sensors

MDPI

Article

An Efficient Certificateless Aggregate Signature Scheme for Blockchain-Based Medical Cyber Physical Systems

Hong Shu [1,2], Ping Qi [1], Yongqing Huang [1,3], Fulong Chen [2,4,*], Dong Xie [2,4] and Liping Sun [2,4]

[1] School of Mathematics and Computer, Tongling University, Tongling 244061, China; shuhongtl@126.com (H.S.); qiping929@gmail.com (P.Q.); hyq@tlu.edu.cn (Y.H.)
[2] Anhui Provincial Key Lab of Network and Information Security, Wuhu 241002, China; xiedong@ahnu.edu.cn (D.X.); slp620@163.com (L.S.)
[3] Institute of Information Technology & Engineering Management, Tongling University, Tongling 244061, China
[4] School of Computer and Information, Anhui Normal University, Wuhu 241002, China
[*] Correspondence: long005@mail.ahnu.edu.cn; Tel.: +86-553-591-0371

Received: 4 February 2020; Accepted: 8 March 2020; Published: 10 March 2020

Abstract: Different from the traditional healthcare field, Medical Cyber Physical Systems (MCPS) rely more on wireless wearable devices and medical applications to provide better medical services. The secure storage and sharing of medical data are facing great challenges. Blockchain technology with decentralization, security, credibility and tamper-proof is an effective way to solve this problem. However, capacity limitation is one of the main reasons affecting the improvement of blockchain performance. Certificateless aggregation signature schemes can greatly tackle the difficulty of blockchain expansion. In this paper, we describe a two-layer system model in which medical records are stored off-blockchain and shared on-blockchain. Furthermore, a multi-trapdoor hash function is proposed. Based on the proposed multi-trapdoor hash function, we present a certificateless aggregate signature scheme for blockchain-based MCPS. The purpose is to realize the authentication of related medical staffs, medical equipment, and medical apps, ensure the integrity of medical records, and support the secure storage and sharing of medical information. The proposed scheme is highly computationally efficient because it does not use bilinear maps and exponential operations. Many certificateless aggregate signature schemes without bilinear maps in Internet of things (IoT) have been proposed in recent years, but they are not applied to the medical field, and they do not consider the security requirements of medical data. The proposed scheme in this paper has high computing and storage efficiency, while meeting the security requirements in MCPS.

Keywords: blockchain; privacy protection; certificateless aggregate signature; trapdoor hash function; MCPS

1. Introduction

In the big data era, with the development of Internet of Things, smart healthcare provides people with more convenient and high-quality healthcare services [1]. The Medical Cyber Physical System (MCPS) [2] is a special type of Cyber Physical System (CPS) based on the application background of the smart healthcare field, which consists of physical space and cyber space. Physical space includes wearable devices, medical diagnostic equipment, and user space consisting of doctors, nurses, etc. Cyber space is the nerve center of MCPS. It receives sensing information from physical space through a network transmission system. Then cyber space identifies, stores, analyzes, processes, and generates feedback control information. Finally, it sends control information to physical space through a network transmission system.

MCPS continuously collects the patient's physical signs data through various wearable devices and medical devices, so that the patient's physical condition can be better detected [3]. In order to provide patients with a more accurate and timely diagnosis, different medical institutions need to share a large amount of physical data collected by the sensors and healthcare staff [4]. At the same time, patient privacy should be protected. Thus, blockchain is needed to utilize peer-to-peer network and cryptography technology to achieve tamper proof, unforgeable, non-repudiation, and verifiable medical records. The combination of MCPS and blockchain [5] promotes the sharing of medical services and resources [6]. However, the block capacity limit is one of the main factors that affects the performance improvement of blockchain.

MCPS controls the embedded medical equipment through a wireless network, which senses and monitors the patient's physical data in real time. When the patient has an abnormal situation, the medical equipment sends the early warning information to the medical institution in time. Once MCPS is under cyberattacks, such as data inconsistency, unauthorized access, and data breaches [7], patients' lives and health will be seriously threatened. In practice, medical institutions need to check the accuracy and integrity of shared and sensed medical data before making medical diagnoses. The medical data, which is collected from wearable devices, medical equipment, medical apps, and healthcare staff needs the responsible healthcare provider to sign on it. A large number of signatures and verifications result in high time and space overheads. At the same time, considering the capacity limitation of the blockchain, the certificateless aggregate signature is an effective method because of its compression characteristics. In recent years, some certificateless aggregate signature schemes [8–10] have been proposed. However, the performance of these schemes is not ideal because they use more time-consuming bilinear maps. At the same security level, the Elliptic Curve Cryptography (ECC) is more efficient than bilinear maps [11]. Therefore, with the characteristics of low computation, low storage, high reliability, privacy protection, and timeliness, the certificateless aggregate signature scheme based on ECC is suitable for blockchain-based MCPS.

The contributions of this paper are as follows:

- A two-layer storage model in which medical data is stored off-blockchain and shared on-blockchain is proposed. The model meets security and privacy requirements of MCPS.
- Based on ECC, we present the multi-trapdoor hash function, which is secure and efficient to construct the certificateless aggregate signature scheme.
- The certificateless aggregate signature scheme based on the multi-trapdoor hash function is proposed in this paper. It can reduce the computation cost of wearable medical devices and miners.

The rest of this paper is organized as follows. Related works are discussed in Section 2. The necessary preliminaries are presented in Section 3. Section 4 presents a multi-trapdoor hash function. In Section 5, we describe the certificateless aggregate signature scheme. A security discussion of the proposed scheme is given in Section 6. Then, we make an efficiency analysis in Section 7. Finally, the conclusion is offered in Section 8.

2. Related Work

2.1. Blockchain

Blockchain is a decentralized, anonymous, untrusted, tamper proof, and traceable distributed data storage technology [5]. With the development of the medical industry, health data is growing exponentially. How to effectively store, share, and manage medical data involving a large number of patients' privacy has become an obstacle to the development of the healthcare industry. Due to the characteristics of blockchain [12], such as non-tamperability, traceability, and multi private key authorization management, it is possible to share medical data securely among different institutions [13].

According to the difference of open objects, blockchain can be divided into Public Blockchain, Private Blockchain, and Consortium Blockchain. These three types of blockchains are compared in Table 1. In the special field of MCPS, medical data contains both a large amount of private information and has the need to be shared between different institutions, therefore the Consortium Blockchain is more suitable for the secure storage and sharing of medical data.

Table 1. The comparison of three types of blockchains.

Blockchain Type	Public Blockchain	Private Blockchain	Consortium Blockchain
Open objects	All	Individuals or inside company	Authorized companies or organizations
Consensus mechanism	PoW, PoS, DPoS	PBFT	PBFT, Raft
Centralization	Decentralization	Centralization	Partial centralization
Typical application	Bitcoin, Ethereum	Overstock	Hyperledger, R3CEV
Characteristics	Self-building of trust	Traceability	Improvement of efficiency

Xue et al. [14] divided the existing medical institutions into medical institution federate servers (MIPS) and audit federate servers (AFS) according to their credit scores. Through the improved consensus mechanism, the medical data sharing model based on blockchain was realized. In the untrusted environment, Xia et al. [15] designed a sensitive medical data sharing model between cloud service providers based on blockchain through a smart contract and access control mechanism. The security requirements of medical records on integrity, confidentiality, and traceability can be realized by digital signature technology in the blockchain-based medical data sharing system.

In recent years, researchers have conducted in-depth research around blockchain-based multi-signatures [16], aggregate signatures [17,18], ring signatures [19], and homomorphic signatures [20]. Among them, aggregate signatures are favored for their advantages, such as fast computing speed, small storage space, and bandwidth saving. Moreover, some scholars have carried out in-depth research on the combination of quantum computing and the security of blockchain [21]. Gao et al. [21] proposed a lattice-based signature scheme and presented a cryptocurrency scheme based on post-quantum blockchain, which could resist quantum computing attacks.

2.2. *Certificateless Aggregate Signature*

In order to solve the management problems of certificate distribution and storage in the traditional PKI-based (Public Key Infrastructure) public key cryptosystem, Shamir proposed the identity-based public key cryptosystem (ID-PKC) in 1984 [22]. In ID-PKC, the public key is denoted by user information, such as mailbox, address, telephone number, etc. The private key is provided by the key generation center (KGC), a third-party trusted organization. Different from traditional public key cryptosystems, users cannot generate their own private key. For KGC, the user's private key is known, and KGC can decrypt ciphertext and forge identity at will. Therefore, ID-PKC has the defect of key escrow [23], which is only applicable to the environment with low security requirements.

To solve this problem, Al-Riyami and Paterson proposed the notion of certificateless public key cryptography (CL-PKC) in 2003 [24]. Unlike ID-PKC, the private key in CL-PKC consists of a partial private key generated by KGC and the secret value selected by the user. KGC only knows partial private key but cannot get the secret key. It can effectively solve the key escrow problem [25]. Moreover, the public key in CL-PKC does not need certificate verification, so the problem of public key authentication is solved. CL-PKC has neither the certificate management problem nor the key escrow problem. Its calculation efficiency is higher than traditional public key cryptosystems, and its security is higher than ID-PKC. Therefore, it is suitable for application scenarios with higher requirements for computing, storage efficiency, and security.

Boneh et al. first proposed the concept of aggregate signature [26] on EUROCRYPT 2003, which greatly promoted the development of digital signature cryptography. Aggregate signature [26] is suitable for compressing many signatures generated by many different users to many different messages into one short signature, and simplifying the verification of multiple signatures into one verification. Aggregation signature greatly improves storage efficiency and verification time.

In recent years, certificateless aggregate signatures (CLAS) have attracted many scholars' research interests because of the advantages of both a certificateless public key cryptosystem and aggregate signatures. Based on different theoretical foundations, scholars have proposed corresponding certificateless aggregate signature schemes. For example, most researchers proposed certificateless aggregate signature schemes based on bilinear maps [8–10]. For the first time, Gong et al. [9] proposed two certificateless identity-based aggregate signature schemes (denoted as CAS-1 and CAS-2 in [9]). In these two schemes, the aggregation verification of CAS-1 used $2n + 1$ pairing operations on an elliptic curve. CAS-2 used $n + 2$ pairing operations and n scalar point multiplication operations on elliptic curves. It is clear that the verification efficiency was very low. Xiong et al. designed a more efficient certificateless aggregate signature scheme [8]. The verification of this scheme used only three pairing operations and $2n$ scalar multiplication operations. The efficiency of the scheme was not related to the number of signers. Moreover, it did not require a synchronized clock. As such, this scheme was more efficient than the Gong's scheme [9]. However, He et al. [27] and Zhang [10] et al. pointed out that Xiong et al.'s scheme was not secure. He et al. [27] proved that Xiong et al.'s scheme was not resistant to forge attacks from $\mathcal{A}_{\mathrm{II}}$ adversary. Zhang et al. proved that Xiong et al.'s scheme could not resist coalition attacks from the honest-but-curious KGC, malicious-but-passive KGC, and inside signers.

Some scholars did not use bilinear pairs to construct certificateless aggregate signatures. Zhou et al. proposed two certificateless aggregate signature schemes without bilinear maps [28]. Based on the Elliptic Curve Discrete Logarithm Problem (ECDLP), the schemes both used $2n + 1$ scalar multiplication operations. The difference is that CLAS-2 provides a shorter constant-level signature length than CLAS-1. Cui et al. [29] proposed a certificateless aggregate signature scheme based on ECC and applied it to vehicular ad hoc networks (VANETs) communication. The verification of this scheme used n scalar multiplications. Since the computational overhead of bilinear pairs is significantly higher than that of scalar multiplication under ECC [11], Zhou's scheme and Cui's scheme had higher computational efficiency.

