Mathematical Biosciences and Engineering

MBE, 18(6): 7539–7560.
DOI: 10.3934/mbe.2021373
Received: 10 April 2021
Accepted: 12 August 2021
Published: 02 September 2021

http://www.aimspress.com/journal/MBE

Research article

A novel fault-tolerant privacy-preserving cloud-based data aggregation scheme for lightweight health data

Fawza A. Al-Zumia[1],*, Yuan Tian[2],* and Mznah Al-Rodhaan[1]

[1] Computer Science Department, King Saud University, Riyadh, Kingdom of Saudi Arabia
[2] Nanjing Institute of Technology, Nanjing, China

* **Correspondence:** Email: f.alzumia@gmail.com, ytian@njit.edu.cn.

Abstract: Mobile health networks (MHNWs) have facilitated instant medical health care and remote health monitoring for patients. Currently, a vast amount of health data needs to be quickly collected, processed and analyzed. The main barrier to doing so is the limited amount of the computational storage resources that are required for MHNWs. Therefore, health data must be outsourced to the cloud. Although the cloud has the benefits of powerful computation capabilities and intensive storage resources, security and privacy concerns exist. Therefore, our study examines how to collect and aggregate these health data securely and efficiently, with a focus on the theoretical importance and application potential of the aggregated data. In this work, we propose a novel design for a private and fault-tolerant cloud-based data aggregation scheme. Our design is based on a future ciphertext mechanism for improving the fault tolerance capabilities of MHNWs. Our scheme is privatized via differential privacy, which is achieved by encrypting noisy health data and enabling the cloud to obtain the results of only the noisy sum. Our scheme is efficient, reliable and secure and combines different approaches and algorithms to improve the security and efficiency of the system. Our proposed scheme is evaluated with an extensive simulation study, and the simulation results show that it is efficient and reliable. The computational cost of our scheme is significantly less than that of the related scheme. The aggregation error is minimized from $O(\sqrt{\mathbf{w+1}})$ in the related scheme to O(1) in our scheme.

Keywords: lightweight data; fault tolerance; privacy; data aggregation; cloud computing

1. Introduction

In the last few decades, technology has significantly dominated our lives and is currently considered the driving force of recent improvements in the medical health care area. Wireless sensor networks (WSNs) have demonstrated considerable importance due to their usage in many different aspects of human lives, such as medical health care, surveillance, environmental monitoring, military fields, and many other useful applications [1–3].

With the widespread use of smartphones, researchers have concentrated on the use of mobile technology for mobile medical health care, focusing on systems of medical data aggregation that are used to collect and send health data from patient smartphones directly to health care organizations.

With the exponential growth of health data, the processes of aggregating and analyzing vast amounts of data require immense storage capabilities, powerful computational resources, and fast and secure means of communication. Achieving these requirements by relying only on traditional WSNs is difficult and expensive for health care organizations [4].

Cloud-based solutions have proliferated in the medical health care field due to their extensive benefits [5]. These benefits include large-scale and on-demand storage, agility, cost-effectiveness and continuous service availability for information processing. Therefore, cloud-based solutions have considerable potential to enhance collaboration among the various participating entities in medical health care, such as patients and health care organizations.

Despite these benefits, cloud-based solutions are associated with elevated threat levels in terms of security and privacy. These threats include identity spoofing; data tampering; information disclosure; and violations of data integrity, confidentiality, authenticity, and accountability [6,7].

Mobile health networks (MHNWs) consist of small and inexpensive sensors that are deployed in unsupervised environments and are easily exposed to malfunctions and malicious attacks. Thus, fault tolerance is an important characteristic that must be considered when designing sensor network schemes [8].

In data aggregation schemes, sensor failures can cause the collection and transfer of incorrect data without a guarantee of excellence. Fault tolerance is defined as “the ability of the network to sustain its functionalities properly, even in the presence of failures in some of its nodes”. Fault tolerance aims to eliminate critical privacy threats and assure strong privacy protection for users who contribute their data to aggregators to ensure that the applied technology can deliver excellent service quality [9].

