Journal of
Sensor and Actuator Networks

MDPI

Article

An Efficient Certificateless Forward-Secure Signature Scheme for Secure Deployments of the Internet of Things

Tahir Ali Shah [1], Insaf Ullah [2], Muhammad Asghar Khan [2], Pascal Lorenz [3,*] and Nisreen Innab [4]

[1] School of Computer, Jiangsu University of Science and Technology, Zhenjiang 212000, China
[2] Hamdard Institute of Engineering & Technology, Hamdard University, Islamabad 44000, Pakistan
[3] IRIMAS Institute, University of Haute Alsace, 68008 Colmar, France
[4] Department of Computer Science and Information Systems, College of Applied Sciences, AlMaarefa University, P.O. Box 71666, Riyadh 11597, Saudi Arabia
[*] Correspondence: pascal.lorenz@uha.fr

Abstract: As an extension of the wired network, the use of the wireless communication network has considerably boosted users' productivity at work and in their daily lives. The most notable aspect of the wireless communication network is that it overcomes the constraints of the wired network, reduces the amount of cost spent on wire maintenance, and distributes itself in a manner that is both more extensive and flexible. Combining wireless communication with the Internet of Things (IoT) can be used in several applications, including smart cities, smart traffic, smart farming, smart drones, etc. However, when exchanging data, wireless communication networks use an open network, allowing unauthorized users to engage in communication that is seriously destructive. Therefore, authentication through a digital signature will be the best solution to tackle such problems. Several digital signatures are contributing to the authentication process in a wireless communication network; however, they are suffering from several problems, including forward security, key escrow, certificate management, revocations, and high computational and communication costs, respectively. Keeping in view the above problems, in this paper we proposed an efficient certificateless forward-secure signature scheme for secure deployments in wireless communication networks. The security analysis of the proposed scheme is carried out using the random oracle model (ROM), which shows that it is unforgeable against type 1 and type 2 adversaries. Moreover, the computational and communication cost analyses are carried out by using major operations, major operations cost in milliseconds, and extra communication bits. The comparative analysis with the existing scheme shows that the proposed scheme reduces the computational cost from 19.23% to 97.54% and the communication overhead from 11.90% to 83.48%, which means that the proposed scheme is efficient, faster, and more secure for communication in the wireless communication network.

Keywords: IoT; certificateless forward-secure signature; hyperelliptic curve cryptography; ROM

check for updates

Citation: Shah, T.A.; Ullah, I.; Khan, M.A.; Lorenz, P.; Innab, N. An Efficient Certificateless Forward-Secure Signature Scheme for Secure Deployments of the Internet of Things. *J. Sens. Actuator Netw.* **2023**, *12*, 10. https://doi.org/10.3390/jsan12010010

Academic Editor: Jordi Mongay Batalla

Received: 14 November 2022
Revised: 5 January 2023
Accepted: 9 January 2023
Published: 23 January 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The Internet of Things (IoT) is a rapidly expanding field that involves connecting millions of physical objects (called "things") to networked sensors and smart devices that allow them to create, collect, and share different kinds of information [1,2]. As demonstrated in Figure 1, IoT has various applications in several industries, including smart cities, smart traffic, smart farming, and smart drones. In smart cities, IoT enhances people's lives by increasing traffic control, tracking the availability of parking places, evaluating the quality of the air, and even warning inhabitants when trash cans are full. In addition, it makes the traffic intelligent and employs sensors to collect raw traffic data, informing the driver of traffic updates to help him choose a better route while keeping his private information secure [3]. Farming is the second useful use of IoT devices, wherein data are gathered and analyzed to advise the owner of the need for water, pesticides, manure, fertilizer, or treatment for ill plants based on factors such as temperature, soil moisture, leaf wetness, and sun radiation [4]. The third application is the Internet of drones, in which smart drones could play an important role in multiple contexts, such as in smart cities, where they can be used for customer order delivery, accident surveillance and road traffic monitoring, private and police investigations, prison surveillance, drone taxis, ambulances drone, pollution control drone, surveillance and monitoring of large crowds at gatherings and protests, etc. [5].

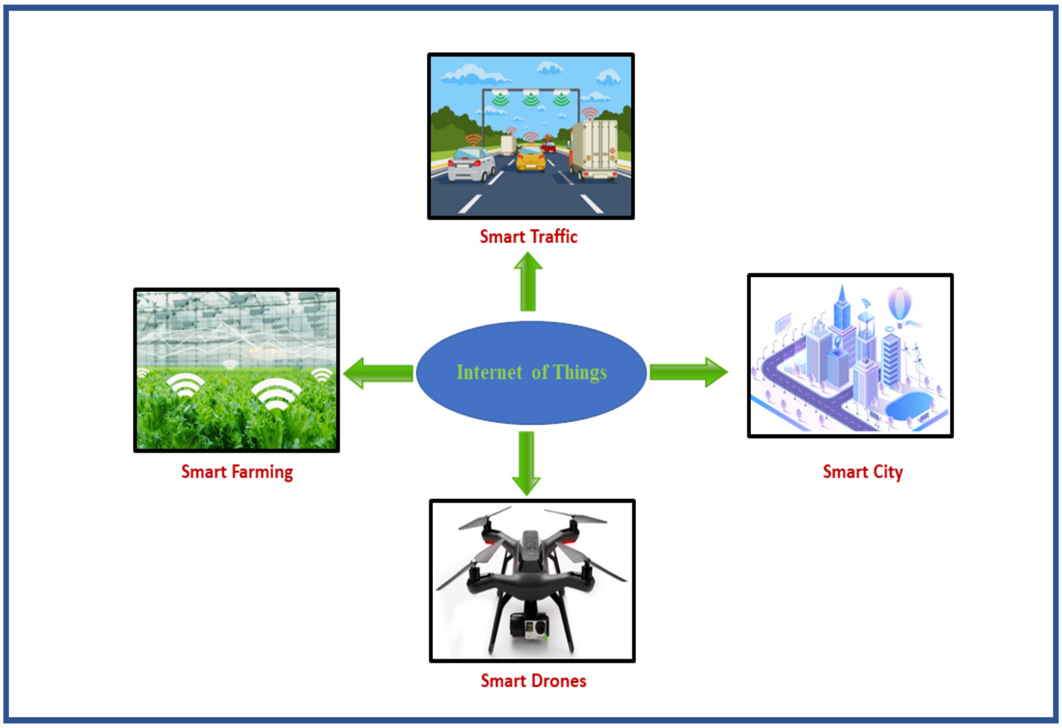

Figure 1. Applications of Internet of Things.

Moreover, drones are also used by most border patrol officers that monitor criminal activity on the border, mainly smuggling of drugs. This huge variation, the increasing management and interaction of devices, and the usage of public networks for the transfer of massive volumes of data make IoT systems an ideal target for hacker attacks [6]. IoT privacy and the safety of devices are linked, e.g., producing accidents by interrupting automotive networks, placing farms in danger by tampering with a farming network, invasion of privacy, power consumption, and poor data security in smart cities, the brick-sized batteries consumed by drones being heavy and losing energy quickly, memory limitations, and chances of malware and virus threats in the information shared by drones, etc. [7]. To counter such attacks, authentication is the most effective strategy, and it allows two or more network participants to verify each other's identity before exchanging data. In cryptography, the attractive technique for authentication is a digital signature, which is a mathematical method that is used to authenticate the identity of the sender through its private key, which it sends to the receiver, and then the receiver uses the public key of the sender and verifies the signature [8]. In a conventional digital signature technique, the signature key cannot be changed for every session, so there is a risk of exposure to the private key. The forward-secure digital signature was introduced to tackle the exposure problem of private keys where private keys are updated for every session [9]. The forward signature may be public key infrastructure-based (PKI-based) or identity-based (ID-based); however, in a PKI-based digital forward signature, there are certificate revocation and certificate management issues, and in ID-based digital forward signature schemes, there is a key escrow problem [10]. The abovementioned problems may be avoided using a forward secure certificateless digital signature, which combines the working structure of forward security with a certificateless signature. Though several forward-secure signatures are contributed, they are based on an elliptic curve, RSA, and bilinear pairing that are suffering from extra computational burdens on small IoT devices during the execution process and require more bandwidth because they need more bits to be transferred. Hyper elliptic curve cryptography (HECC) is the replacement for elliptic curve cryptography (ECC), and it uses only 80-bit keys. HECC is a subclass of algebraic curves that comprises genus g ≥ 1, and the field of the HECC is a quadratic extension of the field of rational functions, so in this sense, it is the simplest field of algebraic functions except for the field of rational functions [11]. HECC consists of the divisor D, which refers to the finite formal sum of points on a hyperelliptic curve, and the divisor D forms an Abelian group referred to as the Jacobian group $J_c\left(F_q\right)$ [12].