In recent years, with the development of blockchain technology, more and more scholars have focused on the research of the aggregation signature algorithm based on blockchain [17,18,30]. Gao et al. [18] designed a fair and efficient multi-party contract signing scheme based on blockchain by conducting a certificateless aggregation verifiable encryption signature scheme. Wang et al. [30] realized the full anonymous blockchain by homomorphic encryption, and aggregate signature technology, which effectively protected the privacy of the user's identity and the transaction amount. Neither of these schemes [18,30] is computationally efficient because they both used bilinear maps. Based on the gamma signature proposed by Yao et al. [31], Zhao [17] constructed an aggregate signature scheme without bilinear maps. By applying Zhao's scheme [17] to Bitcoin, it could be found that both computation and storage overhead have decreased to some extent, however the length of this aggregate signature scheme increased with the number of signers. Due to their low computing or communication efficiency, these schemes [17,18,30] were not suitable for wearable medical devices with limited computing and storage resources. On the other hand, these schemes [17,18,30] did not focus on the security requirements of MCPS, such as timeliness and privacy protection.

Some scholars focused on the research of digital signatures in blockchain-based Internet of things (IoT) applications [32,33]. In order to reduce the time cost of transmitting authentication information from blockchain nodes to IoT devices, Danzi et al. [32] proposed a repeat-authenticate scheme. In which blockchain information that consists of a copy of the block header and the signatures of blockchain nodes is multicasted. Kaga et al. [33] proposed a biometrics-based fuzzy signature scheme and applied it into the IoT blockchain system. This scheme achieved the verification of a creator of a transaction. These two schemes payed more attention to authentication of transaction creators or blocks in IoT scenario. However, they did not focus on the effective storage of a large number of digital signatures and the privacy protection of medical data in MCPS scenario. When a patient goes to the hospital, a great deal of medical records will be generated. The digital signatures of these medical records will occupy a large amount of block space, which will seriously affect the performance of the blockchain. At the same time, medical data involves personal privacy, and it is necessary to protect the private data.

The blockchain-based schemes mentioned above are compared in Table 2. From Table 2, we can conclude that none of these solutions [17,18,30,32,33] provide both high computing and communication efficiency. Furthermore, nowadays, certificateless aggregate signatures based on blockchain have not been widely used in MCPS. In this paper, we combine ECC and the multi-trapdoor hash function to propose a certificateless aggregate signature scheme and apply it to secure storage and sharing of MCPS. The proposed scheme provides high computing efficiency and low space occupation, which is suitable for blockchain-based MCPS scenario with limited blockchain capacity and low computing power wearable devices.

Table 2. The comparison of relevant blockchain-based schemes.

Scheme	Integrity	Authentication	Bilinear Maps	Relevance to Number of Users	MCPS	Aggregate Signature
Zhao et al. [17]	Y	Y	N	Y	N	Y
Gao et al. [18]	Y	Y	Y	Y	N	Y
Wang et al. [30]	Y	Y	Y	N	N	Y
Danzi et al. [32]	Y	Y	N	N	N	N
Kaga et al. [33]	N	Y	N	Y	N	N
Our scheme	Y	Y	N	N	Y	Y

3. Preliminaries

3.1. Elliptic Curve Discrete Logarithm

Let p, q be two large prime numbers, F_p be a finite field determined by p, and $E(F_p)$ be an elliptic curve over F_p, which is defined by the equation: $y^2 = x^3 + ax + b$ mod p, where a, b$\in F_p$ and $4a^3 + 27b^2 \neq 0$. If the additive group G consists of the infinity point O and all points on $E(F_p)$, P is a generator of group G with the order q, we have the following definition.

Definition 1 (Elliptic curve discrete logarithm problem (ECDLP) [34])**.** *Given $Q = mP$ and $Q \in E(F_p)$, the task of ECDLP is to find the integer m, where $0 \leq m \leq q - 1$.*

3.2. Trapdoor Hash Function

The trapdoor hash function is also called the chameleon function [35]. Different from general hash functions, it has a hash/trapdoor key (HaK, TrK). The hash key (HaK) is public, while the trapdoor key (TrK) is private. The trapdoor hash function uses some special information to generate a fixed hash value, and its collision resistance depends on the user's knowledge of trapdoor information (TrK) [36]. That is, without knowing the trapdoor key TrK, the trapdoor hash function is collision resistant. However, when the hash/trapdoor key is known, the trapdoor collision can be computed [37]. This property of the trapdoor hash function is suitable to construct various digital signature schemes [36–39].

The trapdoor hash function consists of the following four algorithms [37]:

- **ParG**: Inputs security parameter k, outputs system parameter *params*;
- **KeyG**: Inputs *params*, outputs hash/trapdoor key <*HaK*, *TrK*>;
- **HashG**: Inputs *params*, message m and auxiliary parameter r, outputs trapdoor value $TH_{HaK}(m, r)$;
- **TrapColG**: Inputs *params*, <*HaK*, *TrK*>, m, r, and new message $m'(\neq m)$, outputs r' and HaK' such that $TH_{HaK}(m,\ r) = \ TH_{HaK'}(m',\ r')$;

According to the number of trapdoor information (*TrK*), trapdoor hash functions include the single trapdoor hash function [35], the double trapdoor hash function [39], and the multi-trapdoor hash function [37,38]. A double trapdoor hash function usually has two pairs of hash/trapdoor keys, named long-term hash/trapdoor key and temporary hash/trapdoor key. Double trapdoor hash function protects the long-term trapdoor key from being leaked by sacrificing the temporary trapdoor key. The multi-trapdoor hash function has multiple hash/trapdoor keys, which combines multiple collisions generated by multiple entities to conduct a single collision. As a result, the multi-trapdoor hash function has the advantage of computing efficiency as well as storage space and bandwidth saving. In this paper, we build a certificateless aggregate signature scheme based on the multi-trapdoor hash function, with which a blockchain- based MCPS data storage and sharing model is proposed.

3.3. Certificateless Aggregate Signature

3.3.1. Definition of Certificateless Aggregate Signature

A certificateless aggregate signature consists of the following six algorithms [40]:

- **Setup**: Inputs the security parameter k, KGC outputs the system public parameter K_{pub} and system master key λ.
- **Partial-Private-Key-Gen**: Inputs k, K_{pub}, λ, and user's identity ID_i, KGC outputs the partial private key θ_i and sends it to the user ID_i through a secure channel.
- **User-Key-Gen**: Inputs k, the user ID_i outputs secret/public key pair (α_i, X_i).
- **Sign**: Inputs k, ID_i, (α_i, X_i), and message m_i, the user ID_i outputs a signature σ_i.
- **Agg-Sign-Gen**: Inputs k, $\{ ID_i \}_{i=1}^{n}$, $\{ \sigma_i \}_{i=1}^{n}$, the aggregator outputs the aggregate signature σ on $\{ m_i \}_{i=1}^{n}$.
- **Agg-Ver-Gen**: Inputs k, $\{ ID_i \}_{i=1}^{n}$, σ, $\{ m_i \}_{i=1}^{n}$, and public key sets $\{ X_i \}_{i=1}^{n}$, if the verification is correct, the verifier outputs 1, otherwise, the verifier outputs 0.

3.3.2. Security Models of Certificateless Aggregate Signature

According to different capabilities, two types of adversaries are considered in certificateless aggregate signature schemes [9]. In addition, certificateless aggregate signature schemes should be existentially unforgeable under these adversaries, $\mathcal{A}_{\mathbf{I}}$ and $\mathcal{A}_{\mathbf{II}}$.

$\mathcal{A}_{\mathbf{I}}$ adversary cannot get the system master key, but they can replace the public keys of legitimate users. Usually, $\mathcal{A}_{\mathbf{I}}$ adversary acts as malicious KGC.

$\mathcal{A}_{\mathbf{II}}$ adversary can obtain the system master key, however they cannot replace the public keys of legitimate users. $\mathcal{A}_{\mathbf{II}}$ adversary is often regarded as malicious inside signers.

For these types of adversaries, we define the following two games:

(1) Game I:

Setup: Challenger Z inputs security parameters k, generates system parameter *pars* and system master key λ, sends *pars* to adversary $\mathcal{A}_{\mathbf{I}}$, and keeps λ secretly.

Query: $\mathcal{A}_{\mathbf{I}}$ adaptively performs the following oracle queries:

- *Hash queries*: $\mathcal{A}_{\mathbf{I}}$ sends a hash oracle query for all hash values in the scheme, and challenger Z returns the corresponding value.
- *Partial-Key-Gen query*: When $\mathcal{A}_{\mathbf{I}}$ makes a partial private key query on the user ID_i, the challenger Z runs the partial private key generation algorithm to generate the corresponding partial private key θ_i and returns it to $\mathcal{A}_{\mathbf{I}}$**.**
- *Secret-Key-Gen query*: When $\mathcal{A}_{\mathbf{I}}$ makes a secret key query on the user ID_i, the challenger Z runs the secret key generation algorithm to generate the corresponding secret key α_i and returns it to $\mathcal{A}_{\mathbf{I}}$**.**
- *Public-Key-Gen query*: When $\mathcal{A}_{\mathbf{I}}$ makes a public key query on the user ID_i, the challenger Z runs the public key generation algorithm to generate the corresponding public key (X_i , V_i) and returns it to $\mathcal{A}_{\mathbf{I}}$**.**
- *Public-Key-Replacement query*: When $\mathcal{A}_{\mathbf{I}}$ queries user ID_i for public key replacement, Z replaces the corresponding public key of user ID_i with a randomly selected $PK^{*}_{DAU_i} = (X^{*}_i, V^{*}_i)$ and saves it.

- Signature queries: Inputs message s_i', user ID_i and corresponding private key $(\alpha_i,\ \theta_i\ )$ and status information Ω_i , Z runs the signature algorithm to generate the corresponding signature σ_i and returns it to $\mathcal{A}_{\mathbf{I}}$**.**

Forge: After the above polynomial bounded queries, Z outputs the forged aggregate signature σ^* = (ω^*, D^*). The adversary wins the game if and only if:

- Forged signature σ^* is a valid signature.
- $\mathcal{A}_{\mathbf{I}}$ cannot query at least one of n users for partial private key.

(2) Game II:

Setup: Challenger Z inputs security parameters k, generates system parameter *pars* and system master key λ, sends *pars* and λ to adversary $\mathcal{A}_{\mathbf{II}}$.

Query: In this stage, adversary $\mathcal{A}_{\mathbf{II}}$ adaptively performs the polynomial bounded oracle queries which are similar to Game I. The difference is that $\mathcal{A}_{\mathbf{II}}$ does not perform the public key replacement query and partial private key query.

Forge: Z outputs the forged aggregate signature σ^* = (ω^*, D^*). The adversary $\mathcal{A}_{\mathbf{II}}$ wins the game if and only if:

- Forged signature σ^* is a valid signature.
- $\mathcal{A}_{\mathbf{II}}$ cannot query at least one of n users for secret value.

3.4. System Model

In this paper, a two-layer system model is used to describe the secure storage and sharing of medical records in MCPS. As shown in Figure 1, the off-blockchain layer completes the acquisition, aggregation, and storage of medical data. In our proposed system model, every doctor, nurse, medical device, and medical app has a pseudonym, partial private key, secret value, and public key. The pseudonym is distributed by the Registry Center, and partial private keys are allocated by the KGC. Doctors, nurses, medical equipment, and medical apps are noted as data acquisition units (DAU). The medical record of a patient consists of several medical record items (MRI). Each MRI is signed by the DAU who is responsible for it. A patient's diagnosis and treatment process corresponds to a Central Hospital. When a patient goes to different Central Hospitals, it corresponds to different treatment processes. Each DAU encrypts the collected MRIs with the public key of the Central Hospital, and calculates the hash value of MRIs it is responsible for as digital digest. The DAU's private key is used to individually sign on the digest information. Then, the encrypted MRIs, digest information, and individual signatures are sent to the Central Hospital. The Central Hospital verifies the correctness of the individual signature. If it is correct, the encrypted original medical data is stored in the Medical Cloud. Finally, the Central Hospital combines the individual signatures into an aggregate signature, and sends the digest, aggregate signature, access control, and location index of the original MRIs to the Medical Blockchain.

