The Journal of Supercomputing (2023) 79:18820–18848
https://doi.org/10.1007/s11227-023-05363-4

A secure and lightweight cloud-centric intelligent medical system based on Internet of Medical Things

Tong Mu[1] · Qiaochuan Ren[1] · BiLin Shao[1] · Genqing Bian[1] · Jing Song[1]

Accepted: 28 April 2023 / Published online: 19 May 2023

Abstract
Internet of Medical Things (IoMT) technology is widely used in intelligent medical treatment; however, massive mobile data transmission makes the cellular network overburden. The combination of medical Internet of Things technology and cloud storage can greatly improve this problem. Most of the existing medical Internet of Things systems cannot be adapted to the environment with limited resources after considering security and privacy, or after ensuring implementation efficiency, it will lead to reduced system security; that is, the mutual authentication function of the system is easily destroyed. The combination of medical Internet of Things technology and medical cloud storage technology can greatly improve the current smart medical environment. We propose an escrow-free identity-based scheme (EF-IDS) to ensure the function of mutual authentication between system entities, propose a secure lightweight cloud-centric smart medical system based on the medical Internet of Things based on EF-IDS and prove that our system ensures the privacy and security of users' personal health information. It also provides the ability to publicly verify the integrity of the data stored on the medical cloud server. Finally, the performance analysis shows that our proposed system has significant advantages in both communication overhead and computational cost.

Keywords Escrow-free identity-based scheme (EF-IDS) · Internet of Medical Things (IoMT) · Public verifiability

1 Introduction

The Internet of Things is a system that relies on autonomous communication between groups of physical objects, including a group of object networks, such as intelligent machines, intelligent cars, and intelligent home appliances, which

✉ Tong Mu
mu_tong@xauat.edu.cn

[1] School of Information and Control Engineering, Xi'an University of Architecture and Technology, No. 13, Middle Yanta Road, Xi'an 710055, People's Republic of China

communicate with each other and use unique Internet addresses to communicate with external devices or networks [1].

There are many applications based on the Internet of Things, such as smart home, smart city, smart industrial automation, and smart services. IoT systems deliver improved productivity, efficiency and quality to a wide range of service providers and industries.

Currently, the application of the rapidly developing Internet of Things (IoT) technology in the field of medical testing has attracted considerable research attention. A typical wireless body area network (WBAN) is a network of various tiny sensors that collect a patient's personal health information (PHI) via sensors implanted or placed in the patient's body. Specifically, wireless body area network is a network used in ubiquitous healthcare to collect and remotely transmit a patient's real-time PHI data by connecting and communicating with implanted or worn sensors such as smart sphygmomanometer, smart glucose meter, smart bracelet, smart pacemaker, and smart pulse monitor, including respiratory rate, heart rate, and blood pressure [2] for use by health care providers, doctors, and hospitals to provide better support and medication.

Typically, IOMTs consist of a variety of tiny sensors that have limited battery life, storage space, and computing power. After collecting patient health information data over a period of time, these sensors send it to medical professionals (i.e., data consumers) over a public network [3]. Obviously, the patient's PHI is crucial, as any malicious or controlled sensors or unauthorized access to the patient's PHI can pose a life-threatening risk to the patient's health. Therefore, security and privacy issues are two extremely important challenges facing the further application of wireless body area network [4].

While mobile technology has benefited smart healthcare, the increasing data transfer is overloading cellular networks. The cloud-based Internet of Things shows great promise in the storage and processing of medical data. Cloud server is an outsourcing platform with a large amount of storage memory and computing resources. Cloud services are usually provided by powerful and well-known companies, which provide users with sufficient storage space and powerful computing power [5]. Therefore, patients can use cloud servers to efficiently store, manage, and share massive medical data generated by various medical sensors, which is not only convenient for users to access, but also can improve the storage utilization of health information system. However, in the medical Internet of Things system, patients who outsource their health data to the cloud server will also face data integrity problems [6]. Because when a data file is uploaded to a cloud server, the data owner loses direct control over the file. Sometimes, dishonest cloud servers may inadvertently delete files or actively modify files and hide it to save storage space or gain other financial benefits. Therefore, to prevent such attacks, it is necessary to authenticate the integrity of the data stored in the cloud server.

In addition, ensuring user legitimacy is critical to a data sharing scheme. PHI can be tampered with or falsified by unauthorized users, which poses a health risk to patients, as medical professionals may make incorrect decisions and recommendations based on incorrect information. Therefore, it is necessary to design a low cost and lightweight data sharing scheme to meet the security requirements and reduce energy consumption as much as possible.

To sum up, how to build a secure cloud-assisted medical Internet of Things system is crucial for the future of smart medicine.

1.1 Related work

In 2019, Sun et al. [7] reviewed the security and privacy challenges of IoT in medical systems and discussed future research directions. Akinyele et al. [8] proposed a self-protection electronic medical record system based on attribute encryption on mobile devices.

Hu et al. [9] used attribute-based encryption technology to solve the secure communication between the body area network and the data consumer (final user). Chandrasekaran et al. [10] reported the low efficiency of the system [9] in multidata transmission and proposed a secure data communication system for multidata transmission in the WBAN.

Li et al. [11] proposed the use of identity-based signature encryption for low-power devices to set up online or offline sensors to satisfy both authentication and confidentiality without additional authentication steps by using the receiver's public key. However, this scheme is vulnerable to the well-known key escrow attack. Therefore, in [12], Omala et al. proposed a lightweight certificate-free signcryption scheme with the help of certificate-free encryption technology. Subsequently, Zhang et al. extended the technique proposed by Omala et al. and discussed the data communication scheme of the electronic medical system using a generalized signcryption scheme [13]. However, Zhou revealed that the protocol proposed in [13] is vulnerable to internal attacks reported in [14]. Thus, protocol is fragile in data confidentiality and not secure.

In 2020, Kumar and Chand proposed a cloud-centric intelligent medical system (KC system) based on the medical IoT [15]. Specifically, they proposed escrow-free identity-based aggregated signcryption (EF-IBASC) public key encryption to ensure the privacy and identity verification of PHI and developed a device to device using the KC system. The security of the system is based on the underlying EF-IBASC scheme. As stated by Kumar, the health care system has numerous advantages, including privacy protection of PHI and the mutual authenticity of authentication entities because encryption and signature functions can be provided by the underlying signcipher scheme.

However, in subsequent studies, Kumar et al. found that the KC system was unsafe [16] because the attacker can calculate the private key of the entity from the communication content transmitted in the network. Therefore, entity authentication and registration become meaningless. This result completely invalidates the mutual authentication function between entities. A malicious attacker may obtain the private key of the entity by disguising as a legitimate entity to join the system to break the intelligent medical system. The KC system was improved in a subsequent study [16] to overcome existing security loopholes in the system.

In this study, we proved that the improved KC system is still insecure, and its key authentication function cannot be guaranteed because the private key of the personal auxiliary device (PAD) in the system can still be obtained from the network,

and the content of the transmitted communication is restored. Furthermore, in the KC system, the biomedical sensor (BMS) achieved excellent privacy and authenticity through signcryption. Although the cost of the signcryption operation is smaller than that of first signing and subsequent encrypting or first encrypting and then signing, a BMS is a resource-constrained device. Therefore, the operation of the device should be simplified. We found that the public verification algorithm is ineffective.

In 2021, Zhang et al. [17] designed an efficient and secure electronic personal health record sharing system based on the Boneh–Franklin identity encryption scheme. Their scheme is sufficiently lightweight for use in mobile devices and allows both parties to decrypt the ciphertext without reconstructing the private key. In 2022, Liu et al. [18] proposed the first DSSE scheme that can be satisfactorily applied to personal health record file databases and resists file injection attacks. This affected effective access control to protect the privacy of patients' personal health record files. Wang et al. [19] studied edge computing by introducing the framework of federated learning and designed a lightweight privacy protection protocol based on secret sharing and weight masks. The scheme was extended as a security system. We proved that the system for edge computing can protect the privacy of medical data and simultaneously reduce the communication overhead. Zhou et al. [20] designed a human-in-loop-aided (HITL-aided) scheme to protect privacy in intelligent healthcare. In this scheme, block design technology is used to blur various health indicators of hospitals and smart wearable devices. After introducing human-in-the-loop (HITL), the smart medical platform was used to realize privacy access to health reports.

1.2 Our contribution

1. We first analyzed the security of the improved KC system [16] and revealed that the previous system was insecure. Malicious adversaries may join the system disguised as legitimate entities and break the intelligent medical system. Thus, the mutual authentication function of the improved KC system has become invalid.
2. On the basis of the bilinear Diffie–Hellman problem, we proposed an escrow-free identity-based signature scheme (EF-IDS) and proved its security in the random oracle model.
3. A secure intelligent medical system was constructed, and the efficiency and secure data transmission mechanism from BMS to PAD, MCS, and SD were considered. The system can accomplish public verification of the data stored by the user on the MCS.
4. Finally, performance analysis on the proposed system revealed the system is efficient in terms of communication overheads and computation cost compared with those in [15, 16], especially for BMS.

1.3 Organization of research

In Sect. 2, we introduce some basic cryptography techniques to be used in the following paper and introduce the system model of intelligent medical systems. In addition,

we also list the English abbreviations and mathematical symbols used in this paper and their meanings.

In Sect. 3, we analyze whether the improved KC system still has security vulnerabilities and analyze the consequences of its security vulnerabilities.

In Sect. 4, we propose an escrow-free identity-based scheme in view of the security holes in the improved KC system. It is proven that our proposed scheme solves the vulnerability of the KC system.

In Sect. 5, we redesigned the IoMT-based intelligent medical system with detailed protocols.

In Sect. 6, we analyze the security and other attributes of our proposed system. It is proven that our proposed system not only avoids the security vulnerabilities of the improved KC system but also guarantees the integrity of the data stored in the cloud.

In Sect. 7, we analyze the performance of our proposed system, and the analysis shows that our proposed system has a strong performance advantage over Kumar's system, reducing the computing consumption of medical sensors.

In Sect. 8, we summarize the conclusions of this paper and describe future work.

2 Preliminaries

See Table 1.

2.1 Bilinear map

Consider two multiplicative cyclic groups $\mathbb{G}_1$,$\mathbb{G}_2$ with the same prime order q and generator g. Definition $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is a bilinear mapping if it satisfies the following conditions.

- **Computability** It is efficient to compute the value of e.
- **Bilinear** For any $u, v \in \mathbb{G}_1$,$a, b \in Z_q$, It holds that $e(u^a, v^b) = e(u, v)^{ab}$.
- **Nondegenerative** If g is a generator of $\mathbb{G}_1$, it remains $e(g, g) \neq 1_{\mathbb{G}_2}$.

