Hindawi
Security and Communication Networks
Volume 2020, Article ID 7085623, 12 pages
https://doi.org/10.1155/2020/7085623

Research Article

MHCOOS: An Offline-Online Certificateless Signature Scheme for M-Health Devices

Abigail Akosua Addobea,[1] **Jun Hou**,[2] **and Qianmu Li**[1,3,4]

[1]*School of Cyber Science and Technology, Nanjing University of Science and Technology, Nanjing 210094, China*
[2]*School Social Sciences, Nanjing Institute of Industry Technology, Nanjing 210023, China*
[3]*Intelligent Manufacturing Department, Wuyi University, Jiangmen 529020, China*
[4]*Jiangsu Zhongtian Internet Technology Co., Ltd., Nantong 226009, China*

Correspondence should be addressed to Jun Hou; houjunnjust@163.com and Qianmu Li; qianmu@njust.edu.cn

Received 24 May 2019; Revised 11 November 2019; Accepted 13 November 2019; Published 28 January 2020

Academic Editor: Huaizhi Li

Current trends of mobile technology have seen a tremendous growth in its application in smart healthcare. This has resulted in the adoption and implementation of mobile health (m-health) systems by providing health assistance to the aging population. Despite its advantageous benefits, its computational complexities cannot be overlooked. M-health devices are portable processing tiny equipment with limited computational capabilities thereby making them complex for the implementation of public key cryptosystems. In spite of this, an Offline-Online signature scheme called the MHCOOS has been proposed to solve the difficulties in the computational ability. The scheme enjoys the following benefits by splitting the signing part into both offline and online phases. The offline phase performs heavy computations when a message is absent, whereas lighter computations are performed at the online stage when a message is present. Secondly, the online computations are extremely fast due to the already computed offline signature value and lighter pairings involved. Our performance analysis demonstrates how the proposed scheme outperforms other schemes. Finally, the hardness of the scheme is proven under the Bilinear Diffie–Hellman (BDH) and Computational Diffie–Hellman (CDH) problem in the random oracle model.

1. Introduction

M-health is a current technology by which its innovation uses mobile devices or smartphones to support public health and medicinal purposes. It forms a connection between Electronic Health (E-health) and smart phone technology. The practice involves monitoring, capturing, analyzing, and processing body signals recorded from biosensors embedded in the mobile devices and transferring the information onto a virtual cloud system. The ubiquitous advantage of mobile health technology allows patients and healthcare professionals to access their data anywhere and anytime. One of the advantages the m-health program provides is the reduction of the number of outpatient's visits to the hospitals since patients can manage their health problems in their home without the need to travel to the health care units. It is an effective and a better health solution system when the patients' live very far away from their health facilities. Mobile health platforms enable health practitioners to remotely monitor their patients' health and give advice or prescriptions without the patient having to travel to the health center. It is without any doubt that mobile platforms are becoming more and more user friendly, computationally powerful, readily available and this has led innovators to begin to develop mobile apps of increasing complexity to leverage the portability these mobile platforms can offer. Some of the new mobile apps specifically target the assistance of individuals in relation to their own health and wellness management.

Other mobile apps target towards healthcare providers to improve and facilitate the delivery of patient care. With the advent of mobile health, manufacturers incorporate commercial health apps during manufacturing into mobile devices to record health data statistics such as the heart rate,

check pulses, monitor blood pressure, and check the fitness levels of patients, whereas some mobile health sensors are implanted into the body to monitor and observe the physical activity of patients. The European Commission funds a project named the MobiHealth. They explained how patients wear a lightweight monitoring system in accordance with their health needs. Their system requires shorter or longer monitoring where patients need not stay in the hospital for monitoring (http://www.mobihealth.org/).

Despite the enormous advantages m-health has to offer, the problems encountered cannot be overlooked. Most mobile devices that carry out health functions are fragile lightweight devices, with limited computational capabilities and minimal processing power. Its interactivity to large cooperated networks obstructs their functionality. Most public key cryptosystems proposed in the literature involve heavy computations, and its implementation has not been suitable for mobile health devices. Likewise, their limited processing nature makes it difficult to perform excessive computational tasks. Algorithms in security protocols involve heavy computations that impede the security performance of m-health devices.

1.1. Our Contributions. We propose an Offline-Online Certificateless Scheme for m-health devices (mobile health devices). The idea is to split the Certificateless signature into offline and online methods. The motivation for choosing both schemes was influenced by Certificateless cryptography (CL-PKC) as introduced by Al-Riyami [1].He identified the benefits of being suitable for the lightweight infrastructure. The CL-PKC dealt with the elimination of the certificate management problem in the traditional PKI and also eliminated the key escrow problem in Identity-Based Cryptography (IBC). Similarly, CL-PKC is appropriate for low-bandwidth and lower power situations such as the mobile security applications [1]. The offline-online signature methods as presented by Even et al. [2] are useful for storage-limited devices. The execution of their method makes use of the offline phase to execute excessive computations whilst the device is at the idle state and no message is available. It further stores the message without knowing the signed message [2].

MHCOOS scheme has the following advantages:

(i) It is a lightweight signature scheme that incorporates both Certificateless signature and Offline-Online Methods into one signature scheme. Thus, the Certificateless signature scheme is lightweight because the signature part is divided into both Offline and Online signing phases.

(ii) The Offline computations are performed whenever the mobile health device has not recorded any message (thus, there is no message available), and the online computations are performed when the device has recorded a message. Secondly, heavy computations occur at the Offline phase, which an offline-computed signature value is produced whilst lighter computations take place at the online phase with the already computed offline signature value.

(iii) Our scheme is attractive for mobile devices used for health applications because it does not require heavy cryptographic computations especially at the signing stage where most computations take place. Heavy computations such as bilinear pairings were not initiated which present great advantages to our scheme.

(iv) Due to the lighter computations initiated, there is optimum reduction in the overall operational overhead cost. Thus, the operational overhead cost (computation and communication cost) is much lower and insignificant.

The proposed scheme is existential unforgeable under the adaptive chosen message attack against the Type I and Type II adversaries. Furthermore, the scheme is proven to be hard under the CDH and BDH assumptions in the random oracle.

1.2. System Requirements. For every IOT health system, there are some fundamental requirements needed to achieve in the design process which are mentioned and expounded as follows:

(i) Authentication: entities within the system should register and have legitimate access to the medical server

(ii) Device traceability: unauthorized persons should not be able to track messages (health data) sent from the client's mobile device to the server during the online phase

(iii) Message availability: client's health information should be readily available at the server side for easy access by the Healthcare Terminal Point

(iv) Anti-interception attack: no unscrupulous persons can gain access to the system to alter messages between the mobile device and the server as well as the server and the Healthcare Terminal Point

(v) User anonymity: Adversaries should not be able to extract user's identity whilst the users submit their ID to the medical server during the registration phase

1.3. Related Work. Security is a major issue in the implementation of the m-health system. Many public key cryptosystems have been proposed for devices with low operational functionality. An example is the introduction of Elliptic Curve cryptography (ECC). Mana gave several important traditional cryptomethods, which fit into m-health context. He further suggested ECC to be an efficient public key cryptographic system suitable for mobile devices. The use of ECC for devices on the mobile health network is due to its smaller key sizes, but its energy requirements are far higher as compared to symmetric cryptosystems [3]. Tan and Wang [4] proposed a lightweight Identity-Based Encryption (IBE) for Body Sensor Networks (BSN). Their approach had several shortcomings: higher execution time, greater energy consumption due to increased computational overhead, and higher storage requirements because of public key storage. Some other book

of thoughts proposed several schemes desirable for devices with acute bandwidth problems. The notion of the Offline-Online digital signature scheme was proposed by Even et al. [2]. Their scheme was applicable for low power constrained devices, where any digital signature scheme can be converted into an offline and online signing methods.

Liu [5] considered their scheme [2] inefficient because of the quadratic factor increment. Most of the schemes proposed in the literature based on Identity-Based Cryptography (IBC) were suitable for most Sensor Networks but not for devices with limited computational power. However, this approach suffers from the key escrow problem where an untrusted Key Generation Center (KGC) could compute private keys of users since the KGC has the power to generate private keys.

