Computers, Materials & Continua
DOI:10.32604/cmc.2022.017380
Article

Tech Science Press

An Access Control Scheme Using Heterogeneous Signcryption for IoT Environments

Insaf Ullah[1,*], Hira Zahid[2], Fahad Algarni[3] and Muhammad Asghar Khan[1]

[1]Hamdard Institute of Engineering and Technology, Islamabad, 44000, Pakistan
[2]Department of Information Technology, Abbottabad University of Science and Technology, Abbottabad, Pakistan
[3]College of Computing and Information Technology, The University of Bisha, Bisha, Saudi Arabia
[*]Corresponding Author: Insaf Ullah. Email: insafktk@gmail.com
Received: 29 January 2021; Accepted: 06 May 2021

Abstract: When the Wireless Sensor Network (WSN) is combined with the Internet of Things (IoT), it can be employed in a wide range of applications, such as agriculture, industry 4.0, health care, smart homes, among others. Accessing the big data generated by these applications in Cloud Servers (CSs), requires higher levels of authenticity and confidentiality during communication conducted through the Internet. Signcryption is one of the most promising approaches nowadays for overcoming such obstacles, due to its combined nature, i.e., signature and encryption. A number of researchers have developed schemes to address issues related to access control in the IoT literature, however, the majority of these schemes are based on homogeneous nature. This will be neither adequate nor practical for heterogeneous IoT environments. In addition, these schemes are based on bilinear pairing and elliptic curve cryptography, which further requires additional processing time and more communication overheads that is inappropriate for real-time communication. Consequently, this paper aims to solve the above-discussed issues, we proposed an access control scheme for IoT environments using heterogeneous signcryption scheme with the efficiency and security hardiness of hyperelliptic curve. Besides the security services such as replay attack prevention, confidentiality, integrity, unforgeability, non-repudiations, and forward secrecy, the proposed scheme has very low computational and communication costs, when it is compared to existing schemes. This is primarily because of hyperelliptic curve lighter nature of key and other parameters. The AVISPA tool is used to simulate the security requirements of our proposed scheme and the results were under two backbends (Constraint Logic-based Attack Searcher (CL-b-AtSER) and On-the-Fly Model Checker (ON-t-FL-MCR)) proved to be SAFE when the presented scheme is coded in HLPSL language. This scheme was proven to be capable of preventing a variety of attacks,

including confidentiality, integrity, unforgeability, non-repudiation, forward secrecy, and replay attacks.

Keywords: Internet of Things (IoT); access control; big data; heterogeneous signcryption

1 Introduction

The Internet of Things (IoT) represents a system of interconnected objects/things and devices that communicate through the Internet in a continuous manner [1–3]. The notion "things" in this context may refer to any virtual or physical object that can be assigned a unique identity, such as an internet protocol (IP) address or an identity number (ID). Most of these services are equipped with sensors to enable a dynamic communication of information and events [4]. So, the IoT in basic terms can be expressed as a roadmap of things. The majority of IoT devices are referred to as smart because of its ability to communicate data received from their surroundings without the need for human intervention [5]. Besides that, when looking at networks, we find out that people have already witnessed connecting objects or devices through wires, often known as cabled or wired connections, then wireless sensor networks have also been implemented (connected through wireless protocols) [6]. The mobile internet has encountered exponential growth multiple times since the establishment of Wireless Sensor Networks (WSN) and has become the backbone of information networks connecting human society [7]. As a result, it is apparent that WSN is associated with IoT due to certain unique features and functionalities [8].

Prior studies within this field have noted the importance of security as a crucial requirement for IoT communications [9], with an increased emphasis on cryptography, which is described as "the practice and analysis of techniques for secure data communication while being transmitted through networks." There are three main techniques associated with cryptography. These are public key infrastructure (PKI), certificateless cryptosystem (CLC), and identity-based cryptosystem (IBC). The first technique in cryptography types is PKI based method. The most prominent limitation in PKI is its unsuitable traditional implementation in IoT. The projecting factor to this is the certificate management overhead i.e., storage, distribution, and revocation of certificates [10]. The second technique in cryptography types is IBC [11], which was introduced to reduce the burden on traditional PKI. IBC recommends using a publicly recognized string as a public key, which reduces the cost of PKI certificate renewal. The IBC, being Identity-Based, appeared to be more vulnerable to third party hacker attacks (key escrow problem). This is classified as a major obstacle that needs to be tackled [12]. To solve this issue, the third technique of cryptography, called CLC, was developed with certificate-less based cryptography [13]. CLC is a form of ID-based cryptography that addresses the problem of key escrow. The key generation center (KGC) creates a partial private key for users and distributes it over a secure network. The user will then create his/her private and public keys using the partial private key obtained and some randomly generated numbers. All of the above-mentioned debates used homogeneous cryptography, which meant that the sender and receiver shared the same security domain, making the network more vulnerable. The vulnerability necessitates the use of a heterogeneous signcryption scheme, in which the sender and receiver have separate security domains, thus protecting the network from different cyber or intruder attacks [7]. Combining both PKI and CLC techniques is required to generate heterogeneous signcryption keys. The advantage of combining CLC and PKI is that it protects the network from intruders by only disclosing the original keys to the sender and receivers.

