scheme and found that the master secret key of the proposed scheme can be easily compromised.

However, the schemes [18–23] suffer from a heavy pairing operation due to the use of bilinear pairing. In [24], the authors did not provide any sort of solution to the proposed claims.

2.2. Certificateless Generalized Signcryption Schemes. Huifang et al. in 2010 [25] defined the notion of certificateless generalized signcryption (CGS) to solve the key escrow problem of IBGS. Later, Kushwah and Lal in 2012 [26] improved the security flaws of Huifang et al. [25] and proposed a new CGS scheme which is unforgeable against insider attacks. In 2014, Zhou et al. [27] proposed a provable CGS scheme for resource-constrained environment devices. The scheme provides security against malicious, but passive, key generation centre attacks. Zhang et al. in 2016 [28] proposed a CGS scheme for mobile health (M-Health). The scheme reduces the computation and communication costs by the use of the elliptic curve cryptosystem. Zhou et al. in 2017 [29] proposed a GSC scheme for security insurance in cloud storage. Zhang et al. in 2018 [30] proposed an efficient CGS scheme that is suitable for low power and low processor devices due to the use of the elliptic curve cryptosystem. Further, the scheme provides security against ciphertext attacks. In 2019, Zhou [31] improved the scheme of Zhang et al. [30] and proposed a new scheme for the mobile health system that can monitor the human body status in real time. Waheed et al. in 2019 [32] analyzed the proposed scheme of Zhou et al. [29] and proved that the scheme of Zhou et al. [29] is insecure against ciphertext indistinguishability under adaptive chosen-ciphertext attacks (IND-CCA2). Further, the

author proposed a new and improved scheme at the same cost which is secure against the aforementioned attacks. Karati et al. in 2019 [33] proposed a new CGS for resource-constrained IoT devices.

However, the schemes [25–33] suffer from heavy computation and communication costs due to the use of bilinear pairing and an elliptic curve cryptosystem.

2.3. Heterogeneous Signcryption Schemes. In 2011, Huang et al. [7] introduced the concept of heterogeneous signcryption (HS) which uses two different types of cryptosystem such as IBC at the sender side and CBC at the receiver side. The proposed scheme was suitable for the practical scenario of IoT where the sender and receiver belonged to different environments. Li et al. in 2016 [34] proposed a multireceiver heterogeneous signcryption scheme for wireless area network applications. The authors used CLC on the sender side and IBC on the receiver side (CLC-IBC). Raveendranath and Aneesh in 2016 [35] proposed a multireceiver HS scheme by using the elliptic curve cryptosystem to reduce the computation and communication costs of the existing HS scheme. Niu et al. in 2017 [36] proposed a CLC to IBC HS scheme by using bilinear pairings in the random oracle model. In the same year, Niu et al. [37] proposed a hybrid IBC to CLC scheme for multimessage and multireceiver. Li et al. in 2017 [38] proposed a PKI to IBC HS scheme for vehicle ad hoc networks. Niu et al. in 2017 [39] proposed a CLC to IBC HS scheme for the privacy-preserving multiparty aggregate scheme. Saeed et al. in 2017 [40] proposed a CLC to PKI online/offline HS scheme for IoT. Furthermore, the authors practically deployed the scheme on healthcare and the smart grid. Wang et al. in 2017 [41] proposed a PKI to IBC HS scheme for broadcast communication in ad hoc networks. Jin et al. in 2018 [42] proposed an IBC to PKI HS scheme for secure communication in the smart grid. Liu et al. in 2018 [43] proposed two HS schemes, such as PKI to CLC and CLC to PKI for secure communications between 5G network slicing. Liu and Ma in 2018 [44] proposed a cross domain of the PKI and IBC HS scheme for the medical information system. The authors use an elliptic curve cryptosystem to reduce computation and communication resources. Omala et al. in 2018 [45] proposed a CLC to IBC heterogeneous access control scheme for body area networks. Zhou et al. in 2019 [46] proposed a PKI to IBC HS scheme for vehicular ad hoc networks.

The aforementioned schemes [7], [34–46] suffer from heavy computation and communication costs due to the use of bilinear pairing and an elliptic curve cryptosystem. Moreover, these schemes are not suitable for NDN-based IoT due to its nongeneralized nature.

3. Preliminaries and Construction of HGSC Scheme

In this section, we will discuss the background of the hyperelliptic curve, threat model, and construction of our proposed HGSC scheme.

3.1. Hyperelliptic Curve (HEC). First, we will define the basic mathematics of hyperelliptic curves (HEC). Let $\mathscr{A}$ be a finite set and G be a genus of HEC with an order $G \geq 2$. Suppose (u), $f(u)\, \varepsilon\, d\,[u]$ and deg $(\mathrm{H}(u)) \leq G$, and $f(u)$ is a monic polynomial possessing deg $(f(u)) = 2d + 1$ [47]. Furthermore, HEC of genus $G \geq 2$ over d is a set of points (u), $d * d$ as in the mentioned equation:

$$\mathrm{HEC} : y^2 + (u)y = f(u). \tag{1}$$

Note: the point of HEC is not the same as elliptic curves [48]. It forms a divisor (D) that is the formal sum of finite integers such as $D = \sum \varkappa_i z_{\varkappa_i}$ where $\varkappa_i\, \varepsilon\, d$ and $z_{\varkappa_i}\, \varepsilon$ HEC. Additionally, HEC over the Jacobian group J_{HEC} has the brief order mentioned in the following equation:

$$(u_{\tau-1})^{2\mathrm{G}} \leq J_{\mathrm{HEC}}(\mathscr{A}) \leq (u_{\tau+1})^{2G}. \tag{2}$$

3.1.1. Hyperelliptic Curve Discrete Logarithm Problem. Assume D is a divisor, which is publicly known to everyone, and ℓ is a private number that is randomly chosen from $\mathscr{A}$ where finding ℓ from $d\,\ell = D$ is known to be an HEC discrete logarithm problem.

3.2. Syntax of Proposed Heterogeneous Generalized Signcryption Scheme. Here, we first explain different notations in Table 1, which can be used in the syntax and our proposed HGSC algorithms.

The syntax proposed scheme consists of 10 algorithms such as setup, generate secrete value, generate public key, generate partial private key, generate full private key, consumer private key generation, signcryption, unsigncryption, signature, and signature verification.

(1) *Setup*: the Key Generation Centre (KGC) executes this algorithm by taking the security parameter $\mathscr{k}$ to generate the master secret key $\mathscr{W}$, master public key $\mathscr{X}$, and public parameter set φ, then publishes φ and $\mathscr{X}$ openly in the network.

(2) *Generate secret value (GSVL)*: the producer takes (k, φ) and generates a secret value ∂.

(3) *Generate public key (GPK)*: in this algorithm, the producer then takes $(\mathscr{k}, \varphi, \partial)$ and generates a public key $\mathscr{B}_p$.

(4) *Generate partial private key (GPPK)*: in this algorithm, the KGC takes $(\mathscr{k}, \varphi, I\mathscr{D}_p, \mathscr{W}, \mathscr{B}_p)$ and generates a partial private key $(\mathscr{N}, \mathscr{J})$.

(5) *Generate full private key (GFPTK)*: in this algorithm, the producer takes $(\mathscr{k}, \varphi, I\mathscr{D}_p, \mathscr{N}, \mathscr{J}, \partial)$ as an input and generates his own full private key $(\mathscr{A}_p)$.

(6) *Consumer private key generation (CPKG)*: in this algorithm, the KGC takes $(ID_c, \mathscr{W})$ as an input by using IBS and a produces private key $(\mathscr{A}_c)$ and a public key $(\mathscr{B}_c)$ for the consumer.

Table 1: Abbreviations used in these algorithms.

Abbreviation	Definition
KGC	Key generation centre
$\mathscr{W}$	Master secret key
$\mathscr{k}$	Security parameter
$\mathscr{X}$	Master public key
φ	Public parameter set
$\mathscr{D}$	Divisor of the hyperelliptic curve
∂	Secret value
$\mathscr{B}_p$	Producer public key
$\mathbf{ID}_p$	Producer identity
$\mathscr{N}, \mathscr{J}$	Partial private key
$\mathscr{A}_p$	Producer full private key
$\mathbf{ID}_c$	Consumer identity
$\mathscr{L}$	Random number
$\mathscr{B}_c$	Consumer public key
$\mathscr{A}_c$	Consumer private key
$\mathscr{m}$	Message
$\mathscr{S}$	Producer digital signature
$\mathscr{H}_1$	Hash
$\mathscr{C}$	Content
Ψ	Singcrypted message/content
$\mathscr{C}$	Fresh nonce
Φ	Sign message/content

(7) *Signcryption*: in this algorithm, the producer takes $(ID_c, \mathscr{B}_c, \mathscr{A}_p, \mathscr{m}, \mathscr{X})$ and generates a signcrypted content/messageΨ and sends it to consumers.

(8) *Unsigncryption*: in this algorithm, the consumer unsigncrypts the Ψ by using $(ID_c, \mathscr{B}_c, \mathscr{A}_c, \mathscr{C}, \mu, \mathscr{S})$.

(9) *Signature*: in this algorithm, the producer takes $(\mathscr{A}_p, \mathscr{m})$ and generates a content/message sign Φ and sends it to the consumer.

(10) *Signature verification*: in this algorithm, the consumer verifies Φ by using $(ID_c, \mathscr{B}_c, \mathscr{A}_c, \mathscr{C}, \mu, \mathscr{S})$.

3.3. Threat Model. In our proposed HGSC scheme, we consider the Dolev-Yao (DY) [49] threat model. According to DY, the communication between two or more entities is not reliable and secure, as the attacker has full commands to reveal the contents of the ciphertext and inject a false signcryption/signature text to the network. The NDN-based IoT environment possesses different types of estimated security threats; it means that the adversaries can easily modify or delete the user's sensitive information. To maintain the security and authentication of NDN-based IoT devices, it is necessary to perform authentic and secure communication among entities in the NDN-based environment. The basic security requirements used in HGSC scheme are as follows: (1) Confidentiality: it means to keep the information secret from unauthorized users. The attackers can break the confidentiality of the HGSC scheme if he/she gets access to the encryption of decryption keys. The attacker here cannot access the original content in the message without having the encryption or decryption keys which are called confidentiality. (2) Unforgeability: it means that the signature could not be reproduced by any other party. Here, the attacker can generate a forged signature if he/she gets access to the digital signature generation secret key. If the attacker fails to do so, then it is called unforgeability. (3) Forward secrecy: forward secrecy means that if one of the session keys gets compromised by any malicious user, the data from the other session could not be affected. Here, the attacker cannot get access to the encryption or decryption keys even if the attacker got access to the sender private key. If the attacker is not able to access the encryption/decryption key of the user, it is called forward secrecy. (4) Antireplay attack: an antireplay attack means the attacker can resend a copy of an authenticated message again. Here, the attacker cannot reply to the existing message again if the sender and receiver use nonce and time stamping techniques for the freshness of a message.