To solve the key escrow problem, Al-riyami and Paterson [1] proposed the Certificateless cryptography where users need not worry about the compromise of their private keys. In Certificateless cryptography, the KGC computes the partial private keys after the user sends their identity. The user then computes the full private keys. It also stated in their literature that their scheme supports lightweight infrastructure with low-bandwidth requirements.

It is difficult to find a cryptographic scheme suitable for m-health, and a number of literatures written focus more on the security and privacy aspect. Other literature studies barely focused on the proposal of the cryptographic scheme for m-health devices. Zhou [6] proposed a lightweight Signcryption protocol (CLGSC) designed for data transmission in m-health systems. In our work, we focused on proposing a technique for m-health devices by splitting our Certificateless scheme into both offline and online phases to further lessen the computational time during the device operation.

1.4. Organization of the Paper. The rest of the paper is divided into the following sections. Section 2 highlights on the preliminary and complexity assumptions. In Section 3, a brief description of the Offline-Online Certificateless Signatures model is given. The formal model of the MHCOOS scheme is introduced in Section 4. Section 5 deals with the performance comparison of our scheme with other schemes in the literature. Section 6 presents the conclusion.

2. Preliminaries

This section highlights the conceptual properties of bilinear pairings. Let G_1 be an additive group of order $q(G_1, +)$ and G_2 a multiplicative group of the same order $(G_2, \times)$ and P being a generator. The structure of bilinear pairing is represented as $\hat{e}$: $G_1 \times G_1 \longrightarrow G_2$ with the following properties:

(1) Bilinearity: $\forall \ R, S, T \in G_1, \hat{e}\ (R+S, T) = \hat{e}\ (R, T)\hat{e}\ (S, T)$ and $\hat{e}\ (R, S+T) = \hat{e}\ (R, S)\hat{e}\ (R, T)$

(2) Nondegeneracy: $\hat{e}\ (P, P) \neq 1_{G_2}$

(3) Computability: there exists an efficient algorithm $\hat{e}\ (P, Q)$ for all $P, Q \in G_1$

(4) For all $u \in G_1, v \in G_2, a, b \in \mathrm{Z}, \hat{e}(u^a, v^b) = \hat{e}(u, v)^{ab}$

The bilinear maps are derived from both Weil and Tate Pairing of an elliptic curve over a finite field. Boneh and Franklin [7] gave a more detailed approach on Bilinear Pairings on Tate and Weil pairings and elliptic curves for efficiency and security.

2.1. Complexity Assumptions. This paper is based on the following computational assumptions which are assumed to be hard to break by an attacker by any probabilistic polynomial time (PPT) algorithm:

(a) *Discrete Logarithmic Problem (DLP).* Given an instance $(g, g^a) \in G_1$ with g as the generator and $a \in Z_r^*$, where a is unknown. The discrete logarithmic problem (DLP) in G requires the value of a to be computed. Thus, the advantage for any probabilistic polynomial time algorithm $\mathcal{A}$, computing a is negligibly small.

(b) *Computational Diffie–Hellman Problem (CDH).* Given $(g, g^a, g^b) \in G_1$ with generator g and $a, b \in Z_r^*$, where a, b are unknowns. Our task is to compute $C = g^{ab}$ in G_1. The CDH problem is assumed to be a computationally hard problem. This means that for any probabilistic polynomial time algorithm $\mathcal{A}$, the advantage of computing the algorithm is negligibly small.

(c) *Bilinear Diffie–Hellman Parameter Generator (BDH-PG).* A Bilinear Diffie–Hellman parameter generator (BDH-PG) is defined as the probabilistic polynomial time- (PPT-) bounded algorithm that takes the security parameter $k \in Z_r^*$ as the input and generates a tuple $(r, G_1, G_2, \hat{e}, P)$.

(d) MHCOOS scheme is secure against Type i adversary if the probability that an adaptively chosen message $\mathrm{Adv}^{\mathrm{BDH-CMA}}_{\mathrm{MLCOOS},A_i}(k)$ can win Game i where $i = 1, 2$. The MHCOOS scheme is secure if $\mathrm{Adv}^{\mathrm{BDH-CMA}}_{\mathrm{MLCOOS},A_i}(k)$ is negligible. Thus, $\mathrm{Adv}^{\mathrm{BDH-CMA}}_{\mathrm{MLCOOS},A_i}(k) \leq \varepsilon$.

(e) MHCOOS is existentially unforgeable against adaptive message attack if it is secure against adversary i. Thus, $\mathrm{Adv}^{\mathrm{BDH-CMA}}_{\mathrm{MLCOOS},A_i}(k) \leq \varepsilon$ holds, respectively.

3. Formal Model of the Offline-Online Certificateless Signature Scheme

In this section, we provide a conventional model of an Offline-Online Certificateless Signature (OOCS) Scheme. The OOCS scheme consists of six polynomial time algorithms. Table 1 presents the symbols and notations used in this paper with their corresponding meanings.

3.1. Syntax

(1) *Setup*. KGC chooses 1^k as a security parameter, returns a master secret key msk, and publishes a list of system public parameters list l.

Table 1: Key symbols used in the paper.

Symbols	Meaning
$(G_1, +)$	Additive notation in group 1
$(G_2, \times)$	Multiplication notation in group 2
H_1, H_2, H_3	Three one-way hash functions
s	Secret value selected by KGC
msk, mpk	Master secret keys and master public keys
ID_i	Identity of the user
L	Secret value of the user in the MHCOOS scheme
SK_{ID}	Private key
x_i	Secret value of the OOCS scheme
PK_{ID}	Public key
prime ord_r	Prime order r
D_{ID}	Partial private key
l	System public parameter list published by the KGC
σ	Offline signature value
δ	Online signature value
MS	Medical server unit
MD	User's mobile device
HTP	Healthcare Terminal Point

(2) *Partial-Private-Key-Extract*. This algorithm takes as inputs system public parameter list l, msk the identity of a user $\mathrm{ID}_i \in \{0,1\}^*$, and returns an output D_{ID} as the partial private key.

(3) *Set-Secret-Value*. User performs this algorithm by taking system public parameters l and a user's $\mathrm{ID}_i \in \{0,1\}^*$ as inputs and returns a secret value x_i.

(4) *Set-Private-Key*. The algorithm takes system public parameters l, the secret value x_i, the partial private key D_{ID}, and returns private key SK_{ID}.

(5) *Set-Public-Key*. The algorithm takes system public parameters l, the secret value x_i, and returns public key PK_{ID}.

(6) *CL-OffSign*. Using system public parameters l, the private key SK_{ID} of the user with identity $\mathrm{ID}_i \in \{0,1\}^*$ and without the availability of the message, this algorithm generates an offline component value σ.

(7) *CL-OnSign*. Given the message, $m \in \{0,1\}^*$, the signer's identity ID_i, the full private key SK_{ID}, and the offline component σ as the input, the signer executes this algorithm in the online phase with the availability of the message and generates the signature value δ.

(8) *Verify*. The verification algorithm performed to determine if the signature is valid or not. It takes the identity ID_i of the signer, the message $m \in \{0,1\}^*$, the Certificateless Signature δ, and the Public key PK_{ID} of the signer. The algorithm generates true if the signature δ is valid and null $\perp$ if it is invalid.

Figure 1 gives a diagrammatic approach of the respective phases of an Offline-Online scheme in the ordinary literature.

3.2. System Model. We provide a description of the entities within the MHCOOS model and their functionalities within the system in Figure 2. The MHCOOS system consists of the user's mobile device (MD), medical server collection unit (MS), and the Healthcare Terminal Point (HTP).

(a) The user's mobile device (MD) has installed sensor nodes that read, sense, and collect all vital information and store them onto to the mobile device. The MD first registers and authenticates itself to the MS. The mobile device further transfers all collected vital data to the medical server collection unit.