In addition, previous studies of access control for IoT environments have developed various schemes that encountered the mutual shortcoming in terms of their roots as mathematical algorithms, their massive costs and huge computations. Bilinear pairing method is the first algorithm that has contributed significantly in this context [14], which experiences huge pairing and RSA ("Rivest-Shamir-Adleman). The Bilinear Pairing method appeared to be worse than RSA since it requires large pairing computations and passes through a map-to-function calculation [14]. In order to address the mutual inefficiencies in both RSA and Bilinear, a recent approach called "Elliptic Curve Cryptography," or "ECC," was developed [15–17]. The most distinguished attributes of ECC seem to be its small size of parameter, private key, identity, public key and certificate. The inflexibility and efficiency of security in ECC is based on small key size of 160 bits [18]. For devices that highly require resources, the 160 bit key-size of ECC is insufficient, as it was not suitable and affordable. To address this issue, we propose a new method called "Hyper Elliptic Curve Cryptography," or "HECC," which is a generalized form of ECC. It provides the same security level as RSA, Bilinear, and ECC, but with a smaller key, identity and certificate size of just 80 bits [19]. For energy-constrained devices, HECC is proved to be the most appropriate, cost-effective, and efficient scheme. As a result, we have incorporated the following new features to this paper:

- We designed a heterogeneous signcryption (Users belongs to CLC and the sensor nodes uses the concept of IBC) based on Hyper elliptic curve.
- The new scheme assures that the security properties of Replay Attack, confidentiality, integrity, Unforgeability, Non-repudiations, and forward secrecy, respectively.
- The AVISPA Tool is used to simulate the security requirements of the proposed scheme and the result under two backbends (Constraint Logic-based Attack Searcher (CL-b-AtSER) and On-the-Fly Model Checker (ON-t-FL-MCR)) are SAFE when the proposed scheme is coded in HLPSL language.
- By applying the concept of hyper elliptic curve, this scheme will significantly reduce the computational cost timing and require smaller amount of bits for communication.

The paper is organized as follows: Section 1 contains a brief introduction, Section 2 encompasses the advantages and disadvantages of related work, Section 3 includes the syntax of heterogeneous signcryption, Section 4 represents the network model, Section 5 comprises the proposed heterogeneous signcryption for IoT, Section 6 covers the security analysis, Section 7 covers the computational cost, and Section 8 involves the communication cost, Section 9 includes scheme simulation, and Section 10 presents the conclusion.

2 Literature Review

Recently, access control techniques for IoT environments have attracted a considerable amount of scholars due to its vital roles in achieving higher levels of security. Li et al. [20], have developed a new concept about an access control strategy for IoT environments. The study incorporated the heterogeneous signcryption (e.g., the sender belongs to CLC and the receiver uses the concept of IBC) on the basis of bilinear pairing cryptosystem. However, since bilinear pairing requires additional resources, this scheme must be slower in terms of computational time and communication delay time. Challa et al. [21], proposed an ECC based scheme to provide an access control mechanism to contemporary IoT environments. Then, Chaudhry et al. [22], claimed that the Challa et al. scheme has higher correctness rates and capable of address certain issues. After that, Luo et al. [8], developed a new scheme using signcryption in heterogeneous nature (e.g., the sender belongs to CLC and the receiver uses the concept of IBC). However, due to more

resources demanding nature of bilinear pairing, the presented scheme suffers from the issur of slow computational time and communication delays. Das et al. [4] designed a new approach for device-to-device access control in IoT on the bases of ECC. Nevertheless, Chaudhry et al. [23], proved that Das et al. scheme was vulnerable to impersonation and man-in-middle attacks. The Authors then proposed a new scheme to address such issues. Malani et al. [24], offered an anonymous scheme which provide access control policy for IoT devices. ECC is also used in this scheme. As a result of ECC's higher resource requirements, the proposed schemes in [4,21,23,24] must be slower in computational time and communication delay time, and are not suitable for heterogeneous IoT environments, because they used the same nature cryptography for sender and receiver, which can be vulnerable at certain times. As a result, providing a heterogeneous access control scheme based on heterogeneous signcryption has become vital (e.g., the sender belongs to CLC and the receiver uses the concept of PKI) using the difficult problem of a hyper elliptic curve, that requires smaller keys and parameters. As a result, such a scheme is expected to achieve higher levels of security for IoT environments.

3 Syntax of Heterogeneous Signcryption

Heterogeneous signcryption contains the steps such as Setup, PKI Key Generation, Certificateless (CL) Key Generation (CLKG), CL-Partial Private Key Processing (CL-PPKG), CL-Secret Value Selection (CL-SVS), CL- Private Key Processing (CL-PKG), CL-Public Key Processing (CL-PBKG), CL-Signcrypt (CL-SCT), and CL-Un-Signcrypt (CL-Un-SCT), respectively. The definition of each step is explained in the following sections.

3.1 Setup

Given J as a security parameter, the application provider (AP), first choose ζ as his secret key and make his public key as δ. Then, it makes β as a public parameter param and keeps secret ζ then published β.

3.2 PKI Key Generation

A receiver with PKI picks a private key R^{pr} with a random manner and calculates his/her public key as R^{pb}.

3.3 Certificateless (CL) Key Generation (CLKG)

3.3.1 CL-Partial Private Key Processing (CL-PPKG)

The application provider (AP) picks a random number Φ and generates X, η, and $\mathcal{W}$. It sets $\mathcal{W}$ is a partial private key and send the tuple (X, $\mathcal{W}$) via a secure channel to sender.

3.3.2 CL-Secret Value Selection (CL-SVS)

The sender picks a random number ω^s and set ω^s is a secret value.

3.3.3 CL- Private Key Processing (CL-PKG)

The sender makes his private key like that $S^{pr} = (\mathcal{W}, \omega^s)$.

3.3.4 CL-Public Key Processing (CL-PBKG)

The sender makes his public key like that $S^{pb} = (\mathrm{X}, \mu^s)$.

3.3.5 CL-Signcrypt (CL-SCT)

By using the message (M), R^{pb}, and S^{pr} as an input, the sender can make and send ψ to the receiver.

3.3.6 CL-Un-Signcrypt (CL-Un-SCT)

By using ψ, S^{pb}, and R^{pr} as an input, the receiver can verify ψ that it is either valid or not.