3.4. Proposed Network Model. Here, we explain the workflow of our proposed HGSC scheme for NDN-enabled IoT. In our proposed scheme, we consider four entities such as the producer, consumer, NDN node, and Key Generation Centre (KGC) as shown in Figure 3. Here, we consider that the consumer belongs from IBC while the producer belongs from CLC. For registration of consumers and producers with KGC, the KGC announces the public parameter set and master public key.

3.4.1. Role of KGC. In the producer registration phase, the producer generates its public key from the public parameter set and sends it to the KGC. The KGC then generates a partial private key for the producer and sends it to the producer in reverse order using a secure network. After receiving the partial private key, the producer generates its full private key.

In the consumer registration phase, the consumer sends its identities to the KGC. The KGC after receiving the identities of the consumer generates private as well as public keys for the consumer and sends them back to the consumer using a secured network.

3.4.2. Role of Consumer. Suppose a consumer sends an interest for some content/message in the NDN-based IoT environment to any producer.

3.4.3. Role of Producer. After receiving the interest, the producer then signs/signcrypts the content using its private key and sends it back to the requested consumer. However, the NDN node will store the copy content/message in their CS according to the caching policies of NDN. After receiving the content/message, the consumer verifies the signature or unsigncrypts the respective content/message.

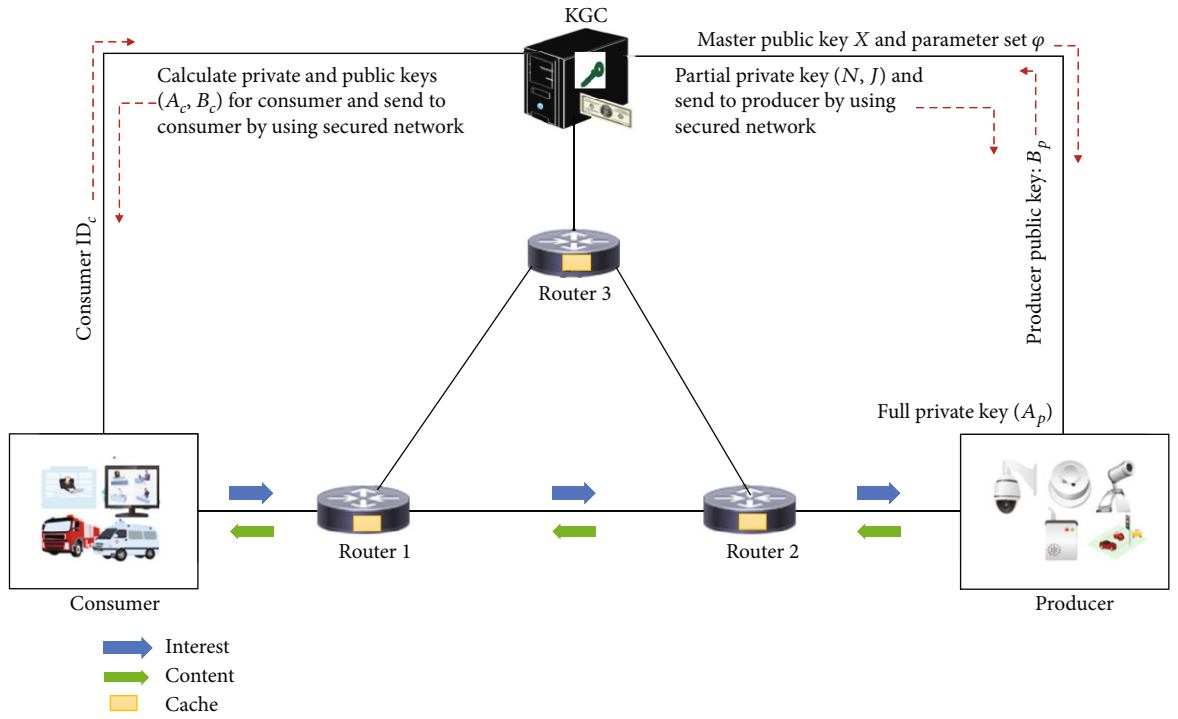

Figure 3: Proposed network model.

Setup:It is processed by KGC
Input: Security parameter $\mathscr{k}$
Output:Master secret key $\mathscr{W}$, master public key $\mathscr{X}$, and public parameter set $\varphi = \{\mathscr{X}, \mathscr{D}, \mathrm{G} \geq 2, \mathrm{HEC}, \mathscr{H}_1, \mathscr{H}_2, \mathscr{H}_3, \mathscr{H}_4\}$.
Process: KGC first produces public parameter set φ, then after randomly pick a master secret key $\mathscr{W} \in \{1, 2, 3, \cdots, z-1\}$ where $\varkappa = 2^{80}$, computes a master public key $\mathscr{X} = \mathscr{W}.\mathscr{D}$, where $\mathscr{D}$ is a divisor of the hyperelliptic curve.
Publishing KGC published φ and $\mathscr{X}$ openly in the network.

Algorithm 1: Setup.

3.5. Proposed Heterogeneous Generalized Signcryption Algorithms. The proposed HGSC consists of the following eight steps.

3.5.1. Algorithm 1. In this step, the KGC generates a master public key, master secret key, and public parameter as shown in Algorithm 1.

3.5.2. Algorithm 2. In this step, the producer generates secrete value as shown in Algorithm 2.

3.5.3. Algorithm 3. In this step, the KGC generates a partial private key for the producer as shown in Algorithm 3.

3.5.4. Algorithm 4. In this step, the producer generates its full private key as shown in Algorithm 4.

3.5.5. Algorithm 5. In this step, the KGC public as well as the private key for the consumer are shown in Algorithm 5.

3.5.6. Algorithm 6. In this step, the producer sign/signcrypts the requested contents as shown in Algorithm 6.

3.5.7. Algorithm 7. In this step, the consumer verifies the sign contents or unsigncrypts the signcrypted contents as shown in Algorithm 7.

4. Security and Cost Analyses

In this section, we briefly discuss the informal analysis and computation and communication cost analyses of our proposed scheme.

4.1. Informal Analysis. This section describes the contribution in upholding the security properties of confidentiality, unforgeability, forward secrecy, and antireplay attack.

4.1.1. Confidentiality. Confidentiality means to keep the contents secret; the attacker (ζ) cannot calculate the plaintext from signcrypted ciphertext. Let the ζ want to break the confidentiality of our proposed scheme and generate the plaintext from signcrypted ciphertext $\Psi = (\mathscr{C}, \mu, \mathscr{S})$. For this purpose, the ζ needs to calculate $\mathscr{C}$ from $\Psi = (\mathscr{C}, \mu, \mathscr{S})$, and to do so, ζ needs δ, β, and α from $K = \mathscr{H}_3(\delta, \alpha, \beta, ID_c$,

Generate secret value (GSVL):It is run by producer
Input: (k, φ)
Output:Secret value ∂
Process: Producer randomly picks a secret value $\partial \in \{1, 2, 3, \cdots, z-1\}$

Algorithm 2: Secret value generation.

Generate partial private key (GPPK):It is executed by KGC
Input: $(\mathbb{k}, \varphi, I\mathbb{D}_p, \mathbb{W}, \mathbb{B}_p)$
Output:Partial private key $(\mathbb{N}, \mathbb{J})$
Process:KGC randomly picks a number $\mathbb{Q} \in \{1, 2, 3, \cdots, z-1\}$, compute $\mathbb{N} = \mathbb{Q} \cdot \mathbb{D}$, compute $\mathbb{J} = \mathbb{Q} + \mathbb{W} \cdot \mathbb{H}_1(ID_p, \mathbb{Q}, \mathbb{B}_p) \mod z$, and send $(\mathbb{N}, \mathbb{J})$ to the producer by using secured network.

Algorithm 3: Partial private key generation.

Generate full private key (GFPTK):It is executed by the producer
Input: $(\mathbb{k}, \varphi, I\mathbb{D}_p, \mathbb{N}, \mathbb{J}, \partial)$
Output:Full private key $(\mathbb{A}_p)$
Process: Producer computes $\mathbb{A}_p = (\mathbb{J}, \partial)$.

Algorithm 4: Partial private key generation.

Consumer private key generation (CPKG):It is executed by KGC and note that KGC acts like a private key generator in an identity-based cryptosystem
Input: $(ID_c, \mathbb{W})$
Output:Private key $(\mathbb{A}_c)$, public key $(\mathbb{B}_c)$
Process: KGC randomly picks a number $\mathbb{L} \in \{1, 2, 3, \cdots, z-1\}$, computes $\mathbb{B}_c = \mathbb{L} \cdot \mathbb{D}$, computes $\mathbb{A}_c = \mathbb{L} + \mathbb{W} \cdot \mathbb{H}_1(I\mathbb{D}_c, \mathbb{L}, \mathbb{B}_c) \mod \mathbb{x}$, and sends $(\mathbb{A}_c, \mathbb{B}_c)$ to the consumer by using a secured network.

Algorithm 5: Consumer's key generation.

Heterogeneous generalized signcryption (HGSN):It is executed by producer
Input: $(ID_c, \mathbb{B}_c, \mathbb{A}_p, m, \mathbb{X})$
Output:Signcryption Ψ
Process: Producer randomly picks a number $\mathbb{R} \in \{1, 2, 3, \cdots, z-1\}$,
If $(\mathbb{C} = m)$
{
(1) Select a fresh nonce $\mathbb{T}$
(2) Compute $\mu = \mathbb{H}_2(\mathbb{T}, m)$
(3) Compute $\mathbb{S} = \mathbb{R} + \mu(\mathbb{J} + \partial)$ and go to step 11
}
Else
{
(4) Compute $\delta = \mathbb{R} \cdot \mathbb{D}$
(5) Compute $\Upsilon = \mathbb{H}_3(ID_c, \mathbb{B}_c, \mathbb{X})$
(6) Compute $\alpha = \mathbb{B}_c + \mathbb{X} \cdot Y$
(7) compute $\beta = \alpha \cdot \mathbb{D}$
(8) compute $K = \mathbb{H}_4(\delta, \alpha, \beta, ID_c, \mathbb{B}_c)$
(9) Repeat step 1 and compute $\mathbb{C} = \mathbb{E}_K(\mathbb{T}, m)$
(10) Repeat steps 2 and 3 and go to step 11
}
(11) Send $\Psi = (\mathbb{C}, \mu, \mathbb{S}, \delta)$ to the consumer by using an open network

Algorithm 6: Signcryption and signature generation.

Heterogeneous generalized unsigncryption (HGUSN)It is executed by consumer
Input: $(ID_c, \mathscr{B}_c, \mathscr{A}_c, \mathscr{C}, \mu, \mathscr{S}, \delta)$
Output and verifications:$(\mathscr{T}, m)$ and $\mu' \stackrel{?}{=} \mu$
Process: (1) Consumer computes $\beta = \delta \cdot \mathscr{A}_c$
(2) Calculates $K = \mathscr{H}_4(\delta, \alpha, \beta, ID_c, \mathscr{B}_c)$
(3) Perform decryption $(\mathscr{T}, m) = D_K(\mathscr{C})$
(4) Compute $\mu' = \mathscr{H}_2(\mathscr{T}, m)'$
(5) Compare $\mu' \stackrel{?}{=} \mu$; if it holds, then accept; otherwise, reject

Algorithm 7. Unsigncryption and signature verification.