(b) The medical server collection unit (MS) stores the received vital information from the user's mobile device. It is responsible for the registration and authentication of the mobile clients as well as the users (doctors and nurses) from the Healthcare Terminal Point.

(c) The Healthcare Terminal Point requests for the vital information of users from the medical server collection unit. It further provides the necessary prescription in case of any detected health disorder.

4. Proposed Scheme

We propose the MHCOOS Scheme in this section. The scheme consists of six algorithms.

4.1. System Initialization Phase. The medical server firstly initializes the system by setting up the following processes using a security parameter 1^k to perform the following steps:

(a) Given two cyclic groups $(G_1, +)$ and $(G_2, \times)$ of prime order r, a pairing map $\hat{e}$: $G_1 \times G_1 \longrightarrow G_2$.

(b) $\langle P \rangle$ becomes a generator of an additive group $(G_1, +)$ of prime $\mathrm{ord}_r(P)$.

(c) The MS selects its secret value, $s \in {}_R\mathbb{Z}_r^*$ and sets $P_{\mathrm{pub}} = sP$.

(d) Chooses three one-way hash functions H_1: $\{0,1\}^* \longrightarrow G_1$ and H_2: $\{0,1\}^* \times G_1 \longrightarrow Z_r^*$, H_3: $\{0,1\}^* \times G_1 \times G_1 \longrightarrow Z_r^*$.

(e) MS performs this algorithm to generate $\{\mathrm{msk}, \mathrm{mpk}\}$: master secret keys and master public keys, respectively. Then, publishes in the public directory list $l = G_1, G_2, \hat{e}, r, P_{\mathrm{pub}}, H_1, H_2, H_3$.

4.2. Registration Phase. The mobile user registers its identity, ID with the medical server MS. The MS fetches the public directory list l, its master secret key, msk , and obtains the user's identity, $\mathrm{ID} \in \{0,1\}^*$ from the user to register the user's details in the system by making the following computations:

(a) Compute $Q_{\mathrm{ID}} = H_1(\mathrm{ID})$; hashes the user's identity

(b) Compute partial private key, $D_{\mathrm{ID}} = sH_1(\mathrm{ID}) = sQ_{\mathrm{ID}}$

4.3. Key Setup Phase. The user obtains the already computed Partial Private Key from MS and further sets up its device registration by firstly generating a secret value. It then

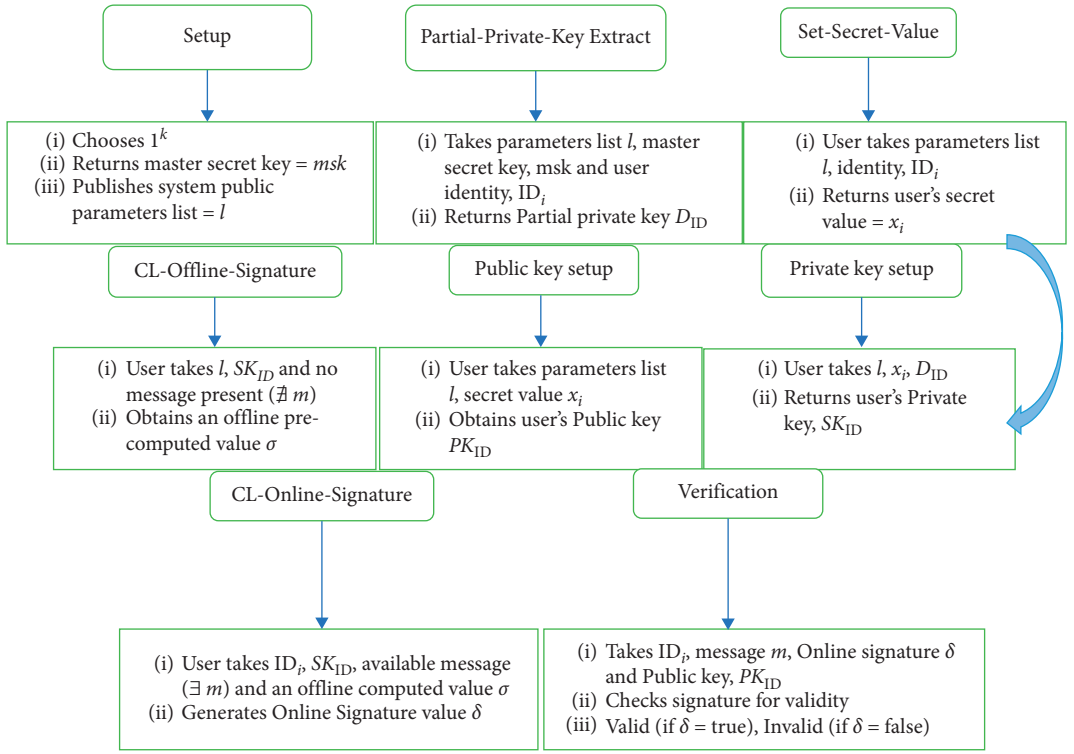


Figure 1: Descriptive model of the OOCS scheme. The diagram describes the respective phases of an ordinary Offline-Online scheme in the literature.

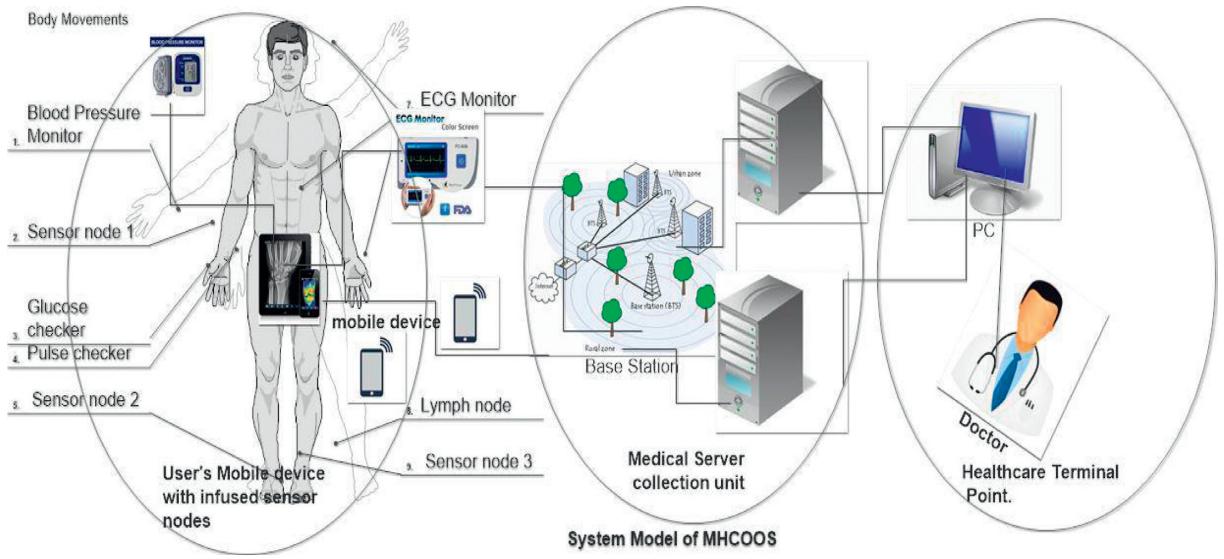


Figure 2: A typical mobile health (m-health) model.

further computes its full private key and public key, respectively.

(a) *Set-Secret-Value*. The user ID randomly picks a secret value $L \in Z_r^*$.

(b) *Set-Private-Keys*. With the secret value L and with partial Private key D_{ID}, user generates its full, Private key $SK_{ID} = (1/(L + sH_1(ID)))P$

(c) *Set-Public-Key*. User sets its public key $PK_{ID} = LP_{Pub}$

4.4. Authentication Phase. The device of the mobile user performs various signing processes at both stages to authenticate itself and transmit the captured health data to the medical server (MS).

4.5. Signing Phase. This stage of the algorithm is split into two, namely, CL-Offline signature and CL-Online signature, respectively. The algorithm works as follows.