4 Network Model

Fig. 1 illustrates our new model for access control of wireless sensor network within the IOT environments utilizing heterogeneous signcryption (Certificateless to PKI). Itcontains six participants, named, the internet users, cloud server, network manager, sensor nodes, the Internet, and controller, respectively. When users require data from sensor nodes, they send their identity to the network manager, who then generates a partial private key for them and transfer it back to them through a secure network. After that, using the concept of a certificateless based Cryptosystem, users perform the signcryption process on the data request query and transmit it to the controller through an open network. The controller first verifies the public key of the receiver from the network manager and then verifies the received signcrypted query by performing the unsigncryption process. Note that for the unsigncryption process the controller used the functionality of PKI. After verifying the signcrypted query, the controller collects the data from sensors and encrypt this data by using the "Advanced Encryption Standard (AES)" algorithm and transmits the encrypted data to the users. In this case, the cloud server is responsible for storing the vast amount of data generated by the relevant users.

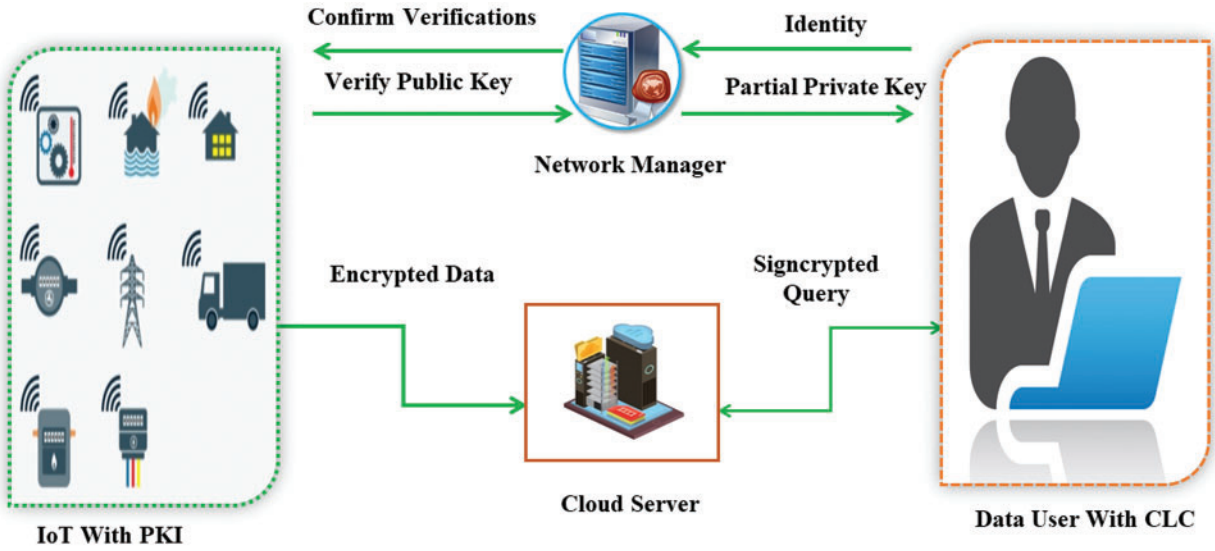

Figure 1: Proposed network model

5 Construction of Proposed Heterogeneous Signcryption for IoT

The explanation of each step-in construction of the proposed scheme is described in the following subsections.

5.1 Setup

Given J as a security parameter, the application provider (AP), first choose $\zeta \in \{1, 2, 3, \ldots, n-1\}$ his secret key and make his public key as $\delta = \zeta \cdot \mathcal{D}$. Then, it selects a triple $(\hbar_x, \hbar_y)$ as a hash function and set $\beta = (\hbar_x, \hbar_y, \hbar_y, \delta, \mathcal{D}, J, HEC)$ as a public parameter param. Then, AP keeps secret ζ and published β.

5.2 PKI Key Generation

A receiver with PKI pick a private key $R^{pr} \in \{1, 2, 3, \ldots, n-1\}$ with a random manner and calculates his/her public key as $R^{pb} = \dfrac{\mathcal{D}}{R^{pr}}$.

5.3 Certificateless (CL) Key Generation (CLKG)

It contains the following four steps:

5.3.1 CL-Partial Private Key Processing (CL-PPKG)

The application provider (AP) picks a random number $\Phi \in \{1, 2, 3, \ldots, n-1\}$ and make $X = \Phi \cdot \mathcal{D}$, $\eta = \hbar_x(id, \mathrm{X})$, and $\mathcal{W} = \Phi + \zeta\eta$. It sets $\mathcal{W}$ is a partial private key and send the tuple $(X, \mathcal{W})$ via a secure channel to sender.

5.3.2 CL-Secret Value Selection (CL-SVS)

The sender picks a random number $\omega^s \in \{1, 2, 3, \ldots, n-1\}$ and set ω^s is a secret value.

5.3.3 CL- Private Key Processing (CL-PKG)

The sender makes his private key like that $S^{pr} = (\mathcal{W}, \omega^s)$.

5.3.4 CL-Public Key Processing (CL-PBKG)

The sender makes his public key like that $S^{pb} = (\mathrm{X}, \mu^s = \omega^s \cdot \mathcal{D})$.

5.4 CL-Signcrypt (CL-SCT)

By using the message (M), R^{pb}, and S^{pr} as an input, the sender can do the following process:

a) It picks $\mathcal{U} \in \{1, 2, 3, \ldots, n-1\}$ uniformly
b) Compute $\mathcal{S} = \mathcal{U} \cdot \mathcal{D}$ and $\mathcal{Q} = r \cdot \mathcal{D}$, where $r = \hbar_y(\boldsymbol{M}, \mathcal{S})$
c) Calculate $\mathcal{Z} = \boldsymbol{M} \oplus \hbar_z(\mathcal{Q})$ and $\nabla = \dfrac{r - \mathcal{U}}{\mathcal{W} + \omega^s}$
d) Calculate $\mathcal{P} = \mathcal{U} \cdot R^{pb}$, set $\psi = (\mathcal{Z}, \nabla, \mathcal{P})$, and send ψ to the receiver.