2.2 Bilinear Diffie–Hellman problem

Let the additive cyclic group of the same order $\mathbb{G}_1$ and multiplicative cyclic group $\mathbb{G}_2$, where q is a very large prime number. Let P denote the generator of the group $\mathbb{G}_1$ of length q bits, and $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is a bilinear mapping. Given a tuple $T = \langle P, xP, yP, zP \rangle \in \mathbb{G}_1$, in the absence of information $x, y, Z \in Z_q$, in the case of any probabilistic polynomial time (PPT) algorithm A for the calculation $Z = e(P, P)^{xyz} \in \mathbb{G}_1$, it is difficult to calculate. This formula ensures that the advantage of algorithm A is utilized in solving the problem.

$$\left| \Pr\left[A = e(P, P)^{xyz} \middle| P, xP, yP, zP \in \mathbb{G}_1, x, y, z \in Z_q \right] \right| \geq \varepsilon. \tag{1}$$

Table 1 Symbols and abbreviations

Name	Value
IoMT	Internet of Medical Things
IoT	Internet of Things
WBAN	Wireless body area network
PHI	Personal health information
MCS	Medical cloud server
KC system	Kumar and Chand proposed a cloud-centric intelligent medical system
EF-IBASC	ESCROW-free identity-based aggregated signcryption
PAD	Personal auxiliary device
BMS	Biomedical sensor
EF-IDS	ESCROW-free identity-based signature scheme
PPT	probabilistic polynomial time
NM	Network manager
KPSs	Key protection servers
SD	Service device
PV	Public verifier
BDH	Bilinear Diffie–Hellman
CDH	Computational Diffie–Hellman
IND-CCA	Indistinguishability under chosen-ciphertext attack
EUF-CMA	Existence unforgeable under chosen message attack
λ	Security parameter
q	Order of $\mathbb{G}_1$
e	Bilinear map
$\mathbb{G}_1$	Additive cyclic group
$\mathbb{G}_2$	Multiplicative cyclic group
P	Generator of $\mathbb{G}_1$
BMS_j	The jth BMS
H_1, H_2	Hash functions

2.3 CDH assumption

Let the additive cyclic group $\mathbb{G}_1$ of order q, where q is a very large prime, and let P denote the generator of groups of length q bits. Given a tuple $T = \langle P, aP, bP \rangle \in \mathbb{G}_1$, in the absence of information $a, b \in Z_q$, in the case of any PPT algorithm A for correct output abP, the possibility is negligible.

2.4 System model

The system model based on the IoMT health care system was introduced. As illustrated in Fig. 1, seven types of entities are present in the system. Their details are as follows:

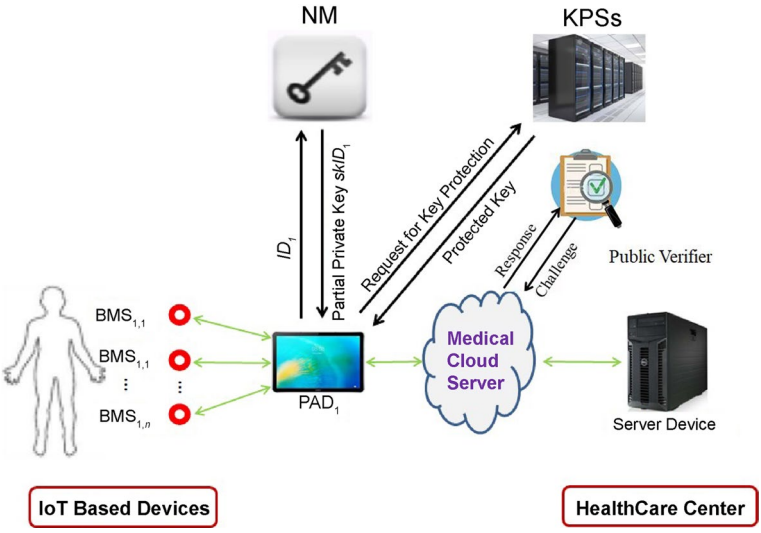

Fig. 1 System model of the IoMT intelligent medical system. NM, Network manager; KPSs, key protection servers; BMS, biomedical sensor; IoMT, Internet of Medical Things; PAD, personal auxiliary device; and IoT, Internet of Things

Network manager (NM): The system would be initialized, and the master key and public key would be calculated through this half-trusted organization. It authenticates the entity and issues a partial private key to it.

Key protection servers (KPSs): With their own keys, KPSs protect the private keys of entities and then send out private keys that are protected shares to them. Calculation is performed on the cloud to reduce computational costs.

BMS: BMSs are miniature sensors with limited storage, typical battery life, and computing power. BMSs are typically placed on/outside of the patient's body or deployed in the patient's tissues. Patient's PHI would be collected, and the symmetric key that is shared by the device for serving is used to encrypt and identify PHI and finally sent to the PAD.

Personal-assisted device (PAD): The data receiver with sufficient storage space and computing power. After gaining some incomplete private key, the PAD completes it by selecting a secret value. The encrypted PHI value that is sent by BMS is verified PAD using BMS. PAD is an entity that is not trusted because, as per Kumar et al., for some opponents, the sensitive data of patients can be easily stolen through statistical attacks or physical attacks.

MCS: MCS is not only an entity can be used to offer storage services to other entities but also has a storage capacity and provides access of encrypted PHI to the server device. Because the possibility that some users' data stored could be lost, MCS is a half-trusted entity.

SD: The PHI value of patients stored on MCS could be visited by the medical organizations and based on the recovered PHI value and patient diagnosis can be performed using this device. Furthermore, with the diagnosis, according to the transition of MCS and PAD, a prescription would be sent to the BMS.

Public verifier (PV): This entity can audit the integrity of data stored in MCS. Specifically, to detect whether the MCS has lost any data blocks stored by the user, PV initiates a challenge to the MCS and receives a response from the MCS. The validator can then verify that the MCS is storing user data.

3 Security analysis of the improved kc system

We analyzed the security of the improved KC system [16]. We first proved that the PAD private key can be easily recovered and subsequently analyzed its consequences.

3.1 Private Key of the entity is not secure

In the final stage of the "entity authentication and registration" algorithm, each entity is $E \in \{BMS, PAD, SD\}$. The private key of each entity is obtained as follows:

$$d_E = s_0\left(s_1 + s_2 + \cdots s_n\right)H_1\left(ID_E\right) \in \mathbb{G}_1 \tag{2}$$

Thus, the secret of each entity cannot be obtained by others. We analyzed and proved that anyone can recover the PAD private key by eavesdropping on the transmission parameters d_{PAD}.

Specifically, recalling the algorithms "PHI Aggregate Signcryption" and "PHI ReAggregation," the j BMS signcrypts the original PHI data $M_{1,j}, M_{2,j}, \ldots, M_{m,j}$ into the signciphertext $CT_j = \langle A_j, B_j, C_j, \sum_{i=1}^{n} D_{i,j}, E_j \rangle$ and subsequently sends the signcryption text to PAD for reaggregation through $c_{PAD} = H_4\left(C_1, C_2, \ldots, C_m\right) \in Z_q^*$ and uses its private key to sign through $C_{\mathrm{PAD}} = c_{\mathrm{PAD}}Y$ and $F = c_{\mathrm{PAD}}d_{\mathrm{PAD}}$. The PAD then sends the $CT_{\mathrm{PAD}} = \langle A_j, B_j, C_j, C_{PAD}, \sum_{i=1}^{n} D_{I,j}, E_j, F \rangle$ go to the MCS.

Any eavesdropper can view the transmitted CT_{PAD}, where the eavesdropper can recover the private key d_{PAD} of this PAD by F and each C_j in the CT_{PAD}.

First, the attacker computes $\mathrm{c}_{\mathrm{PAD}} = H_4\left(C_1, C_2, \ldots, C_m\right)$ with all C_j. From $c_{\mathrm{PAD}} \in Z_q^*$ and prime q, an integer μ exists such that $\mu c_{\mathrm{PAD}} \equiv 1 mod q$, μ can be obtained by the extended Euclidean algorithm. Thus, we have $\mu F = \mu c_{\mathrm{PAD}} d_{\mathrm{PAD}} = \left(\mu c_{\mathrm{PAD}}\right) d_{\mathrm{PAD}} = 1 d_{\mathrm{PAD}} = d_{\mathrm{PAD}}$. A hacker focusing on the CT_{PAD} content can easily compute and recover the PAD's private key d_{PAD}.

3.2 Consequence 1

First, in the improved KC system [16], the NM establishes the system in the "system-setup" stage. Then, first authentication is performed, and the entity is registered through identity ID_E. Finally, the partial private key is calculated using the master private key, which is protected by multiple KPSs. KPSs calculate the protected private keys and forward them to the entity, which merges and shares them to obtain their private keys d_E. This measure ensures that the entity can obtain the correct and

authenticated private key d_E. However, the private key of the PAD, d_{PAD}, can be easily recovered. Therefore, the function of the "Entity's Authentication and Registration" phase is invalid.

3.3 Consequence 2

The private key d_{PAD} of the PAD was exposed. d_{PAD}, which affected the certification stage of the entire medical system. Next, we proved that an attacker can compromise the mutual authentication of the medical system.

Assume a malicious entity or adversary *adv* pretends to pose as a real entity, joins this network to destroy the entire medical system *adv*, and pretends to be an entity *BMS*. The following formula is then performed:

1. For any $a_j^{\mathrm{adv}}, x \in Z_q^*$, and calculate $A_j^{\mathrm{adv}} = a_j^{\mathrm{adv}} xP, B_j^{\mathrm{adv}} = a_j^{\mathrm{adv}} P$, where $Q_{\mathrm{BMS}}^j = H_1\left(ID_{\mathrm{BMS}}^j\right)$.
2. Set $Q_{\mathrm{SD}} = H_1\left(ID_{\mathrm{SD}}\right)$ and $K_j^{\mathrm{adv}} = e\left(a_j^{\mathrm{adv}} xY, Q_{\mathrm{SD}}\right)$, and then calculate the signature key $sk_j^{\mathrm{adv}} = H_2\left(ID_{\mathrm{SD}}, K_j^{\mathrm{adv}}, sk_j^{-\mathrm{adv}}\right)$, where $sk_j^{-\mathrm{adv}}$ is the previous secret key.
3. Set $h_{i,j}^{\mathrm{adv}} = H_3\left(M_{i,j}^{\mathrm{adv}}, A_j^{\mathrm{adv}}, T_{i,j}^{\mathrm{adv}}\right), C_{i,j}^{\mathrm{adv}} = \left(a_j^{\mathrm{adv}} + h_{i,j}^{\mathrm{adv}}\right) xY$.
4. Signcryption $D_{i,j}^{\mathrm{adv}} = M_{i,j}^{\mathrm{adv}} || C_{i,j}^{\mathrm{adv}} || ID_{\mathrm{BMS}}^j || T_{i,j}^{\mathrm{adv}} \oplus sk_j^{\mathrm{adv}}$.
5. Calculate $C_{\mathrm{aggr},j}^{\mathrm{adv}} = H_4\left(C_{1,j}^{\mathrm{adv}}, C_{2,j}^{\mathrm{adv}}, \ldots, C_{m,j}^{\mathrm{adv}}\right), E_j^{\mathrm{adv}} = C_{\mathrm{aggr},j}^{\mathrm{adv}} xY$ and $C_j^{\mathrm{adv}} = C_{\mathrm{aggr},j}^{\mathrm{adv}} Y$.
6. Let *adv* $sendCT_j^{\mathrm{adv}} = \langle A_j^{\mathrm{adv}}, B_j^{\mathrm{adv}}, C_j^{\mathrm{adv}}, \sum_{i=1}^n D_{i,j}^{adv}, E_j^{\mathrm{adv}} \rangle$ to PAD.