4.5.1. CL-Offline Signature. Usually, there is no message present; thus, the mobile device has not recorded any health activity such as checking pulses or the heart rate and any other activities. It performs the following minor operations to generate an offline signature value σ used to authenticate itself to the MS.

This part of the signing algorithm uses the following parameter public directory list l, SK_{ID}, user $\mathrm{ID} \in \{0,1\}^*$, without the presence of a message, $(m = \varnothing)$ to perform the following operations to generate an offline signature value, σ.

(a) Choose randomly $s_1, t \in {}_R Z_r^*$

(b) Compute $U = s_1 P$

(c) Set $Y = H_2(U, \mathrm{ID}, PK_{\mathrm{ID}})$

(d) Compute $X = tSK_{\mathrm{ID}}$

Returns Offline signature value σ, where $\sigma = (U, Y, t, s_1)$.

4.5.2. CL-Online Signature. During the online signature phase, when the mobile device has recorded some health activities, thus with the presence of a message $(m \neq \varnothing)$, it performs the following online operations with the already offline computed signature value and transmits them securely on to the medical server, MS. The MS further stores these values in a secure form till information is requested.

(a) Compute $h = H_3(m, U, \mathrm{ID}_i, SK_{\mathrm{ID}})$

(b) Compute $\theta = s_1 h t^{-1} \bmod p$

(c) Output online signature value $\delta = (U, X, \theta)$

4.6. Verify. At this stage, the Healthcare Terminal Point accesses the MS to request for the user's data and also verifies the veracity of user's health data.

(a) Compute $h = H_3(m \in \{0,1\}^*, U, \mathrm{ID}_i, SK_{\mathrm{ID}})$

(b) If $\hat{e}(X\delta, LP + P_{\mathrm{pub}}) = \hat{e}(U, P)^h$, accept signature

(c) If $\hat{e}(X\delta, LP + P_{\mathrm{pub}}) \neq \hat{e}(U, P)^h$, reject signature

4.7. Correctness for Signature. The HTP further verifies using the correctness signature which is as follows:

$$\begin{aligned}
\hat{e}:\ \left(X\theta, LP + P_{\mathrm{pub}}\right) &= \hat{e}:\ (U, P)^h \\
&= \hat{e}:\ \left(tSK_{\mathrm{ID}} s_1 h t^{-1} P, LP + sP\right) \\
&= \hat{e}:\ \left(tSK_{\mathrm{ID}} s_1 h t^{-1} P, (L + s)P\right) \\
&= \hat{e}:\ \left(t \frac{1}{(L+s)} P s h t^{-1}, (L + s)P\right) \quad (1) \\
&= \hat{e}:\ \left(s_1 hP, P\right) \\
&= \hat{e}:\ \left(s_1 P, P\right)^h \\
&= \hat{e}:\ (U, P)^h.
\end{aligned}$$

The proposed algorithm MHCOOS scheme performs better in the sense that the offline-online approach introduced at the signature stage is to reduce excess computational cost and communication overhead. No pairing computation is adopted at the signature stage owing to the fact that pairing computations are time consuming and are slower to execute when compared to other cryptographic computations like the scalar multiplication and hashing. At the offline stage, there is no message computation whilst minimal offline computations take place to generate an offline-computed value. When the mobile device records a message (health data), the online signature uses the message and the precomputed offline value to generate the online signature. This method promotes faster and quicker signature execution process.

4.8. Security Analysis

Theorem 1. *MHCOOS Scheme is proved to be existentially unforgeable (EUF-CMA) in the random oracle under the CDH assumption problem in G_1; if Type 1 adversary A_I can win the game with advantage ε at time T, it can make the following queries q_{H_i} to the Hash oracles H_i (where $i = 1, 2, 3$), q_E queries to the private-key extraction oracle, q_{PK} queries to the public-key request oracle, and q_{sig} queries to the signing oracle, and then the BDH problem can be solved with probability.*

$$\begin{aligned}
\epsilon' &> \left(\epsilon - \frac{3^k q_{\mathrm{sig}} \left(q H_2 + q_{\mathrm{sig}} q_E\right) + 2^{(2 - q_{H1})}}{q_E \left(q_E q_{H_1} + 1\right)} 2^{-k} \right), \\
T &= t' + O\left(q_{\mathrm{sig}} + k\right)^{t_p} + O\left(q_{H1} q H_2 + q_E q_{H1} q H_2\right) t_e,
\end{aligned} \quad (2)$$

where T represents the total running time; the adversary would perform various queries. t_p is the time to perform one pairing operation and t_e is the time to compute one exponentiation in G_2.

Proof. The main purpose of the Challenger C is to compute $abcP$ from a tuple (P, aP, bP, cp) with the assumption that there exists an adversary A_I capable of attacking the MHCOOS scheme with the above advantage. □

4.8.1. System Initialization Phase. Let P be a generator of the group and a be an unknown master key. The Challenger C sets $P_{\mathrm{pub}} = aP$. The Challenger then updates an initially empty list l_i containing the tuple $l_i = (\mathrm{ID}_i, D_{\mathrm{ID}}, SK_{\mathrm{ID}}, PK_{\mathrm{ID}})$. During the game, A_I starts issuing various queries in q_{Hi} as follows:

(i) H_1 queries: the adversary A_I is allowed to make q_{Hi} number of queries to the oracle H_i with a list identity ID_i. A_I selects $j \in {}_R[1, q_{H1}]$, where q_{H1} denotes the maximum number of queries. An identity ID_i is submitted to the oracle H_1, where $i \in {}_R[1, q_{H1}]$. The Challenger C checks if $i = j$ and $\mathrm{ID}_i = \mathrm{ID}^*$; if this is true, it updates a list l_1 containing the tuple $l_1 = (\mathrm{ID}_i, Q_i, y_i)$ and set $Q_i = bP$ and $y_i = \perp$ (to indicate failure). If $i \neq j$ and $\mathrm{ID}_i \neq \mathrm{ID}^*$, the challenger gets y_i

and randomly sets $Q_i = y_i P$ and saves the tuple $l_1 = (ID_i, Q_i, y_i)$.

4.9. Key Setup Extraction Queries

(a) Partial key extraction queries: if $\mathrm{ID}_i = \mathrm{ID}^*$, C performs a number of tasks and updates l with $(SK_{\mathrm{ID}}, PK_{\mathrm{ID}})$, respectively, after getting an identity ID_i query from A_I. The tasks are as follows: C checks if $\{l = (\mathrm{ID}_i, D_{\mathrm{ID}}, SK_{\mathrm{ID}}, PK_{\mathrm{ID}}), D_{\mathrm{ID}} = \perp\}$. If both conditions are true, C returns D_{ID} to the adversary A_I. If the conditions are false, C sets partial private key $\{D_{\mathrm{ID}} = y_i, P_{\mathrm{pub}} = y_i(aP)\}$ and returns D_{ID} to A_I and updates the list l.

By inspection, if the list $l \neq (\mathrm{ID}_i, D_{\mathrm{ID}}, SK_{\mathrm{ID}}, PK_{\mathrm{ID}})$, C updates the list $l = (\mathrm{ID}_i, D_{\mathrm{ID}}, SK_{\mathrm{ID}}, PK_{\mathrm{ID}})$ by setting the following $\{D_{\mathrm{ID}} = y_i, P_{\mathrm{pub}} = y_i(aP)$ and $(SK_{\mathrm{ID}}, PK_{\mathrm{ID}}) = \perp\}$ and adds them to the list, l.

(b) Public key extraction queries: C performs a number of tasks and updates l, respectively, based on a query made by A_I on identity ID_i. The tasks are as follows: C checks the following: $\{l = (\mathrm{ID}_i, D_{\mathrm{ID}}, SK_{\mathrm{ID}}, PK_{\mathrm{ID}})$ and $PK_{\mathrm{ID}} \neq \perp\}$. If both conditions are true, C returns PK_{ID} to the adversary A_I . If the conditions are false, C selects $L \in {}_R\mathbb{Z}_r^*$ and sets the following $\{PK_{\mathrm{ID}} = LP_{\mathrm{pub}}, SK_{\mathrm{ID}} = L\}$ and returns PK_{ID} to A_I, and then updates the list, l_1.