As a result of the above discussion, the following contributions have been made to this work:

1. We propose a certificateless, forward-secure HECC-based digital signature scheme that provides privacy, gets rid of the key escrow problem, and ensures its forward security.
2. A comprehensive security analysis is conducted to demonstrate that the proposed scheme is secure against various types of cyber-attacks.
3. Finally, the efficiency of the proposed scheme is evaluated by comparing it to other existing schemes in terms of its computation and communication costs. The results reveal that the proposed scheme is more efficient.

2. Literature Review

In recent years, the issues of privacy protection and forward security for the IoT have drawn more and more attention, and that is why security and privacy concerns may occur at multiple levels of smart IoT systems, so it needs to settle the problems mentioned above. Therefore, many signatures and authentication schemes have been proposed; for example, Malkin et al. [13] constructed a new forward-secure digital signature for the first time in which the existing schemes were combined to form a new forward secure digital signature scheme without being aware of the total number of periods. This scheme not only can take any digital signature scheme as the underlying module, but it also does not rely on any assumptions. They proved that this scheme achieves excellent performance overall, is very competitive with previous schemes with respect to all parameters and outperforms each of the previous schemes in at least one parameter. Itkis and Reyzin [14] developed a digital signature technique with forward secrecy using four modular exponentials and proved the security of their scheme based on the random oracle model (ROM). Kozlov and Reyzin [15] constructed a system for digital signatures that requires only a single modular exponential in the key update. The Fiat–Shamir transformation and the strong Rivest–Shamir–Adleman (RSA) assumption were used to demonstrate that this technique is secure against different types of attacks. McCullagh and Barreto [16] suggested a new forward-secured, efficient digital signature technique, which is based on pairing cryptography, that is both transferable and non-transferable. They pointed out semantic security problems in previous schemes and showed that this scheme is more secure than the previously proposed schemes. Boyen et al. [17] were the first to introduce the forward security digital signature with malicious updates in 2006. They introduced the concept of forward-secure signatures with an untrusted update, where the key update can be performed on an encrypted version of the key, and they demonstrated that forward-secure signatures with an untrusted update allow us to add forward security to signatures, while keeping passwords as a second factor of security. The security analysis of their scheme proved that the scheme has better performance as compared to the existing forward-secure signature schemes. The forward-secure ring signatures scheme was proposed by Liu and Wong [18] to resolve the key exposure problem. In their scheme, they reduced the damage of exposure of any secret key of users in a ring signature; even if a secret key is compromised, previously generated ring signatures remain valid and do not need to be regenerated. They demonstrated the security of their system using the ROM. Next, Das et al. [19] presented a new user authentication scheme that supports dynamic node addition. In this scheme, the user authenticates itself at both the base station and the cluster heads inside wireless sensor networks (WSN), so after successful authentication, both the user and the cluster head from which the user wants to access real-time data in the target field will be able to establish a secret session key between them. They showed that this scheme has better security performance. Taking into consideration the restricted sensor resources and time restrictions, a forward-secure Certificateless digital signature scheme was first introduced by Xu et al. [20] based on random lattice in the standard model, and they claimed that the scheme's strong unforgeability was based on the small integer solution problem. Kim et al. [21] constructed the Fast-Bellare–Miner (Fast-BM) and Fast-Abdalla–Reyzin (Fast-AR) fast forward-secure digital signature schemes, which allow fast signing and key updating with constant size public and secret keys and a short constant size signature. They proved that their approach is suitable for both real-time surveillance streaming applications and standard forward-secure signature systems. However, the computation cost was high because it was based on the elliptic curve. Oh et al. [22] designed an ID-based digital signature technique with a forward-secure private key generator. Based on the bilinear Diffie–Hellman inversion assumption (BDHI), they developed its concept and demonstrated its implementation by giving construction and security proof in the standard model (without random oracles). However, this scheme was based on bilinear pairing and required more computing power due to heavy pairing operations. Based on the RSA assumption, Ko et al. [23] developed a forward-secure ID-based digital signature technique with a forward-secure private key generator. They described its concept and presented practical constructions as well as its security proof in the random oracle model under the factoring assumption. Their scheme was based on RSA, which has high computation costs and communication costs. Du et al. [24] proposed a new provably secure certificateless signature scheme for IoT with perfect forward secrecy, which concentrated on designing a certificateless signature scheme (CLS) for IoT applications without pairings, which proved to be secure against different kinds of adversaries. Saqib et al. [25] proposed a three-factor authentication (password, identity, and low-cost digital signature) framework suitable for IoT-driven critical applications using ECC that provides mutual entity authentication of the gateway with both remote users (subscriber) and IoT node (publisher). The session key generation is dynamic, which could be changed in every session, which makes the scheme resistant to known session key attacks and guarantees pure forward secrecy. In 2022, based on an elliptic curve, a forward-secure digital signature scheme was proposed by Ping et al. [26] for privacy protection in wireless communication networks and proved its forward security and unforgeability in the random oracle model. However, this scheme suffers from three major flaws: (1) high computational cost, (2) more communication overhead, and (3) a key escrow problem. So, we have concluded three main limitations from the above literature survey, i.e., they are suffering from high computational cost, more communication overhead, and a key escrow problem, respectively.

To remove the above limitations, we are going to introduce a new method called the certificateless forward signature based on the hyperelliptic curve, which removes the key escrow problem, provides communication with very low bandwidth, and processes algorithms with very little time.

3. Preliminaries

This section discusses the proposed network model used in this scheme, the syntax of the proposed certificateless forward signature scheme, and the hyper elliptic curve discrete logarithm problem (HECDLP), respectively.

3.1. Network Model

This section describes the proposed network model for the proposed certificateless forward signature scheme used in the IoT environment. Figure 2 shows that our network model contains five entities: trusted authority, IoT devices, key update devices, controller, Internet, and receiver, which perform different functions during the communication process, respectively. Here, the role of a trusted third party is that when it receives the identity and request for a partial private key from IoT devices and receivers, it makes the partial key. By using their identities and delivering them on a secure network, the IoT devices receive a partial private key from a trusted third party and make their own private and public keys. After that, the key update device receives the request for signature key updating from IoT devices and sends back the updated key to the IoT devices after performing the updating process. Then, IoT devices give the updated key and generated data to the controller by using Bluetooth technology. Bluetooth technology enables wireless communication between devices without the use of wires or cables [6]. It is based on short-range radio frequency, and any device equipped with the technology can communicate if it is within a specified distance. This technology is essentially a wireless networking protocol for a broad range of devices, such as notebook computers, as well as cooking ovens, PDAs, mobile phones, and refrigerators, in the residential, workplace, and other similar aspects. After the above process, the controller generates a forward signature and sends it to the receiver using 5G communication with the open network. When the signature tuple is received by the receiver, it performs the verification process; if the verification is successful, it accepts the signature and data; otherwise, it rejects it.

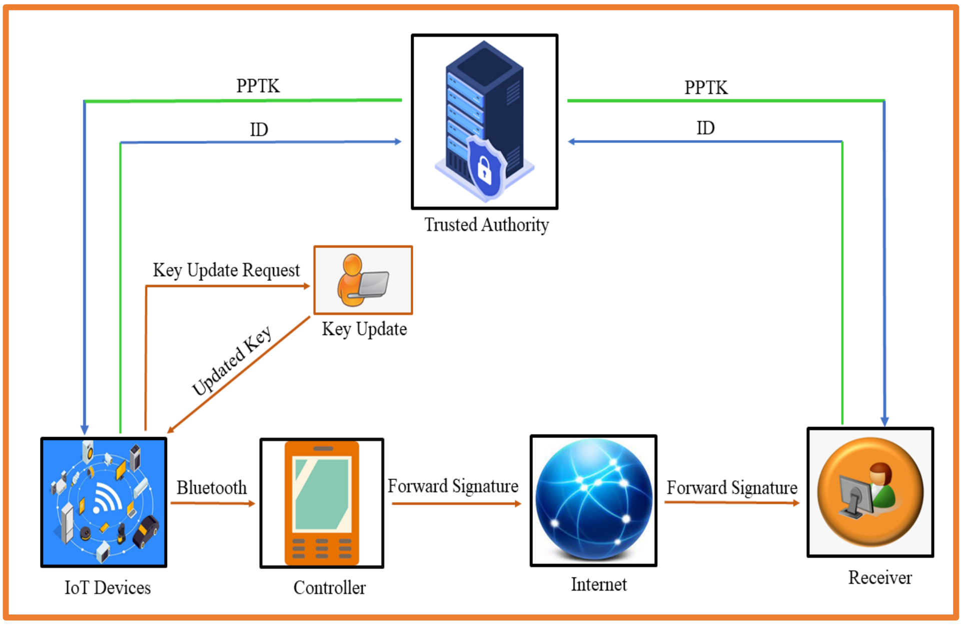

Figure 2. Network model for our proposed system.