Similarly, *adv* disguised as a real one *PAD* communicates with other entities by generating transmission parameters. An adversary can easily obtain all *PAD* private keys d_{PAD} by calculating $c_{\mathrm{PAD}}^{\mathrm{adv}} = H_4\left(C_1^{\mathrm{adv}}, C_2^{\mathrm{adv}}, ..., C_m^{\mathrm{adv}}\right)$, $C_{\mathrm{PAD}}^{\mathrm{adv}} = c_{\mathrm{PAD}}^{\mathrm{adv}} Y$ and $F_{\mathrm{adv}} = c_{\mathrm{PAD}}^{\mathrm{adv}} d_{\mathrm{PAD}}$.

Therefore,*adv* to *MCS* upload $CTPADadv = \langle A_j^{\mathrm{adv}}, B_j^{\mathrm{adv}}, C_j^{\mathrm{adv}}, C_{\mathrm{PAD}}^{\mathrm{adv}}, \sum_{i=1}^n D_{i,j}^{\mathrm{adv}}, E_j^{\mathrm{adv}}, F_{\mathrm{adv}} \rangle$.

Two public verification formulas (3) and (4) can pass the verification.

$$e\left(E_j^{\mathrm{adv}}, B_j^{\mathrm{adv}}\right) = e\left(C_{\mathrm{aggr},j}^{\mathrm{adv}} xY, a_j^{\mathrm{adv}} P\right) = e\left(a_j^{\mathrm{adv}} xP, C_{\mathrm{aggr},j}^{\mathrm{adv}} Y\right) = e\left(A_j^{\mathrm{adv}}, C_j^{\mathrm{adv}}\right) \quad (3)$$

$$\begin{aligned} e\left(F_{\mathrm{adv}}, P\right) &= e\left(c_{\mathrm{PAD}}^{\mathrm{adv}} d_{\mathrm{PAD}}, P\right) \\ &= e\left(s_0\left(s_1 + s_2 + \cdots s_n\right) H_1\left(ID_{\mathrm{PAD}}\right), c_{\mathrm{PAD}}^{\mathrm{adv}} P\right) \\ &= e\left(H_1\left(ID_{\mathrm{PAD}}\right), c_{\mathrm{PAD}}^{\mathrm{adv}} s_0\left(s_1 + s_2 + \cdots s_n\right) P\right) \\ &= e\left(H_1\left(ID_{\mathrm{PAD}}\right), C_{\mathrm{PAD}}^{\mathrm{adv}}\right) \end{aligned} \quad (4)$$

Although malicious adversaries *Adv* can replace BMSs or PADs, the improved KC system could not detect or avoid it. Therefore, the mutual authentication function of the system of Kumar et al. [16] fails.

3.4 Invalid public verifiable algorithm

As demonstrated by Zhou et al. [19], even if an adversarial cloud server cannot satisfactorily maintain outsourcing data satisfactorily, it can still pass the audit of the "public verifiability" algorithm. This problem exists in both [15, 16]. The detailed description of the proof is omitted in this study.

4 Escrow-free identity-based scheme and its security

Kumar et al.'s cloud-centric healthcare IoT system has many problems. We designed an escrow-free identity-based scheme to ensure mutual authentication and nonforgerability of PHI uploaded by patients to the cloud. In this section, we demonstrate the security of EF-IDS because EF-IDS solves the private key leak problem of 3.1, thus avoiding the consequences of 3.2 and 3.3.

The following BDH-based escrow-free identity-scheme, namely EF-IDS, was introduced:

Setup This is the master key generation algorithm and outputsNM, master key of KPS_s, and system parameters. Specifically, given a given security parameter$\left(1^{\lambda}\right)$, the algorithm generates bilinear maps$e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$, where $\mathbb{G}_1$ and $\mathbb{G}_2$ are additive and multiplicative cyclic groups with the same prime order$q(|q| \geq \lambda)$, where P is a generator of $\mathbb{G}_1$. Furthermore, $u \in \mathbb{G}_1$ is selected randomly. H_1 is defined as the hash function from $\{0,1\}^*$ to $\mathbb{G}_1$ and H_2 as the hash function from $\{0,1\}^*$ to Z_q^*.

NP and KPS each randomly select $< s_0, s_1, s_2, \ldots, s_n > \in Z_q^*$, where s_0 is the master key of NM and $< s_1, s_2, \ldots, s_n >$ is the key of KPS_s. NM computes $P_0 = s_0 P$ and sends P_0 to KPS_i. KPS_i calculates the $P_i = s_i P_0$ and keeps the s_i secret and responds the P_i back to NM. NM computes the system public key $Y = \sum_{i=1}^{i=n} P_i = s_0\left(s_1 + s_2 + \cdots s_n\right)P$. The system parameters $params = \left(e, \mathbb{G}_1, \mathbb{G}_2, P, u, H_1, H_2, Y\right)$ are exposed, and their master key is kept secret. **KeyGen** On the identity of a given entity ID_E,

1. Select $x_E \in Z_q^*$ at random, calculate $X_E = x_E P$, $D_E = x_E Q_E$, where $Q_E = H_1(ID_E)$, and send $< X_E, ID_E, D_E >$ to NM.
2. NM calculates $D_{E0} = s_0 D_E$ and returns to E.
3. E requests KPS_i for key protection and sends D_{E0} to KPS_i.
4. KPS_i calculates the protected partial private key sharing $D_{Ei} = s_i D_{E0}$ and sends D_{Ei} to E.

5. Entity E computes its private key $d_E = x_E^{-1} \sum_{i=1}^{i=n} D_{Ei} = s_0(s_1 + s_2 + \cdots s_n)Q_E$。

Sign This is a signature algorithm run by signers who identify as $\mathrm{ID_S}$. Specifically, given a message $M = (M_1, M_2, \ldots, M_m)$, the signer randomly selects a secret value $s \in Z_q^*$ and computes $A_s = sQ_s$, $B_s = sP$,$C_s = sQ_R$, and $F_s = sH_2(M_1, M_2, \ldots, M_m)(Y + d_S)$ as the signature of the message M.

Verify Given system parameters and message signature (M, A_s, B_s, C_s, F_s), the receiver uses its own private key DK to verify whether the following three equations are true.

$$e(C_s, Y) = e(d_R, B_s) \tag{5}$$

$$e(A_s, P) = e(Q_s, B_s) \tag{6}$$

$$e(F_s, Q_R) = e(A_s + B_s, H_2(M_1, M_2, \ldots, M_m)d_R) \tag{7}$$

For validating the message. If so, output is 1. Otherwise, the output is 0.

The correctness of (5) can be verified as follows:

$$e(C_s, Y) = e(sQ_R, s_0(s_1 + s_2 + \cdots s_n)P) = e(s_0(s_1 + s_2 + \cdots s_n)Q_R, sP) = e(d_R, B_s) \tag{8}$$

The correctness of (6) can be verified as follows:

$$e(A_s, P) = e(sQ_s, P) = e(Q_s, sP) = e(Q_s, B_s) \tag{9}$$

The correctness of (7) can be verified as follows:

$$\begin{aligned} e(F_s, Q_R) &= e(sH_2(M_1, M_2, \ldots, M_m)Y + d_S, Q_R) \\ &= e(ss_0(s_1 + s_2 + \cdots s_n)P + s_0(s_1 + s_2 + \cdots s_n)Q_s, H_2(M_1, M_2, \ldots, M_m)Q_R) \\ &= e(sP + Q_s, H_2(M_1, M_2, \ldots, M_m)s_0(s_1 + s_2 + \cdots s_n)Q_R) \\ &= e(A_s + B_s, H_2(M_1, M_2, \ldots, M_m)d_R) \end{aligned} \tag{10}$$

Regarding its security, we have the following points:

Theorem 1 *If the BDH assumption holds for $\mathbb{G}_1$, the scheme EF-IDS is safe in the random oracle. Here,H_1 and H_2 were modeled as random oracle s, respectively. Outputting an effective forgery is unfeasible. Thus, for any PPT adversary*

Proof. We discuss two types of adversaries in the EF-IDS scheme, which are called Type-I and Type-II. Informally, Type-I adversary A_i represents a general adversary (i.e., non-NM and KPS) and cannot access the master keys of NM and KPS. Class II adversary A_{II} represents a malicious NM, which can also collude with $(n-1)$ of the n KPSs and is not allowed to change the public key of any user. If the second type of adversary succeeds, a key escrow problem occurs. Next, a reduction from the BDH assumption to the security of EF-IDS is established for A_i and A_{II}.

Type-I Adversary Assume that the generator of $\mathbb{G}_1$ is P. Let B_I be the algorithm that attacks the BDH assumption on $\mathbb{G}_1$. Here, B_I simulates the environment for the PPT adversary A_I. Specifically, a given tuple is given (P, aP, bP, cP), B_I defined a linear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$.

Here, B_I does not know that $a, b, c \in \mathbb{Z}_q$. The hash function H_1 and H_2 are modeled as a random oracle simulated by B_I. B_I set up $Y = cP \in \mathbb{G}_1$ and sends $\left(e, \mathbb{G}_1, \mathbb{G}_2, P, Y\right)$ to A_I.