By inspection, if the list $l \neq (\mathrm{ID}_i, D_{\mathrm{ID}}, SK_{\mathrm{ID}}, Pk_{\mathrm{ID}})$, C updates the list l with $(SK_{\mathrm{ID}}, PK_{\mathrm{ID}})$. C selects $L^* \in {}_R\mathbb{Z}_r^*$ and sets the following $\{PK_{\mathrm{ID}} = LP_{\mathrm{pub}}, SK_{\mathrm{ID}} = L\}$ and then updates l with $(SK_{\mathrm{ID}}, PK_{\mathrm{ID}})$.

(c) Secret value extraction queries: if $\mathrm{ID}_i = \mathrm{ID}^*$, C performs a number of tasks and updates the list, l with $(SK_{\mathrm{ID}}, D_{\mathrm{ID}})$ after obtaining an identity ID_i query from A_I. C checks the following: $\{l = (\mathrm{ID}_i, D_{\mathrm{ID}}, SK_{\mathrm{ID}}, PK_{\mathrm{ID}}), PK_{\mathrm{ID}} = \perp, D_{\mathrm{ID}} = \perp\}$. If these conditions are true, C executes Partial Key Extraction and Public Key Extraction Queries to obtain $\{D_{\mathrm{ID}}, PK_{\mathrm{ID}} = L^* P_{\mathrm{pub}}, SK_{\mathrm{ID}} = L^*\}$, respectively.

By inspection, if the list $l \neq (\mathrm{ID}, D_{\mathrm{ID}}, SK_{\mathrm{ID}}, PK_{\mathrm{ID}})$, C executes Partial Key Extraction and Public Key Extraction Queries to obtain $\{D_{\mathrm{ID}}, (PK_{\mathrm{ID}}, SK_{\mathrm{ID}})\}$ and updates the list l with full private keys $(D_{\mathrm{ID}}, SK_{\mathrm{ID}})$, respectively.

(d) Public key replacement $(\mathrm{ID}_i, PK'_{\mathrm{ID}})$ queries: C performs the following operations and updates the list when A_I makes the query on $(\mathrm{ID}_i, PK'_{\mathrm{ID}})$. C sets $\{PK_{\mathrm{ID}} = PK'_{\mathrm{ID}}, SK_{\mathrm{ID}}\}$ if the list l contains $(\mathrm{ID}_i, D_{\mathrm{ID}}, SK_{\mathrm{ID}}, PK_{\mathrm{ID}})$. Otherwise, C sets $D_{\mathrm{ID}}, PK_{\mathrm{ID}} = PK'_{\mathrm{ID}}, SK_{\mathrm{ID}} = \perp$ and updates the list l accordingly.

(i) H_2 queries: C checks the list $l_2 = (\mathrm{ID}_i, m, \theta^*, Pk_{\mathrm{ID}}, b_i)$, following a query from A_I on $(m, \theta, PK_{\mathrm{ID}})$. It then returns the list, l_2 to A_I if the list exists. Otherwise, it adds b_i as a hash value to the list l_2 by selecting $b_i \in {}_R\mathbb{Z}_r^*$.

(ii) H_3 queries: C checks the list $l_3 = (\mathrm{ID}_i, m, \theta, PK_{\mathrm{ID}}, b_i, c_j)$, following query from A_I on $(\mathrm{ID}_i, m, \theta, Pk_{\mathrm{ID}}, c_j)$. C then returns the list, l_3 to A_I if l_3 exists. Otherwise, C adds c_j as a hash value to the list L_3 by selecting $c_j \in {}_R\mathbb{Z}_r^*$.

4.10. Queries at the Authentication Phase

(a) Signature queries: A_I queries the challenger C, for a signature on an adaptive chosen message m_i of a user ID_i. The Challenger C checks the list, $l = (\mathrm{ID}_i, D_{\mathrm{ID}}, SK_{\mathrm{ID}}, PK_{\mathrm{ID}})$. C runs Partial Key Extraction and Public Key Extraction queries, respectively, if $\{D_{\mathrm{ID}} \neq \varnothing, (SK_{\mathrm{ID}}, PK_{\mathrm{ID}}) \neq \varnothing\}$. A_I is also allowed to generate a corresponding signature of any arbitrary length message m_i with its full private key $(D_{\mathrm{ID}}, Sk_{\mathrm{ID}})$ under the condition that $\mathrm{ID}_i = \mathrm{ID}^*$ and PK_{ID} are the public key and $SK_{\mathrm{ID}} = 1/(L + a)$ as the private key, where $a, L \in \mathbb{Z}_r^*$. The signature value returned from the Challenger is not a valid signature since the public key has been replaced by A_I, and the Challenger may not know the corresponding public key.

The Challenger computes the following:

4.10.1. CL-Offline Signature

(a) Choose randomly $s_1, t, a, b \in {}_R Z_r$

(b) Compute $U = s_1^* P$ and set $s_1^* = ab$

(c) Set $Y = H_2(U, \mathrm{ID}_i, PK_{\mathrm{ID}})$

(d) Compute $X = tSK_{\mathrm{ID}}$

(e) Output offline signature σ, where $\sigma = (U, Y, t, s_1^*)$

4.10.2. CL-Online Signature

(a) Compute $c_j = H_3(m, U, \mathrm{ID}_i, SK_{\mathrm{ID}})$

(b) Compute $\theta^{**} = s_1^* ct^{-1} \bmod p$

(c) Output online signature value $\delta = (U, X, \theta)$

For hash queries, $l_3 = (\mathrm{ID}_i, m, \theta, PK_{\mathrm{ID}}, b_i, c_j)$, set $\theta^{**} = s_1^* ct^{-1} \bmod p$, and update $\theta = \theta^{**}$.

4.11. Correctness for Signature. The Correctness for Signature is depicted as follows:

$$\begin{aligned}
&\hat{e}: \left(X\theta^{**}, LP + P_{\mathrm{pub}}\right) \\
&= \hat{e}: \left(tSK_{\mathrm{ID}} s_1^* ct^{-1} P, LP + aP\right) \\
&= \hat{e}: \left(tSK_{\mathrm{ID}} s_1^* ct^{-1} P, (L + a)P\right) \\
&= \hat{e}: \left(t\frac{1}{(L + a)} Ps_1^* ct^{-1}, (L + a)P\right) \\
&= \hat{e}: (abcP, P) \\
&= \hat{e}: (P, P)^{abc}.
\end{aligned} \tag{3}$$

Hence, this is the BDH instance to the above problem which is solved for the given random list (P, aP, bP, cP), where $a, b, c \in {}_R Z_r^*$. It is assumed that the BDH problem is difficult to break by any probabilistic polynomial time (PPT) algorithm. Therefore, the MHCOOS scheme is secure under adaptive chosen message attacker A_I in the random oracle.

Theorem 2. *MHCOOS Scheme is proved to be existentially unforgeable (EUF-CMA) in the random oracle under the CDH assumption problem in G_1 if the Type II adversary A_{II} can win the game with advantage ε at time T can make the following queries q_{H_i} to the Hash oracles (H_i, where $i = 1, 2, 3$), q_E queries to the private-key extraction oracle, q_{PK} queries to the public-key request oracle, and q_{sig} queries to the signing oracle, then the CDH problem can be solved with probability.*

$$\varepsilon' > \left(\varepsilon - \frac{3^k q_{\text{sig}}\left(qH_2 + q_{\text{sig}} q_E\right) + 2^{(2-qH_1)}}{q_E\left(q_E qH_1 + 1\right)} 2^{-k} \right). \tag{4}$$

Proof. The theorem relies on the assumption that there exists an adversary A_{II} with considerable powers having the advantage to attack the scheme without any constraint. The goal is to compute abP from a tuple (P, aP, bP) with assumption that there exists an adversary A_{II} capable of attacking the MHCOOS. □

4.12. System Initialization Phase. At the Setup phase, Challenger, C sets P as the generator G_1 and sets $P_{\text{pub}} = sP$, where s is the master key of the KGC. Adversary, A_{II} can act as the dishonest KGC. C then updates an initially empty list l_i containing the list $(\text{ID}_i, SK_{\text{ID}}, PK_{\text{ID}})$ during the game and responds to the various queries in q_{H_i} as follows:

(i) H_1 queries: the adversary A_{II} makes q_{H1} number of queries to the oracle H_1 with an identity ID_i. A_{II} selects $j \in {}_R[1, q_{H1}]$, where q_{H1} denotes the maximum number of queries. The Challenger C checks if $i = j$ and $\text{ID}_i = \text{ID}^*$; if this true, it updates a list l_1 containing the tuple (ID_i, Q_i, y_i) and sets $Q_i = aP$ and $y_i = \perp$ for failure. If $i \neq j$ and $\text{ID}_i \neq \text{ID}^*$, the challenger gets y_i randomly and sets $Q_i = y_i P$ and updates the tuple (ID_i, Q_i, y_i).