3.2. Syntax of Certificateless Forward Signature

The syntax contains the subsections that are Initialization, Generate Private Number, Generate Partial Private Key, Generate Private Key, Generate Public Key, Key Update, Generate Forward Signature, and Forward Signature Verification. So, the explanations of each subsection are as follows:

1. Initialization: The trusted authority (TA) generates public parameter param, his private key (∂), and public key (Γ) by taking as input the security parameter of hyperelliptic curve.
2. Generate Private Number: Given the security parameter and param, the user (U_i) selects ϕ_i as his private number.

3. Generate Partial Private Key: Given user identity (ID_i), public key of TA (Γ), and public parameter param, TA generates the tuple ($\mathcal{I}_i$, ω_i) as a partial private key for user with identity (ID_i).
4. Generate Private Key: Given a private number (ϕ_i) and the tuple ($\mathcal{I}_i$, ω_i), a ser (U_i) sets (ω_i, ϕ_i) as his private key.
5. Key Update: In this phase, it renews the signature key pair by replacing (ω_i, ϕ_i) on (${\omega_i}^{new}$, ${\phi_i}^{new}$) before signature generations and also renews the verification public key as ($\mathcal{Q}_i^{new}$, $\mathcal{I}_i^{new}$).
6. *Generate Public Key:* Given a private number (ϕ_i) and the tuple ($\mathcal{I}_i$, ω_i), the user (U_i) sets ($\mathcal{I}_i, \mathcal{Q}_i$) as his public key, where $\mathcal{Q}_i = \phi_i.\mathcal{D}$.
7. *Generate Forward Signature:* Given a message m, the updated signature key pair (${\omega_i}^{new}$, ${\phi_i}^{new}$), param, signer identity (ID_i), and $\mathcal{I}_i$, generate and send the signature tuple (K, β) to the verifier.
8. *Forward Signature Verification:* Given a message m, the public key pair ($\mathcal{I}_i$, $\mathcal{Q}_i$), param, signer identity (ID_i), and (K, β), the verifier verifies the received signature tuple.

3.3. Hyperelliptic Curve Discrete Logarithm Problem (HECDLP)

In place of elliptic curve cryptography (ECC), hyper elliptic curve cryptography (HECC) uses keys that are just 80 bits long. The field of the HECC is a quadratic extension of the field of rational functions, making it the simplest field of algebraic functions, except for the field of rational functions. The HECC is a subclass of algebraic curves that includes genus g 1. The Jacobian group is an Abelian group that contains the divisor D, which is the finite formal sum of points on a hyperelliptic curve.

Supposing $\mathrm{Y} = \partial.\mathcal{D}$, finding the value of ∂ from Y is called the hyper elliptic curve discrete logarithm problem.

4. Certificateless Forward-Secure Signature Scheme

The following seven sub algorithmic steps can make our proposed certificateless forward-secure signature scheme, and Table 1 contains the symbols that are used to make up the whole algorithm's mathematical steps.

Table 1. Symbols used in the proposed algorithm.

No	Symbol	Description
1	$H_{G=2}$	Represents a hyper elliptic curve with genus 2
2	F_p	Represents a finite field of order p, where its range is not more than 80 bits
3	$\mathcal{D}$	Represents a devisor, where its range is not more then 80 bits
4	H_j, H_k, H_l	Represent three irreversible, one-way, and collision-resistant hash functions from the SHA family
5	Γ	The public key of TA, and it is made from the combination of secret key and devisor
6	∂	The secret key of TA, and it is randomly selected from F_p
7	U_i	This symbol is used to indicate user
8	ω_i, ϕ_i	These two symbols are used to indicate the private key of U_i
9	ϕ_i	This is used to represent the private number of U_i
10	ID_i	This is used to represent the identity of U_i
11	${\omega_i}^{new}$, ${\phi_i}^{new}$	This is used to represent the update private key pair of U_i
12	$\mathcal{I}_i, \mathcal{Q}_i$	This is used to represent the public key pair of U_i
13	$\mathcal{Q}_i^{new}$, $\mathcal{I}_i^{new}$	This is used to represent the update public key pair of U_i
14	K, β	This is used to represent the signature pair generated by signer

Table 1. *Cont.*

No	Symbol	Description
15	$\mathcal{BM}$	This is used to represent bilinear pairing-based multiplication
16	Xe	This is used to represent the exponential
17	$\mathcal{ECM}$	This is used to represent elliptic curve multiplication
18	$\mathcal{HECM}$	This is used to represent hyperelliptic curve multiplication
19	$\mathcal{B}$Ƥ	This is used to represent the bilinear pairing operation
20	C_n	This is used to represent the challenger, which will support the adversary during security analysis
21	A_n	This is used to represent the type 1 adversary
22	A_m	This is used to represent the type 2 adversary
23	$\mathcal{E}$	This is used to represent the non-negligible probability type 1 and type 2 adversaries
24	QH_l	This is used to represent the query for H_l
25	Q_{ppt}	This is used to represent partial private key query
26	Q_U	This is used to represent user creation query
27	QH_k	This is used to denote the query for H_k
28	QH_j	This is used to denote the query for H_j

1. *Initialization:* Here, the trusted authority performs the following mathematical computations:
   - Select hyper elliptic curve ($H_{G=2}$) with genus 2.
   - Suggest the finite field (F_p) of order p, where its range is not more than 80 bits.
   - Suggest the devisor $(\mathcal{D})$ of $H_{G=2}$, where its range is not more than 80 bits.
   - Suggest three irreversible, one-way, and collision-resistant hash functions (H_j, H_k, H_l) from the SHA family.
   - TA computes the public key $\Gamma = \partial.\mathcal{D}$, where ∂ is the randomly selected private key from F_p.
   - TA publishes the public parameter set $\{\Gamma,\ \mathcal{D},\ F_p, H_{G=2}, H_j,\ H_k,\ H_l\}$.
2. *Generate Private Number:* User (U_i) selects ϕ_i from F_p as a private number.
3. *Generate Partial Private Key:* Upon the request of U_i with identity ID_i, TA selects γ_i from F_p and computes $\mathcal{I}_i = \gamma_i.\mathcal{D}$, $\Delta_i = H_j(ID_i, \Gamma, \mathcal{I}_i)$, and $\omega_i = \partial + \gamma_i.\mathcal{I}_i$.
4. *Generate Private Key:* The User (U_i) sets (ω_i, ϕ_i) as his private key.
5. *Key Update:* In this phase, it renews the signature key pair by replacing (ω_i, ϕ_i) on $({\omega_i}^{new}, {\phi_i}^{new})$ before signature generations and also renews the verification public key as $({\mathcal{Q}_i}^{new}, {\mathcal{I}_i}^{new})$.
6. *Generate Public Key:* The user (U_i) sets $(\mathcal{I}_i, \mathcal{Q}_i)$ as his public key, where $\mathcal{Q}_i = \phi_i.\mathcal{D}$.
7. *Generate Forward Signature:* Given a message m, the updated signature key pair $({\omega_i}^{new}, {\phi_i}^{new})$, $\{\Gamma,\ \mathcal{D},\ F_p,\ H_{G=2}, H_j, H_k, H_l\}$, signer identity (ID_i), and $\mathcal{I}_i$, the signer performs the following computations:
   - Signer selects $ꝁ$ from F_p and computes $K = ꝁ.\mathcal{D}$.
   - Compute $r_1 = H_k(m, K)$ and $r_2 = H_l(m, K, \Gamma, \mathcal{Q}_i)$.
   - Compute $\beta = {\phi_i}^{new} + r_1 ꝁ + r_2 {\omega_i}^{new}$ and send (K, β, r_1) to verifier.
8. *Forward Signature Verification:* Given a message m, the public key pair $(\mathcal{I}_i, \mathcal{Q}_i)$, $\{\Gamma,\ \mathcal{D},\ F_p,\ H_{G=2}, H_j, H_k, H_l\}$, signer identity (ID_i), and (K, β, r_1), the verifier performs the following computations:

   Verifier computes $\Delta_i = H_j(ID_i, \Gamma, \mathcal{I}_i)$, $r_1 = H_k(m, K)$, and $r_2 = H_l(m, K, \Gamma, \mathcal{Q}_i)$.

Verifier checks the validity of the signature by computing $\beta.\mathcal{D} = \mathcal{Q}_i + r_1 K + r_2(\Gamma + \Delta_i {\mathcal{I}_i}^{new})$; if it is satisfied, accept.