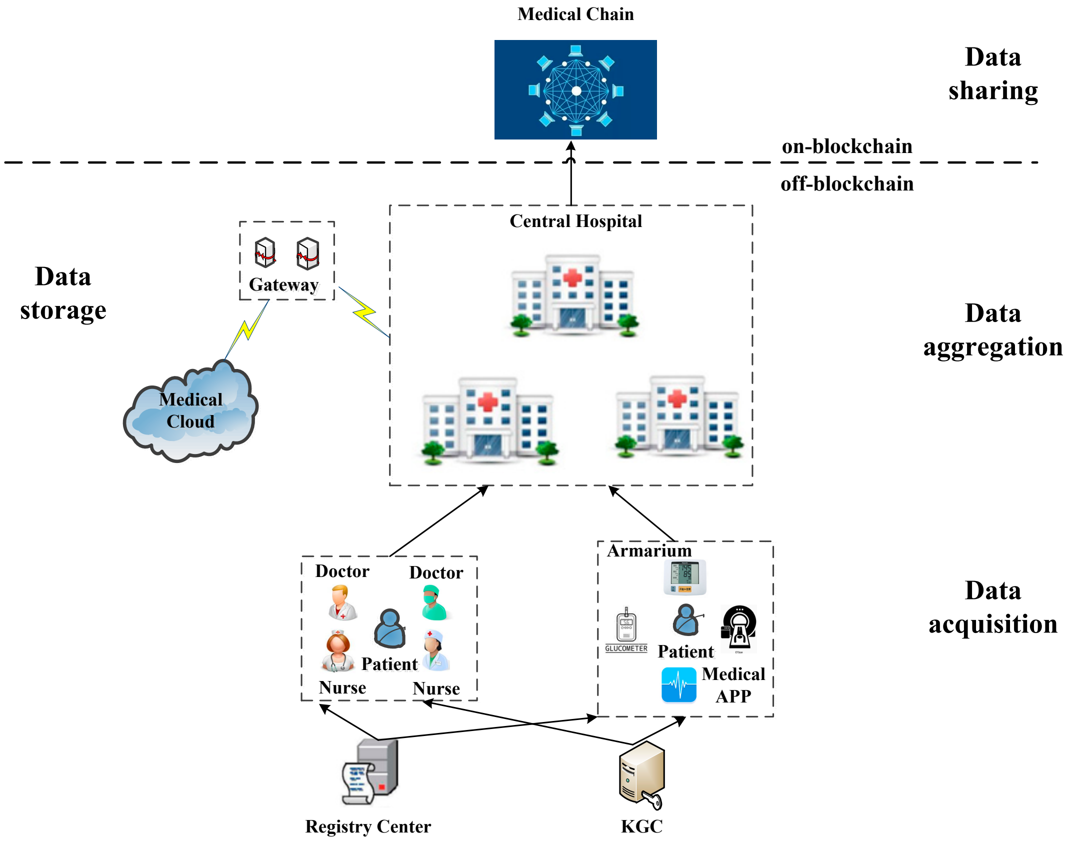

Figure 1. System model.

The on-blockchain layer completes the sharing of medical data. Figure 2 shows that each transaction of the Medical Chain contains a digest of the P_i's MRIs, an aggregate signature, access control, and a specific location index of the original medical data stored in the Medical Cloud. Each block contains a hash value linked to the previous block. This hash value can be used to retrieve the block. The Medical Chain uses time stamps to ensure that the blocks are linked in time. The latest generated blocks are broadcast to the entire network. The nodes receiving the information verify the correctness according to the consensus algorithm. If it is correct, they pass the information to other nodes. After most nodes verify the correctness, the miner adds the block into the main chain to form the permanent storage and sharing of medical records. The patient is the owner of medical data, who grants an entity (doctor, institution, researcher, etc.) access to original medical records through access control protocol. When an entity gains access, they look up on the Medical Chain, obtains the position index of medical data in cloud, then they can access the original medical records.

In the above model, one block contains multiple transactions, and one transaction relates to all medical records of one medical treatment process of a patient. By using blockchain to store the digest and aggregate signature, the unforgeability of DAU's service and the integrity of medical data can be guaranteed. Meanwhile, the block capacity limitation can be greatly eased. On the other hand, the encrypted original medical data is stored in the cloud, which is retrieved through the data location index on the blockchain. The access rights of entities are managed through the access control on the blockchain. Therefore, the secure storage and sharing of medical data in MCPS is realized.

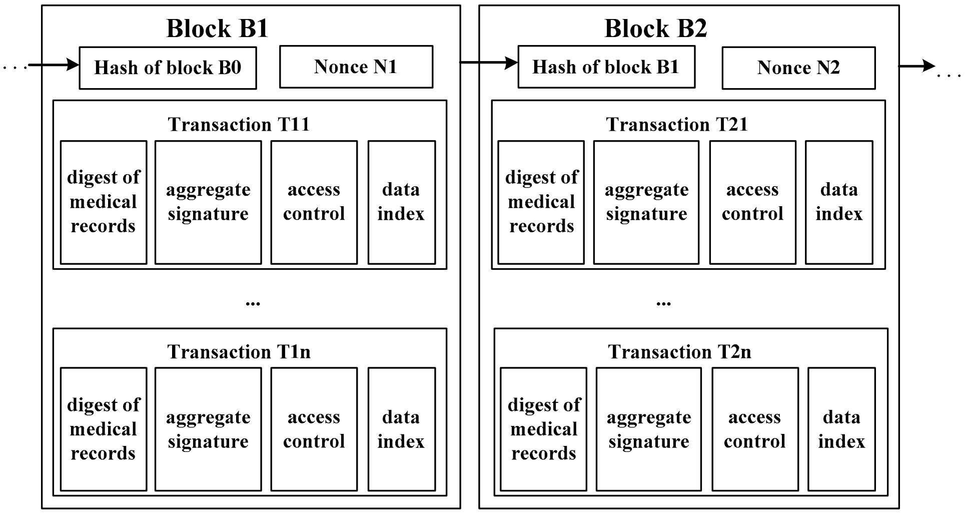

Figure 2. Medical Chain model.

3.5. Security Requirements

The following security requirements are important for medical data in MCPS:

- Non-repudiation: Medical data is the record of treatment process, which has the function of legal evidence. Any modification of a medical record should be non-repudiation;
- Integrity: As an important record of the patient's treatment, medical data should be guaranteed to be accurate, which means it cannot be tampered by anyone in any way. In other words, any data tampering can be detected;
- Privacy: Medical data involves patient's personal privacy, which should be kept confidential. It could not be allowed to be disclosed at will, only the authorized users can access it;
- Traceability: When medical disputes occur between doctors and patients, medical data should be traceable as legal evidence;
- Timeliness: Time factor is one of the key points in the whole treatment process. It is necessary to make effective time judgment on each sensitive link in the treatment process, so as to ensure the authenticity and effectiveness of medical data.

Among these security requirements, tamper-proofing, data integrity, and privacy protection are crucial issues in MCPS [4]. It is necessary to use relevant technical means, such as identity authentication, blockchain technology, digital signatures, to achieve secure storage and sharing of medical information.

3.6. System Framework

The certificateless aggregate signature scheme based on the trapdoor hash function proposed in this paper consists of the following algorithms:

- Setup: The algorithm is completed by KGC. Inputs security parameter k, outputs master key λ, system parameter *pars*.
- Pseudonym-Gen: The algorithm generates pseudonyms for each entity by Registry Center. Inputs the real identity of each DAU_i or patient P_j (denoted as RID_{DAU_i} and RID_{P_j}), outputs its pseudonym PID_{DAU_i} or PID_{P_j}.
- DAU_i Key-Gen: DAU_i generates its secret value-public key pair $(\alpha_i,\ X_i)$ and sends X_i to KGC through the secure channel. After receiving DAU_i's pseudonym RID_{DAU_i}, system parameters *pars*, public key X_i and master key λ, KGC outputs the DAU_i's partial private key θ_i. The public key (long-term hash key) of the DAU_i is X_i, the long-term trapdoor key is α_i, and the private key is θ_i.

- Hash-Gen: In this algorithm, the trapdoor hash value of DAU_i is generated. Inputs system parameter *pars*, original message s_i, DAU_i's hash key X_i, auxiliary parameter u_i, outputs DAU_i's trapdoor hash value $TH_{X_i}(\ s_i,\ u_i)$.
- Individual-Sign: In this algorithm, DAU_i generates its individual signature. Inputs system parameter *pars*, digest of MRIs in the charge of DAU_i (denoted as s_i'), status information Ω_i of DAU_i, trapdoor key α_i, hash key X_i, and outputs DAU_i's individual signature σ_i.
- Individual-Verify: The Central Hospital verifies the correctness of individual signature. Inputs DAU_i's individual signature σ_i, hash key X_i, check the correctness of σ_i. If correct, accepts σ_i and outputs 1, otherwise, rejects σ_i and outputs 0.
- Aggregate-Sign: The Central Hospital produces aggregate signature. The Central Hospital aggregates the verified individual signatures $\{\ \sigma_i\ \}_{i\,=\,1}^{n}$ into a single short signature σ.
- Aggregate-Verify: The algorithm is responsible for verifying the correctness of the aggregate signature by miner nodes. Inputs aggregate signature σ, all related DAU_i's trapdoor keys $\{\ X_i\ \}_{i\,=\,1}^{n}$, verifies the correctness of σ. If correct, accepts σ and outputs 1, otherwise, rejects σ and outputs 0.

4. The Proposed Multi-Trapdoor Hash Function

The proposed multi-trapdoor hash function based on ECC is presented in this section.

- **ParG**: Suppose the security parameter k, KGC selects large prime numbers p, q and elliptic curves over finite fields $y^2 = x^3 + ax + b \bmod p,\ a,\ b \in F_p$. Given G is a cyclic subgroup of E(F_p), P is a q-order generator of G, KGC takes secure hash function: $W = G \rightarrow Z_q^*$. KGC outputs the system parameter *pars* = (G, P, q, W).
- **KeyG**: Each DAU_i selects randomly trapdoor key $\alpha_i \in Z_q^*$ and computes hash key: $X_i = \alpha_i P$, then outputs $\{\alpha_i\ ,\ X_i\ \}_1^n$.
- **HashG**: Each DAU_i randomly selects the auxiliary parameter u_i, computes trapdoor hash value:

$$T_i = TH_{X_i}(s_i,\ u_i) = W(\ s_i,\ X_i)X_i + u_iP.$$

Finally, the Central Hospital calculates multi-trapdoor hash value:

$$T = \sum\nolimits_{i=1}^{n} T_i$$

- **TrapColG**: Each DAU_i randomly selects temporary trapdoor key $\beta_i \in Z_q^*$ and computes temporary hash key $Y_i = \beta_i P$. The collision parameter is given as

$$u_i' = \alpha_i W(s_i,\ X_i) - \beta_i W(s_i', Y_i) + u_i.$$

Trapdoor collision is one of the properties of trapdoor hash functions [37]. Given hash keys (X_i, Y_i), trapdoor keys ($\alpha_i,\ \beta_i$), message/auxiliary parameter pair $(s_i,\ u_i)$, and new message s_i', collision parameter is given by $u_i' = \alpha_i W(s_i,\ X_i) - \beta_i W(s_i', Y_i) + u_i$ which satisfies

$$TH_{X_i}(s_i,\ u_i) = TH_{Y_i}(s_i',\ u_i').$$

That is

$$W(\ s_i,\ X_i)X_i + u_iP\ =\ W(s_i', Y_i)Y_i + u_i'P$$
$$W(\ s_i,\ X_i)\ \alpha_i + u_i = W(s_i', Y_i)\ \beta_i + u_i'.$$

From the above proof process, we can conclude that the owner of the trapdoor key can compute the trapdoor collision based on the given input. The proposed multi-trapdoor hash function aggregates multiple trapdoor collisions into one trapdoor collision, which improves the calculation efficiency. On the other hand, people who do not know the trapdoor key cannot calculate the trapdoor collision. Therefore, the proposed multi-trapdoor hash function is secure and efficient to construct the certificateless aggregate signature scheme.

5. The Proposed Certificateless Aggregate Signature Scheme

The proposed certificateless aggregate signature scheme based on the multiple trapdoor hash function is presented in this section. We introduce an attribute-based signature [41] and state the information, so that the requirements for medical data in blockchain-based MCPS can be better satisfied.