5.5 CL-Un-Signcrypt (CL-Un-SCT)

By using ψ, S^{pb}, and R^{pr} as an input, the receiver can do the following process:

a) Calculate $\mathcal{S} = R^{pr} \cdot \mathcal{P}$ and $\mathcal{Q} = \mathcal{S} + \nabla(\mu^s + X + \eta \cdot \delta))$
b) Calculate $\mathcal{Z} = \boldsymbol{M} \oplus \hbar_z(\mathcal{Q})$ and $r = \hbar_y(\boldsymbol{M}, \mathcal{S})$

c) Accept only ψ, if $\mathcal{S} = r \cdot \mathcal{D} - \nabla(\mu^s + X + \eta \cdot \delta)$ otherwise display $\Vdash$.

5.6 *Security Analysis*

It contains the correctness and the descriptive analysis about replay attack, confidentiality, integrity, unforgeability, non-repudiations, and forward secrecy. Most of the security services are based on hyper elliptic curve discrete logarithm problem. Suppose a $\mathcal{D}$ is the devisor belonging to hyper elliptic curve (HEC) and σ is the point from prime field of 80 bits, so, finding σ from $\mathcal{F} = \sigma \cdot \mathcal{D}$ is called hyper elliptic curve discrete logarithm problem.

5.6.1 *Correctness*

The receiver first checks the correctness of $\mathcal{S} = R^{pr} \cdot \mathcal{P}$ as follows:

$$R^{pr} \cdot \mathcal{P} = \mathcal{S} = R^{pr} \cdot \left(\mathcal{U} \cdot \frac{\mathcal{D}}{R^{pr}}\right) = \mathcal{U} \cdot \mathcal{D} = \mathcal{S}$$

Then it checks the correctness of $\mathcal{Q} = \mathcal{S} + \nabla(\mu^s + X + \eta \cdot \delta))$ as follows:

$$\mathcal{S} + \nabla(\mu^s + X + \eta \cdot \delta) = \mathcal{U} \cdot \mathcal{D} + \left(\frac{r - \mathcal{U}}{\mathcal{W} + \omega^s}\right)(\mu^s + X + \eta \cdot \delta)$$

$$= \mathcal{U} \cdot \mathcal{D} + \left(\frac{r - \mathcal{U}}{\mathcal{W} + \omega^s}\right)(\omega^s \cdot \mathcal{D} + \Phi \cdot \mathcal{D} + \eta \cdot \zeta \cdot \mathcal{D})$$

$$= \mathcal{U} \cdot \mathcal{D} + \left(\frac{r - \mathcal{U}}{\mathcal{W} + \omega^s}\right)(\omega^s + \Phi + \eta \cdot \zeta) \cdot \mathcal{D} = \mathcal{U} \cdot \mathcal{D} + \left(\frac{r - \mathcal{U}}{\mathcal{W} + \omega^s}\right)(\omega^s + \mathcal{W}) \cdot \mathcal{D}$$

$$= \mathcal{U} \cdot \mathcal{D} + (r - \mathcal{U}) \cdot \mathcal{D} = (\mathcal{U} + (r - \mathcal{U})) \cdot \mathcal{D} = (r \cdot \mathcal{D}) = \mathcal{Q}$$

Finally it accepts only ψ, if $\mathcal{S} = r \cdot \mathcal{D} - \nabla(\mu^s + X + \eta \cdot \delta)$, the correctness as follows:

$$= r \cdot \mathcal{D} - \nabla(\mu^s + X + \eta \cdot \delta) = r \cdot \mathcal{D} - \left(\frac{r - \mathcal{U}}{\mathcal{W} + \omega^s}\right)(\mu^s + X + \eta \cdot \delta)$$

$$= r \cdot \mathcal{D} - \left(\frac{r - \mathcal{U}}{\mathcal{W} + \omega^s}\right)(\omega^s \cdot \mathcal{D} + \Phi \cdot \mathcal{D} + \eta \cdot \zeta \cdot \mathcal{D})$$

$$= r \cdot \mathcal{D} - \left(\frac{r - \mathcal{U}}{\mathcal{W} + \omega^s}\right)(\omega^s + \Phi + \eta \cdot \zeta) \cdot \mathcal{D}$$

$$= r \cdot \mathcal{D} - \left(\frac{r - \mathcal{U}}{\mathcal{W} + \omega^s}\right)(\omega^s + \mathcal{W}) \cdot \mathcal{D}$$

5.6.2 *Replay Attack*

A replay attack occurs when someone attempts to capture an old message and replay to it. In our scheme, a replay attack is impossible because we add a NC to the message prior to sending it. In this case, NC is included within the message. The receiver then can check whether a NC is new, thus, a replay attack is unachievable in our scheme.

5.6.3 Confidentiality

Confidentiality means no one can see the original contents of message other than sender and receiver. In our scheme, sender at the first step encrypts the message ($\mathcal{Z} = \mathcal{M} \oplus \hbar_z(\mathcal{Q})$) through secret key ($\mathcal{Q}$). The secret key is as follows in Eq. (1):

$$\mathcal{Q} = r \cdot \mathcal{D} \tag{1}$$

The attacker has to solve Eq. (1) in order to access the original contents. After solving this equation they have to solve for r because in this, r is private number and it will be calculated by the following Eq. (2):

$$r = \hbar_y(\mathcal{M}, \mathcal{S}) \tag{2}$$

To solve Eq. (2), the attacker has to generate a real value for r, which is not possible due to the one way nature of hash function. So, it is quite impossible for an attacker to solve this Eq. (1) because hyper elliptic curve discrete algorithms are required to be solved and this is infeasible for attacker. Hence it is proved that this scheme provides higher levels of confidentiality.