4.13. Key Setup Extraction Queries

(a) Public key extraction queries: C performs number of tasks and updates l with $(SK_{\text{ID}}, PK_{\text{ID}})$ after getting an identity ID_i query from A_{II}. The tasks are as follows: C checks the following: $\{l = (\text{ID}_i, SK_{\text{ID}}, PK_{\text{ID}}), PK_{\text{ID}} = \perp\}$. If both conditions are true, C returns PK_{ID} to the adversary A_I. If the conditions are false, it sets $PK_{\text{ID}} \neq \perp$. C selects $L \in \mathbb{Z}_r^*$ and sets $\left\{ PK_{\text{ID}} = bP_{\text{pub}}, SK_{\text{ID}} = L \right\}$ and returns PK_{ID} to A_{II}. By inspection, if the tuple does not contain $(\text{ID}_i, SK_{\text{ID}}, PK_{\text{ID}})$, C updates the list l with $(SK_{\text{ID}}, PK_{\text{ID}})$ by selecting $L \in \mathbb{Z}_r^*$ and sets $\left\{ PK_{\text{ID}} = bP_{\text{pub}}, SK_{\text{ID}} = L \right\}$ and returns PK_{ID} to A_{II}.

(b) Secret value extraction queries: if $\text{ID}_i = \text{ID}^*$, C performs some tasks and updates l with SK_{ID} after getting an identity ID_i query from A_{II}. The tasks are as follows: C checks the following: $\{l = (\text{ID}_i, SK_{\text{ID}}, PK_{\text{ID}}) PK_{\text{ID}} = \perp\}$. If the conditions return true, C executes Public Key Extraction Queries to obtain $\left\{SK_{\text{ID}} = L, PK_{\text{ID}} = LP_{\text{pub}}\right\}$. By inspection, if $l \neq (\text{ID}_i, SK_{\text{ID}}, PK_{\text{ID}})$, C executes Public Key Extraction Queries to obtain $(PK_{\text{ID}}, SK_{\text{ID}})$ and updates the list l with full private keys, SK_{ID}.

(i) H_2 queries: C searches a list l_2 if it contains the tuple $(m, \theta, PK_{\text{ID}}, h_i)$, following A_{II} query on $(m, \theta, PK_{\text{ID}})$. C then returns the tuple to A_{II} if the tuple exists. Otherwise, C adds b_i as a hash value to the tuple l_2 by selecting $b_i \in {}_R\mathbb{Z}_r^*$.

(ii) H_3 queries: C searches the list $l_3 = (m, \theta, Pk_{\text{ID}}, b_i, c_j)$, following query from A_{II} on $(m, \theta, PK_{\text{ID}}, b_i)$. C then returns the list, l_3 to A_I if l_3 exists. Otherwise, C adds c_j as a hash value to the list l_3 by selecting $c_j \in {}_R\mathbb{Z}_r^*$.

4.14. Queries at the Authentication Phase

(a) Signature queries: A_{II} obtains (ID_i, m_i) and allowed query the Challenger C for a corresponding signature under the condition that $(\text{ID}_i \neq \text{ID}^*)$.

The Challenger C then searches for a list l, containing the tuple $(\text{ID}_i, SK_{\text{ID}}, PK_{\text{ID}})$. C executes Public Key extraction Queries if the following are not found $(SK_{\text{ID}}, PK_{\text{ID}})$. A_{II} is also allowed to generate a corresponding signature on any arbitrary length message m_i with its full private key $(D_{\text{ID}}, SK_{\text{ID}})$ under the condition that $\text{ID}_i = \text{ID}^*$.

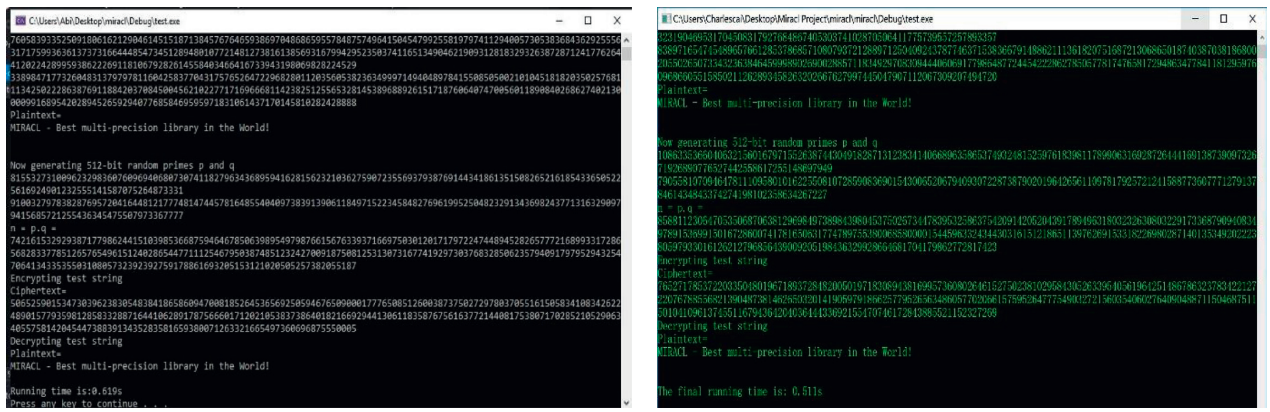
The Challenger computes the following:

$$\begin{aligned} \hat{e}: (U, P) \\ &= \hat{e}: \left(s_1{}^* P, P\right) \\ &= \hat{e}: (abP, P) \\ &= \hat{e}: (P, P)^{ab}. \end{aligned} \tag{5}$$

This is an instance to the CDH problem. It is known that the CDH problem is difficult to break by any probabilistic polynomial time (PPT) algorithm. Hence, the MHCOOS scheme is secure in CDH under adaptive chosen message attacker A_{II} in the random oracle.

5. Performance Analysis

This section presents the performance of the proposed MHCOOS scheme with other similar certificateless schemes in the literature in terms of communication cost, computational cost, and the security performance.



(a) (b)

Figure 3: Simulated results generated from message signature using the MIRACL library.

5.1. Simulation Setup Environment. The simulation environment was setup on Windows 10 Operating system on an Intel (R) Core i5-4210U CPU and 8 GB memory. We implemented our work on a Dev C++ IDE built on MINGW64.

5.1.1. Communication Cost. The simulation environment for the proposed scheme (MHCOOS) was setup on a Dev C++ IDE built on MINGW64 Windows 10 Operating system on an Intel (R) Core i5-4210U CPU using the MIRACL multiprecision library. The pairing operation is defined over a supersingular elliptic curve of $y^2 = x^3 + 1 \bmod r$ over GF (p) with 512 bits using Type 1 pairings.