5.1. Setup

In this subsection, KGC will generate the system parameter and send it to data acquisition units DAU_i, patients P_j, and Central Hospitals. Suppose the security parameter k, KGC selects large prime numbers p, q and elliptic curves over finite fields $y^2 = x^3 + ax + b \bmod p$, a, $b \in F_p$. Given G is a cyclic subgroup of E(F_p), P is a q-order generator of G, KGC takes seven secure hash functions: $W_1 = \{0,1\}^* \to Z_q^*$, $W_2 = G \to Z_q^*$, $W_3 = \{0,1\}^* \to Z_q^*$, $W_4 = G \to Z_q^*$, $W_5 = G \to Z_q^*$, $W_6 = G \to Z_q^*$, $H = G \to Z_q^*$. KGC randomly selects $\lambda \in Z_q^*$ as the system master key. Then, the public key is $K_{pub} = \lambda P$. Finally, KGC outputs the system parameter *pars* = (G, P, q, K_{pub}, W_1, W_2, W_3, W_4, W_5, W_6, H).

5.2. Pseudonym-Gen

In this phase, the Registry Center calculates the pseudonyms for DAU_i and P_j according to their real identities. The pseudonym system [42] is used to provide conditional privacy protection for doctors, nurses, patients, medical devices, etc. When relevant organizations need to know their real identity, the Registry Center can index their real identity. The Registry Center performs the following procedure to generate pseudonyms for DAU_i and P_j.

- The Registry Center accepts DAU_i's real identity RID_{DAU_i} and calculates its pseudo identity $ID'_{DAU_i} = W_1(RID_{DAU_i})$. After selecting a random $a_i \in Z_q^*$, DAU_i calculates $F_i = a_i P$, $PID_{DAU_i,1} = \lambda W_2(F_i)$, and sends $PID_{DAU_i,1}$ to the Registry Center through the secure channel. The Registry Center calculates $PID_{DAU_i,2} = W_3(ID'_{DAU_i}, PID_{DAU_i,1})$, and outputs pseudonym $PID_{DAU_i} = (PID_{DAU_i,1}, PID_{DAU_i,2})$.
- The Registry Center accepts P_j's real identity RID_{P_j} and calculates its pseudo identity $ID'_{P_j} = W_1(RID_{P_j})$. After selecting a random $b_j \in Z_q^*$, P_j calculates $E_j = b_j P$, $PID_{P_j,1} = \lambda W_2(E_j)$, and sends $PID_{P_j,1}$ to the Registry Center through the secure channel. The Registry Center calculates $PID_{P_j,2} = W_3(ID'_{P_j}, PID_{P_j,1})$, and outputs pseudonym $PID_{P_j} = (PID_{P_j,1}, PID_{P_j,2})$.

At the same time, the Registry Center builds an index table between the real identities of DAU_i (P_j) and their pseudonyms, such as (RID_{DAU_i}, PID_{DAU_i}), (RID_{P_j}, PID_{P_j}), so that when relevant organizations need to know the real identities of DAU_i or P_j, the Registry Center could return their real identities.

5.3. DAU_i Key-Gen

In this stage, DAU_i completes secret value/public parameter pair generation and sends the public parameter to KGC. With the received public parameter, KGC computes partial private key/partial public key pair. These two key pairs constitute the public keys and private keys of DAU_i. Because the keys of DAU_i are obtained by two entities (KGC and DAU), it is effective to protect the security of the keys.

DAU_i randomly selects the secret value $\alpha_i \in Z_q^*$, calculates $X_i = \alpha_i P$ as the public parameter. Then, DAU_i sends the public parameter X_i to the KGC and the Central Hospital.

It then inputs the pseudonym PID_{DAU_i} and public parameters X_i of DAU_i, KGC randomly selects $\gamma_i \in Z_q^*$ as the secret value, calculates $V_i = \gamma_i P$ and DAU_i's partial private key $\theta_i = \gamma_i + \lambda W_4(PID_{DAU_i}, X_i, V_i)$, then sends V_i and θ_i to DAU_i through the secure channel. DAU_i verifies the correctness of partial private key θ_i by checking whether the equation $\theta_i P = V_i + K_{pub} W_4(PID_{DAU_i}, X_i, V_i)$ is valid.

DAU_i's public and private keys are: $PK_{DAU_i} = (X_i, V_i)$, $SK_{DAU_i} = (\alpha_i, \theta_i)$. The partial private key and pseudonym effectively protect DAU_i's identity information. It plays a role of privacy protection.

5.4. Hash-Gen

In this section, each DAU_i generates its own trapdoor hash value and sends it to the Central Hospital. Then, the Central Hospital combines all verified trapdoor hash values into a single value. Based on the trapdoor hash value, the trapdoor collision can be calculated, which can be used to achieve the individual signature.

Firstly, it inputs system parameter *pars*, original message s_i, DAU_i's hash key (public parameter) X_i, DAU_i randomly selects auxiliary parameter u_i, and calculates trapdoor hash value $T_i = TH_{X_i}(s_i, u_i) = W_5(s_i, X_i)X_i + u_i P$. Where the original message s_i depends on the attribute value of DAU_i. That is to say, if DAU_i is a doctor or a nurse, then s_i is composed of the ID of the hospital where he or she works, his or her working department, and position titles, etc.; if DAU_i is a medical equipment or app, then s_i is composed of DAU_i's pseudonym PID_{DAU_i}, its manufacturer, categories, the affiliated institutions (hospitals, communities, scientific research institutions, etc.), etc. Using a series of attributes related to the signer to determine their identity can effectively protect the privacy of the signer, such as phone number, home address, email, etc.

When a patient P_j starts data interaction with a DAU_i, the trapdoor hash value T_i of DAU_i is calculated in advance and sent to the Central Hospital. When the treatment of P_j is completed (assuming that P_j generates n MRIs with n DAU_is), the Central Hospital aggregates the trapdoor hash value $T = \sum_{i=1}^{n} T_i$ of all the DAU_is responsible for P_j's MRIs, and sends T to each DAU_i, which interacts with P_j.

5.5. Individual-Sign

In this subsection, each DAU_i that provides medical services to the patient P_j completes an individual signature on the medical data for which it is responsible. We define the state information of DAU_i as Ω_i, that is, the pseudonym of P_j associated with this DAU_i. Only the individual signatures with the same Ω_i (that is, for the same patient) can be aggregated.

DAU_i selects the latest timestamp t_i and calculates $\theta_i' = W_6(t_i, V_i, \Omega_i)$, $y_i = \theta_i' P$. The latest timestamp ensures the timeliness of data collection and resists replay attacks. DAU_i randomly selects temporary trapdoor key $\beta_i \in Z_q^*$, and calculates the temporary hash key $Y_i = \beta_i P$ and the trapdoor hash value $TH_{Y_i}(s_i', u_i') = W_5(s_i', Y_i)Y_i + u_i' P$. s_i' represents the digest of P_j's MRI, which is in the charge of DAU_i during this treatment. According to trapdoor collision (that is $TH_{X_i}(s_i, u_i) = TH_{Y_i}(s_i', u_i')$), it calculates collision parameter $u_i' = \alpha_i W_5(s_i, X_i) - \beta_i W_5(s_i', Y_i) + u_i$, $H^* = H(PID_{DAU_i}, T, u_i')$, $d_i = \theta_i' - (\alpha_i + \theta_i) H^* \bmod q$. DAU_i's individual signature for patient P_j is $\sigma_i = (y_i, d_i)$. DAU_i sends (σ_i, u_i') to the Central Hospital.

5.6. Individual-Verify

In this stage, the Central Hospital achieves the verification of DAU_i's individual signature. When the Central Hospital receives DAU_i's individual signature $\sigma_i = (y_i, d_i)$ and new auxiliary parameter u_i', the Central Hospital performs the following steps:

1. Compute $W_4^* = W_4(PID_{DAU_i}, X_i, V_i)$ and $H^* = H(PID_{DAU_i}, T, u_i')$
2. Check whether $d_i P + (X_i + V_i + K_{pub} W_4^*)H^* = y_i$ holds or not. If it holds, the Central Hospital accepts σ_i and then stores the encrypted original medical data in the Medical Cloud.

Since $X_i = \alpha_i P$, $\theta_i P = V_i + K_{pub} W_4(PID_{DAU_i}, X_i, V_i)$, $y_i = \theta_i' P$, we obtain

$$\begin{aligned} & d_iP + (X_i + V_i + K_{pub}W_4^*)\, H^* \\ = \; & \theta_i'P - (\alpha_i + \theta_i)P \cdot H^* + (X_i + V_i + K_{pub}W_4^*)H^* \\ = \; & \theta_i'P - (X_i + V_i + K_{pub}W_4^*)\, H^* + (X_i + V_i + K_{pub}W_4^*)H^* \\ = \; & \theta_i'P \\ = \; & y_i \end{aligned}$$

5.7. Aggregate-Sign

In this phase, the Central Hospital aggregates the accepted individual signatures for medical data from the same patient. The Central Hospital checks the status information Ω_i of each DAU_i whose individual signature σ_i is accepted. For individual signatures with the same Ω_i, the Central Hospital calculates $\omega = \sum_{i=1}^{n} y_i$, $D = \sum_{i=1}^{n} d_i$, and the aggregate signature $\sigma = (\omega, D)$. Then, the Central Hospital forms a transaction by P_j's MRI digest, aggregation signature, access control, and the specific location of the original medical data in the Medical Cloud. Finally, a transaction request is sent to the Medical Chain.

5.8. Aggregate-Verify

After the miner receives the message, the aggregate signature is verified through the consensus mechanism. If the equation $DP + \sum_{i=1}^{n} (X_i + V_i + K_{pub}W_4^*)\, H^* = \omega$ holds, the information is broadcast to other nodes in the network. The other nodes start consensus verification of the transaction and broadcast on the network. After the verification is successful, the transaction is added to the block.

6. Security Discussion

6.1. Correctness Proof

The correctness proof of the aggregate is verified as follows:

$$\begin{aligned} & DP + \textstyle\sum_{i=1}^{n} (X_i + V_i + K_{pub}W_4^*)\, H^* \\ = \; & \textstyle\sum_{i=1}^{n} \left(\theta_i'P - (\alpha_i + \theta_i)P \cdot H^*\right) + \sum_{i=1}^{n} (X_i + V_i + K_{pub}W_4^*)\, H^* \\ = \; & \textstyle\sum_{i=1}^{n} \left[\theta_i'P - (X_i + V_i + K_{pub}W_4^*)\, H^*\right] + \sum_{i=1}^{n} (X_i + V_i + K_{pub}W_4^*)\, H^* \\ = \; & \textstyle\sum_{i=1}^{n} \theta_i' P \\ = \; & \textstyle\sum_{i=1}^{n} y_i \\ = \; & \omega \end{aligned}$$

6.2. Security Proof

Theorem 1. *In the random oracle model, the proposed certificateless aggregate signature scheme is existentially unforgeable against adaptive chosen-message attacks under the assumption that the ECDLP problem is hard.*

This theorem is obtained by combining Lemmas 1 and 2.

Lemma 1. *Given an $\mathcal{A}_{\mathrm{I}}$ type adversary C_1 makes at most q_S Sign queries, q_K Partial-Key-Gen queries, q_{SK} Partial-Key-Gen queries within a period t in the random oracle model, and wins the game with an non-negligible probability ε, that is, successfully forging the signature of the proposed scheme. Then, an algorithm T_1 can be performed in polynomial time, and solve an instance of ECDLP with probability (supposing the number of aggregate signatures is n) $\varepsilon' \geq \frac{\varepsilon}{ne\,(q_S + n)} \left(1 - \frac{q_K}{2^t}\right) \left(1 - \frac{q_{SK}}{2^t}\right)$.*

Proof. Suppose T_1 is a solution of ECDLP and $(P, xP) \epsilon G$ as an instance of ECDLP, the goal of the algorithm T_1 is to compute x. Suppose T_1 makes q_S Sign queries on q_S identities, and generates n

aggregate signatures at the challenge stage, T_1 selects PID_{DAU_k} as the target victim, and the probability of the selection is $\mu \in [\frac{1}{q_S+n}, \frac{1}{q_S+1}]$. We set up a game between adversary C_1 and challenger Z_1, and the detailed interaction process is as follows:

Setup: Given $K_{pub} = xP$, challenger Z_1 inputs security parameters k, generates system parameter $pars = (G, P, q, K_{pub}, W_1, W_2, W_3, W_4, W_5, W_6, H)$, and sends $pars$ to adversary C_1. Z_1 needs to maintain nine lists $(L_{W_4}, L_{W_5}, L_{W_6}, L_H, L_P, L_{PK}, L_{SK}, L_T, L_S)$, whose initial values are empty.