5.6.4 Integrity

Integrity means that the receiver receives the message in the same format which has been sent by the sender. In our scheme, before sending the data, sender calculates the hash function of the message is shown as $= \hbar_y(\mathcal{M}, \mathcal{S})$. Now, if the attacker wants to make any changes to the cipher text ($\mathcal{Z}$), he has to change the plane text ($\mathcal{M}$) as well but he will not be able to do so because he has to solve $r = \hbar_y(\mathcal{M}, \mathcal{S})$ for which he requires to compute $\mathcal{S} = \mathcal{U}.\mathcal{D}$ that was solvable only if it captured $\mathcal{U}$, which is not possible according to HECDLP. And overall, hash functions are irreversible and the attacker cannot generate the same equation again because the hash function produces new values each time it appears in a message and values are never repeated. As a result, our scheme demonstrates that it is provides the required integrity.

5.6.5 Unforgeability

Unforgeability means that no one else than the sender can generate the digital signature. In our scheme, a sender generates digital signature $\nabla = \left(\frac{r - \mathcal{U}}{\mathcal{W} + \omega^s}\right)$ using his three private numbers i.e., ($r, \mathcal{U}, \omega^s$). Now if the attacker wants to forge the signature. First of all he has to solve for r which is solved through $r = \hbar_y(\mathcal{M}, \mathcal{S})$, for which he requires to compute $\mathcal{S} = \mathcal{U} \cdot \mathcal{D}$ that was solvable only if it captured $\mathcal{U}$, which is not possible according to HECDLP. And overall, hash functions are irreversible and the attacker cannot generate the same equation again because the hash function produces new values each time it appears in a message and values are never repeated. Secondly, he has to solve for $\mathcal{U}$ which is solved through $\mathcal{S} = \mathcal{U} \cdot \mathcal{D}$ that was solvable only if it captured $\mathcal{U}$, which is not possible according to HECDLP. Thirdly, he has to solve for ω^s which is solved through $\mu^s = \omega^s \cdot \mathcal{D}$ that was solvable only if it captured ω^s, which is not possible according to HECDLP. Thus, making solution three times for HECDLP is infeasible, so, we claim that our scheme provides unforgeability.

5.6.6 Forward Secrecy

It means that in case if even the private key (ω^s) of sender gets compromised, still the messages the message ($\mathcal{Z} = \mathcal{M} \oplus \hbar_z(\mathcal{Q})$) of the sender remain confidential because sender uses session key ($\mathcal{Q} = r \cdot \mathcal{D}$) for the encryption and decryption. The attacker has to make value for

$\mathcal{Q}$ for accessing the message contents. After making $\mathcal{Q} = r \cdot \mathcal{D}$ they have to solve for r because in this, $r = h_y(\mathcal{M}, \mathcal{S})$ is private number for which attacker requires to compute $\mathcal{S} = \mathcal{U} \cdot \mathcal{D}$ that was solvable only if it captured $\mathcal{U}$, which is not possible according to HECDLP. And overall, hash functions are irreversible and the attacker cannot generate the same equation again because the hash function produces new values each time it appears in a message and values are never repeated. Therefore, our scheme confirmed that it provides forward secrecy.

5.6.7 Non-repudiation

Non-repudiation means that no one can deny something they said did or commit. In the context of our research, it means that the sender can not deny the signatures because he/she uses his/her private key (ω^s), and this is directly associated with the public key of the sender. If he/she denies this signature the network manager can prove it because it is only known by network manager. Hence, it is proved that our scheme also provides non-repudiation.

6 Cost Analysis

Before doing the comparison, one must remember that the computational costs are always the main concern for both the sender and receiver. Now in this case, the existing schemes used elliptic curve point multiplication and bilinear pairing.

6.1 Computational Cost

These have always been considered the costly options for measuring the computational costs. In our mechanism we are using hyper-elliptic curve divisor multiplication which is considered to be very cheaper than others in measuring computational costs. Tab. 1 shows the comparisons of computational cost of Li et al. [20], Challa et al. [21], Luo et al. [8], Das et al. [4], Chaudhry et al. [23], and Malani et al. [24] with the proposed scheme.

Table 1: Computational cost comparisons with the help of major operations and milli seconds

Access control schemes	Total operations	Total cost in milli seconds (ms)
Li et al. [20]	6T-Pair + 3T-P-M	6(11.9845) + 3(1.7090) = 77.034
Challa et al. [21]	14 T-E-M	14(0.0321) = 0.4494
Luo et al. [8]	5T-Pair + 3T-P-M	5(11.9845) + 3(1.7090) = 65.0495
Das et al. [4]	18T-E-M	18(0.0321) = 0.5136
Chaudhry et al. [23]	10T-E-M	10(0.0321) = 0.321
Malani et al. [24]	13T-E-M	13(0.0321) = 0.4173
Proposed	8T-D-M	8(0.01605) = 0.1284

According to the experimental results of [25], the following specifications were used to produce the experiments through a PC.

- Intel Core i7-7700 CPU@3.6 GHz2.0 GHz
- 8GB Random Access memory
- pairing-based cryptography library in VC++ 6.0

So, the single time Pairing Operation (T-Pair), time for multiplication in bilinear pairing (T-P-M), time for multiplication in ECC (T-E-M), are consumed 11.9845, 1.7090, and 0.0321

milliseconds (ms), respectively. Accordingly, the time for multiplication in HECC (T-D-M) will be the half of multiplication in ECC i.e., 0.01605 ms [26–28].