Table 2: Comparative analysis in terms of major operations with CGS schemes.

Schemes	Signcryption	Unsigncryption	Total
Zhang et al. [28]	4 SEPM	5 SEPM	9 SEPM
Zhou et al. [29]	7 SEPM	8 SEPM	15 SEPM
Zhang et al. [30]	5 SEPM	4 SEPM	9 SEPM
Zhou [31]	5 SEPM	7 SEPM	12 SEPM
Waheed et al. [32]	1 SBP + 5 SPBPM	3 SBP + 1 SPBPM	4 SBP + 6 SPBPM
Karati et al. [33]	3 SPBPM + 3 SEXP	2 SPBPM + 2 SBP + 5 SEXP	5 SPBPM + 2 SBP + 8 SEXP
Proposed scheme	2 SHEDM	2 SHEDM	4 SHEDM

$\mathscr{B}_c$). Here, $\delta = \mathscr{R} \cdot \mathscr{D}$, $\beta = \alpha \cdot \mathscr{D}$, and $\alpha = \mathscr{B}_c + \mathscr{X} \cdot Y$ where $\mathscr{R}$ and α are discrete logarithm problems over the hyperelliptic curve cryptosystem which is not possible to calculate. Thus, our proposed scheme provides the property of confidentiality.

4.1.2. Unforgeability. Unforgeability means that no one can sign the content, except the valid provider. To forge the signature, ζ needs to calculate $\mathscr{R}$, μ, and $\mathscr{J} + \partial$. Here, $\mathscr{R}$ is a private number, and for calculating $\mu = \mathscr{H}_2(\mathscr{T}, m)$, ζ needs to calculate a private number $\mathscr{T}$ from $\mu = \mathscr{H}_2(\mathscr{T}, m)$. Further, ζ needs to $\mathscr{J} + \partial$ where $\mathscr{J}$ is a fresh nonce and ∂ is a private number, so to forge the signature $\mathscr{S}$, ζ needs to calculate 3 private numbers $\mathscr{R}$, $\mathscr{T}$, and ∂ with a fresh nonce $\mathscr{J}$ which is not possible to calculate. So, our proposed scheme provides the property of unforgeability.

4.1.3. Forward Secrecy. Forward secrecy means if the private key of the signer is compromised, still it could not affect the respective contents, because the content is encrypted via a session secret key. Here, in our scheme, to break forward secrecy, ζ needs to calculate $K = \mathscr{H}_3(\delta, \alpha, \beta, ID_c, \mathscr{B}_c)$ which requires δ where $\delta = R \cdot D$. So, for this purpose, ζ needs to calculate R, which is a private number, and δ is a discrete logarithm problem over the hyperelliptic curve, which is infeasible for ζ to break.

4.1.4. Antireplay Attack. In our proposed scheme, before communication, the provider generates a $\mathscr{T}$ and stores it in his memory. Then after, it sends the encrypted text as $\mathscr{C} = \mathscr{E}_K(\mathscr{T}, m)$ to the consumer. After receiving the FNs, the consumer, by using secret key K, performs the decryption process on the received ciphertext. Once the $\mathscr{T}$ is recovered, the consumer verifies the freshness, and if it is fresh, then the ciphertext is new. However, ζ cannot replay the old messages because he/she needs fresh FNs for every new session.

4.2. Cost Analysis. In this section, we compare the proposed scheme with existing certificateless generalized signcryption (CGS) and identity-based generalized signcryption (ID-BGS) schemes in terms of computation and communication costs.

4.2.1. Computation Cost. Here, we compare our proposed scheme with existing CGS and ID-BGS schemes in terms of expansive mathematical operations such as single pairing-based point multiplication (SPBPM), single bilinear pairing (SBP), single exponential (SEXP), single elliptic curve point multiplication (SEPM), and hyperelliptic curve point multiplication (SHEDM). Moreover, operations like addition, division, subtraction, encryption, decryption, and hash are neglected, due to its minimal consumption time during the computation.

Furthermore, we compare our scheme with the existing CGS and ID-BGS schemes in milliseconds (ms) by using the above major operation, according to the experiments performed in [50] with the following hardware and software specifications:

(i) Intel Core i7-4510U CPU

(ii) 2 GHz processor

(iii) 8 GB RAM

(iv) Windows 7, 64 bits

(v) Multiprecision integer and rational arithmetic C library

According to [50], an SPBPM will take 4.32 ms, a single SBP will take 14.90 ms, SEXP will take 1.25 ms, and SEPM will take 0.97 ms. Based on the experiments performed in [51, 52], we consider that a SHEDM will take 0.48 ms. On the bases of the above expansive mathematical operations, we conduct the computation cost comparison of our proposed scheme with existing CGS schemes which are Zhang et al. [28], Zhou et al. [29], Zhang et al. [30], Zhou [31], Waheed et al. [32], and Karati et al. [33] as shown in Tables 2 and 3. Further, the computation cost comparison of our proposed scheme with existing ID-BGS schemes which are Wei et al. [21] and Shen et al. [23] is shown in Tables 4 and 5. Moreover, a clear computation reduction is shown in Figures 4 and 5.

(1) Computation Cost Reduction of Our Scheme from CGS Schemes. The following formula will be used to calculate cost reduction

$$\left(\frac{\text{existing scheme} - \text{our scheme}}{\text{existing scheme}}\right) * 100. \tag{3}$$

(i) Computation cost reduction from Zhang et al. [28]:

$$\left(\frac{9\,\text{SEPM} - 4\,\text{SHEDM}}{9\,\text{SEPM}}\right) * 100 = \left(\frac{8.73 - 1.92}{8.73}\right) * 100 = 78.09\% \tag{4}$$

(ii) Computation cost reduction from Zhou et al. [29]:

$$\left(\frac{15\,\text{SEPM} - 4\,\text{SHEDM}}{15\,\text{SEPM}}\right) * 100 = \left(\frac{14.55 - 1.92}{14.55}\right) * 100 = 86.80\% \tag{5}$$

(iii) Computation cost reduction from Zhang et al. [30]:

$$\left(\frac{9\,\text{SEPM} - 4\,\text{SHEDM}}{9\,\text{SEPM}}\right) * 100 = \left(\frac{8.73 - 1.92}{8.73}\right) * 100 = 78.09\% \tag{6}$$

(iv) Computation cost reduction from Zhou [31]:

$$\left(\frac{12\,\text{SEPM} - 4\,\text{SHEDM}}{12\,\text{SEPM}}\right) * 100 = \left(\frac{11.64 - 1.92}{11.64}\right) * 100 = 83.50\% \tag{7}$$

Table 3: Computation cost comparison (CGS) in ms.

Schemes	Signcryption	Unsigncryption	Total
Zhang et al. [28]	3.88	4.85	8.73
Zhou et al. [29]	6.79	7.76	14.55
Zhang et al. [30]	4.85	3.88	8.73
Zhou [31]	4.85	6.79	11.64
Waheed et al. [32]	36.5	49.02	85.52
Karati et al. [33]	16.71	44.69	61.4
Proposed scheme	0.96	0.96	1.92

Table 4: Comparative analysis in terms of major operations with ID-BGS.

Schemes	Signcryption	Unsigncryption	Total
Wei et al. [21]	6 SEXP	2 SEXP + 5 SBP	8 SEXP + 5 SBP
Shen et al. [23]	5 SEXP + 1 SBP	3 SBP	5 SEXP + 4 SBP
Proposed	2 SHEDM	2 SHEDM	4 SHEDM

Table 5: Computation cost comparison (ID-BGS) in ms.

Schemes	Signcryption	Unsigncryption	Total
Wei et al. [21]	7.5	77	84.5
Shen et al. [23]	14.90	29.8	44.7
Proposed	0.96	0.96	1.92

(v) Computation reduction from Waheed et al. [32]:

$$\left(\frac{4\,\text{SBP} + 6\,\text{SPBPM} - 4\,\text{SHEDM}}{4\,\text{SBP} + 6\,\text{SPBPM}}\right) * 100 = \left(\frac{85.52 - 1.92}{85.52}\right) * 100 = 97.75\% \tag{8}$$

(vi) Computation cost reduction from Karati et al. [33]:

$$\left(\frac{5\,\text{SPBPM} + 2\,\text{SBP} + 8\,\text{SEXP} - 4\,\text{SHEDM}}{5\,\text{SPBPM} + 2\,\text{SBP} + 8\,\text{SEXP}}\right) * 100 = \left(\frac{61.4 - 1.92}{61.4}\right) * 100 = 96.87\% \tag{9}$$

(2) Computation Cost Reduction of Our Scheme from ID-BGS Schemes.

(i) Our Computation Cost Reduction from Wei et al. [21]:

$$\left(\frac{8\,\text{SEXP} + 5\,\text{SBP} - 4\,\text{SHEDM}}{8\,\text{SEXP} + 5\,\text{SBP}}\right) * 100 = \left(\frac{84.5 - 1.92}{84.5}\right) * 100 = 97.84\% \tag{10}$$

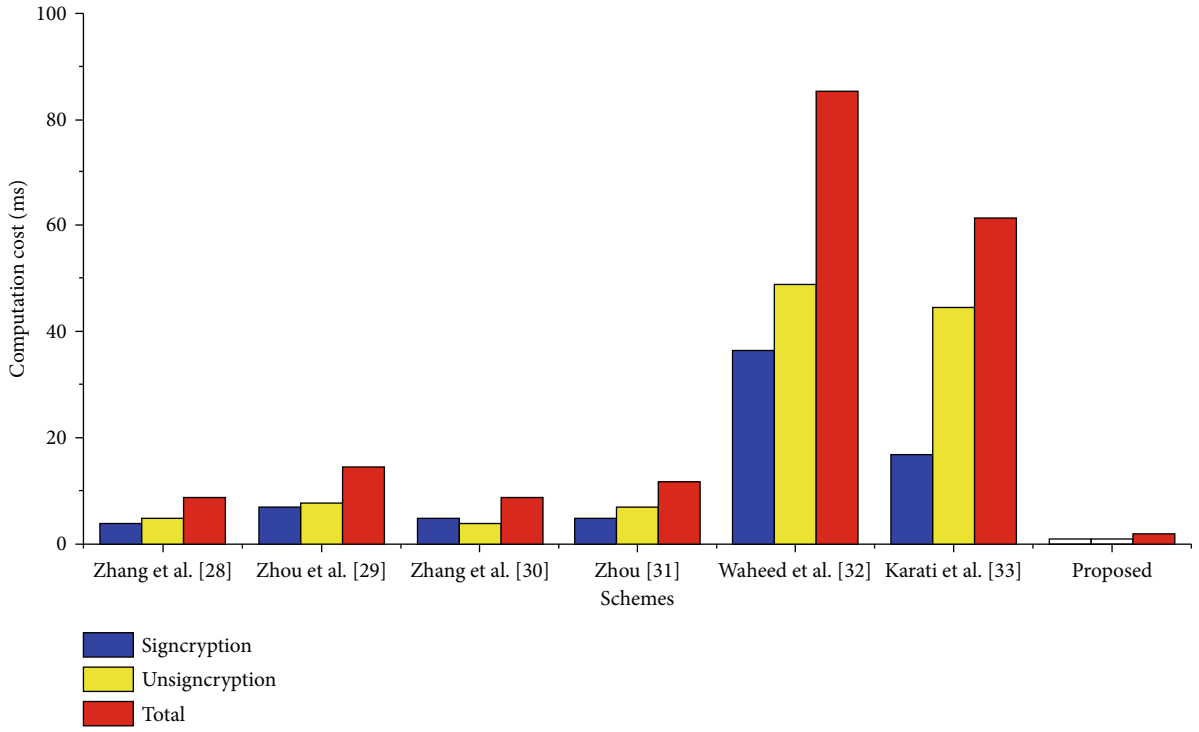

Figure 4: Computational cost reduction from CGS schemes.