5. Correctness

Given a message m, the public key pair $(\mathcal{I}_i, \mathcal{Q}_i)$, $\{\Gamma, \mathcal{D}, F_p,\ H_{G=2}, H_j, H_k, H_l\}$, signer identity (ID_i), and (K, β, r_1), the verifier computes $\Delta_i = H_j(ID_i, \Gamma, \mathcal{I}_i)$, $r_1 = H_k(m, K)$, and $r_2 = H_l(m, K, \Gamma, \mathcal{Q}_i)$. Verifier checks the validity of the signature by computing $\beta.\mathcal{D} = \mathcal{Q}_i + r_1 K + r_2(\Gamma + \Delta_i \mathcal{I}_i{}^{new})$; if it is satisfied, accept.

$$\begin{gathered}
\beta.\mathcal{D} = \mathcal{Q}_i + r_1 K + r_2(\Gamma + \Delta_i \mathcal{I}_i{}^{new}) \\
\beta.\mathcal{D} = (\phi_i{}^{new} + r_1 \boldsymbol{k} + r_2 \omega_i{}^{new}).\mathcal{D} \\
= (\phi_i{}^{new}.\mathcal{D} + r_1 \boldsymbol{k}.\mathcal{D} + r_2 \omega_i{}^{new}.\mathcal{D}) \\
= (\mathcal{Q}_i{}^{new} + r_1 K + r_2(\partial + \gamma_i.\mathcal{I}_i{}^{new}).\mathcal{D}) \\
= (\mathcal{Q}_i{}^{new} + r_1 K + r_2(\partial.\mathcal{D} + \gamma_i.\mathcal{D}.\mathcal{I}_i{}^{new}) \\
= \mathcal{Q}_i + r_1 K + r_2(\Gamma + \Delta_i \mathcal{I}_i{}^{new})
\end{gathered}$$

is hence proved.

6. Security Analysis

Our proposed certificateless forward-secure signature scheme is analyzed for unforgeability under the process of the random oracle model against type 1 and type 2 adversaries based on the crack hyperelliptic curve discrete logarithm problem. The following two theorems (e.g., Theorems 1 and 2) are used for the provable security of the proposed scheme. Both of the theorems, i.e., Theorems 1 and 2, are based on the robustness of hard problem called the hyperelliptic curve discrete logarithm, which is not feasible for type 1 and type 2 adversaries to break its security. Therefore, the following two theorems show that our proposed scheme is unforgeable due to the hardiness of the hyperelliptic curve discrete logarithm problem.

Theorem 1. *In this theorem, we first introduce some players and symbols, A_n, C_n, and $\mathcal{E}$, denoting the type 1 adversary, challenger, and non-negligible advantages of A_n in a polynomial time. Then, we explain the probability of solving the hyperelliptic curve discrete logarithm problem of C_n in the following equations.*

$$\mathcal{E}^{/} = \left(1 - \frac{QH_j}{Q}\right)^{Q_U} + \left(1 - \frac{1}{Q_U}\right)^{Q_{ppt}} \left(\frac{1}{Q_U}\right)\left(1 - \frac{QH_k}{Q}\right)\left(1 - \frac{QH_l}{Q}\right)\mathcal{E}$$

Here, QH_j, QH_k, Q_U, Q_{ppt}, and QH_l denote the query for H_j, H_k, user creation query, partial private key query, and the query for H_l, respectively.

Proof. A_n can win in Theorem 1 with $\mathcal{E}$, and the challenger (C_n) is needed to crack the hyperelliptic curve discrete logarithm problem in which $\Upsilon = \partial.\mathcal{D}$. The challenger ($C_n$) sets $\Upsilon = \Gamma$ and is required to extract ∂. The challenger (C_n) suggests some empty lists at the beginning of this process, which are $L_j, L_k, L_{l,}, L_{CUQ}, L_{PNQ}$, and L_{PPKQ}, that can store the information about H_j query, H_k query, H_l query, and user creation query, private number query, and partial private key query, respectively. □

Phase 1: here, first of all, the challenger (C_n) could suggest the target identity ID^*, generate public parameter set $\{\Gamma = \Upsilon, \mathcal{D}, F_p,\ H_{G=2}, H_j, H_k, H_l\}$, and send it to A_n.

Phase 2: keeping in view the polynomials' bounded nature, it performs the following queries:

1. *H_j Query:* When A_n submits the H_j query with $(ID_i, \Gamma, \mathcal{I}_i)$, the challenger ($C_n$) combs in L_j and returns $(ID_i, \Gamma, \mathcal{I}_i, \Delta_i)$, if it was available previously. Otherwise, it chooses Δ_i from F_p and sends it to A_n.
2. *H_k Query:* When A_n submits the H_k query with (m, K) , the challenger (C_n) combs in L_k and returns (m, K, r_{1i}), if it was available previously. Otherwise, it chooses r_{1i} from F_p and sends it to A_n.

3. *H_l Query*: When A_n submits the H_l query with $(m, K, \Gamma, \mathcal{Q}_i)$, the challenger ($C_n$) combs in L_l and returns $(m, K, \Gamma, \mathcal{Q}_i, r_{2i})$, if it was available previously. Otherwise, it chooses r_{2i} from F_p and sends it to A_n.
4. *User Creation Query:* When A_n submits query with ID_i, the challenger (C_n) combs in L_{UCQ} and returns $(\mathcal{Q}_i{}^{new}, \mathcal{I}_i{}^{new})$ and $(\mathcal{I}_i, \mathcal{Q}_i)$, if they exist. Otherwise, it goes for the following conditions:
   - If $ID_i \neq ID^*$, three variables $\omega_i, \phi_i, \Delta_i$ are chosen by C_n, which computes $\mathcal{I}_i = \frac{\omega_i.\ \mathcal{D}-\Gamma}{\Delta_i}$, and $\mathcal{Q}_i = \phi_i.\mathcal{D}$.
   - If $ID = ID^*$, three variables $\omega_i, \phi_i, \Delta_i$ are chosen by C_n, which computes $\mathcal{I}_i = \gamma_i.\mathcal{D}$, $\mathcal{Q}_i = \phi_i.\ \mathcal{D}$, and sets $\omega_i = null$. Then, it returns $(\mathcal{I}_i, \mathcal{Q}_i)$ and renews $(\mathcal{Q}_i{}^{new}, \mathcal{I}_i{}^{new})$ to A_n and updates L_{UCQ}.

1. *Replace Public Key Query:* When A_n submits a query with ID_i, the challenger (C_n) replaces $(\mathcal{Q}_i{}^{new/}, \mathcal{I}_i{}^{new/})$ and $(\mathcal{Q}_i{}^{/}, \mathcal{I}_i{}^{/})$ and returns them to A_n.
2. *Private Number Query:* When A_n submits a query with ID_i, the challenger (C_n) combs in L_{PNQ} and returns ϕ_i, if it exists. Otherwise, it goes for the following conditions:
   - If $ID_i \neq ID^*$, three variables $\omega_i, \phi_i, \Delta_i$ are chosen by C_n, which computes $\mathcal{I}_i = \frac{\omega_i.\ \mathcal{D}-\Gamma}{\Delta_i}$ and $\mathcal{Q}_i = \phi_i.\mathcal{D}$.
   - If $ID = ID^*$, three variables $\omega_i, \phi_i, \Delta_i$ are chosen by C_n, which computes $\mathcal{I}_i = \gamma_i.\mathcal{D}$, $\mathcal{Q}_i = \phi_i.\ \mathcal{D}$, and sets $\omega_i = null$. Then, it renews $(\omega_i{}^{new}, \phi_i{}^{new})$ and returns to A_n and updates L_{PNQ}.

1. *Partial Private Key Query:* When A_n submits a query with ID_i, the challenger (C_n) checks if $ID_i \neq ID^*$, and then it combs in L_{PPKQ} and returns $\omega_i{}^{new}$, if it exists. Otherwise, it stops the further executions.
2. *Generate Forward Signature Query:* When A_n submits a query with ID_i, the challenger (C_n) combs in $L_j, L_k, L_{l,}, L_{CUQ}, L_{PNQ}$, and L_{PPKQ} for the record of $(ID_i, \omega_i{}^{new}, \phi_i{}^{new}, \mathcal{Q}_i{}^{new}, \mathcal{I}_i{}^{new})$, $(ID_i,\Gamma, \mathcal{I}_i{}^{new})$, (m, K), and $(m, K, \Gamma, \mathcal{Q}_i{}^{new})$. If $ID = ID^*$ or $\omega_i = null$, C_n randomly chooses K and β and sends them to A_n. Otherwise, three variables $\Bbbk, r_1, r_2$ are chosen by C_n, which computes $K = \Bbbk.\mathcal{D}$, $\beta = \phi_i{}^{new} + r_1\Bbbk + r_2\omega_i{}^{new}$ and returns K, β to A_n.