The Tab. 1 represents the major operations used in proposed and those Li et al. [20], Challa et al. [21], Luo et al. [8], Das et al. [4], Chaudhry et al. [23], and Malani et al. [24] as well as the total consumed time in ms. Then, we make Fig. 2 which clearly shows the superiority of our scheme in terms of computational cost.

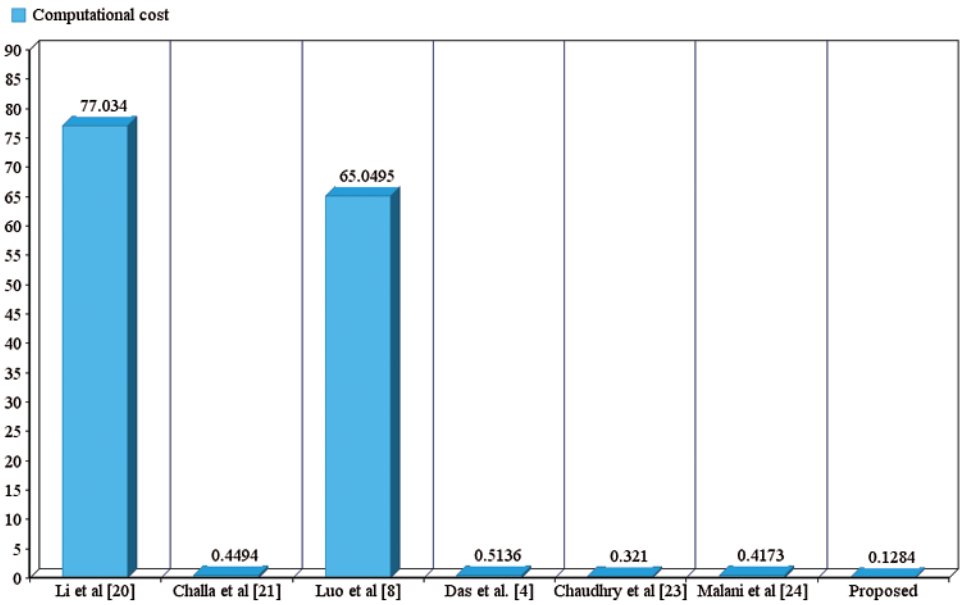

Figure 2: Computational cost comparisons with the help of major operations and milli seconds

6.2 Communication Cost

Here, we perform some computations in Tab. 2 regarding making of communication cost comparisons with existing ones that are Li et al. [20], Challa et al. [21], Luo et al. [8], Das et al. [4], Chaudhry et al. [23], and Malani et al. [24]. So, we suppose the following terms:

- |M| represents plaintext or cipher text size and equals to 60 bits
- |G| the group size of bilinear pairing and equals to 256 bits
- |Q| the size of ECC point and equals to 160 bits
- |N| the size of HECC devisor and equals to 80 bits
- |H| the size of hash value and equals to 512 bits
- |NON/T| the size of nonce or time stamp and equals to 80 bits in hyper elliptic curve environment and 80 bits in elliptic curve based environment
- |ID| represents the size of identity and equals to 80 bits in hyper elliptic curve environment and 160 bits in elliptic curve based environment
- |CERT| represents the size of certificate and equals to 80 bits in hyper elliptic curve environment and 160 bits in elliptic curve-based environment.

Finally, we created Fig. 3, which clearly demonstrates our scheme's superiority in terms of communication costs.

Table 2: Communication cost comparisons with the help of bits

Access control schemes	Communication cost	Communication cost in bits
Li et al. [20]	$\|M\| + 2\|G\|$	$\|60\| + 2\|256\| = 572$
Challa et al. [21]	$2\|M\| + 8\|Q\| + 4\|T\| + 1\|H\|$	$2\|60\| + 8\|160\| + 4\|160\| + 1\|512\| = 2552$
Luo et al. [8]	$\|M\| + 2\|G\|$	$\|60\| + 2\|256\| = 572$
Das et al. [4]	$10\|Q\| + 3\|T\| + 2\|H\| + 2\|ID\|$	$10\|160\| + 3\|160\| + 2\|512\| + 2\|160\| = 3224$
Chaudhry et al. [23]	$9\|Q\| + 2\|T\| + 1\|H\| + 2\|ID\|$	$9\|160\| + 2\|160\| + 1\|160\| + 2\|160\| = 1880$
Malani et al. [24]	$6\|Q\| + 2\|T\| + 1\|H\| + 2\|CERT\|$	$6\|160\| + 2\|160\| + 1\|512\| + 2\|160\| = 2112$
Proposed	$\|M\| + 2\|N\|$	$\|60\| + 2\|80\| = 220$