Query: C_1 adaptively performs the following oracle queries.

- *W_4 hash query*: When C_1 makes a W_4 hash query with parameter (PID_{DAU_i}, X_i, V_i), Z_1 checks whether existing $(PID_{DAU_i}, X_i, V_i, \delta_{W_4}) \in L_{W_4}$ or not, if so, Z_1 sends δ_{W_4} to C_1. Otherwise, Z_1 selects a random $\delta_{W_4} \in Z_q^*$. If the list L_{W_4} does not include the tuple $(*, *, *, \delta_{W_4})$, Z_1 sends δ_{W_4} to C_1 and saves $(PID_{DAU_i}, X_i, V_i, \delta_{W_4})$ into the hash list L_{W_4}.
- *W_5 hash query*: When C_1 makes a W_5 hash query with parameter (s_i, X_i), Z_1 checks whether existing $(s_i, X_i, \delta_{W_5}) \in L_{W_5}$ or not, if so, Z_1 sends δ_{W_5} to C_1. Otherwise, Z_1 selects a random $\delta_{W_5} \in Z_q^*$. If the list L_{W_5} does not include the tuple $(*, *, \delta_{W_5})$, Z_1 sends δ_{W_5} to C_1 and saves (s_i, X_i, δ_{W_5}) into the hash list L_{W_5}.
- *W_6 hash query*: When C_1 makes a W_6 hash query with parameter (t_i, V_i, Ω_i), Z_1 checks whether existing $(t_i, V_i, \Omega_i, \delta_{W_6}) \in L_{W_6}$ or not, if so, Z_1 sends δ_{W_6} to C_1. Otherwise, Z_1 selects a random $\delta_{W_6} \in Z_q^*$. If the list L_{W_6} does not include the tuple $(*, *, *, \delta_{W_6})$, Z_1 sends δ_{W_6} to C_1 and saves $(t_i, V_i, \Omega_i, \delta_{W_6})$ into the hash list L_{W_6}.
- *H hash query*: When C_1 makes an H hash query with parameter (PID_{DAU_i}, T, u_i'), Z_1 checks whether existing $(PID_{DAU_i}, T, u_i', \delta_H) \in L_H$ or not, if so, Z_1 sends δ_H to C_1. Otherwise, then Z_1 selects a random $\delta_H \in Z_q^*$. If the list L_H does not include the tuple $(*, *, *, \delta_H)$, Z_1 sends δ_H to C_1 and saves $(PID_{DAU_i}, T, u_i', \delta_H)$ into the hash list L_H.
- *Partial-Key-Gen query*: When C_1 makes a Partial-Key-Gen query with parameter (PID_{DAU_i}, X_i), Z_1 checks whether existing $(PID_{DAU_i}, \theta_i, V_i) \in L_P$ or not.
    - If L_P includes the tuple $(PID_{DAU_i}, \theta_i, V_i)$, Z_1 sends (θ_i, V_i) to C_1.
    - If L_P does not include the tuple $(PID_{DAU_i}, \theta_i, V_i)$ and $PID_{DAU_i} \neq PID_{DAU_k}$, Z_1 selects a random $\theta_i, \delta_{W_4} \in Z_q^*$, computes $V_i = \theta_i P - K_{pub}\delta_{W_4}$, sends (θ_i, V_i) to C_1 and saves $(PID_{DAU_i}, \theta_i, V_i)$ into the hash list L_P. If list L_{W_4} does not include corresponding tuple, then Z_1 adds tuple $(PID_{DAU_i}, X_i, V_i, \delta_{W_4})$ into L_{W_4}.
    - If L_P does not include the tuple $(PID_{DAU_i}, \theta_i, V_i)$ and $PID_{DAU_i} = PID_{DAU_k}$, Z_1 randomly selects $\theta_i, \delta_{W_4} \in Z_q^*$, lets $V_k = \gamma_r P$ ($\gamma_r \in Z_q^*$ is a known random number to Z_1), then saves $(PID_{DAU_k}, \theta_k, V_k)$ into the hash list L_P and sends (θ_k, V_k) to C_1 If list L_{W_4} does not include corresponding tuple, then Z_1 adds tuple $(PID_{DAU_k}, X_k, V_k, \delta_{W_4})$ into L_{W_4}.
- *Secret-Key-Gen query*: Suppose that the query is on a pseudo identity PID_{DAU_i}. If the list L_{SK} includes $(PID_{DAU_i}, \alpha_i, \theta_i)$, Z_1 sends (α_i, θ_i) to C_1 Otherwise, Z_1 selects a random $\alpha_i \in Z_q^*$ and computes $X_i = \alpha_i P$. Then Z_1 makes a Partial-Key-Gen query by (PID_{DAU_i}, X_i) and adds $(PID_{DAU_i}, \alpha_i, \theta_i)$ into list L_{SK}. Z_1 sends (α_i, θ_i) to C_1 and adds (PID_{DAU_i}, X_i, V_i) into list L_{PK}.
- *Public-Key-Gen query*: Suppose that the query is on a pseudo identity PID_{DAU_i}. If the list L_{PK} includes (PID_{DAU_i}, X_i, V_i), Z_1 sends (X_i, V_i) to C_1 Otherwise, Z_1 selects a random $\alpha_i \in Z_q^*$ and computes $X_i = \alpha_i P$. Then Z_1 makes a *Partial- Key query* by (PID_{DAU_i}, X_i) and adds (PID_{DAU_i}, X_i, V_i) into list L_{PK}. Z_1 sends (X_i, V_i) to C_1 and adds $(PID_{DAU_i}, \alpha_i, \theta_i)$ into list L_{SK}.
- *Public-Key-Replacement query*: C_1 can select a new public key $PK_{DAU_i}^* = (X_i^*, V_i^*)$ to replace the original public key PK_{DAU_i} of any legitimate DAU_i.

- *Hash-Gen query*: When C_1 makes a Hash-Gen query with parameter (s_i , u_i), Z_1 checks whether existing $(s_i , u_i , T_i) \in L_T$ or not, if so, Z_1 returns T_i to C_1. Otherwise, selects a random $\alpha_i \in Z_q^*$ and computes:
$$T_i = W_5(s_i , \alpha_i P)\, \alpha_i P + u_i P.$$
Sends T_i to C_1 and saves (s_i , u_i , T_i) into the hash list L_T.
- *Sign query*: When C_1 makes a sign query with parameter $(\alpha_i , \Omega_i , s_i , s'_i)$, Z_1 checks whether $PID_{DAU_i} = PID_{DAU_k}$ or not, if so, Z_1 randomly selects $t_i \in Z_q^*$ and $\beta_i \in Z_q^*$, and computes:
$$\begin{aligned} &\theta'_i = W_6(t_i , V_i , \Omega_i) \\ &y_i = \theta'_i P \\ &Y_i = \beta_i P \\ &H^* = H(PID_{DAU_i} , T , u'_i) \\ &u'_i = \alpha_i W_5(s_i , X_i) - \beta_i W_5(s'_i , Y_i) + u_i \\ &d_i = \theta'_i - (\alpha_i + \theta_i)\, H^* \bmod q \end{aligned}$$
Then, Z_1 generates individual signature (y_i, d_i) and sends it to C_1. Otherwise, Z_1 outputs failure and halts.
- *Aggregate-Sign query*: When all of the PID_{DAU_i} $(1 \le i \le n)$ satisfies $PID_{DAU_i} \ne PID_{DAU_k}$, Z_1 randomly selects $t_i \in Z_q^*$ and $\beta_i \in Z_q^*$ for every DAU_i $(1 \le i \le n)$. Then Z_1 calculates
$$\begin{aligned} &\theta'_i = W_6(t_i , V_i , \Omega_i) \\ &y_i = \theta'_i P \\ &Y_i = \beta_i P \\ &H^* = H(PID_{DAU_i} , T , u'_i) \\ &u'_i = \alpha_i W_5(s_i , X_i) - \beta_i W_5(s'_i , Y_i) + u_i \\ &d_i = \theta'_i - (\alpha_i + \theta_i)\, H^* \bmod q \\ &\omega = \textstyle\sum_{i=1}^{n} y_i \\ &D = \textstyle\sum_{i=1}^{n} d_i \end{aligned}$$
Then, Z_1 generates aggregate signature (ω, D) and sends it to C_1. Otherwise, if $PID_{DAU_i} = PID_{DAU_k}$, Z_1 outputs failure and halts.
- *Individual-Verify query*: When C_1 makes an Individual-Verify query, Z_1 checks whether the corresponding tuple of PID_{DAU_i} is included in list L_{PK}.
  - If the corresponding tuple of PID_{DAU_i} is included in list L_{PK} and $PID_{DAU_i} \ne PID_{DAU_k}$, Z_1 calculates $W_4^* = W_4(PID_{DAU_i} , X_i , V_i)$, $H^* = H(PID_{DAU_i} , T , u'_i)$ and verifies whether the equation $d_i P = y_i + (X_i + V_i + K_{pub} W_4^*) H^*$ holds or not, if so, Z_1 returns 1 to C_1, otherwise, returns 0 to C_1.
  - If the corresponding tuple of PID_{DAU_i} is included in list L_{PK} and $PID_{DAU_i} = PID_{DAU_k}$, Z_1 returns 1 to C_1 when the list L_H includes the tuple $(PID_{DAU_i} , T, u'_i, \delta_H)$, otherwise, Z_1 returns 0 to C_1
  - If the corresponding tuple of PID_{DAU_i} is not included in list L_{PK}, Z_1 returns 1 to C_1 when the list L_H includes the tuple $(PID_{DAU_i} , T, u'_i, \delta_H)$, otherwise, Z_1 returns 0 to C_1

Forge: After the above polynomial bounded queries, Z_1 outputs the aggregate signature $\sigma^* = (\omega^*, D^*)$ of PID_{DAU_i} $(1 \le i \le n)$, in which at least one PID_{DAU_i} $(i \in [1, n])$ does not make Partial-Key-Gen query and Secret-Key-Gen query, and at least one message s'_i $(i \in [1, n])$ does not make Sign query.

If all the PID_{DAU_i} $(1 \le i \le n)$ satisfies $PID_{DAU_i} \ne PID_{DAU_k}$, then Z_1 outputs failure and halts. Otherwise, if one PID_{DAU_i} $(1 \le i \le n)$ satisfies $PID_{DAU_i} = PID_{DAU_k}$, then Z_1 queries the corresponding tuples of PID_{DAU_i} $(1 \le i \le n)$ in the lists L_{PK}, L_{SK}, L_H and checks whether the equation $DP + \sum_{i=1}^{n} (X_i + V_i + K_{pub} W_4^*)\ H^* = \omega$ holds or not:

- If the equation holds, Z_1 outputs $x = (W_4^* H^*)^{-1}\{ \sum_{i=1, i \ne k}^{n} [\theta_i' - (\alpha_i + \theta_i) H^*] + \theta_k' - (\alpha_k + \gamma_r) H^* - D \}$ as the efficient solution to the ECDLP.
- Otherwise, Z_1 cannot solve the discrete logarithmic problem, because:

$$\begin{aligned} D &= \textstyle\sum_{i=1}^{n} d_i \\ &= \textstyle\sum_{i=1}^{n} [\theta_i' - (\alpha_i + \theta_i) \cdot H^*] \\ &= \textstyle\sum_{i=1, i \ne k}^{n} [\theta_i' - (\alpha_i + \theta_i) \cdot H^*] + \theta_k' - (\alpha_k + \theta_k) \cdot H^* \\ &= \textstyle\sum_{i=1, i \ne k}^{n} [\theta_i' - (\alpha_i + \theta_i) \cdot H^*] + \theta_k' - (\alpha_k + \gamma_r + \mathrm{x}\mathrm{W}_4^*) \cdot H^* \end{aligned}$$

If C_1 queries all PID_{DAU_i} $(1 \le i \le n)$ with Partial-Key-Gen and Secret-Key-Gen, Z_1 will terminate the simulation. Suppose that

- Event E_1 represents that at least a PID_{DAU_k} $(1 \le k \le n)$ does not make Partial-Key-Gen query and Secret-Key-Gen query.
- Event E_2 represents that Z_1 does not terminate at the Sign-query stage.
- Event E_3 represents that Z_1 does not terminate at the challenge stage.