Phase 3: A_n generates a forge signature $(K^{forge}, \beta^{forge})$, C_n checks if it belongs to ID^*, and if it does not, it stops further processing. Otherwise, the challenger (C_n) combs in $L_j, L_k, L_{l,}, L_{CUQ}, L_{PNQ}$, and L_{PPKQ} for the record of $(ID_i, \omega_i{}^{new}, \phi_i{}^{new}, \mathcal{Q}_i{}^{new}, \mathcal{I}_i{}^{new})$, $(ID_i,\Gamma, \mathcal{I}_i{}^{new})$, (m, K), and $(m, K, \Gamma, \mathcal{Q}_i{}^{new})$. If the above records are not found in $L_j, L_k, L_{l,}$, L_{CUQ}, L_{PNQ}, and L_{PPKQ}, it stops further processing. For the forge signature generation, a genuine value of $\Bbbk$, $\phi_i{}^{new}$, and $\omega_i{}^{new}$ needs to be chosen, which will solve the hyperelliptic curve discrete logarithm problem. Suppose the probability of solving the hyperelliptic curve discrete logarithm problem is $Prob(Wins)$ and $rob(Wins) = Prob(Event1 \wedge Event2)$, where $Event1$ represents all the queries, and executions of this theorem are successful, and $Event2$ denotes that A_n generates a forge signature on ID^*. Letting A_n forge a forward signature with probability advantages $\mathcal{E}$, we can calculate $Prob(Wins) = Prob(Event1 \wedge Event2) = Prob(Event1)Prob(Event1.Event2) = Prob(Event1)\mathcal{E}$. We can define some of the probabilities that follow:

1. If there exists no collision during the user creation query, its probability is $\left(1 - \frac{QH_j}{Q}\right)^{Q_U}$.
2. When A_n is not called for the partial private key query on ID^*, its probability is $\left(1 - \frac{1}{Q_U}\right)^{Q_{ppt}}$.
3. A_n can send forward a signature if $ID = ID^*$, and its probability is $\frac{1}{Q_U}$.
4. A_n can find the valid value from L_k, and its probability is $\left(1 - \frac{QH_k}{Q}\right)$.
5. A_n can find the valid value from L_l, and its probability is $(\ 1 - \frac{QH_l}{Q})$.

6. The combined probability will be what follows: $\mathcal{E}^{/} = \left(1 - \frac{QH_j}{Q}\right)^{Q_U} + \left(1 - \frac{1}{Q_U}\right)^{Q_{ppt}} \left(\frac{1}{Q_U}\right)\left(1 - \frac{QH_k}{Q}\right)(\,1 - \frac{QH_l}{Q})\mathcal{E}$.

Using the above probability analysis, we have proved that the proposed scheme resists against the type 1 adversary for forgeability attack, because the adversary is not able to find the solution for the hyperelliptic curve discrete problem.

Theorem 2. *In this theorem, we first introduce some players and symbols, A_m, C_n, and $\mathcal{E}$, denoting the type 2 adversary, challenger, and non-negligible probability of A_m in a polynomial time. Then, we explain the probability of solving the hyperelliptic curve discrete logarithm problem of C_n in the following equations.*

$$\mathcal{E}^{/} = \left(1 - \frac{QH_j}{Q}\right)^{Q_U} + \left(1 - \frac{1}{Q_U}\right)^{Q_{ppt}} \left(\frac{1}{Q_U}\right)\left(1 - \frac{QH_k}{Q}\right)$$

Here, QH_j, QH_k, Q_U, Q_{ppt}, and QH_l denote the query for H_j, H_k, user creation query, partial private key query, and the query for H_l, respectively.

Proof. A_m can win in Theorem 2 with $\mathcal{E}$, and the challenger (C_n) is needed to crack the hyperelliptic curve discrete logarithm problem in which $\Upsilon = \partial.\mathcal{D}$. The challenger ($C_n$) sets $\Upsilon = \Gamma$ and is required to extract ∂. The challenger (C_n suggests some empty lists at the beginning of this process, which are $L_j, L_k, L_{l,}, L_{CUQ}, L_{PNQ}$, and L_{PPKQ}, that can store the information about H_j query, H_k query, H_l query, and user creation query, private number query, and partial private key query, respectively. □

Phase 1: Here, first of all, the challenger (C_n) could suggest the target identity ID^*, generate public parameter set $\{\Gamma = \Upsilon, \mathcal{D},\ F_p,\ H_{G=2}, H_j, H_k, H_l\}$, and send Γ and ∂ to A_m.

Phase 2: keeping in view the polynomials' bounded nature, it performs the following queries:

1. H_j *Query*: This query is performed as in Theorem 1.
2. H_k *Query*: This query is performed as in Theorem 1.
3. H_l *Query*: This query is performed as in Theorem 1.
4. *User Creation Query*: When A_n submits a query with ID_i, the challenger (C_n) combs in L_{UCQ} and returns $(\mathcal{Q}_i{}^{new}, \mathcal{I}_i{}^{new})$ and $(\mathcal{I}_i, \mathcal{Q}_i)$, if they exist. Otherwise, it goes for the followed conditions:
   - If $ID_i \neq ID^*$, three variables $\omega_i, \phi_i, \Delta_i$ are chosen by C_n, which computes $\mathcal{I}_i = \frac{\omega_i.\ \mathcal{D} - \Gamma}{\Delta_i}$ and $\mathcal{Q}_i = \phi_i.\mathcal{D}$.
   - If $ID = ID^*$, three variables $\omega_i, \phi_i, \Delta_i$ are chosen by C_n, which computes $\mathcal{I}_i = \gamma_i.\ \mathcal{D}$, $\mathcal{Q}_i = \phi_i.\mathcal{D}$, and sets $\omega_i = null$. Then, it returns $(\mathcal{I}_i, \mathcal{Q}_i)$ and renews $(\mathcal{Q}_i{}^{new}, \mathcal{I}_i{}^{new})$ to A_m and updates L_{UCQ}.
5. *Private Number Query*: Here, A_m is not allowed to access ϕ_i on ID^*, and C_n will not stop further executions if $ID_i \neq ID^*$. Otherwise, the challenger (C_n) combs in L_{PNQ} and returns ϕ_i if it exists.
6. *Partial Private Key Query:* When A_m submits a query with ID_i, the challenger (C_n) combs in L_{PPKQ} and returns $\omega_i{}^{new}$ if it exists.
7. *Generate Forward Signature Query:* When A_m submits a query with ID_i, the challenger (C_n) combs in $L_j, L_k, L_{l,}, L_{CUQ}, L_{PNQ}$, and L_{PPKQ} for the record of $(ID_i, \omega_i{}^{new}, \phi_i{}^{new}, \mathcal{Q}_i{}^{new}, \mathcal{I}_i{}^{new})$, $(ID_i, \Gamma,\ \mathcal{I}_i{}^{new})$, (m, K), and $(m, K, \Gamma, \mathcal{Q}_i{}^{new})$. If $ID = ID^*$ or $\omega_i = null$, C_n randomly chooses K and β, and sends them to A_m. Otherwise, three variables $\mathscr{k}, r_1, r_2$ are chosen by C_n, which computes $K = \mathscr{k}.\mathcal{D}$, $\beta = \phi_i{}^{new} + r_1\mathscr{k} + r_2\omega_i{}^{new}$, and returns K, β to A_m.

Phase 3: A_m generates a forge signature $(K^{forge},\ \beta^{forge})$, C_n checks if it belongs to ID^*, and if it does not, it stops further processing. Otherwise, the challenger (C_n) combs in $L_j, L_k, L_{l,}, L_{CUQ}, L_{PNQ}$, and L_{PPKQ} for the record of $(ID_i,\ \omega_i{}^{new},\ \phi_i{}^{new},\ \mathcal{Q}_i{}^{new},\ \mathcal{I}_i{}^{new})$, $(ID_i,\Gamma,\ \mathcal{I}_i{}^{new})$, (m,K), and $(m,K,\Gamma,\mathcal{Q}_i{}^{new})$. If the above records are not found in $L_j, L_k, L_{l,}$, L_{CUQ}, L_{PNQ}, and L_{PPKQ}, it stops further processing. For the forge signature generation, a genuine value of $\Bbbk$, $\phi_i{}^{new}$, and $\omega_i{}^{new}$ needs to be chosen, which will the solve hyperelliptic curve discrete logarithm problem. Suppose the probability of solving the hyperelliptic curve discrete logarithm problem is $Prob(Wins)$ and $rob(Wins) = Prob(Event1 \wedge Event2)$, where $Event1$ represents all the queries, and executions of this theorem are successful, and $Event2$ denotes that A_n generates a forge signature on ID^*. Letting A_m forge a forward signature with probability advantages $\mathcal{E}$, we can calculate $Prob(Wins) = Prob(Event1 \wedge Event2) = Prob(Event1)Prob(Event1.Event2) = Prob(Event1)\mathcal{E}$. We can define some of the probabilities that follow:

1. If there exists no collision during the user creation query, its probability is $\left(1 - \frac{QH_j}{Q}\right)^{Q_U}$.
2. When A_m is not called for the partial private key query on ID^*, its probability is $\left(1 - \frac{1}{Q_U}\right)^{Q_{ppt}}$.
3. A_m can send forward a signature if $ID = ID^*$, and its probability is $\frac{1}{Q_U}$.
4. A_m can find the valid value from L_k, and its probability is $\left(1 - \frac{QH_k}{Q}\right)$.
5. A_m can find the valid value from L_l, and its probability is $(\,1 - \frac{QH_l}{Q})$.
6. The combined probability will be what follows: $\mathcal{E}^{/} = \left(1 - \frac{QH_j}{Q}\right)^{Q_U} + \left(1 - \frac{1}{Q_U}\right)^{Q_{ppt}} \left(\frac{1}{Q_U}\right)\left(1 - \frac{QH_k}{Q}\right)(\,1 - \frac{QH_l}{Q})\mathcal{E}$.