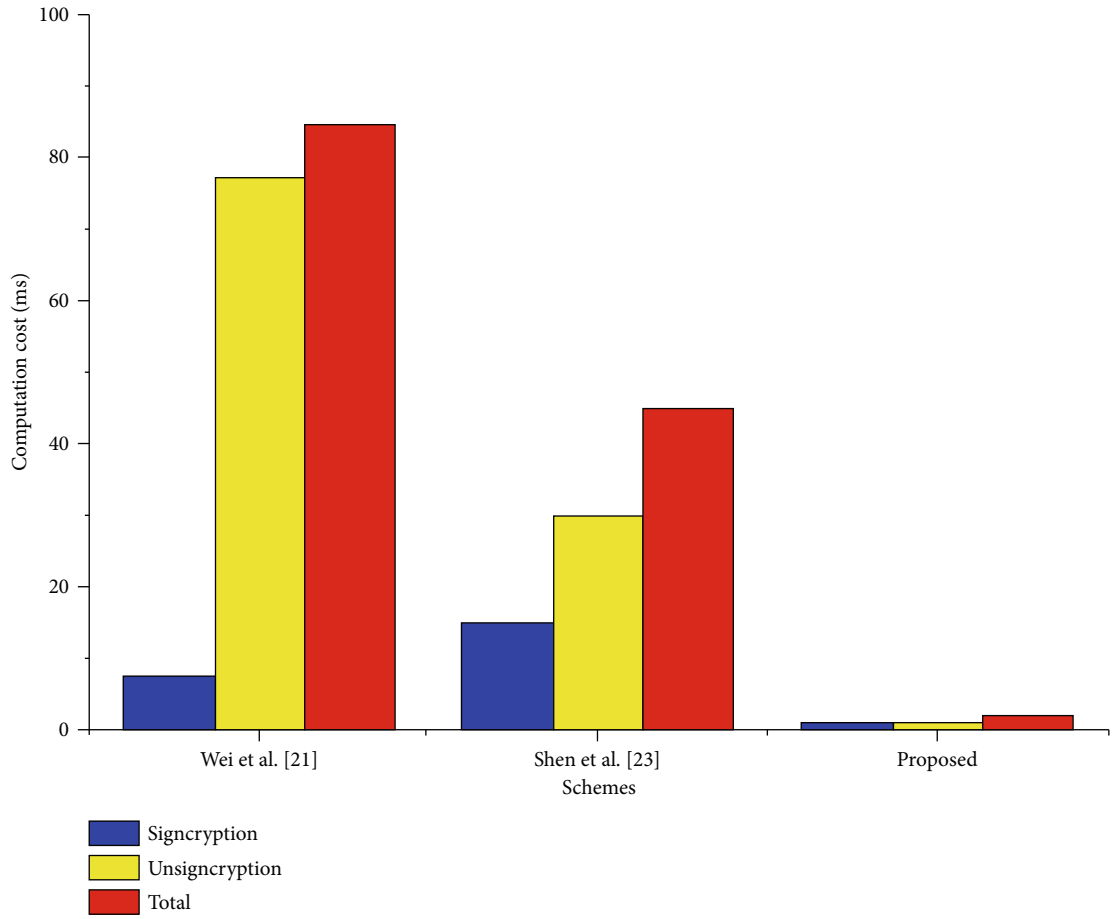

Figure 5: Computational cost reduction from ID-BGS schemes.

Table 6: Communication cost comparison with CGS schemes.

Schemes	Communication cost	Ciphertext size
Zhang et al. [28]	$1 \mid m \mid +3 \mid Q \mid = 1\|100\| + 3\|160\| = 100 + 540$	640 bits
Zhou et al. [29]	$1 \mid m \mid +3 \mid Q \mid = 1\|100\| + 3\|160\| = 100 + 540$	640 bits
Zhang et al. [30]	$1 \mid m \mid +2 \mid Q \mid = 1\|100\| + 2\|160\| = 100 + 360$	460 bits
Zhou [31]	$1 \mid m \mid +3 \mid Q \mid = 1\|100\| + 3\|160\| = 100 + 540$	640 bits
Waheed et al. [32]	$1 \mid m \mid +3 \mid G \mid = 1\|100\| + 3\|1024\| = 100 + 3072$	3172 bits
Karati et al. [33]	$1 \mid m \mid +4 \mid G \mid = 1\|100\| + 4\|1024\| = 100 + 4096$	4196 bits
Proposed	$1 \mid m \mid +3 \mid N \mid = 1\|100\| + 3\|80\| = 100 + 240$	340 bits

(ii) Our computation cost reduction from Shen et al. [23]:

$$\left(\frac{5\,\text{SEXP} + 4\,\text{SBP} - 4\,\text{SHEDM}}{5\,\text{SEXP} + 4\,\text{SBP}}\right) * 100 = \left(\frac{44.7 - 1.92}{44.7}\right) * 100 = 95.70\% \tag{11}$$

4.2.2. Communication Cost. In this section, we compare our proposed scheme with existing CGS and ID-BGS schemes in terms of bits. For this purpose, we suppose elliptic curve $|Q| = 160$ bits, bilinear pairing $|G| = 1024$ bits, hyperelliptic curve $|N| = 80$ bits, and message $|M| = 100$ bits. According to our suppositions, for CGS schemes, the communication cost of the Zhang et al. [28] scheme is $1 \mid m \mid +3|Q|$, of the Zhou et al. [29] scheme is $1 \mid m \mid + 3|Q|$, of the Zhang et al. [30] scheme is $1 \mid m \mid +2|Q|$, of the Zhou [31] scheme is $1 \mid m \mid +3|Q|$, of the Waheed et al. [32] scheme is $1 \mid m \mid +3|G|$, and of the Karati et al. [33] scheme is $1 \mid m \mid +4|G|$, and the communication cost of our proposed scheme is $1 \mid m \mid +3|N|$. Furthermore, Table 6 shows the efficiency of our scheme from Zhang et al. [28], Zhou et al. [29], Zhang et al. [30], Zhou [31], Waheed et al. [32], and Karati et al. [33]. Moreover, a clear communicational cost reduction is shown in Figure 6.

Furthermore, for the ID-BGS schemes, the communication cost of the Wei et al. [21] scheme is $1 \mid m \mid +4 \mid Q \mid$ and of the Shen et al. [23] scheme is $1 \mid m \mid +7 \mid Q \mid$. Furthermore, Table 7 shows the efficiency of our scheme from Wei et al. [21] and Shen et al. [23]. Additionally, a clear communicational cost reduction is shown in Figure 7.

(1) Communication Cost Reduction of Our Scheme from CGS Schemes. The following formula can be used to calculate the cost reduction.

$$\left(\frac{\text{existing scheme} - \text{our scheme}}{\text{existing scheme}}\right) * 100. \tag{12}$$

(i) Our communication cost reduction from Zhang et al. [28]:

$$\left(\frac{1 \mid m\mid+3 \mid Q\mid-1 \mid m\mid+3 \mid N \mid}{1 \mid m\mid+3 \mid Q \mid}\right) * 100 = \left(\frac{1|100| + 3|160| - 1|100| + 3|80|}{1|100| + 3|160|}\right) * 100 = \left(\frac{640\,\text{bits} - 340\,\text{bits}}{640\,\text{bits}}\right) * 100 = 46.87\% \tag{13}$$

(ii) Our communication cost reduction from Zhou et al. [29]:

$$\left(\frac{1 \mid m\mid+3 \mid \text{Q}\mid-1 \mid m\mid+3 \mid N \mid}{1 \mid m\mid+3 \mid Q \mid}\right) * 100 = \left(\frac{1|1\text{OO}| + 3|160| - 1|100| + 3|80|}{1|100| + 3|160|}\right) * 100 = \left(\frac{640\,\text{bits} - 340\,\text{bits}}{640\,\text{bits}}\right) * 100 = 46.87\% \tag{14}$$

(i) Our communication cost reduction from Zhang et al. [30]:

$$\left(\frac{1 \mid m\mid+2 \mid Q\mid-1 \mid m\mid+3 \mid N \mid}{1 \mid m\mid+2 \mid Q \mid}\right) * 100 = \left(\frac{1|1\text{OO}| + 2|160| - 1|1\text{OO}| + 3|80|}{1|1\text{OO}| + 3|160|}\right) * 100 = \left(\frac{460\,\text{bits} - 340\,\text{bits}}{460\,\text{bits}}\right) * 100 = 21.73\% \tag{15}$$

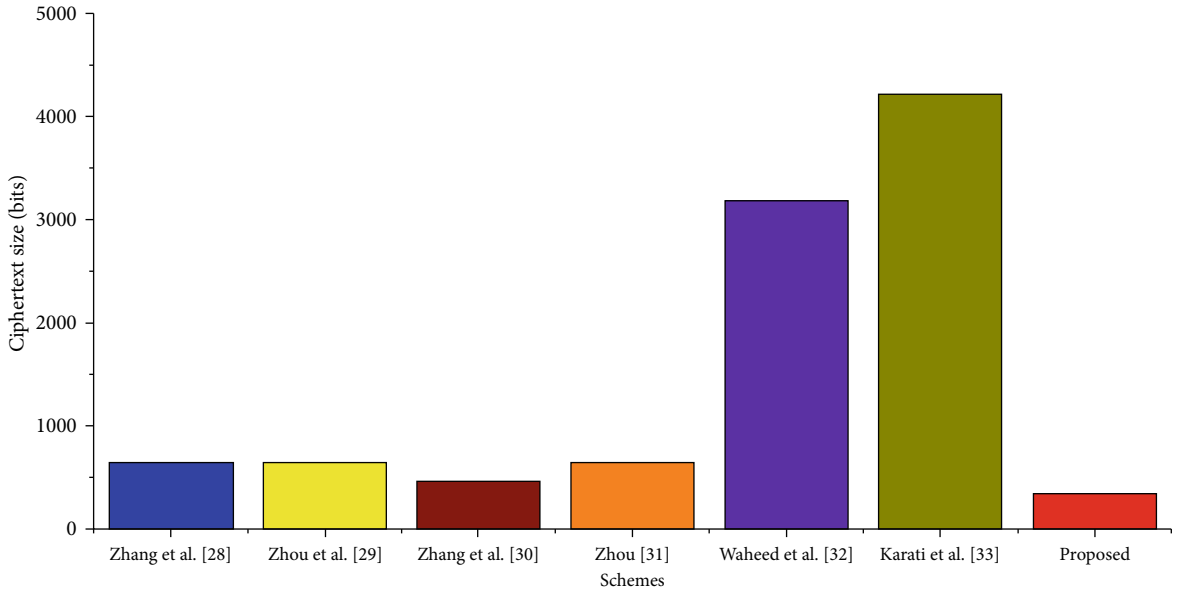

Figure 6: Communication cost reduction from CGS schemes.