The compilation time of the proposed scheme was compared with CL-SDVS [8] in Figure 3 and Table 2**.** The compilation results were generated by using a demo C++ code to test the library. The total execution time of the proposed scheme generated 1.13 s after two rounds of execution and that of the CL-SDVS [8] was 67.93. Both schemes used the MIRACL multiprecision library for its execution. MHCOOS scheme achieved a lower communication cost due to the lighter operations used in the algorithm generation. CL-SDVS [8] used a lot of pairing computations which take longer time to execute. Furthermore, it did not adopt offline/online alternative. We therefore conclude that execution process is faster when algorithms adopt an offline-online approach.

5.1.2. Computation Cost. This section compares the computational operations of the proposed scheme (MHCOOS) with other schemes in the literature. Table 3 elaborates the comparison analysis of our scheme and other schemes in text. We denoted pairing operations: p, hashing operation: h, scalar multiplication: sm, and exp: exponentiation in G_1.

According to Table 3, the proposed scheme (MHCOOS), Selvi [12] and L-OOCLS/HRAAP scheme [9] only included the Offline and Online computations at the signing stage of their algorithm. However, schemes [8, 10, 11] did not adopt offline and online methods in their signing computations. MHCOOS scheme employs 2 scalar multiplications at both offline and online stages which are lesser when compared to schemes [9, 12] at the online phase and schemes [8, 9, 11] at the offline approach except scheme [10] which has the same number of scalar multiplications with the proposed scheme.

At the verification stage, our pairing operation was slightly higher than the pairing operation in schemes [8, 9] but similar to scheme [10]. Schemes [11, 12] had the highest the number of pairing operations. The signing part of the MHCOOS scheme was split into both Offline and Online computations. During the offline computation, an offline-computed value is generated which is used in conjunction with the message (health data) to generate an online signature. No pairing computation was introduced at the signing stage due to the fact that pairing computations based on elliptic curves require heavy computational cost and extra execution time. Execution of the whole signature process is faster and quicker because at the offline stage, the device does not record any message but minute computations take place to generate a precomputed offline value.

As soon as the mobile device records an activity (receives a message), the online computation takes place using the recorded message and the precomputed offline value to generate the online signature. In the MHCOOS scheme, the user need not perform a lot of computations at the verification stage despite its 2 times pairing computation because much of the computations already took place at the signing stage. Overall, the MHCOOS scheme has proven to be of much advantage over scheme [8, 9, 12] at the signing stages and better than [11, 12] at the verification stage because our scheme adopted lesser pairing computations in both stages.

5.2. Application Scenario. In this section, an m-health practical scenario is provided to demonstrate the workflow of a secure data transmission of the entities that employ the MHCOOS scheme. First of all, mobile health (m-health) supported by e-health is a healthcare technology by which entities utilize smart devices to access their healthcare needs. It consists of an already installed mobile medical application which records the daily and fitness activities of its users

Table 2: Performance comparison-communication cost.

Scheme	Execution time for round 1 (s)	Execution time for round 2 (s)	Total Execution time (*s*)
MHCOOS (proposed scheme)	0.619	0.511	1.13
CL-SDVS [8]	—	—	67.93

Table 3: Performance comparison-communication cost.

Scheme	Signing		Verification
	Offline	Online	
L-OOCLS/HRAAP [9]	3M + 1Exp	3M	1P + 1Exp + 1M
MHCOOS scheme	2M	2M	2P + 1Exp
Liu et al. [10]	—	1P + 1Exp + 2M	2P + 1Exp
Kumar et al. [11]	—	3M	3P + 1M
Hafizul Islam and Biswas [8]	—	3P + 3M + Exp	1P + 1M + 1Exp
Selvi [12]	3M	—	6M + 4P

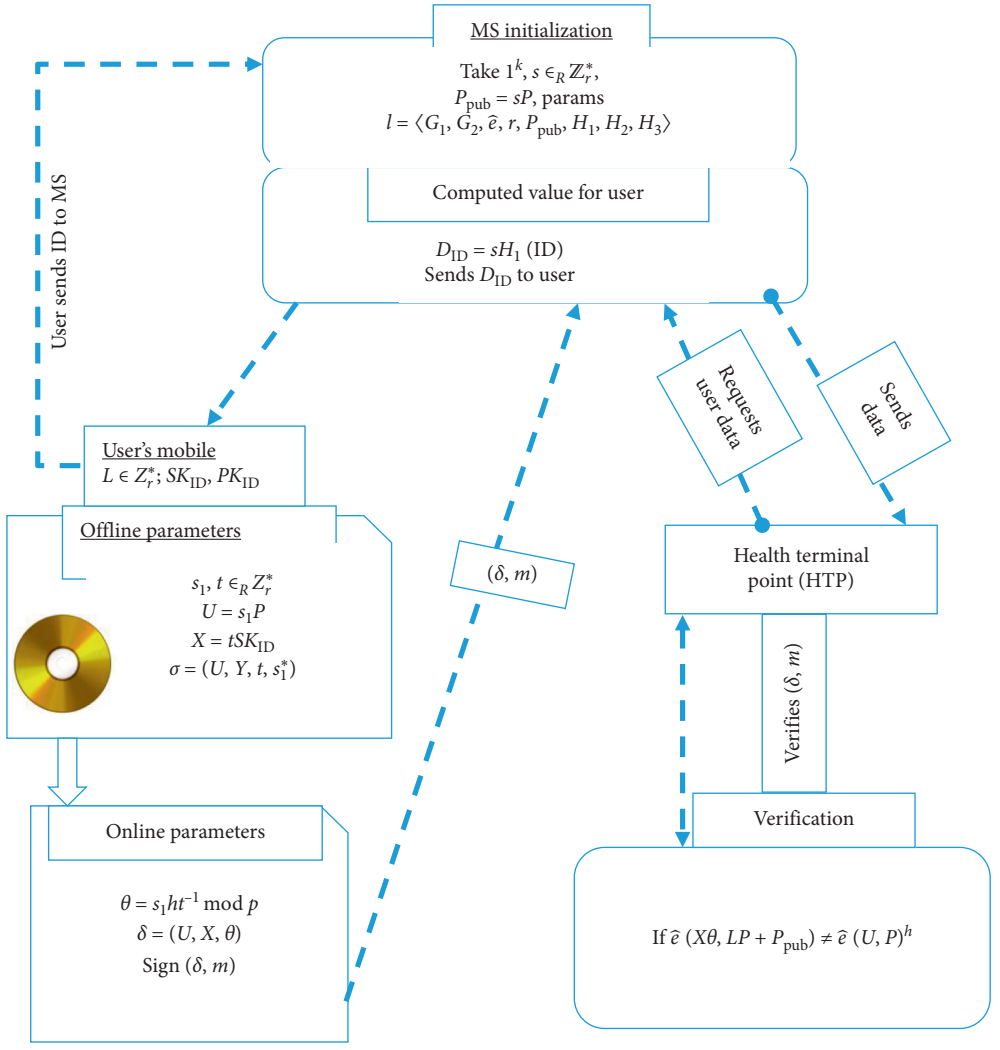


Figure 4: A toy scenario for the m-health model.

simultaneously collecting vital health data. The standard ISO TR 17522 : 2015 developed for health applications on mobile/smart devices is used to establish communication amongst entities.

The data is securely transmitted via a Bluetooth and WLAN medium onto the medical server for storage. The healthcare terminal submits the user's identity to request for their respective stored data. The data is stored at the database of the data center where the health practitioner is able to collect the recorded data of each health respondent. The communication scenario initiates the lightweight MHCOOS algorithm. It performs the offline computations when no health data is present to generate an offline-computed value. It then fully performs the online computations using the detected health data and the already offline-computed value to generate the online signature with the received health data (health data present). The various activities that take place in the MHCOOS system are well expounded in the following steps and diagramatically represented in Figure 4.

(a) The MS initializes the system by generating system setup and other parameters. The user's mobile device sends the identity of the user ID_s to MS to compute $D_{\mathrm{ID}} = sH_1(\mathrm{ID})$ for the user and transmits it securely to the user.