Using the above probability analysis, we have proved that the proposed scheme resists against the type 2 adversary for forgeability attack, because the adversary is not able to find the solution for the hyperelliptic curve discrete problem.

Theorem 3. *In this theorem, we will first prove how our proposed scheme provides the integrity of the message [27].*

Proof. In the proposed scheme, the sender computes $r_1 = H_k(m,K)$ and sends (r_1) to the verifier. At the receiving side, the verifier computes $r_{11} = H_k(m,K)$ and compares if the following equation is satisfied, $r_{11} = r_1$, and then it means that our scheme provides integrity of message. □

Theorem 4. *In this theorem, we will first prove how our proposed scheme provides authentication between the sender and verifier.*

Proof. In the proposed scheme, the signer selects $\Bbbk$ from F_p, computes $K = \Bbbk.\mathcal{D}$, $r_1 = H_k(m,K)$, $r_2 = H_l(m,K,\Gamma,\mathcal{Q}_i)$, $\beta = \phi_i{}^{new} + r_1\Bbbk + r_2\omega_i{}^{new}$, and sends (K,β,r_1) to the verifier. The verifier computes $\Delta_i = H_j(ID_i,\Gamma,\mathcal{I}_i)$, $r_1 = H_k(m,K)$, $r_2 = H_l(m,K,\Gamma,\mathcal{Q}_i)$, and checks the validity of the signature by computing $\beta.\mathcal{D} = \mathcal{Q}_i + r_1K + r_2(\Gamma + \Delta_i\mathcal{I}_i{}^{new})$; if it is satisfied, the signature is accepted. In Section 5, Correctness, we have shown equality of the followed equation: $\beta.\mathcal{D} = \mathcal{Q}_i + r_1K + r_2(\Gamma + \Delta_i\mathcal{I}_i{}^{new})$; if it is proved, that means that the proposed schemes provide authentication or authenticity security requirements. □

7. Computational Cost

In this section, we are going to evaluate the efficiency of the proposed scheme with respect to the computational cost based on major operations. Normally, the major operations in cryptographic scheme are considered the operation, such as elliptic curve point multiplication, bilinear pairing operation, exponentiations, and hyperelliptic curve devisor multiplications, respectively. For the evaluation of the proposed scheme with respect to the computational cost, we consider major operations such as exponential (Xe), bilinear pairing-based multiplication ($\mathcal{BM}$), hyperelliptic curve multiplication ($\mathcal{HECM}$), bilinear pairing operation ($\mathcal{BP}$), and elliptic curve multiplication ($\mathcal{ECM}$) in the proposed scheme and those of Kim et al. [21], Oh et al. [22], Ko et al. [23], and Zhang et al. [26], respectively. The comparative outcomes are presented in Table 2, based on major operations in the proposed scheme and those of Kim et al. [21], Oh et al. [22], Ko et al. [23], and Zhang et al. [26]. The analysis based on time in milliseconds (ms) is included in Table 3, between Kim et al. [21], Oh et al. [22], Ko et al. [23], and Zhang et al. [26], and the proposed scheme. Note that we have calculated the values used in Table 3 based on the experimental setup of [28], which includes hardware and software specifications such as a PC Intel Corei7, random access memory (RAM) of 8 GB, and a multi-precision integer and rational arithmetic C library, in which Xe needs 1.25 ms, $\mathcal{BM}$ consumes 4.31 ms, $\mathcal{HECM}$ requires 0.48 ms, and $\mathcal{BP}$ needs 14.90 ms, respectively. By using the values contained in Table 3, we generated Figure 3, which clearly indicates that the proposed scheme is efficient as compared to Kim et al. [21], Oh et al. [22], Ko et al. [23], and Zhang et al. [26]. In comparison with the schemes of Kim et al. [21], Oh et al. [22], Ko et al. [23], and Zhang et al. [26], Tables 2 and 3 and Figure 3 demonstrate that the new approach consumed fewer computing resources by using the hyperelliptic curve cryptography, which uses only 80 bits of key size and provides the same security level as the RSA, as well as elliptic curve cryptography.

Table 2. Comparison of computation cost in terms of major operations between Our Scheme and those Kim et al. [21], Oh et al. [22], Ko et al. [23], and Zhang et al. [26].

Schemes	Key Update	Sender	Receiver	Total
Kim et al. [21]	8Xe $+ 5\mathcal{BM}$	$5\mathcal{BM} +$ 6Xe	$3\mathcal{BM} +$ 3Xe $+ 4\mathcal{BP}$	17Xe $+ 13\mathcal{BM} + 4\mathcal{BP}$
Oh et al. [22]	1Xe	3Xe	2Xe	6Xe
Ko et al. [23]	1Xe	2Xe	3Xe	6Xe
Zhang et al. [26]	1 Xe	$2\mathcal{ECM}$	$1\mathcal{ECM}$	1Xe $+ 3\mathcal{ECM}$
Our Scheme	-	$3\mathcal{HECM}$	$4\mathcal{HECM}$	7 $\mathcal{HECM}$

Table 3. Computation cost comparison in milliseconds between Our Scheme and those Kim et al. [21], Oh et al. [22], Ko et al. [23], and Zhang et al. [26].

Schemes	Key Update	Sender	Receiver	Total
Kim et al. [21]	8 × 1.25 + 5 × 4.31 = 31.55	5 × 4.31 + 6 × 1.25 = 29.05	3 × 4.31 + 3 × 1.25 + 4 × 14.90 = 76.28	17 × 1.25 + 13 × 4.31 + 4 × 14.90 = 136.88
Oh et al. [22]	1 × 1.25 = 1.25	3 × 1.25 = 3.75	2 × 1.25 = 2.5	6 × 1.25 = 7.5
Ko et al. [23]	1 × 1.25 = 1.25	2 × 1.25 = 2.5	3 × 1.25 = 3.75	6 × 1.25 = 7.5
Zhang et al. [26]	1 × 1.25 = 1.25	2 × 0.97 = 1.94	1 × 0.97 = 0.97	1 × 1.25 + 3 × 0.97 = 4.16
Our Scheme	-	3 × 0.48 = 1.44	4 × 0.48 = 1.92	7 × 0.48 = 3.36

For more details, we used the following cost reduction formula: $\frac{\textit{Existing Scheme}-\textit{Newly Proposed Scheme}}{\textit{Existing Scheme}} * 100$ [29]. The following computation shows how the proposed scheme provides secure communication with a reduced amount of computation compared to the schemes that are proposed in Kim et al. [21], Oh et al. [22], Ko et al. [23], and Zhang et al. [26], respectively.

1. Computational cost reduction process between the newly proposed scheme and Kim et al. [21], which is represented and processed as $\frac{\text{Kim et al.}-\textit{Newly Proposed Scheme}}{\text{Kim et al.}} * 100 = \frac{136.88-3.36}{136.88} * 100 = 97.54\%$.

2. Computational cost reduction process between the newly proposed scheme and Oh et al. [22], which is represented and processed as $\frac{\text{Oh et al.}-\textit{Newly Proposed Scheme}}{\text{Oh et al.}} * 100 = \frac{7.5-3.36}{7.5} * 100 = 55.2\ \%$.
3. Computational cost reduction process between the newly proposed scheme and Ko et al. [23], which is represented and processed as $\frac{\text{Ko et al.}-\textit{Newly Proposed Scheme}}{\text{Ko et al.}} * 100 = \frac{7.5-3.36}{7.5} * 100 = 55.2\ \%$.
4. Computational cost reduction process between the newly proposed scheme and Ping et al. [26], which is represented and processed as $\frac{\text{Zhang et al. }-\textit{Newly Proposed Scheme}}{\text{Zhang et al.}} * 100 = \frac{4.16-3.36}{4.16} * 100 = 19.23\ \%$.