Table 7: Communication cost comparison with CGS schemes.

Schemes	Communication cost	Ciphertext size
Wei et al. [21]	$1 \mid m \mid +4 \mid Q \mid = 1\|100\| + 4\|160\| = 100 + 640$	740 bits
Shen et al. [23]	$1 \mid m \mid +7 \mid Q \mid = 1\|100\| + 7\|160\| = 100 + 1120$	1220 bits
Proposed	$1 \mid m \mid +3 \mid N \mid = 1\|100\| + 3\|80\| = 100 + 240$	340 bits

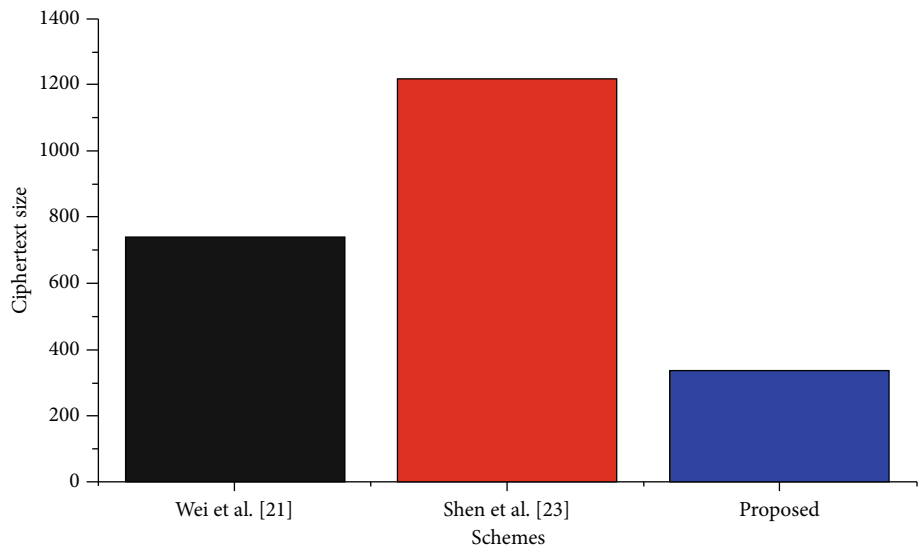

Figure 7: Communication cost reduction from ID-BGS schemes.

(ii) Our communication cost reduction from Zhou [31]:

$$\left(\frac{1\,|\,m|+3\,|\,Q|-1\,|\,\mathrm{m}|+3\,|\,\mathrm{N}\,|}{1\,|\,\mathrm{m}|+3\,|\,\mathrm{Q}\,|}\right)*100 = \left(\frac{1|1\mathrm{OO}|+3|160|-1|1\mathrm{OO}|+3|80|}{1|1\mathrm{OO}|+3|160|}\right)*100 = \left(\frac{640\text{ bits}-340\text{ bits}}{640\text{ bits}}\right)*100 = 46.87\% \tag{16}$$

(iii) Our communication cost reduction from Waheed et al. [32]:

$$\left(\frac{1\,|\,\mathrm{m}|+3\,|\,\mathrm{G}|-1\,|\,m|+3\,|\,N\,|}{1\,|\,m|+3\,|\,G\,|}\right)*100 = \left(\frac{1|1\mathrm{OO}|+3|1024|-1|1\mathrm{OO}|+3|80|}{1|1\mathrm{OO}|+3|1024|}\right)*100 = \left(\frac{3172\text{ bits}-340\text{ bits}}{3172\text{ bits}}\right)*100 = 89.28\% \tag{17}$$

(iv) Our communication cost reduction from Karati et al. [33]:

$$\left(\frac{1\,|\,m|+4\,|\,G|-1\,|\,m|+3\,|\,N\,|}{1\,|\,m|+4\,|\,G\,|}\right)*100 = \left(\frac{1|1\mathrm{OO}|+4|1024|-1|1\mathrm{OO}|+3|80|}{1|1\mathrm{OO}|+4|1024|}\right)*100 = \left(\frac{4196\text{ bits}-340\text{ bits}}{4196\text{ bits}}\right)*100 = 91.89\% \tag{18}$$

(2) Communication Cost Reduction of Our Scheme from ID-BGS Schemes.

(i) Our communication cost reduction from Wei et al. [21]:

$$\left(\frac{1\,|\,m|+4\,|\,Q|-1\,|\,m|+3\,|\,N\,|}{1\,|\,m|+4\,|\,Q\,|}\right)*100 = \left(\frac{1|100|+4|160|-1|1\mathrm{OO}|+3|80|}{1|100|+4|160|}\right)*100 = \left(\frac{740\text{ bits}-340\text{ bits}}{740\text{ bits}}\right)*100 = 54.05\% \tag{19}$$

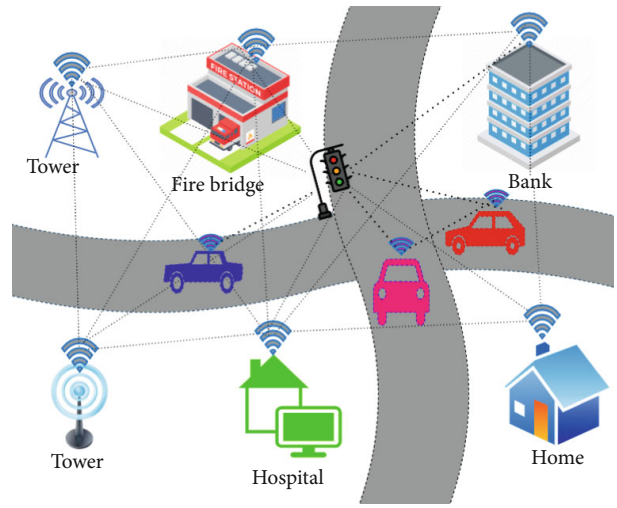

Figure 8: Deployment in smart city.

(ii) Our communication cost reduction from Shen et al. [23]:

$$\left(\frac{1\,|\,m|+7\,|\,Q|-1\,|\,m|+3\,|\,N\,|}{1\,|\,m|+7\,|\,Q\,|}\right)*100 = \left(\frac{1|1\mathrm{OO}|+7|160|-1|1\mathrm{OO}|+3|80|}{1|1\mathrm{OO}|+7|160|}\right)*100 = \left(\frac{1220\text{ bits}-340\text{ bits}}{1220\text{ bits}}\right)*100 = 72.13\% \tag{20}$$

5. Practical Scenario on NDN-Based Smart City

Assume an NDN-based smart city, where the number of sensors deployed for monitoring environmental conditions is shown in Figure 8. The sensors can monitor some emergency parameters such as fire, leakage of water, and vehicle accident, which require authentication as well as confidentiality. Furthermore, these sensors can sense some normal parameters (e.g., temperature, humidity, and energy consumption) which require authentication only.

These sensed parameters are forwarded through NDN routers using the following transmission modes.

(1) *Pull-based mode*: in this mode, a consumer sends an interest in some content/message. The sensor nodes provide the requested contents according to given interest.

(2) *Push-based mode*: in this mode, the sensor nodes intermittently forward content/message without receiving any interests of the consumer. This mode better suits the secure transfer of emergency contents/messages to a specific destination in run time.

Our deployment consists of entities such as KGC (authorization provider), content/message producer (sensors and

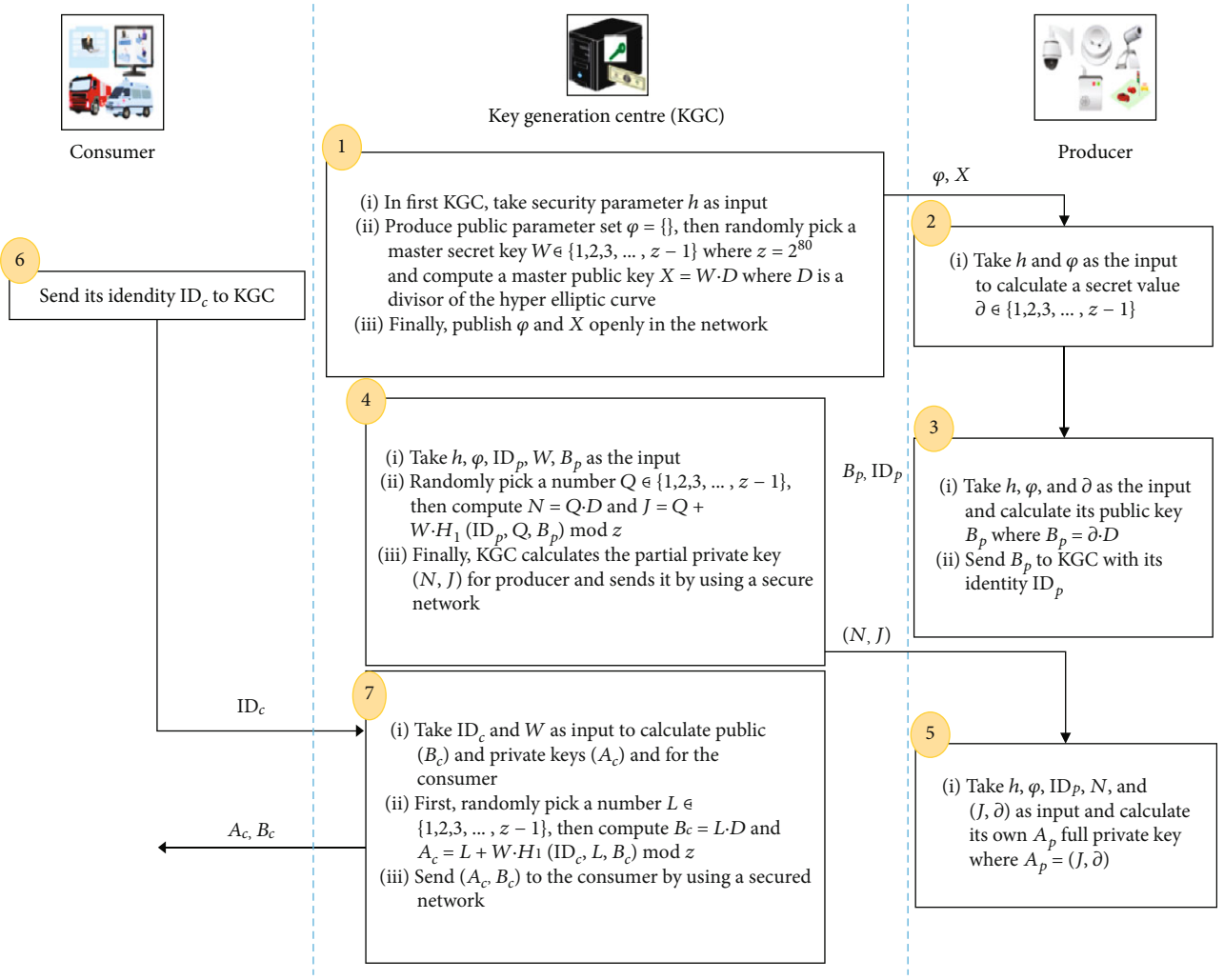

Figure 9: Registration and key generation process.