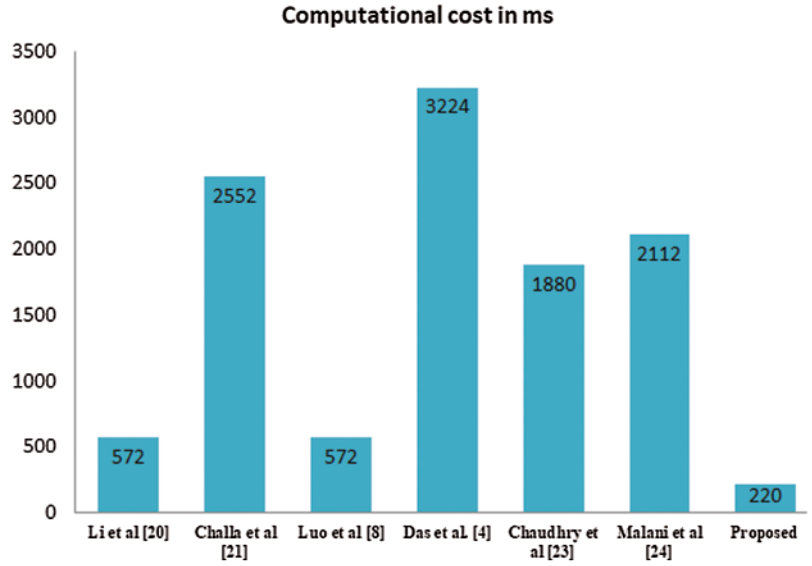

Figure 3: Communication cost comparisons with the help of bits

```
role
role_Sender(Sender:agent,Reciever:agent,Spb:public_key,Rpb:public_key,SND
,RCV:channel(dy))
played_by Sender
def=
      local

      State:nat,Nns:text,Minuss:hash_func,U:text,R:text,W:text,M:text,E:h
ash_func,Q:symmetric_key
      init
            State := 0
      transition
           1. State=0 /\ RCV(start) =|> State':=1 /\ SND(Sender.Reciever)
            2. State=1 /\ RCV(Reciever.{Nns'}_Rpb) =|> State':=2 /\
W':=new() /\ U':=new() /\ R':=new() /\ Q':=new() /\ M':=new() /\
secret(M',sec_2,{Sender}) /\ witness(Sender,Reciever,auth_1,M') /\
SND(Sender.{E(M')}_Q'.{Minuss(R'.U'.W')}_inv(Spb))
end role
```

Figure 4: HLPSL code for sender

```
role
role_Reciever(Sender:agent,Reciever:agent,Spb:public_key,Rpb:public_key,S
ND,RCV:channel(dy))
played_by Reciever
def=
      local

      State:nat,Nns:text,Minuss:hash_func,U:text,R:text,W:text,M:text,E:h
ash_func,Q:symmetric_key
      init
            State := 0
      transition
            1. State=0 /\ RCV(Sender.Reciever) =|> State':=1 /\
Nns':=new() /\ SND(Reciever.{Nns'}_Rpb)
            6. State=1 /\
RCV(Sender.{E(M')}_Q'.{Minuss(R'.U'.W')}_inv(Spb)) =|> State':=2 /\
request(Reciever,Sender,auth_1,M') /\ secret(M',sec_2,{Sender})
end role
```

Figure 5: HLPSL code for receiver

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/hlpslGenFile.if

GOAL
  As Specified

BACKEND
  OFMC

COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.13s
  visitedNodes: 6 nodes
  depth: 4 plies
```

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/hlpslGenFile.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 3 states
  Reachable  : 2 states
  Translation: 0.01 seconds
  Computation: 0.00 seconds
```

Figure 6: Simulation results of proposed scheme with Constraint Logic-based Attack Searcher (CL-b-AtSER) and On-the-Fly Model Checker (ON-t-FL-MCR)

7 Simulation Results and Analysis

By analyzing the security requirement of our scheme regarding man in the middle attack (confidentiality, integrity, Unforgeability, Non-repudiations, and forward secrecy) and Replay Attack, we used AVISPA tool to simulate. AVISPA working under four backend protocol (SAT-based Model Checker (SAT-b-MCR), Constraint Logic-based Attack Searcher (CL-b-AtSER), On-the-Fly Model Checker (ON-t-FL-MCR), and Tree automata based on Automatic Approximations for Analysis of Security Protocol (TA-4-SP)) when the scheme is pseudo code is written in High-Level-Protocol-Specification-Language (H-L-P-S-L) and converted to intermediate format (IF) [19]. So, we first convert our scheme algorithm into H-L-P-S-L code which contains two main roles that are *Sender and Receiver* in which we used the public and private keys of sender and receiver. The code for Sender and Receiver roles is represented in Figs. 4 and 5. We also used

nonce and hash functions for sender and receiver. We also set two goals that are authentication on auth_1 and secrecy of sec_2, which mean that security and authenticity. As we mentioned above the proposed scheme ensures the security services of confidentiality, integrity, Unforgeability, Non-repudiations, forward secrecy, and replay attack. So, in this regard, the goal "authentication on auth_1" ensures integrity, Unforgeability, and Non-repudiations and goal "secrecy of sec_2" ensures confidentiality, forward secrecy, and replay attack. We show the simulation result of our scheme in Fig. 6. and it is confirmed that the scheme is secured under the functionality of SAT-b-MCR and CL-b-AtSER.

8 Conclusion

Achieving higher levels of security in IoT environments is critical for protecting users' privacy and enhancing the overall functionality of such interconnected systems. In this work, we have proposed "an efficient heterogeneous signcryption scheme for access control within IoT environments to address the computational and communication cost issues of the existing approaches. We demonstrated that the proposed scheme prevented various attacks such as confidentiality, integrity, Unforgeability, Non-repudiations, Forward secrecy, and Replay attacks. AVISPA was utilized to perform formal security simulations, and the results supported our claim. We then compared the proposed scheme to existing schemes in terms of "computational costs" and "communication costs". As a result, our proposed scheme efficiently reduced both computational and communication costs. Accordingly, the proposed scheme proved to be more practical and appropriate than existing schemes for heterogeneous IoT applications.

Acknowledgement: Authors would like to thanks their universities for the support provided during this research.

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

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

References

[1] A. E. Guerrero-Sanchez, E. A. Rivas-Araiza, J. L. Gonzalez-Cordoba, M. Toledano-Ayala and A. Takacs, "Blockchain mechanism and symmetric encryption in a wireless sensor network," *Sensors*, vol. 20, no. 10, 2798, pp. 2020.

[2] K. A. Abuhasel and M. A. Khan, "A secure industrial internet of things (IIoT) framework for resource management in smart manufacturing," *IEEE Access*, vol. 8, pp. 117354–117364, 2020.