So, we can conclude that the proposed scheme is significantly more efficient by 97.54% compared to [21], 55.2% compared to [22], 55.2% compared to [23], and 19.23% compared to [26] regarding computational cost.

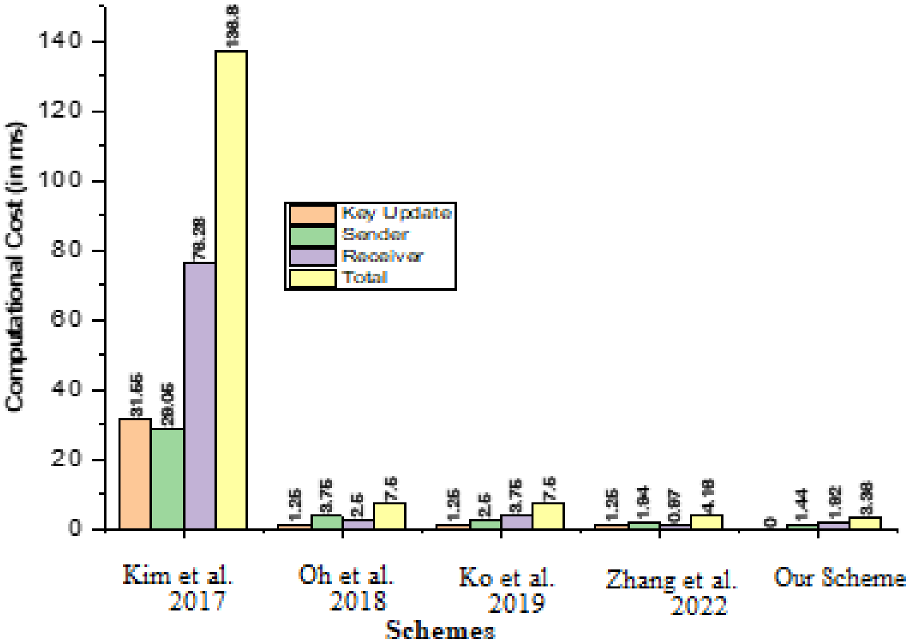

Figure 3. Computation cost comparison in milliseconds between Our Scheme and those Kim et al. [21], Oh et al. [22], Ko et al. [23], and Zhang et al. [26].

8. Communication Overhead

This section compares the efficiency of the proposed scheme with the other relevant schemes of Kim et al. [21], Oh et al. [22], Ko et al. [23], and Zhang et al. [26] in term of communication overhead. This comparison is based on extra parameters being sent with the message, which include the current timestamp size, bilinear pairing ($|\Psi|$), parameter size ($|\boldsymbol{G}|$), hash value ($|\boldsymbol{\mathcal{H}}|$), elliptic-curve point size ($|\boldsymbol{\mathcal{Q}}|$), and hyperelliptic-curve ($|\boldsymbol{n}|$) divisor size, respectively. We assume $|\mathcal{M}| = 1024\ bits$, $|\Psi| = 1024\ bits$, $|G| = 1024\ bits$, $|\mathcal{H}| = 256$ $|\mathcal{Q}| = 160\ bits,\ and\ |n| = 80\ bits$. The comparative analysis is performed in Table 4 using the above values between the proposed scheme, Kim et al. [21], Oh et al. [22], Ko et al. [23], and Zhang et al. [26]. We can conclude from Table 4 and Figure 4 that our proposed strategy clearly outperforms the [21–23,26] schemes in both characteristics.

Table 4. Communication overhead analysis between Our Scheme and those Kim et al. [21], Oh et al. [22], Ko et al. [23], and Zhang et al. [26].

Schemes	Communication Overheads	Communication Overheads in Bits
Kim et al. [21]	$\lvert\mathcal{M}\rvert + 6\lvert G\rvert$	$6 * 1024 + 1024 = 7168\ bits$
Oh et al. [22]	$\lvert\mathcal{M}\rvert + 2\lvert\mathfrak{P}\rvert + \lvert\mathcal{H}\rvert$	$1024 + 2 * 1024 + 256 = 3328\ bits$
Ko et al. [23]	$\lvert\mathcal{M}\rvert + 3\lvert\mathfrak{P}\rvert$	$1024 + 3 * 1024 = 4096\ bits$
Zhang et al. [26]	$\lvert\mathcal{M}\rvert + 2\lvert\mathcal{Q}\rvert$	$1024 + 2 * 160 = 1344\ bits$
Our Scheme	$\lvert\mathcal{M}\rvert + 2\lvert\mathfrak{n}\rvert$	$1024 + 2 * 80 = 1184\ bits$

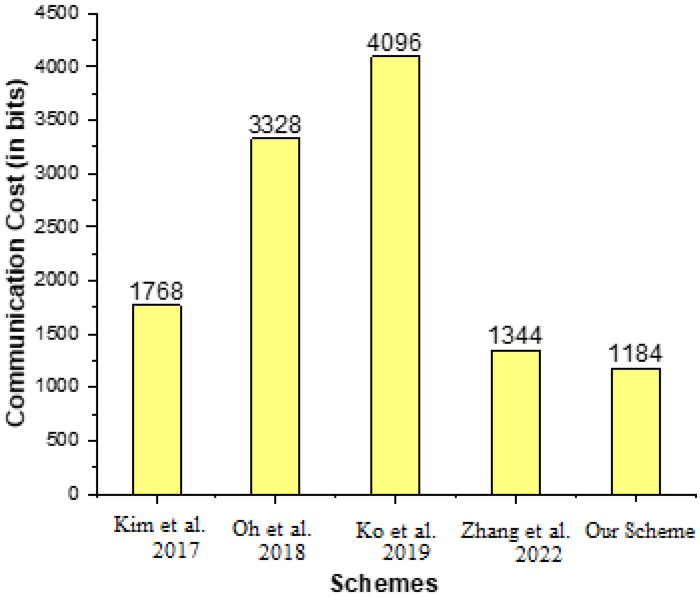

Figure 4. Communication cost comparison in bits between Our Scheme and those Kim et al. [21], Oh et al. [22], Ko et al. [23], and Zhang et al. [26].

For more details, we used the following overhead reduction formula: $\frac{\textit{Existing Scheme}-\textit{Newly Proposed Scheme}}{\textit{Existing Scheme}} * 100$ [29]. The following computation shows how the proposed scheme provides secure communication with a reduced amount of computation compared to the schemes that are proposed in Kim et al. [21], Oh et al. [22], Ko et al. [23], and Zhang et al. [26], respectively.

1. Communication overheads reduction process between the newly proposed scheme and Kim et al. [21], which is represented and processed as $\frac{\text{Kim et al.}-\textit{Newly Proposed Scheme}}{\text{Kim et al.}} * 100 = \frac{7168-1184}{7168} * 100 = 83.48\%$.
2. Communication overheads reduction process between the newly proposed scheme and Oh et al. [22], which is represented and processed as $\frac{\text{Oh et al } -\textit{Newly Proposed Scheme}}{\text{Kim et al.}} * 100 = \frac{3328-1184}{3328} * 100 = 64.42\%$.
3. Communication overheads reduction process between the newly proposed scheme and Ko et al. [23], which is represented and processed as $\frac{\text{Ko et al.}-\textit{Newly Proposed Scheme}}{\text{Ko et al.}} * 100 = \frac{4096-1184}{4096} * 100 = 71.09\%$.
4. Communication overheads reduction process between the newly proposed scheme and Zhang et al. [26], which is represented and processed as $\frac{\text{Zhang et al. } -\textit{Newly Proposed Scheme}}{\text{Zhang et al.}} * 100 = \frac{1344-1184}{1344} * 100 = 11.90\%$.

So, we can conclude that the proposed scheme is significantly more efficient by 83.48% compared to [21], 64.42% compared to [22], 71.09% compared to [23], and 11.90% compared to [26] regarding communication overheads.

9. Conclusions

To remove the problem of key escrow in existing forward-secure signature schemes, in this paper we have proposed a certificateless forward-secure signature scheme based on the hyperelliptic curve for the Internet-of-Things environment. The security analysis of this newly designed scheme is performed under the random oracle model (ROM), in which we have shown the proposed scheme safeguarded from type 1 and type 2 adversaries regarding forgeability and forward security requirements. The computational cost and communication overheads comparisons show that the proposed scheme is significantly efficient compared to existing similar schemes. From the above discussion, we have concluded that the proposed scheme has good quality such as being key-escrow-free, unforgeable, forward-secure, and having low computational cost and low communication overheads. With these qualities, it would be a suitable approach for resource-hungry IoT devices which can communicate with each other using the open Internet.