NDN router), and consumer (mobile user, fire centre, hospital, etc.).

The overall process is discussed below.

5.1. Registration and Key Generation Phase. In Figure 9, we explain the registration and key generation of consumers and providers. In step 1, the KGC takes security parameters $\mathscr{k}$ as input and produces public parameter set φ for generating master secret key $\mathscr{W} \in \{1, 2, 3, \cdots, z-1\}$ and master public key $\mathscr{X} = \mathscr{W} \cdot \mathscr{D}$. Then, publish φ and $\mathscr{X}$ in the entire network. In step 2, the producer takes (k, φ) as an input and generates a secret value $\partial \in \{1, 2, 3, \cdots, \varkappa - 1\}$.

In step 3, the producer then takes the parameters $(\mathscr{k}, \varphi, \partial)$ as input and computes its public key $\mathscr{B}_p = \partial \cdot \mathscr{D}$. After computing $\mathscr{B}_p$, the producer sends it alongside with his identity ID_p to the KGC. In step 4, after receiving the $\mathscr{B}_p$ and ID_p, the KGC takes $(\mathscr{k}, \varphi, ID_p, \mathscr{W}, \mathscr{B}_p)$ as input and randomly picks a number from $\mathscr{Q} \in \{1, 2, 3, \cdots, z-1\}$, computes $\mathscr{N} = \mathscr{Q} \cdot \mathscr{D}$ and $\mathscr{J} = \mathscr{Q} + \mathscr{W} \cdot \mathscr{H}_1(ID_p, \mathscr{Q}, \mathscr{B}_p)$, and generates a partial private key $(\mathscr{N}, \mathscr{J})$ for the producer. The KGC then sends $(\mathscr{N}, \mathscr{J})$ to the producer using a secure network. In step 5, upon receiving $(\mathscr{N}, \mathscr{J})$, the producer takes $(\mathscr{k}, \varphi, ID_p, \mathscr{N}, \mathscr{J}, \partial)$ as an input and computes its own full private key $(\mathscr{A}_p)$.

In step 6, the consumer sends the identity ID_c to KGC for registration. In step 7, upon receiving the ID_c, the KGC takes $(ID_c, \mathscr{W})$ as input and randomly picks a number from $\mathscr{L} \in \{1, 2, 3, \cdots, \varkappa - 1\}$ to calculate the public key $(\mathscr{B}_c)$ and private key $(\mathscr{A}_c)$ for the consumer. The KGC then sends (B_c, A_c) to the consumer using a secure channel.

5.2. Communication Phase. In Figure 10, we explain the secure communication of the consumer and provider after a successful registration and key generation phase. If the consumer wants the signed/signcrypted contents from the producer or the producer wants to deliver signed/signcrypted contents to the consumer securely, first, for the signcrypted content, the producer takes content (m) and$(ID_c, \mathscr{B}_c, \mathscr{A}_p, \mathscr{X})$ with a randomly picked number from $\mathscr{R} \in \{1, 2, 3, \cdots, z-1\}$; computes the secret value δ, hash of $(ID_c, \mathscr{B}_c, \mathscr{X})$, a fresh nonce $\mathscr{T}$, encrypted contents $\mathscr{C} = \mathscr{E}_K(\mathscr{T}, m)$, and a hash of the encrypted contents $\mu = \mathscr{H}_2(\mathscr{T}, m)$; and applies signature $\mathscr{S} = \mathscr{R} + \mu(\mathscr{J} + \partial)$ on it. Finally, generate the signcrypted contents $\Psi = (\mathscr{C}, \mu, \mathscr{S}, \delta)$ and send it to

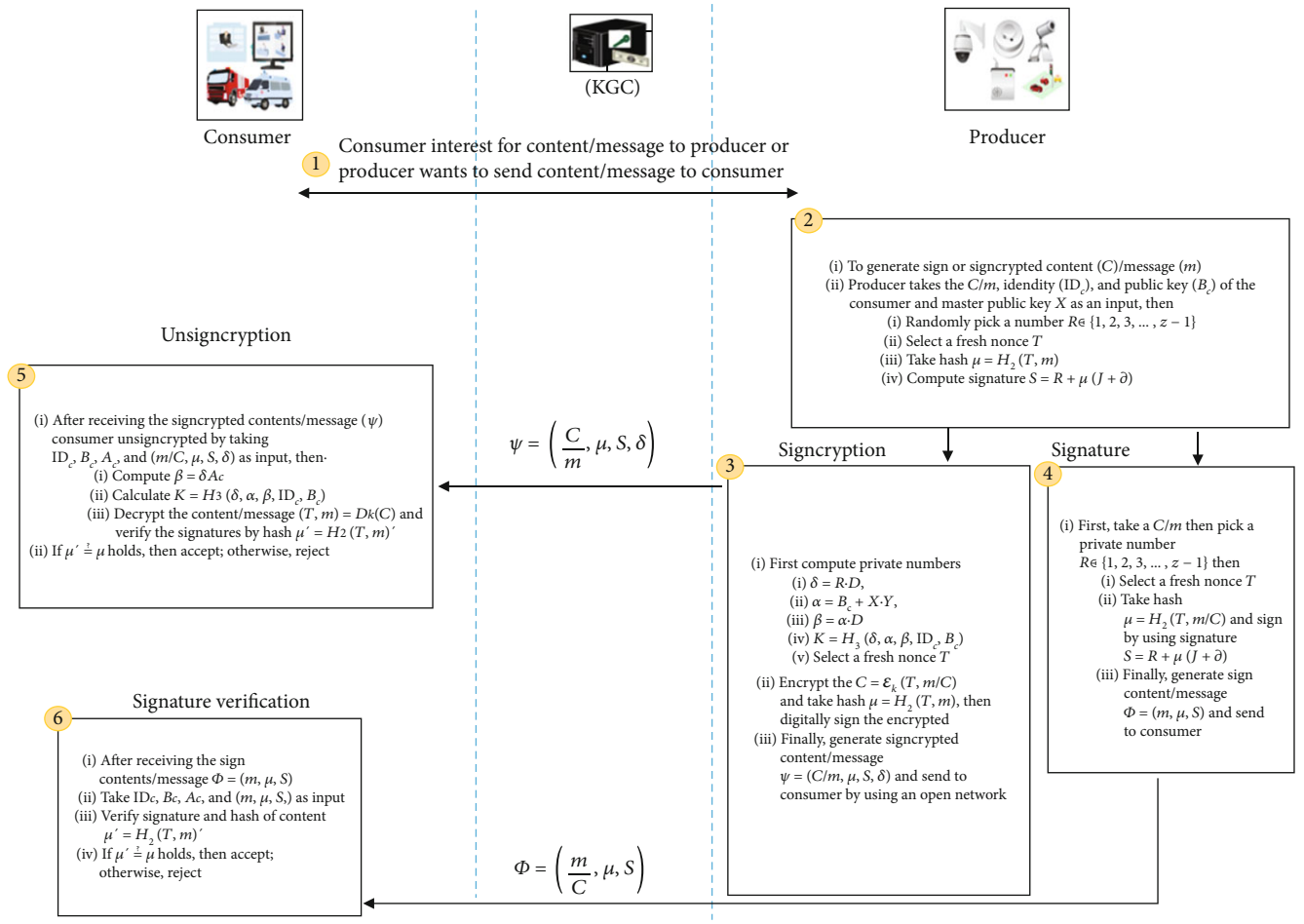

Figure 10: Communication process of the proposed scheme.

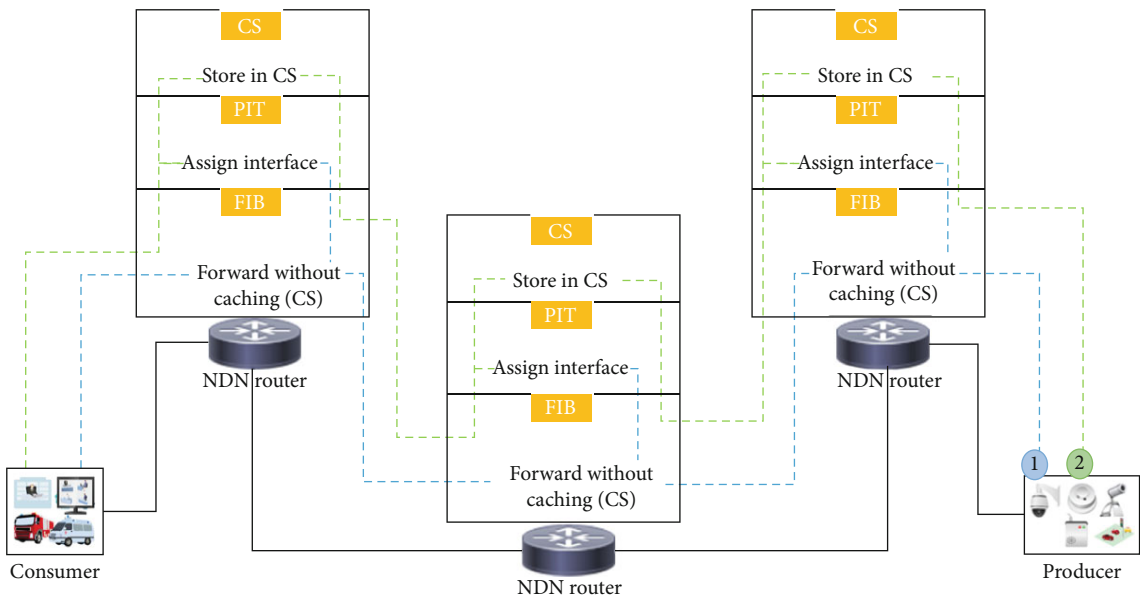

Figure 11: Workflow process of the proposed scheme in NDN architecture.

the consumer. For signed contents, the producer takes content (m) with randomly picked numbers from $\mathscr{R} \in \{1, 2, 3, \cdots, z-1\}$ and selects a fresh nonce $\mathscr{T}$, takes hash of $\mu = \mathscr{H}_2(\mathscr{T}, m)$, and applies signature $\mathscr{S} = \mathscr{R} + \mu(\mathscr{J} + \partial)$. Finally, it generates signed contents $\Phi = (m, \mu, \mathscr{S}, \delta)$ and sends it to the consumer.

After receiving the signcrypted contents Ψ, the consumer unsigncrypts the contents by taking $(ID_c, \mathscr{B}_c, \mathscr{A}_c, \mathscr{C}, \mu, \mathscr{S},)$ as an input and computing $\beta = \delta \cdot \mathscr{A}_c$, calculates the hash of signature $K = \mathscr{H}_3(\delta, \alpha, \beta, ID_c, \mathscr{B}_c)$, decrypts the content $(\mathscr{T}, m) = D_K(\mathscr{C})$, and computes the hash of the content $\mu' = \mathscr{H}_2(\mathscr{T}, m)'$; if $\mu' \stackrel{?}{=} \mu$ holds, then the contents are accepted; otherwise, they are rejected. In the case of signed contents Φ, the consumer takes $(ID_c, \mathscr{B}_c, \mathscr{A}_c, \mathscr{C}, \mu, \mathscr{S}, \delta)$ as input and calculates hash $\mu' = \mathscr{H}_2(\mathscr{T}, m)'$; if $\mu' \stackrel{?}{=} \mu$ holds, then the contents are accepted; otherwise, they are rejected.