The probability of solving the ECDLP by algorithm T_1 is as follows:

$$\begin{aligned} &\Pr[E_1] \ge \tfrac{1}{n}(1 - \tfrac{q_K}{2^t})(1 - \tfrac{q_{SK}}{2^t}) \\ &\Pr[E_2 \mid E_1] = (1 - \varphi)^{q_S} \\ &\Pr[E_2 \wedge E_1] = \Pr[E_2 \mid E_1]\Pr[E_1] \ge \tfrac{1}{n}(1 - \tfrac{q_K}{2^t})(1 - \tfrac{q_{SK}}{2^t})(1 - \varphi)^{q_S} \\ &\Pr[E_3] = \mu \end{aligned}$$

The probability that Z_1 does not terminate during the whole simulation is at least

$$\frac{1}{n}(1 - \frac{q_K}{2^t})(1 - \frac{q_{SK}}{2^t})(1 - \varphi)^{q_S}\ \mu$$

Since $\mu \in [\frac{1}{q_S + n}, \frac{1}{q_S + 1}]$, when q_S is large enough, $(1 - \varphi)^{q_S}$ tends to e^{-1}, so the probability that Z_1 does not terminate during the simulation is at least

$$\frac{1}{ne\,(q_S + n)}(1 - \frac{q_K}{2^t})(1 - \frac{q_{SK}}{2^t})$$

In summary, if Z_1 is not terminated during the simulation, and C_1 breaks the unforgeability of the proposed scheme with a non-negligible probability ε, T_1 can successfully solve ECDLP with a non-negligible probability:

$$\varepsilon' \ge \frac{\varepsilon}{ne\,(q_S + n)}(1 - \frac{q_K}{2^t})\,(1 - \frac{q_{SK}}{2^t})$$

□

Lemma 2. *Given an $\mathcal{A}_{\mathrm{II}}$ type adversary C_2 makes at most q_S Sign queries, q_K Partial-Key-Gen queries, q_{SK} Partial-Key-Gen queries within a period t in the random oracle model, and wins the game with an non-negligible probability ε, that is, successfully forging the signature of the proposed scheme. Then, an algorithm T_2 can be performed in polynomial time, and solve an instance of ECDLP with probability (supposing the number of aggregate signatures is n) $\varepsilon' \ge \frac{\varepsilon}{ne\,(q_S + n)}(1 - \frac{q_K}{2^t})\,(1 - \frac{q_{SK}}{2^t})$.*

Proof. Suppose T_2 is a solution of ECDLP and $(P, xP)\ \epsilon\ G$ as an instance of ECDLP. The goal of the algorithm T_2 is to compute x. T_2 selects PID_{DAU_k} as the target victim, and the probability of the selection is $\mu \in [\frac{1}{q_S + n}, \frac{1}{q_S + 1}]$. We set up a game between adversary C_2 and challenger Z_2, and the detailed interaction process is as follows:

Setup: Challenger Z_2 inputs security parameters k, generates system parameter *pars*, and sends *pars* = $(G, P, q, K_{pub}, W_1, W_2, W_3, W_4, W_5, W_6, H)$ to adversary C_2. Z_2 needs to maintain nine lists $(L_{W_4}, L_{W_5}, L_{W_6}, L_H, L_P, L_{PK}, L_{SK}, L_T, L_S)$, whose initial values are empty.

Query: Adversary C_2 makes the same queries as that of W_4 hash, W_5 hash, W_6 hash, H hash, Secret-Key-Gen, Public-Key-Gen, Hash-Gen, Sign query, Aggregate-Sign query in Lemma 1.

- *Partial-Key-Gen query*: When C_2 makes a Partial-Key-Gen query with parameter (PID_{DAU_i} , X_i), Z_2 checks whether existing $(PID_{DAU_i} , \theta_i , V_i) \in L_P$ or not.
  - If the tuple $(PID_{DAU_i} , \theta_i , V_i)$ is included in the list L_P, Z_2 sends (θ_i , V_i) to C_2.
  - If the tuple $(PID_{DAU_i} , \theta_i , V_i)$ is not included in the list L_P and $PID_{DAU_i} \neq PID_{DAU_k}$, Z_2 selects a random θ_i , $\delta_{W_4} \in Z_q^*$, computes $V_i = \theta_i P - K_{pub} \delta_{W_4}$, sends (θ_i , V_i) to C_2 and saves $(PID_{DAU_i} , \theta_i , V_i)$ into the list L_P. Then Z_1 adds tuple $(PID_{DAU_i} , X_i , V_i , \delta_{W_4})$ into L_{W_4}.
  - If the tuple $(PID_{DAU_i} , \theta_i , V_i)$ is not included in the list L_P and $PID_{DAU_i} = PID_{DAU_k}$, Z_2 randomly selects θ_i, $\delta_{W_4} \in Z_q^*$, lets $V_k = xP$, then saves $(PID_{DAU_k} , \theta_k , V_k)$ into the hash list L_P and sends (θ_k , V_k) to C_2 Then Z_2 adds tuple $(PID_{DAU_k} , X_k , V_k , \delta_{W_4})$ into L_{W_4}.
- *Individual-Verify query*: When C_2 makes an Individual-Verify query with parameter (PID_{DAU_i} , s_i'), Z_2 checks whether the corresponding tuple of PID_{DAU_i} is included in list L_{PK}.
  - If the corresponding tuple of PID_{DAU_i} is included in list L_{PK} and $PID_{DAU_i} \neq PID_{DAU_k}$, Z_2 calculates $W_4^* = W_4(PID_{DAU_i} , X_i , V_i)$, $H^* = H(PID_{DAU_i} , T , u_i')$ and verifies whether the equation $d_iP + (X_i + V_i + K_{pub}W_4^*)H^* = y_i$ holds or not, if so, Z_2 returns 1 to C_2, otherwise, returns 0 to C_2.
  - If the corresponding tuple of PID_{DAU_i} is included in list L_{PK} and $PID_{DAU_i} = PID_{DAU_k}$, Z_2 returns 1 to C_2 when the list L_H includes the tuple $(PID_{DAU_i} , T, u_i', \delta_H)$, otherwise, Z_2 returns 0 to C_2

Forge: After the above polynomial bounded queries, Z_2 outputs the aggregate signature $\sigma^* = (\omega^*, D^*)$ of PID_{DAU_i} $(1 \leq i \leq n)$, in which at least one PID_{DAU_i} $(i \in [1, n])$ does not perform the Partial-Key-Gen query and Secret-Key-Gen query, and at least one message, s_i' $(i \in [1, n])$ does not make Sign query.

If all the PID_{DAU_i} $(1 \leq i \leq n)$ satisfy $PID_{DAU_i} \neq PID_{DAU_k}$, then Z_2 outputs failure and halts. Otherwise, if one PID_{DAU_K} $(1 \leq K \leq n)$ satisfies $PID_{DAU_K} = PID_{DAU_k}$, then Z_2 queries the corresponding tuples of PID_{DAU_i} $(1 \leq i \leq n)$ in the lists L_{PK}, L_{SK}, L_H, L_{W_4} and checks whether the equation $DP + \sum_{i=1}^{n} (X_i + V_i + K_{pub}W_4^*) H^* = \omega$ holds or not:

- If the equation holds, Z_2 outputs $x = (H^*)^{-1}\{ \sum_{i=1,i\neq k}^{n} [\theta_i' - (\alpha_i + \theta_i)H^*] + \theta_k' - (\alpha_k + \lambda W_4^*)H^* - D \}$ as the solution to the ECDLP.
- Otherwise, Z_2 cannot solve the discrete logarithmic problem, because:

$$\begin{aligned} D &= \textstyle\sum_{i=1}^{n} d_i \\ &= \textstyle\sum_{i=1}^{n} [\theta_i' - (\alpha_i + \theta_i) \cdot H^*] \\ &= \textstyle\sum_{i=1,i\neq k}^{n} [\theta_i' - (\alpha_i + \theta_i) \cdot H^*] + \theta_k' - (\alpha_k + \theta_k) \cdot H^* \\ &= \textstyle\sum_{i=1,\, i\neq k}^{n} [\theta_i' - (\alpha_i + \theta_i) \cdot H^*] + \theta_k' - (\alpha_k + x + \lambda W_4^*) \cdot H^* \end{aligned}$$

It can be seen from the proof of Lemma 1 that the probability that Z_2 does not terminate during the simulation is at least

$$\frac{1}{ne\ (q_S+n)}\ (1-\frac{q_K}{2^t})(1-\frac{q_{SK}}{2^t})$$

Therefore, if Z_2 is not terminated during the simulation, and C_2 breaks the unforgeability of the proposed scheme with a non-negligible probability, T_2 can successfully solve ECDLP with a non-negligible probability:

$$\varepsilon' \geq \frac{\varepsilon}{ne\ (q_S+n)}\ (1-\frac{q_K}{2^t})\ (1-\frac{q_{SK}}{2^t})$$

□

6.3. Security Analysis

- **Message authentication**: As Theorem 1 states, no polynomial adversary could forge a valid message under the assumption that the ECDLP problem is hard. Therefore, the Central Hospital verifies the validity and integrity of the message $(PID_{DAU_i}\ ,\ X_i,\ V_i,\ t_i\ ,\ u_i',\sigma_i)$ by checking whether the equation $d_iP\ =\ y_i+\ (X_i+V_i+K_{pub}W_4^*)H^*$ holds or not, where $W_4^*=\ W_4(PID_{DAU_i}\ ,\ X_i\ ,\ V_i)$ and $H^*=\ H(PID_{DAU_i}\ ,\ T\ ,\ u_i')$. Thus, the proposed scheme for MCPS provides message authentication.
- **Identity privacy protection**: The pseudonym proposed in this paper is divided into two types: the pseudonym of DAUs $(PID_{DAU_i}\ ,\ 1\ \leq\ i\ \leq\ n\ )$ and the pseudonym of patients $(PID_{P_j},\ 1\ \leq\ j\ \leq\ n)$. PID_{DAU_i} and PID_{P_j} are generated by combining the randomly chosen secret value a_i or b_j and the system master key λ. No adversary could compute the real identity from the pseudonym without knowing the secret a_i or b_i and λ. Thus, the pseudonym proposed in this paper can protect the identity privacy of DAUs and patients.
- **Resistance to replay attack**: Whenever DAU_i makes an individual signature, it chooses a latest timestamp t_i. The Central Hospital will check the freshness of the timestamp t_i in order to detect the replay attacks.
- **Resistance to modification attack**: According to Theorem 1, the Central Hospital can protect the integrity of message $(PID_{DAU_i}\ ,\ X_i,\ V_i,\ t_i\ ,\ u_i',\sigma_i)$. Therefore, any modification on the message will be detected by checking whether the equation $d_iP\ =\ y_i+\ (X_i+V_i+K_{pub}W_4^*)H^*$ holds or not.
- **Resistance to spam attack [17]**: Because of natural compression property of the aggregate signature, the proposed signature scheme can combine n individual signature into one short signature. The length of the aggregate signature will not increase with the increase of the number of signers. Therefore, in the blockchain-based MCPS, more transactions can be added into a block. However, the attacker has to send more transactions to congest the network. It will spend more transaction fee which will increase the cost of spam attacks.

7. Efficiency Analysis

Certificateless aggregate signatures can be classified into pairing-based certificateless aggregate signatures and ECC-based certificateless aggregate signatures. In this paper, we adopt the same efficiency evaluation method as reference [11,29], in which the simulations are conducted on an Intel I7 3.4 GHz, 4 GB machine with Windows 7. Pairing-based aggregate signature schemes can be simulated on the bilinear pairing $e\ :\ G_1\times G_1\ \rightarrow\ G_2$. G_1 is an additive group generated with the order q_1 on the type A elliptic curve $E_1:\ y^2=\ x^3+x \bmod p_1$, where p_1 and q_1 are 512-bit and 160-bit prime number, respectively [11]. For ECC-based aggregate signature schemes, the simulation can be conducted over the non-singular elliptic curve $E\ :\ y^2=\ x^3+ax\ +\ b \bmod p_2$. G is an additive group generated on E with the order q_2, where p_2, q_2 are two 160-bit prime numbers, respectively. The above mentioned bilinear pairing and elliptic curve constructed in the experiments are on the same security level of 80 bits. As shown in Tables 3 and 4, the running time of these encryption operations has been presented.