Author Contributions: Conceptualization, T.A.S., I.U. and M.A.K.; methodology, T.A.S., I.U., M.A.K., P.L. and N.I.; software, I.U.; M.A.K. and P.L.; validation, T.A.S., P.L. and I.U.; formal analysis, I.U. and M.A.K.; investigation, I.U. and M.A.K.; resources, P.L. and N.I.; data curation, M.A.K. and I.U.; writing—original draft preparation, T.A.S., I.U., M.A.K., P.L. and N.I.; writing—review and editing, M.A.K. and I.U.; visualization, P.L.; funds acquisitions, N.I.; supervision, I.U. and M.A.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by AlMaarefa University, Riyadh, Saudi Arabia (TUMA-2021- 57).

Data Availability Statement: Not applicable.

Acknowledgments: Nisreen Innab would like to express her gratitude to AlMaarefa University, Riyadh, Saudi Arabia, for providing funding (TUMA-2021-57) to conduct this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fazeldehkordi, E.; Grønli, T.-M. A Survey of Security Architectures for Edge Computing-Based IoT. *IoT* **2022**, *3*, 332–365. [CrossRef]
2. Dilberoglu, U.M.; Gharehpapagh, B.; Yaman, U.; Dolen, M. The Role of Additive Manufacturing in the Era of Industry 4.0. *Procedia Manuf.* **2017**, *11*, 545–554. [CrossRef]
3. Williams, P.; Dutta, I.K.; Daoud, H.; Bayoumi, M. A Survey on Security in Internet of Things with a Focus on the Impact of Emerging Technologies. *Internet Things* **2022**, *19*, 100564. [CrossRef]
4. Villa-Henriksen, A.; Edwards, G.T.C.; Pesonen, L.A.; Green, O.; Sørensen, C.A.G. Internet of Things in Arable Farming: Implementation, Applications, Challenges and Potential. *Biosyst. Eng.* **2020**, *191*, 60–84. [CrossRef]
5. Khan, M.A.; Kumar, N.; Mohsan, S.A.H.; Khan, W.U.; Nasralla, M.M.; Alsharif, M.H.; Zywiolek, J.; Ullah, I. Swarm of UAVs for Network Management in 6G: A Technical Review. *IEEE Trans. Netw. Serv. Manag.* **2022**. [CrossRef]
6. Ullah, I.; Alkhalifah, A.; Althobaiti, M.M.; Al-Wesabi, F.N.; Hilal, A.M.; Khan, M.A.; Ming-Tai Wu, J. Certificate-Based Signature Scheme for Industrial Internet of Things Using Hyperelliptic Curve Cryptography. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 7336279. [CrossRef]
7. Majeed, R.; Abdullah, N.A.; Mushtaq, M.F.; Kazmi, R. Drone Security: Issues and Challenges. *Int. J. Adv. Comput. Sci. Appl.* **2021**, *12*. [CrossRef]
8. Xiang, D.; Li, X.; Gao, J.; Zhang, X. A Secure and Efficient Certificateless Signature Scheme for Internet of Things. *Ad. Hoc. Netw.* **2022**, *124*, 102702. [CrossRef]
9. Cao, Y.; Xu, S.; Chen, X.; He, Y.; Jiang, S. A Forward-Secure and Efficient Authentication Protocol through Lattice-Based Group Signature in VANETs Scenarios. *Comput. Netw.* **2022**, *214*, 109149. [CrossRef]
10. Yadav, V.K.; Andola, N.; Verma, S.; Venkatesan, S. PSCLS: Provably Secure Certificateless Signature Scheme for IoT Device on Cloud. *J. Supercomput.* **2022**. [CrossRef]
11. Ullah, I.; Khan, M.A.; Abdullah, A.M.; Mohsan, S.A.H.; Noor, F.; Algarni, F.; Innab, N. A Conditional Privacy Preserving Generalized Ring Signcryption Scheme for Micro Aerial Vehicles. *Micromachines* **2022**, *13*, 1926. [CrossRef] [PubMed]
12. Ullah, I.; Khan, M.A.; Kumar, N.; Abdullah, A.M.; AlSanad, A.A.; Noor, F. A Conditional Privacy Preserving Heterogeneous Signcryption Scheme for Internet of Vehicles. *IEEE Trans. Veh. Technol.* **2022**, 1–10. [CrossRef]
13. Malkin, T.; Micciancio, D.; Miner, S. Composition and Efficiency Tradeoffs for Forward-Secure Digital Signatures. *Cryptol. Eprint Arch.* **2001**. Available online: https://eprint.iacr.org/2001/034 (accessed on 13 November 2022).

14. Itkis, G.; Reyzin, L. Forward-Secure Signatures with Optimal Signing and Verifying. *Adv. Cryptol. CRYPTO* **2001**, *2001*, 332–354. [CrossRef]
15. Kozlov, A.; Reyzin, L. Forward-Secure Signatures with Fast Key Update. *Secur. Commun. Netw.* **2003**, *2576*, 241–256. [CrossRef]
16. McCullagh, N.; Barreto, P.S.L.M. Efficient and Forward-Secure Identity-Based Signcryption. *Cryptol. Eprint Arch.* **2004**. Available online: https://eprint.iacr.org/ (accessed on 13 November 2022).
17. Boyen, X.; Shacham, H.; Shen, E.; Waters, B. Forward-Secure Signatures with Untrusted Update. In Proceedings of the 13th ACM conference on Computer and Communications Security CCS '06 2006, Alexandria, VI, USA, 30 October–3 November 2006. [CrossRef]
18. Liu, J.K.; Wong, D.S. Solutions to Key Exposure Problem in Ring Signature. *Cryptol. Eprint Arch.* **2005**. Available online: https://eprint.iacr.org/2005/427 (accessed on 13 November 2022).
19. Das, A.K.; Sharma, P.; Chatterjee, S.; Sing, J.K. A Dynamic Password-Based User Authentication Scheme for Hierarchical Wireless Sensor Networks. *J. Netw. Comput. Appl.* **2012**, *35*, 1646–1656. [CrossRef]
20. Qian, X.; Chengxiang, T.; Jun, F.; Zhijie, F.; Wenye, Z. Lattice-Based Forward Secure and Certificateless Signature Scheme. *J. Comput. Res. Dev.* **2017**, *54*, 1510. [CrossRef]
21. Kim, J.; Oh, H. Forward-Secure Digital Signature Schemes with Optimal Computation and Storage of Signers. *ICT Syst. Secur. Priv. Prot.* **2017**, *502*, 523–537. [CrossRef]
22. Oh, H.; Kim, J.; Shin, J.S. Forward-Secure ID Based Digital Signature Scheme with Forward-Secure Private Key Generator. *Inf. Sci.* **2018**, *454–455*, 96–109. [CrossRef]
23. Ko, H.; Jeong, G.; Kim, J.; Kim, J.; Oh, H. Forward Secure Identity-Based Signature Scheme with RSA. *ICT Syst. Secur. Priv. Prot.* **2019**, *562*, 314–327. [CrossRef]
24. Du, H.; Wen, Q.; Zhang, S.; Gao, M. A New Provably Secure Certificateless Signature Scheme for Internet of Things. *Ad. Hoc. Netw.* **2020**, *100*, 102074. [CrossRef]
25. Saqib, M.; Jasra, B.; Moon, A.H. A Lightweight Three Factor Authentication Framework for IoT Based Critical Applications. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 6925–6937. [CrossRef]
26. Zhang, P.; Li, Y.; Liu, M.; Shang, Y.; Fu, Z. An ECC-Based Digital Signature Scheme for Privacy Protection in Wireless Communication Network. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 1977798. [CrossRef]
27. Lu, Y.; Wang, D.; Obaidat, M.S.; Vijayakumar, P. Edge-Assisted Intelligent Device Authentication in Cyber-Physical Systems. *IEEE Internet Things J.* **2022**, 1. [CrossRef]
28. Ullah, I.; Khan, M.A.; Khan, F.; Jan, M.A.; Srinivasan, R.; Mastorakis, S.; Hussain, S.; Khattak, H. An Efficient and Secure Multi-Message and Multi-Receiver Signcryption Scheme for Edge Enabled Internet of Vehicles. *IEEE Internet Things J.* **2021**, *9*, 2688–2697. [CrossRef]
29. Ullah, I.; Amin, N.U.; Khan, M.A.; Khattak, H.; Kumari, S. An Efficient and Provable Secure Certificate-Based Combined Signature, Encryption and Signcryption Scheme for Internet of Things (IoT) in Mobile Health (M-Health) System. *J. Med. Syst.* **2020**, *45*, 4. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.