Suppose A_i makes at most q_i queries to $H_i (i = 1, 2) -$ Oracle, q_k private key queries, and q_s signature queries. Then, A_i wins the EUF-CMA-I game by a nonnegligible advantage ε. Algorithm B_i can solve the BDH problem in polynomial time t' with advantages $\mathrm{Adv}\left(A_i\right)$ and time t', where t_{B_I} is the running time of algorithm B_i. Here, B_i selects two numbers $S, R \in \left\{1, 2, \ldots, q_1\right\}$ randomly as B_i's guess on the identities of the final sender and receiver, where S is the sender and R is the receiver.

Phase 1 Here,A_I asks the following query.

1. $\boldsymbol{H_1 - Oracle}$ **queries:** A_I requests H_1 query on ID_i, B_I randomly selects $x_i \in \mathbb{Z}_q$, where $i \in \left\{1, 2, \ldots, q_1\right\}$, and checks whether it is the Sth or Rth queries. If not, compute $Q_i = H_1\left(ID_i\right) = x_i P$. Otherwise, B_I outputs $H_1\left(ID_S\right) = x_S aP$ and $Q_R = H_1\left(ID_R\right) = x_R bP$ and to A_I.
2. $\boldsymbol{H_2 - Oracle}$ queries: Answers to the $H_2 - Oracle$ query are simply sampled by delay.
3. **KeyGen queries:** For ID_i of the query, B_I performs a H_1 query on it. If it is not the Sth or Rth H_1 queries, B_I computes and returns $d_i = x_i cP$; otherwise, $\perp$ is returned.
4. **Signature query:** A_I commits tuple $\left(M, ID_i, ID_j\right)$ to this $Oracle$. Then, for the query from A_I, it sends identity ID_i and receives identity ID_j, and B_I checks whether ID_i and ID_j are the S th or R th H_1 queries. If not, B_I randomly selects $s \in Z_q^*$, sets $A = sx_i P, B = sP, C = sx_j P$, and $F = sH_2(M)\left(Y + d_i\right)$ and returns $\langle ID_i, ID_j, M, A, B, C, F\rangle$ to A_I Otherwise, $\perp$ is returned.

Forge Finally, the A_I responds to the forged message signature pair $\langle ID_i^*, ID_j^*, M^*, A^*, B^*, C^*, F^*\rangle$, which satisfies $e(C^*, Y) = e\left(d_j^*, B^*\right)$, $e(A^*, P) = e\left(Q_i^*, B^*\right)$ and $e\left(F^*, Q_j^*\right) = e\left(A^* + B^*, H_2(M^*) d_j^*\right)$. If the sending identity ID_i^* and receiving identity ID_j^* are not ID_S and ID_R, the process is aborted. Every other case has $A^* = sx_S aP$, $B^* = sP$, $C^* = sx_R bP$, and $F^* = sH_2(M^*)\left(Y + d_S\right) = sH_2(M^*)(c + x_S ac)P$. B_I computes formula (11) where formula (12) is verified.

$$W = \left(\frac{e(F^*, C^*)}{e\left(sH_2(M^*)Y, C^*\right)}\right)^{\frac{1}{ssH_2(M^*)x_S x_R}} \tag{11}$$

$$
\begin{aligned}
W &= \left(\frac{e(F^*, C^*)}{e\left(sH_2(M^*)Y, C^*\right)} \right)^{\frac{1}{ssH_2(M^*)x_S x_R}} \\
&= \left(\frac{e\left(sH_2(M^*)c + x_S acP, sx_R bP\right)}{e\left(sH_2(M^*)cP, sx_R bP\right)} \right)^{\frac{1}{ssH_2(M^*)x_S x_R}} \\
&= \left(\frac{e\left(sH_2(M^*)cP, sx_R bP\right) \cdot \left(sH_2(M^*)x_S acP, sx_R bP\right)}{e\left(sH_2(M^*)cP, sx_R bP\right)} \right)^{\frac{1}{ssH_2(M^*)x_S x_R}} \\
&= e\left(sH_2(M^*)x_S acP, sx_R bP\right)^{\frac{1}{ssH_2(M^*)x_S x_R}} = e(P, P)^{\mathrm{abc}}
\end{aligned}
\tag{12}
$$

So we have formula (13) to the BDH problem as the solution.

$$
W = \left(\frac{e(F^*, C^*)}{e\left(C^*, sH_2(M^*)Y\right)} \right)^{\frac{1}{ssH_2(M^*)x_S x_R}} = e(P, P)^{abc} \tag{13}
$$

If simulator B_I guesses S, R, the signature query of tuple $\left(M, ID_i, ID_j\right) = \left(M, ID_S, ID_R\right)$ can be simulated, and the forged signature can be reducible because the message of the choice of the signature query should differ from ID_S and ID_R. Thus, the probability of successful simulation and useful attack is $1/q_1(q_1 - 1)$. Assume adversary A_I cracks this scheme with $(t, 1, \varepsilon)$ by executing q_1th $H_1 - Oracle$ query, the advantage of solving the BDH problem is $\varepsilon/q_1\left(q_1 - 1\right)$. Here, T_S represents the time cost of the simulation, and we have $T_S = O(1)$. Furthermore, $B_I\left(t + T_S, \varepsilon/q_1(q_1 - 1)\right)$ is used to solve the BDH problem.

Type-II Adversary We constructed another algorithm $\mathrm{B_{II}}$. Here, B_{II} uses A_{II} as a subroutine to attack the BDH hypothesis. Given tuple (P, aP, bP, cP), B_{II} simulates system parameters B_i, in addition to generating Y. Specifically, B_{II} selects $h, r \in Z_q^*$ as the master key randomly; then, $Y = hrP + hcP \in \mathbb{G}_1$. Subsequently, we provide $\left(e, \mathbb{G}_1, \mathbb{G}_2, P, Y, h, r\right)$ to A_{II} and select two numbers $S, R \in \left\{1, 2, \ldots, q_1\right\}$ as B_{II}'s guess on the identities of the final sender and receiver, where S is the sender and R is the receiver.

- **Phase 1**: A_{II} asks the following query.
- $\boldsymbol{H_1 - Oracle}$ **queries:** A_{II} runs the same query as Theorem 1.
- $\boldsymbol{H_2 - Oracle}$ **queries:**A_{II} runs the same query as Theorem 1.
- **KeyGen queries:** For ID_i of the query, B_{II} performs a H_1 query on it. If it is not the Sth or Rth queries, B_{II} computes and returns$d_i = x_i hrP + x_i hcP$; otherwise, $\perp$ is returned.
- **Signature query:** A_{II} commits tuple $\left(M, ID_i\right)$ to this Oracle. Then, for the query from A_{II}, it sends identity ID_i and receives identity $\mathrm{ID_j}$, and B_i checks whether ID_i and $\mathrm{ID_j}$ are S th or R th H_1 queries. If not, B_{II} randomly selects $\mathrm{s} \in Z_{\mathrm{q}}^*$, sets $\mathrm{A} = \mathrm{sx}_i P, \mathrm{B} = \mathrm{sP}, \mathrm{C} = \mathrm{sx_j} P$, and $F = sH_2(M)\left(Y + d_i\right)$, and returns $\langle ID_i, ID_j, M, A, B, C, F\rangle$ to A_{II} Otherwise, $\perp$ is returned.

Forge Finally, A_{II} responds to the forged message signature pair $\langle ID_i^*, ID_j^*, M^*, A^*, B^*, C^*, F^* \rangle$, which satisfies $e(C^*, Y) = e\left(d_j^*, B^*\right)$, $e(A^*, P) = e\left(Q_i^*, B^*\right)$ and $e\left(F^*, Q_j^*\right) = e\left(A^* + B^*, H_2(M^*)d_j^*\right)$. If the sending identity ID_i^* and receiving identity ID_j^* are not ID_S and ID_R, the process is aborted. Every other case has $A^* = sx_S aP$, $B^* = sP$, $C^* = sx_R bP$, and $F^* = sH_2(M^*)\left(Y + d_S\right) = sH_2(M^*)(h(r+c) + x_S ah(r+c))P$. B_{II} computes formula (14) where formula (15) is verified.

$$W = \left(\frac{e(F^*, C^*)}{e\left(sH_2(M^*)Y, C^*\right) \cdot e\left(H_2(M^*)hrA^*, C^*\right)} \right)^{\frac{1}{ssH_2(M^*)hx_S x_R}} \tag{14}$$

$$\begin{aligned} W &= \left(\frac{e(F^*, C^*)}{e\left(sH_2(M^*)Y, C^*\right) \cdot \left(H_2(M^*)hrA^*, C^*\right)} \right)^{\frac{1}{ssH_2(M^*)hx_S x_R}} \\ &= \left(\frac{e\left(sH_2(M^*)h(r+c) + x_S ah(r+c)P, C^*\right)}{e\left(sH_2(M^*)Y, C^*\right) \cdot \left(H_2(M^*)hrA^*, C^*\right)} \right)^{\frac{1}{ssH_2(M^*)hx_S x_R}} \\ &= \left(\frac{e\left(sH_2(M^*)Y + sH_2(M^*)x_S ah(r+c)P, C^*\right)}{e\left(sH_2(M^*)Y, C^*\right) \cdot \left(H_2(M^*)hrA^*, C^*\right)} \right)^{\frac{1}{ssH_2(M^*)hx_S x_R}} \\ &= \left(\frac{e\left(H_2(M^*)hrA^* + sH_2(M^*)x_S ahcP, C^*\right)}{e\left(H_2(M^*)hrA^*, C^*\right)} \right)^{\frac{1}{ssH_2(M^*)hx_S x_R}} \\ &= e\left(sH_2(M^*)x_S ahcP, sx_R bP\right)^{\frac{1}{ssH_2(M^*)hx_S x_R}} \\ &= e(P, P)^{\text{abc}} \end{aligned} \tag{15}$$

We have $W = e(P, P)^{\text{abc}}$ to the BDH problem as the solution.

If the simulator B_I successfully guesses S, R, the signature query of tuple $\left(M, ID_i, ID_j\right) = \left(M, ID_S, ID_R\right)$ can be simulated, and the forged signature can also be reducible too because the message of the choice of the signature query should differ from ID_S and ID_R. Therefore, the probability of successful simulation and useful attack is $1/q_1(q_1 - 1)$. Assume that adversary A_I cracks this scheme with $(t, 1, \varepsilon)$ by executing q_1th $H_1 - Oracle$ query. The advantage of solving the BDH problem is $\varepsilon/q_1(q_1 - 1)$.

Here, T_S represents the time cost of the simulation, and we have $T_S = O(1)$, B_I $\left(t + T_S, \varepsilon/q_1\left(q_1 - 1\right)\right)$ is used to solve the BDH problem.

5 Proposed IoMT-based intelligent medical system

In this section, we provide the specific algorithmic implementation of the proposed smart healthcare system, which consists of the following seven stages.