Table 3. Different encryption operation running time [11,29,37].

Encryption Operation	Description	Time (ms)
t_p	The bilinear pair operation	4.2110
t_{mp}	The scalar multiplication in the bilinear pair	1.7090
t_{ap}	The bilinear pair-to-midpoint addition	0.0071
t_{hp}	The hash-to-point operation in bilinear pair	4.4060
t_{mecc}	The scalar multiplication in elliptic curve	0.4420
t_{aecc}	The point addition operation in elliptic curve	0.0018
t_h	The general hash operation	0.0001

Table 4. Group parameter [11,29,37].

Symbol	Description	Length (bytes)
$\|G_1\|$	The size of elements in group G_1	128
$\|G\|$	The size of elements in group G	40
$\|q\|$	The size of the elements in Z_q^*	20

The computation cost and communication cost are two important factors to evaluate certificateless aggregate signature schemes. In this section, the efficiency analysis is divided into two parts. First, we compare the proposed scheme with related certificateless aggregate signature schemes. Second, we compare the proposed scheme with related aggregate signature schemes based on blockchains.

1. The efficiency analysis of certificateless aggregate signature schemes

Table 5 compares the computation cost of the proposed scheme and related certificateless aggregate signature schemes [9,29].

- In the individual sign algorithm, DAU_i needs three scalar multiplications in the elliptic curve and two general hash operations to generate individual signature. The computation cost of our scheme in individual signature is smaller than related certificateless aggregate signature schemes [9,29].
- In the individual-verify algorithm, the Central Hospital needs three scalar multiplications, three point addition operations in the elliptic curve, and two general hash operations to verify the DAU_i's individual signature. The computation cost of our scheme in individual verification is smaller than that of Gong et al.'s scheme [9], but slightly higher than that of Cui et al.'s scheme [29].
- As shown in Figure 3, in the aggregate verify algorithm, the Central Hospital needs ($2n$+1) scalar multiplications, ($2n$ + 1) point addition operations in the elliptic curve, and $2n$ general hash operations to verify the aggregate signature. The computation cost of our scheme in aggregate verification is smaller than Gong et al.'s scheme [9], but slightly higher than that in Cui et al.'s scheme [29].

Table 5. The comparison of computation cost.

Scheme	Individual Sign	Individual Verify	Aggregate Verify
Gong et al. -1 [9]	$2t_{mp} + t_{ap} + t_{hp}$ $\approx$ 7.8311ms	$3t_p + 2t_{hp}$ $\approx$ 21.445ms	$(2n + 1)\ t_p + 2nt_{hp}$ $\approx$ 17.234n + 4.211ms
Gong et al. -2 [9]	$3t_{mp} + 2t_{ap} + 2t_{hp}$ $\approx$ 13.9532ms	$3t_p + t_{mp} + t_{ap} + 3t_{hp}$ $\approx$ 27.5671ms	$(n + 2)\ t_p + nt_{mp} + nt_{ap} + 2nt_{hp}$ $\approx$ 14.7391n + 8.422ms
Cui et al. [29]	$t_{mecc} + t_{aecc} + t_h$ $\approx$ 0.4439ms	$3t_{mecc} + 2t_{aecc} + 2t_h$ $\approx$ 1.3298ms	$(n + 2)t_{mecc} + 2nt_{aecc} + 2nt_h$ $\approx$ 0.4458n + 0.884 ms
Our scheme	$t_{mecc} + 3t_h$ $\approx$ 0.4423ms	$3t_{mecc} + 3t_{aecc} + 2t_h$ $\approx$ 1.3316ms	$(2n + 1)t_{mecc} + (2n + 1)t_{aecc} + 2nT_H$ $\approx$ 0.8878n + 0.4438ms

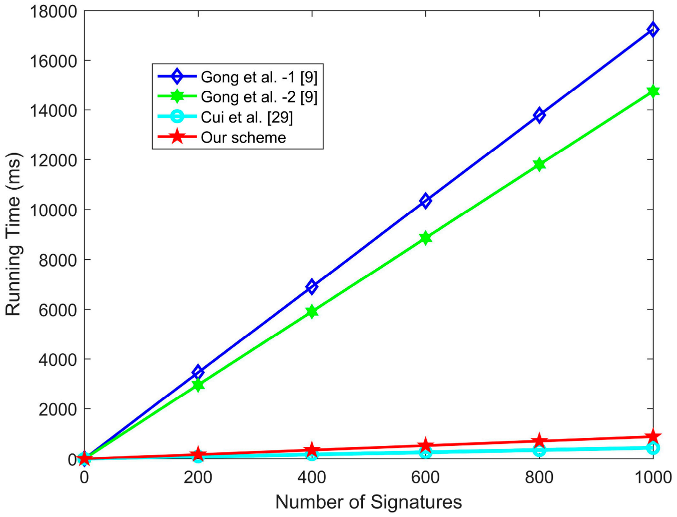

Figure 3. The comparison of aggregate verification time.

Table 6 shows the communication cost of our scheme and related certificateless aggregate signature schemes. In the proposed scheme, the aggregate signature length, such as that of CAS-2 in [9], is a constant, which does not increase with the number of individual signatures.

Table 6. The comparison of communication cost.

Scheme	Aggregate Signature Length	Correlation between Signature Length and *n*
Gong et al. −1 [9]	$(n+1)\lvert G_1\rvert$	Yes
Gong et al. −2 [9]	$2\lvert G_1\rvert$	No
Cui et al. [29]	$(n+1)\lvert G\rvert$	Yes
Our scheme	$\lvert G\rvert + \lvert q\rvert$	No

From Figure 4, we can see that the communication cost of the proposed scheme is obviously smaller than that of CAS-1 [9] and Cui et al.'s scheme [29], and slightly smaller than that of CAS-2 [9].

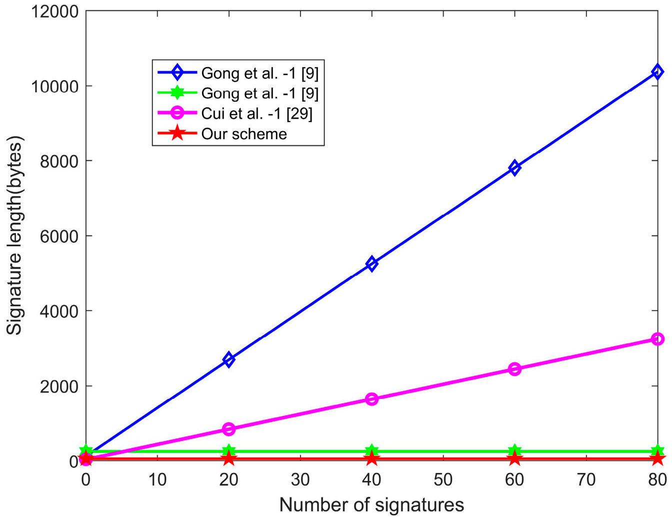

Figure 4. The comparison of signature length.

2. The comparison of certificateless aggregate signatures based on blockchain

In this subsection, we compare the computation cost and communication cost of the proposed scheme with two most recently proposed certificateless aggregate signature schemes based on blockchain [17,18]. As shown in Table 7 and Figure 5, in the individual sign algorithm and aggregate verify algorithm, the computation cost of the proposed scheme is lower than that of Gao et al.'s scheme [18], but it is close to Zhao et al.'s scheme [17]. In the individual verify algorithm, the computation cost of the proposed scheme is lower than Gao et al.'s scheme [18] but slightly higher than that of Zhao et al.'s scheme [17].

Table 7. Computation cost of schemes based on blockchain.

Scheme	Individual Sign	Individual Verify	Aggregate Verify
Zhao et al. [17]	$t_{mecc} + 2t_h$ ≈ 0.4422ms	$2t_{mecc} + 2t_h$ ≈ 0.8842ms	$(2n + 1)\, t_{mecc} + 2nt_h$ $\approx 0.8842n + 0.442$ms
Gao et al. [18]	$5t_{mp} + 3t_{ap} + 2t_{hp}$ ≈ 17.3783ms	$5t_p + 3t_{hp}$ ≈ 34.273ms	$(n + 4)t_p + (2n + 1)t_{hp}$ $\approx 13.023n + 21.25$ ms
Our scheme	$t_{mecc} + 3t_h$ ≈ 0.4423ms	$3t_{mecc} + 3t_{aecc} + 2t_h$ ≈ 1.3316ms	$(2n + 1)t_{mecc} + (2n + 1)t_{aecc} + 2nT_h$ $\approx 0.8878n + 0.4438$ms

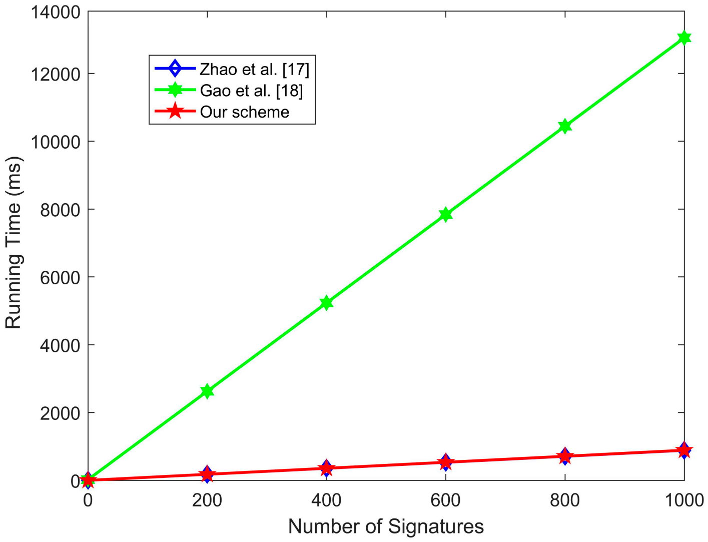

Figure 5. The aggregate verification cost of schemes based on blockchain.

As shown in Table 8 and Figure 6, the aggregate signature length of the two most recently proposed certificateless aggregate signature schemes [17,18] based on blockchain is correlated to the individual signature number. However, the aggregate signature length of our scheme is $|G|+|q|$, which is a constant and is obviously lower than the other two schemes [17,18]. That is to say, the storage capacity of the aggregate signature does not increase with the increase of the DAU_i's in each transaction, which can effectively improve the storage efficiency of each block.

Table 8. Communication cost of schemes based on blockchain.

Scheme	Aggregate Signature Length	Correlation between Signature Length and n
Zhao et al. [17]	$2n\,|G|+|q|$	Yes
Gao et al. [18]	$(2n + 1)|G_1|$	Yes
Our scheme	$|G|+|q|$	No

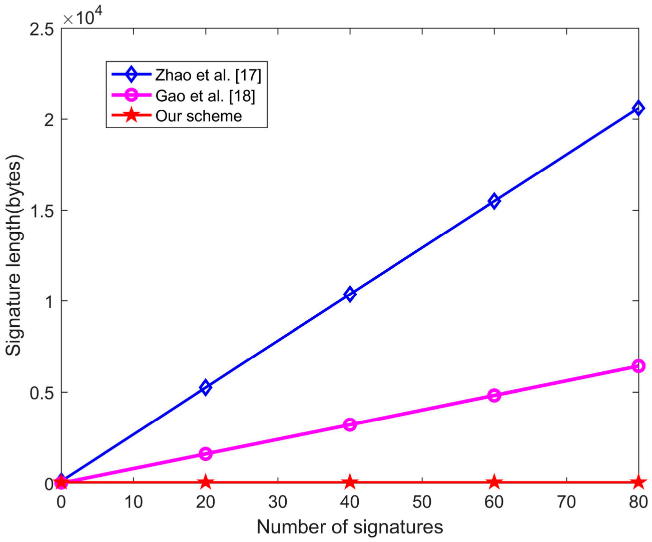

Figure 6. The signature length of schemes based on blockchain.