5.3. The Workflow in NDN Architecture. NDN provides in-network caching, which means that the router of NDN will store and forward every message. Here, we divide the overall scenario into two types such as emergency situation and routine-based situation. In case of an emergency situation (fire, leakage of water, vehicle accident, etc.) that requires signcryption (confidentiality and authentication) for successful delivery to the intended destination in run time, the signcryption algorithm will execute and the NDN routers must not store these messages in the CS as shown in step 1 (Figure 11). The storage of emergency messages in CS does not facilitate any consumer later with the expense of latency.

In the routine-based situation, some parameters like, e.g., temperature, humidity, energy consumption, and video streaming, require authentication only and facilitate a number of consumers at a time. For this type of situation, the signature algorithm will execute and the NDN routers will store the copy of these contents/messages in its CS for future use as shown in step 2.

6. Conclusion

In this paper, we introduce the concept of lightweight in a natural heterogeneous generalized signcryption for the NDN-based Internet of Things (IoT). The proposed scheme provides the security properties of unforgeability, confidentiality, forward secrecy, and antireplay attack. We did the computation and communication cost comparisons with existing schemes, and the results give a satisfactory output due to the use of the hyperelliptic curve. So, our scheme reduced the computation cost of certificateless generalized signcryption (CGS) schemes from 78.09 to 97.23% and the communication from 21.73 to 91.89%. Furthermore, our scheme reduced the computation cost of identity-based generalized signcryption (ID-BGS) schemes from 95.70 to 97.84% and the communication cost from 54.05 to 72.13%. In addition, we practically deployed our scheme in the NDN-based smart city. Additionally, the scheme is validated through a security verification tool called AVISPA. The simulation results show that our scheme is valid and safe under the back-end protocols (OFMC, ATSE) of AVISPA.

Appendix

In this section, we discuss the simulation and validation of our proposed scheme in AVISPA. The simulation tool, code, and results are shown in the subsection below.

A. Automated Validation of Internet Security Protocols and Applications (AVISPA)

Automated Validation of Internet Security Protocols and Applications (AVISPA) is a security simulation tool used to check the validity of cryptographic schemes [39]. The AVISPA tools work under two states such as "safe" if the scheme resists against security threats and "unsafe" if the scheme cannot resist against security threats. AVISPA uses a role-oriented language called a high-level protocol specification language (HLPSL) for a specification of a cryptographic scheme. For checking the security, the user needs to convert the pseudocode of the proposed algorithm into the HLPSL. Then, the HLPSL2IF translator translates it to the intermediate format (IF). HLPSL2IF then verifies the security of the proposed scheme under four back-end tools called on-the-fly model checker (OFMC), CL-based attack searcher (CL-AtSe), SAT-based model checker (SATMC), and tree-automata-based protocol analyzer (TA4SP). According to the requirement of the scheme, each backed tool has its own functionality as further discussed in [40, 41], as shown in Figure 12.

B. Simulation Code

Here, we divide the simulation code according to the entities that participate in our scheme such as the producer and consumer. Note: for simulation of the proposed algorithm, the pseudocode of the proposed algorithm needs to be changed for the HLPSL library. Moreover, the notation used in the proposed algorithm is different as compared to the notation used in HLPSL. Further, the simulation code is shown in Pseudocodes B.1 and B.2.

C. Simulation Results

This section contains the simulation results of the proposed scheme according to the back-end protocols of the AVISPA tool such as OFMC and ATSE.

C.1. OFMC. The results of the proposed scheme after applying the OFMC protocol show that our scheme is safe against malicious attacks as shown in Figure 13.

C.2. ATSE. The results of the proposed scheme after applying the ATSE protocol show that our scheme is safe against malicious attacks as shown in Figure 14.

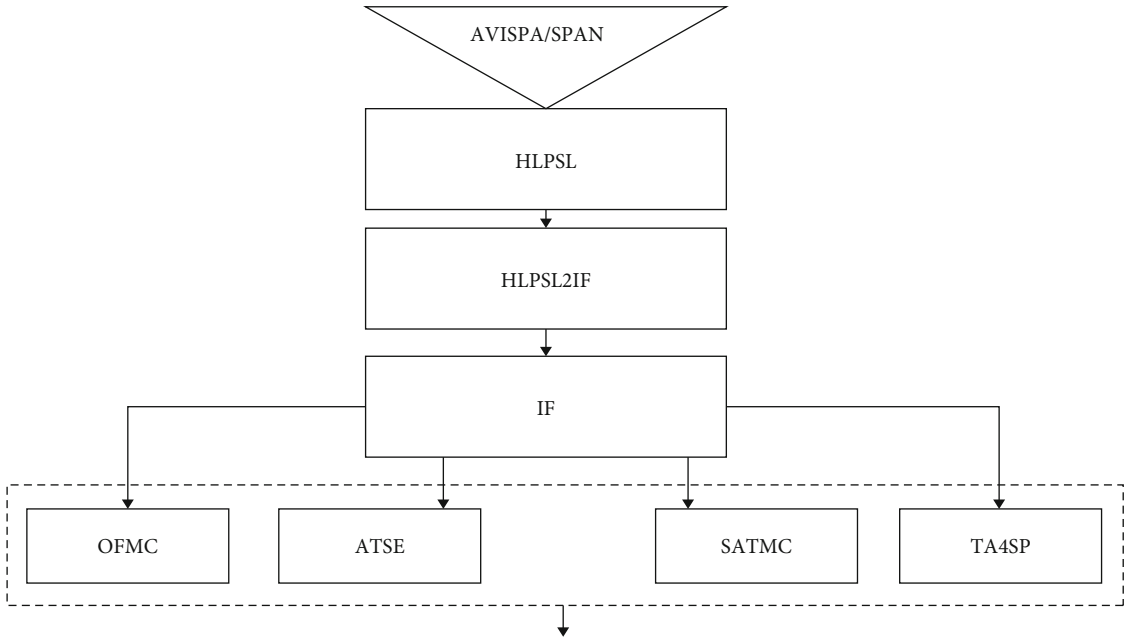

Figure 12: AVISPA workflow.

```
role role_Producer(Producer:agent, Consumer:agent, Bp:public_key, Bc:public_key, SND, RCV:channel(dy))
played_by Producer
def=
  local
    State:nat, T:text, Plus:hash_func, R:text, U:text, Mmm:text, Encryptionnn:hash_func, Kk:symmetric_key
  init
    State := 0
  transition
    1. State=0 /\ RCV(start) = |> State':=1 /\ SND(Producer.Consumer)
    2. State=1 /\ RCV(Consumer.{T'}_Bc) = |> State':=2 /\ U':=new() /\ R':=new() /\ Kk':=new() /\ Mmm':=new() /\ SND(Produ-
cer.{Encryptionnn(Mmm')}_Kk'.{Plus(R'.U')}_inv(Bp))
end role
```

Pseudocode B.1: HLPSL code for producer role.

```
role role_Consumer(Producer:agent, Consumer:agent, Bp:public_key, Bc:public_key, SND, RCV:channel(dy))
played_by Consumer
def=
  local
  State:nat, T:text, Plus:hash_func, R:text, U:text, Mmm:text, Encryptionnn:hash_func, Kk:symmetric_key
  init
    State := 0
  transition
    1. State=0 /\ RCV(Producer.Consumer) = |> State':=1 /\ T':=new() /\ SND(Consumer.{T'}_Bc)
    6. State=1 /\ RCV(Producer.{Encryptionnn(Mmm')}_Kk'.{Plus(R'.U')}_inv(Bp)) = |> State':=2
end role
```

Pseudocode B.2: HLPSL code for consumer role.

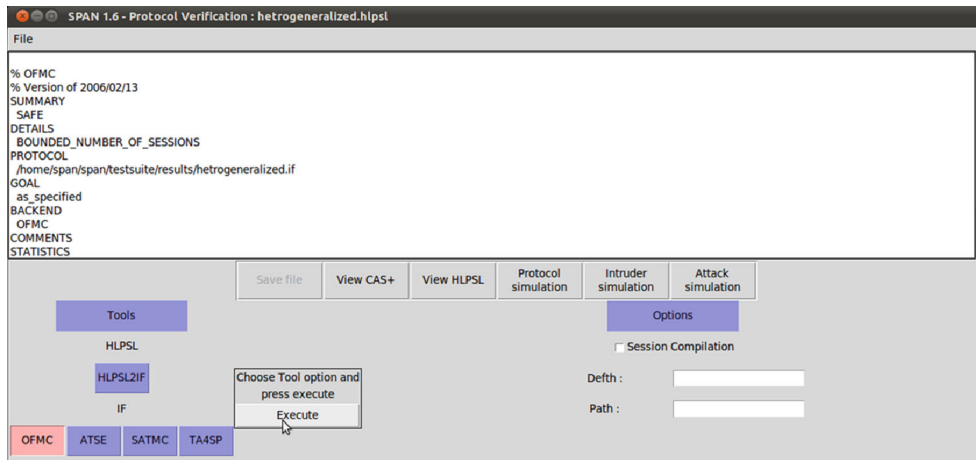

Figure 13: OFMC protocol result of proposed scheme.

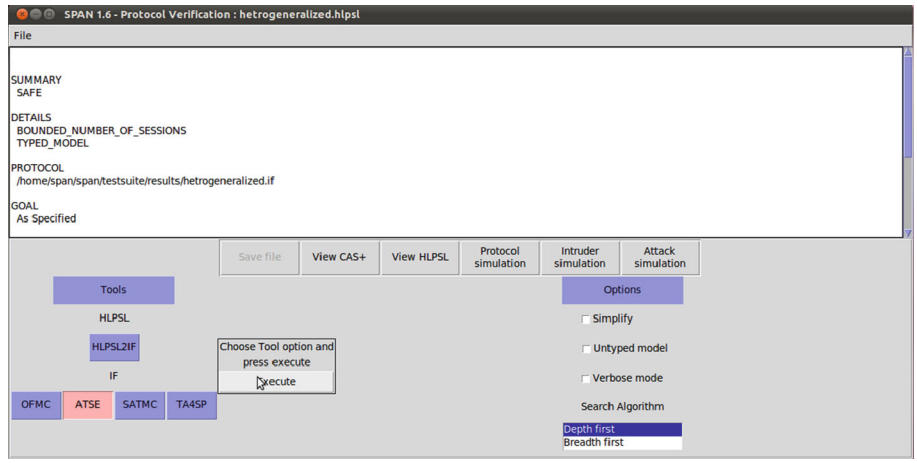

Figure 14: ATSE protocol result of proposed scheme.

Data Availability

All data generated or analyzed during this study are included in this published article.

Conflicts of Interest

The authors declare no conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

[1] D. Mars, S. Mettali Gammar, A. Lahmadi, and L. Azouz Saidane, "Using information centric networking in internet of things: a survey," *Wireless Personal Communications*, vol. 105, no. 1, pp. 87–103, 2019.