5.1 System initialization

First, *NM* and KPS_s follow the steps below to generate the system master key and system public parameters.

Algorithm 1 System initialization

1. Given a security parameter 1^{λ}, NM selects an element $q(|q|) \geq \lambda$). Let addition group $\mathbb{G}_1$ and multiplicative group $\mathbb{G}_2$ of order q, $\mathbb{G}_1$holds the generator p, and then randomly choose an element $u \in \mathbb{G}_1$, pairing function $e: \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. Pseudorandom permutation $\pi: Z_q^* \times \{1, \ldots, n\} \to \{1, \ldots, n\}$ and pseudorandom function $\varphi: Z_q^* \times Z_q^* \to Z_q^*$.
2. Assume two one-way cryptographic hash functions.$H_1: \{0,1\}^* \to \mathbb{G}_1$,$H_2: \{0,1\}^* \to Z_q^*$.
3. NM selects an element $s_0 \in Z_q^*$ and sets the public key $P_0 = s_0 P$ and sends p to KPS_i.
4. KPS_iselects an element $s_i \in Z_q^*$ and calculates $P_i = s_i P_0$ and keeps s_i secret to respond P_i back to NM.
5. NM combines all the received parameters and computes the system public key $Y = \sum_{i=1}^{i=n} P_i = s_0(s_1 + s_2 + \cdots s_n)P$.
6. NM keeps $s_0$secret and releases $params = (q, e, \pi, \varphi, P, \mathbb{G}_1, \mathbb{G}_2, u, Y, H_1, H_2, P_1, P_2, \cdots, P_n)$.

5.2 BMS, PAD, and SD registration

The NM verifies the identity of the new entity that it wants to add to the network and issues a partial private key. The entity then seeks partial private key protection from multiple KPSs. Its private key is extracted.

Algorithm 2 escrow-free identity-based key agreement for BMS, PAD, and SD

Entity $E \in \{BMS, PAD, SD\}$ has an identity ID_E, take an element $x_E \in Z_q^*$, set $X_E = x_E P$, $D_E = x_E Q_E$, where $Q_E = H_1(ID_E)$ and send $< X_E, ID_E, D_E >$ to NM.

NM uses equation $e(Q_E, X_E)? = e(D_E, P)$ to verify the parameters, calculates part of the private key as $D_{E0} = s_0 D_E$,$X_{E0} = s_0 X_E$, and sends D_{E0} to E.

E uses equality$e(D_E, P_0)? = e(D_{E0}, P)$ to check part of the private key and requests KPS_i for key protection

KPS_i verifies the partial private key using equality $e(Q_E, X_{E0})? = e(D_{E0}, P)$, computes the protected partial private key share $D_{Ei} = s_i D_{E0}$, and sends D_{Ei} to E.

E uses equality $e(D_E, P_i)? = e(D_{Ei}, P)$ to check the protected partial private key share.

E unblinds and computes its own private key $d_E = x_E^{-1} \sum_{i=1}^{i=n} D_{Ei} = s_0(s_1 + s_2 + \cdots s_n)Q_E$.

5.3 Data communication from BMS to PAD

We now focus on the authentication communication from the BMS to the PAD. Each PAD is assumed to be connected to nth BMSs in the general case. Here, $m_{i,j}$ can denote PHI collected at time $t_{i,j}$ for the jth BMS, where $1 \le i \le m$ and $1 \le j \le n$. Algorithm 2 describes secure data communication from BMS_j to the PAD. Thus,BMS_j encrypted the collected messages, timestamps, and their identifications. Finally, the authenticated content is transmitted to the PAD.

Algorithm 3 Authentication transmission from $BMS_j$to PAD

1. The jth BMS first selects an element $a_j \in Z_q^*$ and computes $A_j = a_j Q_{BMS}^j$, where $Q_{BMS}^j = H_1(ID_{BMS}^j)$
2. The jth BMS sets $K_j = e\left(a_j d_{BMS}^j, Q_{SD}\right)$, where $Q_{SD} = H_1(ID_{SD})$, and calculates the encryption key as $S_j^k = H_2\left(ID_{SD}, K_j, S_j^{K-1}\right)$, where S_j^{K-1} is the previous encryption key.
3. For PHI elimination $m_{i,j} \in \{0,1\}^{\lambda/3}$, time stamp $t_{i,j} \in \{0,1\}^{\lambda/3}$, and BMS identity $ID_{BMS_j} \in \{0,1\}^{\lambda/3}$. This BMS calculates $M_{i,j} = \left(m_{i,j} \| t_{i,j} \| ID_{BMS}^j\right) \oplus S_j^k$, where $1 \le i \le m$ and $1 \le j \le n$.
4. For $1 \le i \le m$, send $\left(A_j, \{M_{i,j}\}_{i=1}^m\right)$ to the PAD for storage

5.4 Data communication from PAD to MCS

By collecting m authentication data $M_{1,j}, M_{2,j}, \ldots, M_{m,j}$, from BMS_j, the aggregate signature σ_j is generated. Because data $M_{i,j}$ are encrypted by jth BMS, the PAD cannot know the actual PHI. Finally, the PAD uploads the reaggregated label and ciphertext to the MCS for storage.

Algorithm 4 Secure transmission from PAD to MCS

1. Collecting encrypted data $\left(A_j, \{M_{i,j}\}_{i=1}^{m}\right)$.
2. PAD selects the element $s \in Z_q^*$ and computes the signatures $A_{PAD} = sQ_{PAD}$, $B_{PAD} = sP$, $C_{PAD} = sQ_{SD}$, and $F_{PAD} = sH_2\left(M_{1,j}, M_{2,j}, \cdots, M_{m,j}\right)(Y + d_{PAD})$, where $Q_{PAD} = H_1(ID_{PAD})$.
3. To publicly verify the integrity of the data that exists on the cloud, the integrity label generation algorithm is run on the data blocks $\sigma_j = s\left(H_1\left(ID_{PAD} || ID_{SD} || id_j\right) + \sum_{i=1}^{m} M_{i,j}\, u\right) + d_{PAD}$ Id is the unique id_j of $\left(\mathrm{M}_{1,\mathrm{j}}, \mathrm{M}_{2,\mathrm{j}}, \cdots, \mathrm{M}_{\mathrm{m,j}}\right)$.
4. The $\left(id_j, \{M_{i,j}\}_{i=1}^{m}, \sigma_j, A_j, A_{PAD}, B_{PAD}, C_{PAD}, F_{PAD}\right)$ is transferred to the MCS for storage.

5.5 Public verification

Algorithm 6 defines public verifiability: The data need not be downloaded from the MCS but still can verify the integrity of PHI. Therefore, PAD, SD, or other PVs can perform verification tasks. Furthermore, public validation is performed in the classical challenger response model. The verifier informally sends the challenge message *chal* to the MCS to compute and return the proof generated from the stored data and challenge message. Eventually, the validator checks whether Γ is valid. The PHI data are complete if the Γ is valid. Otherwise, the data integrity of PHI is destroyed.

Algorithm 5 Public validation

1. After receiving the verification query, the public verifier obtains the A_{PAD} of the PAD on the MCS
2. The value (k_1, k_2) is randomly selected, where $k_1, k_2 \in Z_q^*$ are the two random seeds of PRP and PRF, respectively. The number of challenge blocks $l \in [1, n]$ is then selected.
3. Send the challenge message $chal = (l, k_1, k_2)$ to the MCS.
4. MCS uses $a_\tau = \pi(k_1, \tau)$ to compute a random index of all challenge blocks and $b_\tau = \varphi(k_2, \tau)$ to obtain random parameters. Compute the aggregated proof $\Gamma = (\{M_i\}_{i=1}^m, T)$ to the verifier, where $M_i := \sum_{\tau=1}^{l} b_\tau M_{i,a_\tau}, T := \sum_{\tau=1}^{l} (b_\tau \sigma_{a_\tau}) \in \mathbb{G}_1$.
5. The verifier uses $a_\tau = \pi(k_1, \tau)$ to challenge the block index and $b_\tau = \varphi(k_2, \tau)$ to calculate random parameters. Check if the equation is true, $e(T, P) = e\left(\sum_{\tau=1}^{l} b_\tau H_1(ID_{PAD} \parallel ID_{SD} \parallel id_j), B_{PAD}\right) \cdot e(\sum_{i=1}^{m} M_i u, B_{PAD}) \cdot e(\sum_{\tau=1}^{l} b_\tau Q_{PAD}, Y)$, and return 1 if it is true; otherwise, return 0.

The correctness of step 5 in Algorithm 4 can be proven as follows.

$$
\begin{aligned}
e(T,P) &= e\left(\sum_{\tau=1}^{l}(b_\tau \sigma_{a_\tau}), P\right) = e\left(\sum_{\tau=1}^{l} b_\tau \left(s\left(H_1(ID_{\mathrm{PAD}} \parallel ID_{\mathrm{SD}} \parallel id_{a_\tau}) + \sum_{i=1}^{m} M_{i,a_\tau} u\right) + d_{\mathrm{PAD}}\right), P\right) \\
&= e\left(\sum_{\tau=1}^{l} s\left(b_\tau H_1(ID_{\mathrm{PAD}} \parallel ID_{\mathrm{SD}} \parallel id_{a_\tau}) + \sum_{i=1}^{m} b_\tau M_{i,a_\tau} u\right) + b_\tau d_{\mathrm{PAD}}, P\right) \\
&= e\left(\sum_{\tau=1}^{l} b_\tau H_1(ID_{\mathrm{PAD}} \parallel ID_{\mathrm{SD}} \parallel id_{a_\tau}), B_{\mathrm{PAD}}\right) \cdot e\left(\sum_{\tau=1}^{l}\sum_{i=1}^{m} b_\tau M_{i,a_\tau} u, B_{\mathrm{PAD}}\right) \cdot e\left(\sum_{\tau=1}^{l} b_\tau Q_{\mathrm{PAD}}, Y\right) \\
&= e\left(\sum_{\tau=1}^{l} b_\tau H_1(ID_{\mathrm{PAD}} \parallel ID_{\mathrm{SD}} \parallel id_j), B_{\mathrm{PAD}}\right) \cdot e\left(\sum_{i=1}^{m} M_i u, B_{\mathrm{PAD}}\right) \cdot e\left(\sum_{\tau=1}^{l} b_\tau Q_{\mathrm{PAD}}, Y\right)
\end{aligned}
\tag{16}
$$

5.6 Decryption of PHI data by SD

To assist in the diagnosis of patients, SD first downloads authenticated PHI data and decrypts it to obtain real information about the patient's status. Algorithm 5 provides a detailed description.