(b) At this stage, the health app installed on the mobile device is termed idle if it is not reading the heart beat or checking the pulse of the patient. It performs offline computations at this idle stage and generates the offline value (σ). As soon as the mobile device detects the presence of any health activity, the application starts to record the vital health data (heart rate or records his pulses). At the online stage, the application performs several computations using the already computed offline parameters with the captured data. The installed health application (health app) signs the online computed value δ on the message, thus sign (δ, m), and sends it to the MS for storage.

(c) During verification, the HTP submits the identity of the mobile user to the MS and requests for the health data and checks for the veracity of signature on the message sign (δ, m).

6. Conclusions

In this paper, we presented an MHCOOS scheme by adopting an Offline-Online approach to Certificateless signatures that are applicable to mobile devices used in the health environment. MHCOOS is a lightweight cryptographic scheme designed to support mobile devices used for health applications. Based on minimum bilinear pairings, the scheme splits the signing part into two phases: the offline phase and the online phase. The offline phase performs a lot of computational processes when a message (no record of health data) is unavailable to generate an offline computed value, whereas the online computations take place during the presence of a message. MHCOOS has been shown to be unforgeable against the Type I and Type II adversaries (A_I and A_{II}), respectively, under the adaptive chosen message attacks whilst it is subsequently proven to be intractable under the BDH and CDH assumptions in the random oracle. The scheme is shown to be lightweight and has wider applicability not only to mobile health (m-health) devices but other wearable devices. In our future works, we will look further to propose a different lightweight scheme useful for devices with wearable technology without the use of heavy cryptographic methods.

Data Availability

The data used in running the simulation were download from the Miracl Github repository from the below website: https://github.com/miracl/MIRACL. A demo code from this site https://github.com/miracl/MIRACL/blob/master/source/pk-demo.cpp was used to test pk-demo.cpp of the library file.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This paper was supported by Fundamental Research Funds for the Central Universities (no. 30918012204), Military Common Information System Equipment Pre-Research Special Technology Project (315075701), 2019 Industrial Internet Innovation and Development Project from the Ministry of Industry and Information Technology of China, and 2018 Jiangsu Province Major Technical Research Project "Information Security Simulation System," Shanghai Aerospace Science and Technology Innovation Fund (SAST2018-103).

References

[1] S. S. Al-riyami and K. G. Paterson, "Certificateless public key cryptography," *Advances in Cryptology—ASIACRYPT 2003*, Springer, Berlin, Germany, 2003.

[2] S. Even, O. Goldreich, and S. Micali, "On-line/off-line digital signatures," *Advances in Cryptology—CRYPTO' 89 Proceedings*, pp. 263–275, 1990.

[3] M. Mana, "Trust key management scheme for wireless body area networks," *International Journal of Network Security*,vol. 12, no. 2, pp. 71–79, 2011.

[4] C. C. Tan and H. Wang, "Body sensor network Security : an identity-based cryptography approach," in *Proceedings of the First ACM Conference on Wireless Network Security—WiSec '08*, Alexandria, VA, USA, April 2008.

[5] J. K. Liu, *Efficient Online/Offline Identity-Based Signature for Wireless Sensor Network*, Institute for Infocomm Research, Singapore, 2010.

[6] C. Zhou, "Comments on "Light-Weight and robust security-aware D2D-assist data transmission protocol for mobile-health systems"," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 7, pp. 1869-1870, 2018.

[7] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," *SIAM Journal on Computing*, vol. 32, no. 3, pp. 586–615, 2003.

[8] S. Hafizul Islam and G. P. Biswas, "Provably secure certificateless strong designated verifier signature scheme based on

elliptic curve bilinear pairings," *Journal of King Saud University—Computer and Information Sciences*, vol. 25, no. 1, pp. 51–61, 2013.

[9] M. E. S. Saeed, Q.-Y. Liu, G. Tian, B. Gao, and F. Li, "Remote authentication schemes for wireless body area networks based on the Internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4926–4944, 2018.

[10] J. Liu, Z. Zhang, X. Chen, K. Sup, and K. Member, "Certificateless remote anonymous authentication schemes for wireless body area networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 332–342, 2014.

[11] P. Kumar, S. Kumari, V. Sharma, A. K. Sangaiah, J. Wei, and X. Li, "A certificateless aggregate signature scheme for healthcare wireless sensor network," *Sustainable Computing: Informatics and Systems*, vol. 18, pp. 80–89, 2018.

[12] S. S. D. Selvi, "Efficient certificateless online/offline signature with tight security," *Journal of Internet Services and Information Security*, vol. 2, no. 3/4, pp. 77–92, 2012.

[13] M. C. Gorantla and A. Saxena, "An efficient certificateless signature scheme," *Computational Intelligence and Security*, pp. 110–116, Springer, Berlin, Germany, 2005.

[14] A. Ge, S. Chen, and X. Huang, "A concrete certificateless signature scheme without pairings," in *Proceedings of the 2009 International Conference on Multimedia Information Networking and Security*, vol. 2, pp. 374–377, Hubei, China, November 2009.

[15] Y.-C. Chen, R. Tso, G. Horng, C.-I. Fan, and R.-H. Hsu, "Strongly secure certificate less signature: cryptanalysis and improvement of two schemes," *Journal of Information Science and Engineering*, vol. 31, no. 1, pp. 297–314, 2015.

[16] A. C.-C. Yao and Y. Yunlei Zhao, "Online/offline signatures for low-power devices," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 2, pp. 283–294, 2013.

[17] Y. Sun, Z. Zhang, and L. Shen, "A revocable certificateless signature scheme without pairing," *Cloud Computing and Security*, vol. 10039, pp. 355–364, springer, Berlin, Germany, 2016.

[18] Y. Xie, S. Zhang, X. Li, Y. Li, and Y. Chai, "CasCP: efficient and secure certificateless authentication scheme for wireless body area networks with conditional privacy-Preserving," *Security and Communication Networks*, vol. 2019, Article ID 5860286, 13 pages, 2019.

[19] S. Li, J. Cui, H. Zhong, Y. Zhang, and Q. He, "LEPA: a lightweight and efficient public auditing scheme for cloud-assisted wireless body sensor networks," *Security and Communication Networks*, vol. 2017, Article ID 4364376, 16 pages, 2017.

[20] A. Adavoudi-Jolfaei, M. Ashouri-Talouki, and S. F. Aghili, "Lightweight and anonymous three-factor authentication and access control scheme for real-time applications in wireless sensor networks," *Peer-to-Peer Networking and Applications*, vol. 12, no. 1, pp. 43–59, 2019.

[21] K.-A. Shim, "Universal forgery attacks on remote authentication schemes for wireless body area networks based on Internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 9211-9212, 2019.

[22] Z. Xu, X. Liu, G. Zhang, and W. He, "McCLS: certificateless signature scheme for emergency mobile wireless cyber-physical systems," *International Journal of Computers Communications & Control*, vol. 3, no. 4, pp. 395–411, 2008.

[23] D. Stebila, An introduction to provable security, 2014.

[24] J. Liu, Z. Zhang, R. Sun, and K. S. Kwak, "An efficient certificateless remote anonymous authentication scheme for wireless body area networks," in *Proceedings of the 2012 IEEE International Conference on Communications (ICC)*, pp. 3404–3408, Ottawa, ON, Canada, June 2012.

[25] J. Hanen, Z. Kechaou, and M. B. Ayed, "An enhanced healthcare system in mobile cloud computing environment," *Vietnam Journal of Computer Science*, vol. 3, no. 4, pp. 267–277, 2016.

[26] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," *Advances in Cryptology - ASIACRYPT 2003*, pp. 1–40, Springer, Berlin, Germany, 2003.

[27] Ernst and Young, *mHealth: Mobile Technology Poised to Enable a New Era in Health Care*, pp. 1–54, 2012, https://www.ey.com/Publication/vwLUAssets/mHealth/$FILE/mHealth%20Report_Final_19%20Nov%2012.pdf.

[28] L. Wu, Z. Xu, D. He, and X. Wang, "New certificateless aggregate signature scheme for healthcare multimedia social network on cloud environment," *Security and Communication Networks*, vol. 2018, Article ID 2595273, 13 pages, 2018.