8. Conclusions

In this paper, a certificateless aggregate signature scheme based on blockchain is proposed, which can be used for secure storage and sharing of medical data in MCPS. To improve performance, the function of trapdoor collision calculation in trapdoor hash function is included in our proposed scheme. The security analysis presents that the proposed scheme is existentially unforgeable against adaptive chosen-message attacks, which is resistant to replay attack and modification attack. The proposed scheme provides message authentication and identity privacy protection, which satisfies the security requirements of MCPS. Compared with pairing-based schemes, the scheme proposed in this paper is based on ECC with better computational efficiency, and the computational cost of our scheme is lower. More importantly, the aggregate signature length of the proposed scheme is independent of the number of signers, which can effectively increase the number of transactions stored in each block. Therefore, the proposed scheme can alleviate the capacity limitation of blockchain and prevent spam attacks to a certain extent.

In the future work, we will focus on the lattice-based digital signature algorithm and combine it with blockchain to improve the security of blockchain. More importantly, we will apply our research to practice and obtain measurement results from practical implementation.

Author Contributions: Conceptualization: P.Q. and L.S.; methodology: H.S.; validation: Y.H.; formal analysis: P.Q. and Y.H.; investigation: H.S. and L.S.; security analysis: D.X.; writing—original draft preparation: H.S.; writing—review and editing: D.X. and F.C.; project administration: F.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (nos. 61972438, and 61801004); the Natural Science Foundation of Anhui Province (no. 1808085QF211); the Natural Science Foundation of Universities of Anhui Province (nos. KJ2019A0702 and KJ2019A0704); Anhui Key Laboratory of Network and Information Security (no. AHNIS2020003).

Acknowledgments: The authors thank for the help of reviewers and editors.

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

References

1. Yang, Y.; Zheng, X.H.; Guo, W.Z.; Liu, X.M.; Chang, V. Privacy-preserving Smart IoT-based Healthcare Big Data Storage and Self-adaptive Access Control System. *Inf. Sci.* **2019**, *479*, 567–592. [CrossRef]
2. Lee, I.; Sokolsky, O. Medical Cyber Physical Systems. In Proceedings of the CPS Demystified Session, DAC 2010, Anaheim, CA, USA, 17 June 2010; pp. 743–748.

3. Zhang, X.J.; Zhao, J.; Mu, L.M.; Tang, Y.; Xu, C.X. Identity-based Proxy-oriented Outsourcing with Public Auditing in Cloud-based Medical Cyber–physical Systems. *Pervasive Mob. Comput.* **2019**, *56*, 18–28. [CrossRef]
4. Yi, C.; Ding, S.; Xu, Z.; Zheng, H.D.; Yang, S.L. Blockchain-based Medical Records Secure Storage and Medical Service Framework. *J. Med. Syst.* **2019**, *43*, 5–13.
5. Liu, A.D.; Du, X.H.; Wang, N.; Li, S.Z. Research Progress of Blockchain Technology and Its Application in Information Security. *J. Softw.* **2018**, *29*, 270–293. (In Chinese)
6. Tsai, W.; Yu, L.; Wang, R.; Liu, N.; Deng, E. Blockchain Application Development Techniques. *J. Softw.* **2017**, *28*, 1474–1487. (In Chinese)
7. Sengupta, J.; Ruj, S.; Das Bit, S. A Comprehensive Survey on Attacks, Security Issues and Blockchain Solutions for IoT and IIoT. *J. Netw. Comput. Appl.* **2020**, *149*, 102481–102500. [CrossRef]
8. Xiong, H.; Guan, Z.; Chen, Z.; Li, F. An Efficient Certificateless Aggregate Signature with Constant Pairing Computations. *Inf. Sci.* **2013**, *219*, 225–235. [CrossRef]
9. Gong, Z.; Long, Y.; Hong, X.; Chen, K. Two Certificateless Aggregate Signatures from Bilinear Maps. In Proceedings of the 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2007), Qingdao, China, 30 July–1 August 2007; pp. 2093–2106.
10. Zhang, F.; Shen, L.; Ge, W. Notes on the Security of Certificateless Aggregate Signature Schemes. *Inf. Sci.* **2014**, *287*, 32–37. [CrossRef]
11. He, D.B.; Zeadally, S.; Xu, B.W.; Huang, X.Y. An Efficient Identity-Based Conditional Privacy-Preserving Authentication Scheme for Vehicular Ad Hoc Networks. *IEEE Trans. Inf. Forensic Secur.* **2015**, *10*, 2681–2691. [CrossRef]
12. Suciu, G.; Nădrag, C.; Istrate, C.; Vulpe, A.; Ditu, M.C.; Subea, O. Comparative Analysis of Distributed Ledger Technologies. In Proceedings of the 6th Global Wireless Summit (GWS 2018), Chiang Rai, Thailand, 25–28 November 2018; pp. 370–373.
13. Fan, K.; Wang, S.; Ren, Y.H.; Li, H.; Yang, Y.T. Medblock: Efficient and Secure Medical Data Sharing via Blockchain. *J. Med Syst.* **2018**, *42*, 136–147. [CrossRef]
14. Xue, T.F.; Fu, Q.C.; Wang, C.; Wang, X.Y. A Medical Data Sharing Model via Blockchain. *Acta Autom. Sin.* **2017**, *43*, 1555–1562. (In Chinese)
15. Xia, Q.; Sifah, E.B.; Asamoah, K.O.; Gao, J.B.; Du, X.J.; Guizani, M. MeDShare: Trust-less Medical Data Sharing among Cloud Service Providers via Blockchain. *IEEE Access* **2017**, *5*, 14757–14767. [CrossRef]
16. Aitzhan, N.Z.; Svetinovic, D. Security and Privacy in Decentralized Energy Trading through Multi-signatures, Blockchain and Anonymous Messaging Streams. *IEEE Trans. Dependable Secur. Comput.* **2016**, *15*, 840–852. [CrossRef]
17. Zhao, Y. Aggregation of Gamma-Signatures and Applications to Bitcoin. *IACR Cryptol. ePrint Arch.* **2018**, *2018*, 414. Available online: https://eprint.iacr.org/eprint-bin/getfile.pl?entry=2018/414&version=20180510:203542&file=414.pdf (accessed on 7 January 2020).
18. Gao, Y.; WU, J.X. Efficient Multi-party Fair Contract Signing Protocol based on Blockchains. *J. Cryptologic Res.* **2018**, *5*, 556–567.
19. Liu, Y.; Li, R.; Liu, X.; Wang, J.; Tang, C.; Kang, H. Enhancing Anonymity of Bitcoin Based on Ring Signature Algorithm. In Proceedings of the 13th International Conference on Computational Intelligence and Security (CIS 2017), HongKong, China, 15–18 December 2017; pp. 317–321.
20. Lin, Q.; Yan, H.; Huang, Z.; Chen, W.; Shen, J.; Tang, Y. An ID-based Linearly Homomorphic Signature Scheme and Its Application in Blockchain. *IEEE Access* **2018**, *6*, 20632–20640. [CrossRef]
21. Gao, Y.L.; Chen, X.B.; Chen, Y.L.; Sun, Y.; Niu, X.X.; Yang, Y.X. A Secure Cryptocurrency Scheme Based on Post-quantum Blockchain. *IEEE Access* **2018**, *6*, 27205–27213. [CrossRef]
22. Shamir, A. Identity-based Cryptosystems and Signature Schemes. In Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques, Paris, France, 9–11 April 1984; pp. 47–53.
23. Kumar, P.; Sharma, V.; Sharma, G. Certificateless Aggregate Signature Schemes: A Review. In Proceedings of the 2016 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 29–30 April 2016; pp. 531–536.

24. Al-Riyami, S.S.; Paterson, K.G. Certificateless Public Key Cryptography. In Proceedings of the Advances in Cryptology—ASIACRYPT 2003, 9th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, 30 November–4 December 2003.
25. Zhang, Z.F.; Wong, D.S.; Xu, J.; Feng, D. Certificateless Public-key Signature: Security Model and Efficient Construction. In Proceedings of the International Conference on Applied Cryptography and Network Security, Singapore, 6–9 June 2006; pp. 293–308.
26. Boneh, D.; Gentry, C.; Lynn, B.; Shacham, H. Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2003), Warsaw, Poland, 4–8 May 2003; pp. 416–432.
27. He, D.B.; Tian, M.; Chen, J. Insecurity of an Efficient Certificateless Aggregate Signature with Constant Pairing Computations. *Inf. Sci.* **2014**, *268*, 458–462. [CrossRef]
28. Zhou, Y.W.; Yang, B.; Zhang, W.Z. Efficient and Provide Security Certificateless Aggregate Signature Scheme. *J. Softw.* **2015**, *26*, 3204–3214. (In Chinese)
29. Cui, J.; Zhang, J.; Zhong, H.; Shi, R.H.; Xu, Y. An Efficient Certificateless Aggregate Signature without Pairings for Vehicular Ad Hoc Networks. *Inf. Sci.* **2018**, *451*, 1–15. [CrossRef]
30. Wang, Z.Y.; Liu, J.W.; Zhang, Z.Y.; Yu, H. Fully Anonymous Blockchain based on Aggregate Signature and Confidential Transaction. *J. Comput. Res. Dev.* **2018**, *55*, 2185–2198. (In Chinese)
31. Yao, A.C.-C.; Zhao, Y.L. Online/offline Signatures for Low-power Devices. *IEEE Trans. Inf. Forensic Secur.* **2012**, *8*, 283–294. [CrossRef]
32. Danzi, P.; Kalør, A.E.; Stefanović, Č.; Popovski, P. Repeat-Authenticate Scheme for Multicasting of Blockchain Information in IoT Systems. *arXiv* **2019**, arXiv:1904.07069.
33. Kaga, Y.; Fujio, M.; Naganuma, K.; Takahashi, K.; Murakami, T.; Ohki, T.; Nishigaki, M. A Secure and Practical Signature Scheme for Blockchain Based on Biometrics. In Proceedings of the Information Security Practice and Experience (ISPEC 2017), Melbourne, VIC, Australia, 13–15 December 2017; pp. 877–891.
34. Johnson, D.; Menezes, A.; Vanstone, S. The Elliptic Curve Digital Signature Algorithm (ECDSA). *Int. J. Inf. Secur.* **2001**, *1*, 36–63. [CrossRef]
35. Krawczyk, H.M.; Rabin, T.D. Chameleon Signatures. In Proceedings of the Network and Distributed System Security Symposium (NDSS 2000), San Diego, CA, USA, 2–4 February 2000; pp. 143–154.
36. Shamir, A.; Tauman, Y. Improved Online/Offline Signature Schemes. In Proceedings of the 21th Annual International Cryptology Conference (CRYPTO 2001), Santa Barbara, CA, USA, 19–23 August 2001; pp. 355–367.
37. Shu, H.; Chen, F.L.; Xie, D.; Sun, L.P.; Qi, P.; Huang, Y.Q. An Aggregate Signature Scheme Based on a Trapdoor Hash Function for the Internet of Things. *Sensors* **2019**, *19*, 4239. [CrossRef]
38. Chandrasekhar, S.; Singhal, M. Efficient and Scalable Aggregate Signcryption Scheme based on Multi-trapdoor Hash Functions. In Proceedings of the 1st Workshop on Security and Privacy in the Cloud, Florence, Italy, 28–30 September 2015; pp. 610–618.
39. Chandrasekhar, S.; Ibrahim, A.; Singhal, M. A Novel Access Control Protocol Using Proxy Signatures for Cloud-based Health Information Exchange. *Comput. Secur.* **2017**, *67*, 73–88. [CrossRef]
40. Cheng, L.; Wen, Q.Y.; Jin, Z.P.; Zhang, H.; Zhou, L.M. Cryptanalysis and Improvement of a Certificateless Aggregate Signature Scheme. *Inf. Sci.* **2015**, *295*, 337–346. [CrossRef]
41. Maji, H.K.; Prabhakaran, M.; Rosulek, M. Attribute-based Signatures. Proceedings of Cryptographers' Track at the RSA conference, San Francisco, CA, USA, 14–18 February 2011; pp. 376–392.
42. Health informatics-Pseudonymization, ISO 25237. 2017. Available online: https://www.iso.org/standard/63553.html (accessed on 7 January 2020).