Algorithm 6 Decryption of original PHI data

1. SD first downloads the stored file $\left(id_j, \{M_{i,j}\}_{i=1}^{m}, \sigma_j, A_j, A_{PAD}, B_{PAD}, C_{PAD}, F_{PAD}\right)$ from MCS
2. Check if the following three equations are true, $e(C_{PAD}, Y) = e(d_{SD}, B_{PAD})$, $e(A_{PAD}, P)e(Q_{PAD}, B_{PAD})$, and $e(F_{PAD}, Q_{SD}) = e(A_{PAD} + B_{PAD}, H_2(M_1, M_2, \ldots, M_m)d_{SD})$.
3. If not, then the SD knows that the packet is not a PAD upload and stops the process. Otherwise, continue with the following validation.
4. Set $K_j' = e\left(A_j, d_{SD}\right)$ and calculate key as $S_j^k = H_2\left(ID_{SD}, K_j', S_j^{K-1}\right)$, where S_j^{K-1} is the previous encryption key.
5. Decryption PHI by $\left(m_{i,j} \| t_{i,j} \| ID_{BMS}^j\right) = M_{i,j} \oplus S_j^k$

5.7 Data communication from the SD to BMS

If SD suspects the integrity of PHI, Algorithm 6 is run to verify it. SD diagnoses the patient after evaluating the true decrypted PHI data and sends the prescription data back to the j-th BMS through identity verification. The reverse process of data transmission from the BMS to the SD can be performed. Specifically, the SD calculates key S_j^k (by using $K_j^{'}$) and encrypts the prescription. The signature is then generated according to the EF-IDS and stored in the MCS. The PAD can download the encrypted prescription and its label (from the MCS), verify the validity of the signature, and send BMS_j back. The original prescription can eventually be recovered by decrypting the ciphertext through the key S_j^k, and commands can be executed according to the SD's recommendations.

6 Security analysis

We analyze the security concerning our IoMT-based health care system.

6.1 Privacy

Theorem 2 *Assume that hash functions H_1 and H_2 are random oracles. If it is difficult to solve the BDH assumption, the aforementioned health care system is probably safe under the indistinguishability under the chosen-ciphertext attack* (*IND-CCA*) *security model.*

Proof Assuming an adversary $\mathcal{A}$ that can crack the encryption scheme by using (t, q_k, q_d, ϵ) under the IND-CCA security model, a simulator $\mathcal{B}$ was constructed to solve the BDH problem. Given a problem instance (P, aP, bP, cP) with a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$, $\mathcal{B}$ controls the random predictor machine and simulates the environment for the PPT adversary $\mathcal{A}$. Then, $\mathcal{B}$ finishes the following step.

Setup $\mathcal{B}$ Set up $Y = cP \in \mathbb{G}_1$ and send $(e, \mathbb{G}_1, \mathbb{G}_2, P, Y)$ to $\mathcal{A}$. Assume that $\mathcal{A}$ has conducted q_i queries to $H_i(i = 1, 2) - Oracle$ at most, q_k private key queries, q_{en} encryption queries, and q_{de} decryption queries. $\mathcal{B}$ randomly selects two numbers $S, R \in \{1, 2, \ldots, q_1\}$ as B's guess randomly on the identity of the final sender and receiver, where S is the sender and R is the receiver.

Stage 1 $\mathcal{A}$ Ask the following query.

1. $\boldsymbol{H_1 - Oracle}$ **Query**:$\mathcal{A}$ requests H_1 query onID_i,$\mathcal{B}$ randomly selects$x_i \in \mathbb{Z}_q$, where$i \in \{1, 2, \ldots, q_1\}$. It is checked whether it is the Sth or Rth query. If not,$Q_i = H_1(ID_i) = x_i P$. Otherwise,$\mathcal{B}$ output $Q_S = H_1(ID_S) = x_S aP$ and $Q_R = H_1(ID_R) = x_R bP$ to $\mathcal{A}$.
2. $\boldsymbol{H_2 - Oracle}$ **Query:** The answer to the $H_2 - Oracle$ query is found only through delayed sampling.
3. **KeyGen query:** For the query ID_i, $\mathcal{B}$ performs an H_1 query on it. If it is not the Sth H_1 query, $\mathcal{B}$ returns $d_i = x_i cP$; otherwise, $\perp$ is returned.
4. **Decryption query:**$\mathcal{A}$ asks (ID_i, CT_i) for the decryption result, let $CT_i = (A_i, M_i)$. Only by $ID_i \neq ID_S$, the simulator $\mathcal{B}$ generate a corresponding private key to decrypt the information; otherwise, $ID_i = ID_S$. Thus, simulator continues decryption only if the decryption inquiry can pass verification.

Challenge $\mathcal{A}$ outputs two messages of equal length $m_0, m_1 \in \{0, 1\}^n$ and sends identity ID_i^* and receiving identityID_j^*. In the hash list of H_1,ID_i^* corresponds to$(i^*, ID_i^*, x_{i^*}, H_1(ID_i^*))$, and ID_j^* corresponds to$\left(j^*, ID_j^*, x_{j^*}, H_1\left(ID_j^*\right)\right)$. If ID_i^* and ID_j^* are not ID_S andID_R, the process is aborted; otherwise,$i^* = S, H_1(ID_i^*) = x_S aP$,$j^* = R$ and $H_1\left(ID_j^*\right) = x_R bP$. Here,$\mathcal{B}$ guesses a bit$b \in \{0, 1\}$, and the challenge ciphertext is calculated as$CT^* = (A^*, M^*)$. Select $s \in Z_q^*$ and set$A^* = sx_S aP$, where A^* from the problem case. Challenge ciphertext is the function of random number s on messagem_b. If $H_2(ID_R || e(P, P)^{sx_S x_R \text{abc}}) = M^* \oplus m_b$, then$CT^* = (sx_S aP, H_2(ID_R || e(P, P)^{sx_S x_R \text{abc}}) \oplus m_b)$.

Therefore, if the random oracle machine H_2 has never been used by inquired $ID_R || e(P, P)^{sx_S x_R \text{abc}}$, from the perspective of adversary $\mathcal{A}$, the challenge ciphertext is the correct ciphertext.

Stage 2 The same as phase 1, but this phase does not allow the ID_S and ID_R to interrogate the private key.

Guess $\mathcal{A}$ output a guessed result $b' \in \{0, 1\}$. If $b' = b$, the adversary $\mathcal{A}$ wins this game.

Probabilistic analysis If the sending identity ID_i^* and receiving identity ID_j^* of the challenge are the i^*th and j^*th identities asked to the random predictor, the adversary cannot ask for its private key. Only then can the interrogation and challenge phases be simulated. Because $q_{H_1} th\ H_1$ inquiries exist in the simulation process, the success probability is $2/q_{H_1}$. Suppose the adversary asks the random predictor for $e(P,P)^{\text{abc}}$ with probability ε, and the simulator calculates $e(P,P)^{\text{abc}}$ with probability ε/q_{H_2}.

Therefore, simulator $\mathcal{B}$ becomes $2\varepsilon/q_{H_1}q_{H_2}$ which benefits solving the aforementioned BDH problem.

6.2 Integrity of PHI on MCS

Theorem 3. *In the IoMT-based health care system, generating an effective forgery in the calculation is difficult if the MCS loses data blocks stored by users. Specifically, if the CDH assumption holds in* $\mathbb{G}_1$*, the CDH assumption is solved by an effective forgery.*

Proof. Suppose *MCS* is a malicious cloud server that outputs valid forged files on corrupted data. We use *MCS* as a subroutine and construct algorithm $\mathcal{B}$ that attacks the CDH assumption.

Specifically, given tuples $(P, aP, bP) \in \mathbb{G}_1^3$,$\mathcal{B}$ give *MCS* simulation environment. Here, B defines a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ and randomly selects $u \in \mathbb{G}_1$. Next, set $Y = aP \in \mathbb{G}_1$. B simulates and models hash functions H_1 and H_2 as random oracles. Next, $\mathcal{B}$ sends $\left(e, \mathbb{G}_1, \mathbb{G}_2, P, u, Y\right)$ to *MCS*. Assume that *MCS* makes p query to $H_1 - Oracle$. Here, $\mathcal{B}$ randomly selects $\tau^* \in [q_1]$ as the number for $\mathcal{B}$ to guess the final identity.

1. $\boldsymbol{H_1 - Oracle}$ **Query**: For the identity ID of the query, $\mathcal{B}$ randomly selects $r \in \mathbb{Z}_q$ and checks whether it is the τ^* query. If not, $H_1(ID) = rP$; otherwise, $H_1(ID) = brP$.
2. **Label generation query:***MCS* submits tuples $\left(\{M_i\}_{i=1}^m, ID_i, id_j\right)$ to this*Oracle*. Then, for query ID_j from*MCS*, $\mathcal{B}$ checks whether ID_i is an $\tau^* H_1$ query. If not, $\mathcal{B}$ selects $s \in Z_q^*$ randomly, sets $B = sP$ and$\sigma = s\left(H_1\left(ID_i||id_j\right) + \sum_{i=1}^m M_i u\right) + arP$, and returns $\langle ID_i, id_j, \{M_i\}_{i=1}^m, B, \sigma\rangle$ to MCS. Otherwise, return$\perp$.

Forgery Finally, the MSC responds to the forged message signature pair $\langle ID_i^*, id_j, \{M_i\}_{i=1}^m, B^*, \sigma^*\rangle$, which satisfies $e(\sigma^*, P) = e\left(H_1\left(ID_i^*||id_j\right), B^*\right) \cdot e\left(\sum_{i=1}^m M_i u, B^*\right) \cdot e\left(H_1(ID), Y\right)$. If the identity ID_i^* is not τ^*queries, it aborts; otherwise, $B^* = sP, \sigma^* = s\left(H_1\left(ID_i^*||id_j\right) + \sum_{i=1}^m M_i u\right) + abrP$.$\mathcal{B}$ calculates $W = r^{-1}\left(\sigma^* - s\left(H_1\left(ID_i^*||id_j\right) + \sum_{i=1}^m M_i u\right)\right)$, where $W = r^{-1}\left(\sigma^* - s\left(H_1\left(ID_i^*||id_j\right) + \sum_{i=1}^m M_i u\right)\right) = r^{-1}(abrP) = abP$.

Therefore, $W = abP$ is used to solve the *CDH* assumption.