[2] A. Khanna and S. Kaur, "Evolution of internet of things (IoT) and its significant impact in the field of precision agriculture," *Computers and electronics in agriculture*, vol. 157, pp. 218–231, 2019.

[3] V. Jacobson, D. K. Smetters, J. D. Thornton, and M. F. Plass, "Networking named content," in *CoNEXT '09: Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pp. 1–12, Rome, Italy, 2009.

[4] C. Fang, F. Yu, T. Huang, J. Liu, and Y. Liu, "A survey of energy-efficient caching in information-centric networking," *IEEE Communications Magazine*, vol. 52, no. 11, pp. 122–129, 2014.

[5] M. Amadeo, G. Ruggeri, C. Campolo, and A. Molinaro, "IoT services allocation at the edge via named data networking: from optimal bounds to practical design," *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 661–674, 2019.

[6] C. Fang, F. R. Yu, T. Huang, J. Liu, and Y. Liu, "A survey of green information-centric networking: research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1455–1472, 2015.

[7] Q. Huang, D. S. Wong, and G. Yang, "Heterogeneous signcryption with key privacy," *The Computer Journal*, vol. 54, no. 4, pp. 525–536, 2011.

[8] Y. Han, X. Yang, P. Wei, Y. Wang, and Y. Hu, "ECGSC: elliptic curve based generalized signcryption," in *Ubiquitous Intelligence and Computing*, pp. 956–965, Springer Berlin Heidelberg, 2006.

[9] M. Suárez-Albela, P. Fraga-Lamas, and T. Fernández-Caramés, "A practical evaluation on RSA and ECC-based cipher suites for IoT high-security energy-efficient fog and mist computing devices," *Sensors*, vol. 18, p. 3868, 2018.

[10] M. Yu, J. Zhang, J. Wang et al., "Internet of things security and privacy-preserving method through nodes differentiation, concrete cluster centers, multi-signature, and blockchain," *International Journal of Distributed Sensor Networks*, vol. 14, 2018.

[11] A. Braeken, "PUF based authentication protocol for IoT," *Symmetry*, vol. 10, no. 8, 2018.

[12] I. Ullah, N. Ul Amin, M. Zareei et al., "A lightweight and provable secured certificateless signcryption approach for crowdsourced IIoT applications," *Symmetry*, vol. 11, p. 1386, 2019.

[13] S. Kumari, M. Karuppiah, A. K. Das, X. Li, F. Wu, and N. Kumar, "A secure authentication scheme based on elliptic curve cryptography for IoT and cloud servers," *The Journal of Supercomputing*, vol. 74, no. 12, pp. 6428–6453, 2018.

[14] Z. Ullah, A. Zeb, I. Ullah et al., "Certificateless proxy reencryption scheme (CPRES) based on hyperelliptic curve for access control in content-centric network (CCN)," *Mobile Information Systems*, vol. 2020, Article ID 4138516, p. 13, 2020.

[15] C. Tamizhselvan and V. Vijayalakshmi, "An energy efficient secure distributed naming service for IoT," *International Journal of Advanced Studies of Scientific Research*, vol. 3, no. 8, 2019.

[16] V. S. Naresh, R. Sivaranjani, and N. V. E. S. Murthy, "Provable secure lightweight hyper elliptic curve-based communication system for wireless sensor network," *International Journal of Communication Systems*, vol. 31, no. 15, article e3763, 2018.

[17] A. Rahman, I. Ullah, M. Naeem, R. Anwar, H. S. Khaǰak, and A. Ullah, "Lightweight multi-message and multi-receiver heterogeneous hybrid signcryption scheme based on hyper elliptic curve," *International Journal of Advanced Computer Science and Applications*, vol. 9, p. 5, 2018.

[18] S. Lal and P. Kushwah, "ID based generalized signcryption," Cryptology ePrint Archive, Report, 2008, http://eprint.iacr.org.

[19] W. Liang, Z. Chuan-Rong, and L.-Q. Zheng, "A key management scheme based generalized Signcryption in mobile ad hoc network," in *2010 International Conference on Communications and Intelligence Information Security*, Nanning, China, 2010.

[20] P. Kushwah and S. Lal, "An efficient identity based generalized signcryption scheme," *Theoretical Computer Science*, vol. 412, no. 45, pp. 6382–6389, 2011.

[21] G. Wei, J. Shao, Y. Xiang, P. Zhu, and R. Lu, "Obtain confidentiality or/and authenticity in big data by ID-based generalized signcryption," *Information Sciences*, vol. 318, pp. 111–122, 2015.

[22] D. Mishra and S. Singh, "A survey on ID based and certificateless generalized signcryption scheme," *International Journal of Innovative Research in Advanced Engineering*, vol. 2, no. 11, 2014.

[23] X. Shen, Y. Ming, and J. Feng, "Identity based generalized signcryption scheme in the standard model," *Entropy*, vol. 19, no. 3, p. 121, 2017.

[24] A. Waheed, A. I. Umar, N. Din, N. U. Amin, S. Abdullah, and P. Kumam, "Cryptanalysis of an authentication scheme using an identity based generalized signcryption," *Mathematics*, vol. 7, no. 9, p. 782, 2019.

[25] J. Huifang, H. Wenbao, and Z. Long, "Certificateless generalized signcryption," Cryptology ePrint Archive, Report, 2010, http://eprint.iacr.org.

[26] P. Kushwah and S. Lal, "Provable secure certificateless generalized signcryption scheme," *Technology & Applications*, vol. 3, pp. 925–939, 2012.

[27] C. Zhou, W. Zhou, and X. Dong, "Provable certificateless generalized signcryption scheme," *Designs, Codes and Cryptography*, vol. 71, no. 2, pp. 331–346, 2014.

[28] A. Zhang, L. Wang, X. Ye, and X. Lin, "Light-weight and robust security-aware D2D-assist data transmission protocol for mobile-health systems," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 3, pp. 662–675, 2016.

[29] M. A. Khan, I. Ullah, S. Nisar et al., "An efficient and provably secure certificateless key-encapsulated signcryption scheme for flying ad-hoc network," *IEEE Access*, vol. 8, pp. 36807–36828, 2020.

[30] B. Zhang, Z. Jia, and C. Zhao, "An efficient certificateless generalized signcryption scheme," *Security and Communication Networks*, vol. 2018, Article ID 3578942, 2018.

[31] C. Zhou, "An improved lightweight certificateless generalized signcryption scheme for mobile-health system," *International Journal of Distributed Sensor Networks*, vol. 15, no. 1, 2019.

[32] A. Waheed, J. Iqbal, N. Din, S. Ul, A. Iqbal, and N. Ul, "Improved cryptanalysis of provable certificateless generalized signcryption," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 4, 2019.

[33] A. Karati, C. I. Fan, and R. H. Hsu, "Provably secure and generalized Signcryption with public verifiability for secure data transmission between resource-constrained IoT devices," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10431–10440, 2019.

[34] Y. Li, C. Wang, Y. Zhang, and S. Niu, "Privacy-preserving multi-receiver signcryption scheme for heterogeneous systems," *Security and Communication Networks*, vol. 9, no. 17, 4584 pages, 2016.

[35] S. Raveendranath and A. Aneesh, "Efficient multi-receiver heterogenous signcryption," in *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 1693–1697, Chennai, India, 2016.

[36] S. Niu, Z. Li, M. Tian, C. Wang, and X. Jia, "An efficient heterogeneous signcryption scheme from certificateless to identity-based cryptosystem," *MATEC Web of Conferences*, vol. 139, article 00037, 2017.

[37] S. Niu, L. Niu, X. Yang, C. Wang, and X. Jia, "Heterogeneous hybrid signcryption for multi-message and multi-receiver," *PloS One*, vol. 12, no. 9, article e0184407, 2017.

[38] Y. Li, Y. Qi, and L. Lu, "Secure and efficient V2V communications for heterogeneous vehicle ad hoc networks," in *2017 International Conference on Networking and Network Applications (NaNA)*, pp. 93–99, Kathmandu, Nepal, 2017.

[39] S. Niu, Z. Li, and C. Wang, "Privacy-preserving multi-party aggregate signcryption for heterogeneous systems," in *International Conference on Cloud Computing and Security*, pp. 216–229, Nanjing, China, 2017.

[40] M. E. Saeed, Q. Liu, G. Tian, B. Gao, and F. Li, "HOOSC: heterogeneous online/offline signcryption for the internet of things," *Wireless Networks*, vol. 24, no. 8, pp. 3141–3160, 2018.

[41] C. Wang, C. Liu, Y. Li, H. Qiao, and L. Chen, "Multi-message and multi-receiver heterogeneous signcryption scheme for ad-hoc networks," *Information Security Journal: A Global Perspective*, vol. 26, no. 3, pp. 136–152, 2017.

[42] C. Jin, G. Chen, C. Yu, J. Shan, J. Zhao, and Y. Jin, "An efficient heterogeneous signcryption for smart grid," *PloS One*, vol. 13, no. 12, article e0208311, 2018.

[43] J. Liu, L. Zhang, R. Sun, X. Du, and M. Guizani, "Mutual heterogeneous signcryption schemes for 5G network slicings," *IEEE Access*, vol. 6, pp. 7854–7863, 2018.

[44] X. Liu and W. Ma, "CDAKA: a provably-secure heterogeneous cross-domain authenticated key agreement protocol with symptoms-matching in TMIS," *Journal of Medical Systems*, vol. 42, no. 8, 2018.

[45] A. A. Omala, A. S. Mbandu, K. D. Mutiria, C. Jin, and F. Li, "Provably secure heterogeneous access control scheme for wireless body area network," *Journal of Medical Systems*, vol. 42, no. 6, 2018.

[46] F. Zhou, Y. Li, and Y. Ding, "Practical V2I secure communication schemes for heterogeneous VANETs," *Applied Sciences*, vol. 9, no. 15, 2019.

[47] I. Ullah, N. U. Amin, M. Naeem, S. J. KhaJak, and H. Ali, "A novel provable secured signcryption scheme????: A hyperelliptic curve-based approach," *Mathematics*, vol. 7, no. 8, p. 686, 2019.

[48] S. Ullah, X.-Y. Li, and L. Zhang, "A Review of signcryption schemes based on hyper elliptic curve," in *2017 3rd International Conference on Big Data Computing and Communications (BIGCOM)*, pp. 10-11, Chengdu, China, 2017.

[49] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[50] C. Zhou, Z. Zhao, W. Zhou, and Y. Mei, "Certificateless key-insulated generalized signcryption scheme without bilinear pairings," *Security and Communication Networks*, vol. 2017, Article ID 8405879, 2017.

[51] S. S. Ullah, I. Ullah, H. Khattak et al., "A lightweight identity-based signature scheme for mitigation of content poisoning attack in named data networking with internet of things," *IEEE Access*, vol. 8, pp. 98910–98928, 2020.

[52] S. Hussain, I. Ullah, H. Khattak et al., "A lightweight and formally secure certificate based Signcryption with proxy re-encryption (CBSRE) for internet of things enabled smart grid," *IEEE Access*, vol. 8, pp. 93230–93248, 2020.