[3] M. A. Khan, M. T. Quasim, F. Algarni and A. Alharthi, "Internet of things: On the opportunities, applications and open challenges in Saudi Arabia," in *Proc. Int. Conf. on Advances in the Emerging Computing Technologies*, Al Madinah Al Munawwarah, Saudi Arabia, pp. 1–5, 2020.

[4] A. K. Das, M. Wazid, A. R. Yannam, J. J. P. C. Rodrigues and Y. Park, "Provably secure ECC-based device access control and key agreement protocol for IoT environment," *IEEE Access*, vol. 7, pp. 55382–55397, 2019.

[5] 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, pp. 1–15, 2018.

[6] A. Lohachab and Karambir, "ECC based inter-device authentication and authorization scheme using MQTT for IoT networks," *Journal of Information Security and Applications*, vol. 46, pp. 1–12, 2019.

[7] 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.
[8] M. Luo, Y. Luo, Y. Wan and Z. Wang, "Secure and efficient access control scheme for wireless sensor networks in the cross-domain context of the IoT," *Security and Commun. Network*, vol. 2018, pp. 1–10, 2018.
[9] M. U. Tariq, M. Babar, M. A. Jan, A. S. Khattak, M. D. Alshehri *et al.*, "Security requirement management for cloud-assisted and internet of things—Enabled smart city," *Computers, Materials & Continua*, vol. 67, no.1, pp. 625–639, 2021.
[10] S. Ullah, L. Marcenaro and B. Rinner, "Secure smart cameras by aggregate-signcryption with decryption fairness for multi receiver IoT applications," *Sensors*, vol. 19, no. 2, pp. 327, 2019.
[11] A. Karati, S. H. Islam, G. P. Biswas, M. Z. A. Bhuiyan, P. Vijayakumar *et al.*, "Provably secure identity-based signcryption scheme for crowdsourced industrial internet of things environments," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2904–2914, 2018.
[12] P. Kumar, S. Kumari, V. Sharma, A. K. Sangaiah, J. Wei *et al.*, "A certificateless aggregate signature scheme for healthcare wireless sensor network," *Sustainable Computing: Informatics and Systems*, vol. 18, pp. 80–89, 2018.
[13] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," *Advances in Cryptology ASIACRYPT of Lecture Notes in Computer Science*, vol. 2894. Springer, pp. 452–473, 2003.
[14] S. Hussain, I. Ullah, H. Khattak, M. Adnan, S. Kumari *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.
[15] S. Kumari, M. Karuppiah, A. K. Das, X. Li, F. Wu *et al.*, "A secure authentication scheme based on elliptic curve cryptography for IoT and cloud servers," *Journal of Supercomputing*, vol. 74, pp. 6428–6453, 2018.
[16] M. Khalifa, F. Algarni, M. A. Khan, A. Ullah and K. Aloufi, "A lightweight cryptography (LWC) framework to secure memory heap in internet of things," *Alexandria Engineering Journal*, vol. 60, no. 1, pp. 1489–1497, 2020.
[17] M. A. Khan, M. T. Quasim, N. S. Alghamdi and M. Y. Khan, "A secure framework for authentication and encryption using improved ECC for IoT-based medical sensor data," *IEEE Access*, vol. 8, pp. 52018–52027, 2020.
[18] A. Rahman, I. Ullah, M. Naeem, R. Anwar, H. Khattak *et al.*, "A lightweight multi-message and multi-receiver heterogeneous hybrid signcryption scheme sased on hyper elliptic curve," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 5, pp. 160–167, 2018.
[19] I. Ullah, N. U. Amin, A. Almogren, M. A. Khan and M. I. Uddin, "A lightweight and secured certificate-based proxy signcryption (CB-pS) scheme for e-prescription systems," *IEEE Access*, vol. 8, pp. 199197–199212, 2020.
[20] F. Li, Y. Han and C. Jin, "Practical access control for sensor networks in the context of the internet of things," *Computer Communications*, vol. 89–90, pp. 154–164, 2016.
[21] S. Challa, M. Wazid, A. K. Das, N. Kumar, A. G. Reddy *et al.*, "Secure signature-based authenticated key establishment scheme for future IoT applications," *IEEE Access*, vol. 5, pp. 3028–3043, 2017.
[22] S. A. Chaudhry, T. Shon, F. Al-Turjman and M. H. Alsharif, "Correcting design flaws: An improved and cloud assisted key agreement scheme in cyber physical systems," *Computer Communications*, vol. 153, pp. 527–537, 2020.
[23] S. A. Chaudhry, K. Yahya, F. Al-Turjman and M. -H. Yang, "A secure and reliable device access control scheme for IoT based sensor cloud systems," *IEEE Access*, vol. 8, pp. 139244–139254, 2020.
[24] S. Malani, J. Srinivas, A. K. Das, K. Srinathan and M. Jo, "Certificate-based anonymous device access control scheme for IoT environment," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9762–9773, 2019.
[25] Y. Zhao, Y. Hou, L. Wang, S. Kumari, M. K. Khan *et al.*, "An efficient certificateless aggregate signature scheme for the internet of vehicles," *Transactions on Emerging Telecommunications Technologies*, vol. 31, pp. e3708, 2020.

[26] M. A. Khan, I. M. Qureshi, I. Ullah, S. Khan, F. Khanzada *et al.*, "An efficient and provably secure certificateless blind signature scheme for flying ad-hoc network based on multi-access edge computing," *Electronics*, vol. 9, no. 30, pp. 1–22, 2020.

[27] M. A. Khan, I. Ullah, S. Nisar, F. Noor, I. M. Qureshi *et al.*, "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, I. Ullah, N. Kumar, I. M. Qureshi, F. Noor *et al.*, "An efficient and secure certificate-based access control and key agreement scheme for flying ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 2, pp. 1–13, 2021.