If simulator $\mathcal{B}$ successfully guessed τ^*, the label generation query of the tuple $\left(\{M_i\}_{i=1}^m, ID_i, id_j\right) = \left(\{M_i\}_{i=1}^m, ID_{\tau^*}, id_j\right)$ is simulable, and the forged label is reducible because the simulator cannot select the message $\{M_i\}_{i=1}^m and \{M_i\}_{i=1}^m$ for the label generation query to be used for the label generation query. Therefore, for q_1 queries, the probability of successful simulation and useful attack is $1/q_1$. Assume the adversary *MCS* cracks the tag generation scheme with $\left(t, q_t, \varepsilon\right)$ after executing q_1 $H_1 - Oracle$ queries. The advantages of solving the CDH problem are ε/q_1. Ensure T_S represents the time cost of the simulation, and we have $T_S = O\left(q_1 + q_t\right)$, and $\mathcal{B}$ is $\left(t + T_S, \varepsilon/q_1\right)$ the advantages of solving the CDH problem.

The error detection probability of the MCS is critical because in the proposed protocol, the random sampling technique is used to detect the damage of the power factor.

Theorem 4. *Suppose a total of n blocks are stored on the MCS, of which* $p(p \le n)$ *blocks have been tampered with. For questioning information* $chal = \left(l, k_1, k_2\right)$, *randomly selected l different blocks are used to generate the integrity proof. Without loss of generality, assume that* $l_1\left(l_1 \le p\right)$ *blocks are selected. Let* P_a *represent the probability of false detection. Then,* $P_a \ge 1 - \left(\frac{n-p}{n}\right)^l$ *is obtained.*

Proof. According to the definition of P_a, we have the following:

$$\begin{aligned} P_a &= \Pr\left\{l_1 \ge 1\right\} = 1 - \Pr\left\{l_1 = 0\right\} \\ &= 1 - \frac{n-p}{n} \cdot \frac{n-p-1}{n-1} \cdots \frac{n-p-(l-1)}{n-l+1} \\ &\ge 1 - \left(\frac{n-p}{n}\right)^l. \end{aligned} \tag{17}$$

The more challenging blocks are, the higher the probability of false detection is. If 1% of the blocks are tampered with, the challenge of 300 blocks can be obtained as $P_a \ge 95\%$. Challenge 460 blocks to obtain $P_a \ge 99\%$. If 5% of the blocks are tampered with, the challenge of 90 blocks can be obtained as $P_a \ge 99\%$. Therefore, the error detection rate of the proposed scheme is high.

6.3 Other properties

Eavesdropping In Theorem 2, we show that our solution is safe in IND-CCA. To intercept the original PHI from the encrypted data, the adversary requires the private key of the SD or BMS, which generates the NM-based master key and the KPS key. Theorem 2 reveals that the master key and secret generation are equivalent for solving the BDH assumption. Therefore, an entity without authentication cannot obtain the original message.

Identity authentication In the registration process of the proposed system, NM authenticates and registers every entity and obtains the private key. In the same system, two entities can communicate with each other if they have registered with the NM before.

Forward and reverse confidentiality If the private key of the BMS and the $k-1$ session keys for each instance S_j^{k-1} are leaked, the random element a_j contained in the session key S_j^k if the kth instance ensurestheconfidentialityofthesessionkey.

Public verifiability In Theorems 3 and 4, we proved that MCS cannot pass the integrity audit of the data stored in the cloud by the verifier in the case of data loss or tampering. In addition, because the verifier can perform the audit without using any private key, the audit task can be completed by anyone.

7 Performance analysis

In this section, we evaluate the performance of the proposed intelligent medical system from computing and communication cost perspectives. The focus is on computing data communication and the energy consumption used by the BMS side during computing because BMS is a resource-limited device. To highlight its efficiency, we compared four other related systems or schemes including Kumar et al.'s system [15, 16].

7.1 Communication overhead

First, in our system, communication contents include partial private keys D_{E0} and D_{Ei} and returns by entities from NM and KPSs, ciphertext from BMS_j to PAD, ciphertext from PAD to MCS, signature and tag pairs, and the encrypted PHI data of SD downloaded from MCS are required. Let us analyze the parts of communication cost.

Specifically, suppose we have one NM and ρ KPS in our system. First, the entity registers with NM and each KPS to obtain the private key. The registration process occurs only once, so the communication overhead of generating the private key for each entity is negligible.

The j TH BMS sends encrypted text $M_{i,j} \in \{0,1\}^{\lambda}$ and $A_j \in \mathbb{G}_1$ to the PAD such that the total communication overhead from the j TH BMS to the PAD is equal to $m\lambda + |\mathbb{G}_1|$ for m messages.

PAD calculation identification $id_j \in \{0,1\}^{\lambda}$, sign $A_{\text{PAD}} \in \mathbb{G}_1, B_{\text{PAD}} \in \mathbb{G}_1$, $C_{\text{PAD}} \in \mathbb{G}_1, F_{\text{PAD}} \in \mathbb{G}_1$ and integrity label $\sigma_j \in \mathbb{G}_1$ Therefore, the total communication overhead from PAD to MCS is $(m+1)\lambda + 6|\mathbb{G}_1|$. The communication overhead of SD download from MCS is the same as that of PAD upload to MCS, which is $(m+1)\lambda + 6|\mathbb{G}_1|$.

Table 2 Comparison of communication overhead

System	BMS- > PAD	PAD- > MCS	MCS- > SD
Kumar et al.'s system [15]	$(m+3)\lvert\mathbb{G}_1\rvert + m\lambda + \lvert\mathbb{Z}_q\rvert$	$(m+4)\lvert\mathbb{G}_1\rvert + m\lambda + 2\lvert\mathbb{Z}_q\rvert$	$(m+4)\lvert\mathbb{G}_1\rvert + m\lambda + 2\lvert\mathbb{Z}_q\rvert$
Kumar et al.'s system [16]	$(m+4)\lvert\mathbb{G}_1\rvert + m\lambda$	$(m+6)\lvert\mathbb{G}_1\rvert + m\lambda$	$(m+6)\lvert\mathbb{G}_1\rvert + m\lambda$
Our proposed system	$m\lambda + \lvert\mathbb{G}_1\rvert$	$(m+1)\lambda + 6\lvert\mathbb{G}_1\rvert$	$(m+1)\lambda + 6\lvert\mathbb{G}_1\rvert$

BMS, biomedical sensor; PAD, personal auxiliary device; and MCS, medical cloud server

Similarly, we evaluate the communication overheads of the corresponding processes in [15, 16] and list them in Table 2, where KeyGen-E represents the communication overheads generated by the key of entity E, and A- > B represents slave A communication overheads to B.

7.2 Computation costs

For simplicity, let T_p, T_{pm}, T_{pa}, and T_H denote the execution times of one pairing, one dot product, one dot addition, and one hash to point on group $\mathbb{G}_1$, respectively, with negligible computational cost for the other operations. Because the registration phase occurs only once, the computational cost is negligible.

In our system, after collecting PHI and timestamp, BMS should be $T_p + T_{pm} + 3T_H$ to perform encryption calculation.

The PAD collects the data $M_{i,j} \in \{0,1\}^{\lambda}$ and $A_j \in \mathbb{G}_1$ from the BMS_j, and the PAD calculates the identity id_j, signature $A_{\text{PAD}}, B_{\text{PAD}}, C_{\text{PAD}}, F_{\text{PAD}}$, and integrity label σ_j. Therefore, the computational cost should equal $(5+m)T_{pm} + (m+2)T_{pa} + 4T_H$.

Finally, SD downloads file $\left(id_j, \{M_{i,j}\}_{i=1}^{m}, \sigma_j, A_j, A_{\text{PAD}}, B_{\text{PAD}}, C_{\text{PAD}}, F_{\text{PAD}}\right)$ from the MCS and decrypts it. This action cost $3T_p + T_{pm} + T_{pa} + 3T_H$.

Similarly, we evaluated the computational costs of the corresponding processes in [15, 16] and listed them in Table 3.

7.3 Experimental comparisons

This section assesses the performance of the intelligent medical system from a computational cost perspective. We compared the corresponding processes of BMS encryption, PAD signature, and SD decryption with other systems or schemes to show our efficiency advantage. The focus is to calculate the computational cost used by the BMS side during data communication and calculation because compared with the machine in the medical institution, the BMS side is a resource-constrained device.

Table 3 Comparison of computational cost

System	BMS-Enc	PAD-Sign	SD-Dec
Kumar et al.'s system [15]	$T_p + (m+4)T_{pm} + \mathrm{m}T_{pa} + (\mathrm{m}+4)T_H$	$T_{pm} + T_H$	$7T_p + mT_{pm} + \mathrm{m}T_{pa} + (\mathrm{m}+2)T_H$
Kumar et al.'s system [16]	$T_p + (m+5)T_{pm} + \mathrm{m}T_{pa} + (\mathrm{m}+4)T_H$	$2T_p + 2T_{pm} + T_H$	$7T_p + mT_{pm} + \mathrm{m}T_{pa} + (\mathrm{m}+2)T_H$
Our proposed system	$T_p + T_{pm} + 3T_H$	$(6+m)T_{pm}$ $(m+2)T_{pa} + 4T_H$	$7T_p + T_{pm} + T_{pa} + 3T_H$

BMS-Enc, PAD-Sign, and SD-Dec represent the time consumption of BMS encryption, PAD signature, and SD decryption operations

BMS, biomedical sensor; and PAD, personal auxiliary device

We conducted experiments on a laptop with an Intel Core i5-8300U CPU at 2.3 GHz and 16 GB RAM. In our implementation, a hypersingular curve $y^2 + y = x^3 + x$ with embedding degree 4 was used and $\eta : E(F_{2^{271}}) \times E(F_{2^{271}}) \to E(F_{2^{4.271}})$ is paired with eta, and the PBC library was used to perform calculations.

Specifically, to better illustrate the advantages of our proposed scheme, we choose the KC system [15] and the improved KC system [16] for comparison and then conduct the following experiments. The experiment is divided into three parts.

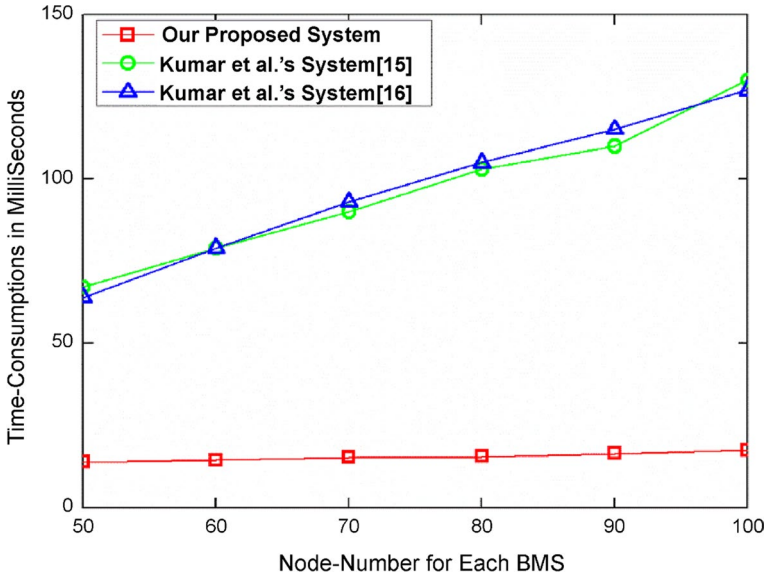

Fig. 2 Time consumption of BMS encryption. BMS, biomedical sensor

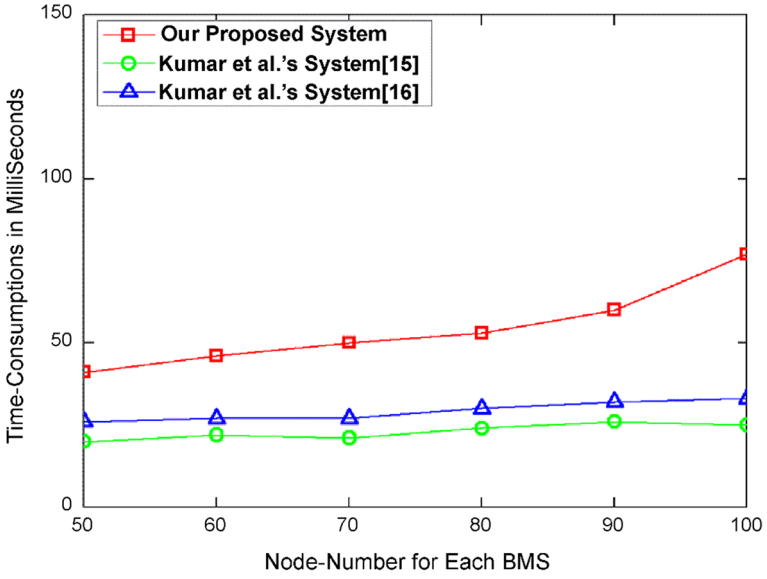

Fig. 3 Time consumption of PAD signing. BMS, biomedical sensor; PAD, personal auxiliary device

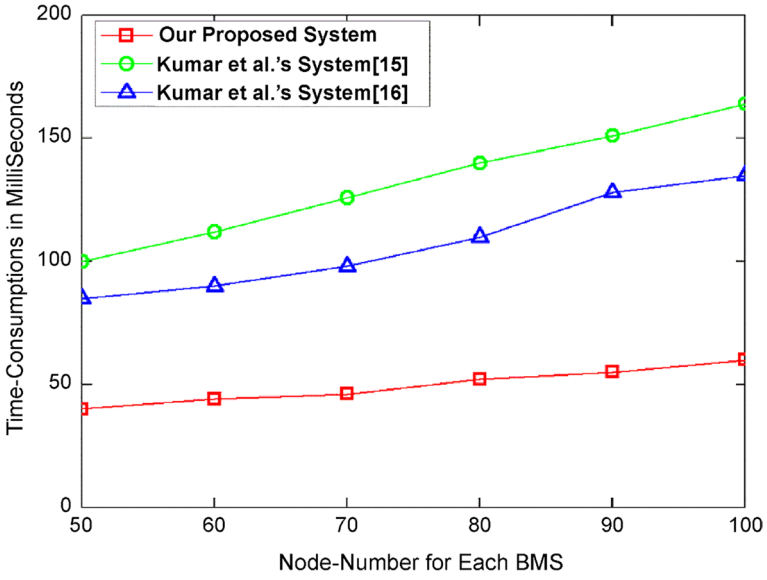

Fig. 4 Time consumption for SD decryption. BMS, biomedical sensor

The first part compares the time cost of different data blocks encrypted by BMS, and the second and third parts compare the time cost of signature and decryption in PAD and SD based on different data blocks encrypted by BMS.

Now, we choose to increase the encrypted data block m per transmission BMS from 50 to 100 blocks in increments of 10. Then, in Fig. 2, we plot the change in the time cost of encrypting blocks of data. It is easy to see that the cost of encryption algorithms running on BMS is significantly less than the KC system and the improved KC system. In our system, it takes approximately 17.5 ms to encrypt 100 blocks of data, which is a significant advantage. In addition, the key distribution process in the system only takes place once, which has little impact on the performance of each entity. In addition, in the second part of the experiment, although the time cost of the PAD signature is larger than the corresponding process of the KC system and the improved KC system, in our system, the PAD is a device with relatively powerful battery resources and computing power, so a slight increase in the time cost is acceptable for the PAD, as shown in Fig. 3. In the third part, the time consumption of SD verification and decryption to obtain PHI also has a significant advantage over the KC system and the improved KC system, as shown in Fig. 4.

Through the analysis of the experimental results, we can see that our proposed intelligent medical system has great advantages in running time, especially for BMS. Therefore, the system is more suitable for intelligent healthcare based on IoMT.

8 Conclusion and future work

8.1 Conclusion

The patient may not have the ability to call for help in an emergency situation. Therefore, if the medical sensor detects obvious abnormal health information that requires immediate rescue, it should be equipped with an alarm or a means of calling for help. We proposed a cloud-centric IoMT-based intelligent medical system that is based on the EF-IDS and ensures the privacy of users' PHI. The proposed method includes mutual authentication and public verification of data integrity. Finally, experimental results demonstrate that our system is more efficient than Kumar et al.'s, especially for resource-constrained BMSs.

8.2 Future work

The first research direction for the future is to explore the application of our proposed technology in other scenarios, such as intelligent transportation and smart cities. Second, we aim to further optimize the key distribution process, which is crucial for practical implementation of intelligent medicine due to its complexity. Finally, we strive to enhance the timeliness of our medical Internet of Things (IoT) system. The patient may lack the capacity to request assistance in an emergency situation. Therefore, a medical sensor detecting significant deviations from normal health parameters should be equipped with either an alarm or a summoning aid.

Acknowledgements No funds, grants, or other support were received.

Authors' contributions TM contributed to conceptualization, formal analysis, and writing—original draft preparation; TM and QR were involved in methodology and writing—review and editing; QR contributed to investigation; BS and GB were involved in resources; JS contributed to data curation; and BS and JS were involved in supervision. All authors have read and agreed to the published version of the manuscript.

Funding Not applicable.

Availability of data and material The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Code availability Not applicable.

Declarations

Conflicts of interest The authors declare that they have no conflicts of interest.

Ethical approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

References

1. Laghari AA, Wu K, Laghari RA, Ali M, Khan AA (2021) A review and state of art of Internet of Things (IoT). Arch Comput Methods Eng 29:1395–1413
2. He D, Zeadally S, Wu L (2015) Certificateless public auditing scheme for cloud-assisted wireless body area networks. IEEE Syst J 12(7):64–73
3. Xiong H, Qin Z (2015) Revocable and scalable certificateless remote authentication protocol with anonymity for wireless body area networks. IEEE Trans Inf Forensics Secur 10(7):1442–1455
4. Sureshkumar V, Amin R, Vijaykumar VR, Sekar SR (2019) Robust secure communication protocol for smart healthcare system with FPGA implementation. Futur Gener Comput Syst 100:938–951
5. Wang H, Feng L, Ji Y, Shao B, Xue R (2021) Toward usable cloud storage auditing, revisited. IEEE Syst J 16(1):693–700
6. Ming Y, Zhang T (2018) Efficient privacy-preserving access control scheme in electronic health records system. Sensors 18(10):3520
7. Sun Y, Lo FP-W, Lo B (2019) Security and privacy for the internet of medical things enabled healthcare systems: a survey. IEEE Access 7:183339–183355. https://doi.org/10.1109/ACCESS.2019.2960617
8. Akinyele JA, Pagano MW, Green MD, Lehmann CU, Peterson ZN, Rubin AD (2011) Securing electronic medical records using attribute-based encryption on mobile devices. In: Proceedings of the 1st ACM workshop on security and privacy in smartphones and mobile devices, pp 75–86.
9. Chandrasekaran B, Balakrishnan R, Nogami Y (2018) Secure data communication using file hierarchy attribute based encryption in wireless body area networks. JCOMSS 14(1):75–81. https://doi.org/10.24138/jcomss.v14i1.446
10. Li F et al (2012) Identity-based online/offline signcryption for low power devices. J Netw Comput Appl 35(1):340–347. https://doi.org/10.1016/j.jnca.2011.08.001
11. Omala AA, Robert N, Li F (2016) A provably secure transmission scheme for wireless body area networks. J Med Syst 40(11):247. https://doi.org/10.1007/s10916-016-0615-1

12. Zhang A et al (2017) Light-weight and robust security-aware D2D-assist data transmission protocol for mobile-health systems. IEEE Trans Inform Forensic Secur Trans IEEE (Trans) 12(3):662–675. https://doi.org/10.1109/TIFS.2016.2631950
13. Zhou C (2018) Comments on "light-weight and robust security-aware D2D-assist data transmission protocol for mobile-health systems." IEEE Trans Inform Forensic Secur Trans IEEE (Trans) 13(7):1869–1870. https://doi.org/10.1109/TIFS.2018.2799582
14. Kumar M, Chand S (2020) A secure and efficient cloud-centric internet-of-medical-things-enabled smart healthcare system with public verifiability. IEEE Internet Things J 7(10):10650–10659. https://doi.org/10.1109/JIOT.2020.3006523
15. Kumar M, Chand S (2021) A provable secure and light weight smart healthcare cyber-physical system with public verifiability. IEEE Syst J 16(4):5501–5508
16. Zhou C (2018) Comments on "light-weight and robust security-aware D2D-assist data transmission protocol for mobile-health systems." IEEE Trans Inf Forensics Secur 13(7):1869–1870
17. Liu Y, Yu J, Fan J, Vijayakumar P, Chang V (2021) Achieving privacy-preserving DSSE for intelligent IoT healthcare system. IEEE Trans Industr Inf 18(3):2010–2020
18. Wang R et al (2022) Privacy-preserving federated learning for Internet of medical things under edge computing. IEEE J Biomed Health Inform 27:854–865. https://doi.org/10.1109/JBHI.2022.3157725
19. Zhou T et al (2022) Human-in-the-loop-aided privacy-preserving scheme for smart healthcare. IEEE Trans Emerg Top Comput Intell 6(1):6–15. https://doi.org/10.1109/TETCI.2020.2993841
20. Xu R, Ren Q (2022) Cryptoanalysis on a cloud-centric Internet-of-medical-things-enabled smart healthcare system. IEEE Access 10:23618–23624. https://doi.org/10.1109/ACCESS.2022.3154466