Two approaches can be employed to achieve fault tolerance. The first is a reactive approach, in which a system can recover from failures when they occur [10]. A small error can be recovered by a state-of-the-art protocol despite failures [11]. Unfortunately, these protocols can tolerate only partial failures and are not efficient in terms of bandwidth and delay. The second is a proactive approach that handles failures using multiple message exchanges between the nodes and the aggregator before faults occur. This approach substantially reduces the required recovery time, as the information needed for fault recovery is available.

Despite the fact that the state-of-the-art binary proactive protocol achieves a low delay, it suffers from communication overhead, bandwidth costs and large errors [10]. Won et al. [10] presented a novel design for a future ciphertext mechanism; this design supports differential privacy and achieves a higher bandwidth than the state-of-the-art binary proactive protocol. Chen et al. [12] presented a data aggregation scheme that preserves user privacy and guarantees data integrity by adopting a future ciphertext mechanism to provide fault tolerance capability.

Due to the confidential nature of health information and the importance of protecting and preserving the confidentiality of data, information security systems should be designed and developed with consideration of legal, ethical and security issues. Therefore, to design a workable data aggregation scheme for medical health care, the following issues must be addressed. The first issue is how to protect and preserve the security and privacy of data and maintain data confidentiality. The second issue is how to protect a system against failures.

Therefore, we propose a novel design for a fault-tolerant privacy-preserving data aggregation scheme. We use the cloud to aggregate, store and process data. Our contributions can be summarized as follows:

- The proposed architecture achieves a fault-tolerant privacy-preserving data aggregation scheme for lightweight health data with end-to-end verification. Moreover, when some failures occur, the cloud can compute the aggregation result, and health care institutions (HCs) can verify the correctness of the aggregated result.
- We modify the future ciphertext mechanism by adding a threshold for the number of faulty nodes. This modification avoids scenarios in which the cloud continues to compute meaningless aggregation when a serious abnormality occurs in the system.
- For secure aggregation and identity protection, we use homomorphic encryption, as it enables aggregation functions to be performed on encrypted data. We use random noise to achieve differential privacy.
- We provide a security and privacy analysis to show that our proposed scheme supports privacy preservation, fault tolerance, and data integrity verification. Additionally, we evaluate the efficiency, robustness and reliability of our scheme to confirm that it has good real-time performance and low aggregation error.

The remainder of the paper is organized as follows: Section 2 provides a background of health data aggregation and investigates the privacy and security challenges of data aggregation. Section 3 reviews related studies. Preliminaries and the proposed scheme are presented in Sections 4 and 5, respectively. The security analysis is provided in Section 6, followed by the performance analysis in Section 7. The conclusion of the study and future research ideas are discussed in Section 8.

2. Background

WSNs are formed by hundreds of thousands of sensor nodes that are used to measure and transmit physical or environmental changes, such as temperature and pressure or motion within a monitoring environment. Each sensor node consists of a sensing unit, memory, a processing unit, a power supply and a wireless communication unit [13]. The characteristics of WSNs include limited power, mobility, ability to cope with node failures, and low cost. These features have prompted researchers to introduce a new research area in the medical health care field: MHNWs. Recently, wearable devices and smartphones have been extensively applied in offering health monitoring services based on health data gathered from users. As these health data are very sensitive, any data leakage may violate user privacy [2]. In this section, we present an overview of the different uses of cloud computing in terms of data aggregation and address the security and privacy issues associated with MHNWs.

2.1. Security and privacy challenges

The usage of WSNs has rapidly increased over the past few years in various fields. A massive amount of data is being collected, transmitted and aggregated to perform processing operations. A large number of threats surround WSNs. Consequently, the design of a privacy-preserving data aggregation protocol should address these threats [13], which include the following:

a) Privacy Preservation and Eavesdropping: Eavesdropping is a type of attack in which the intruder tries to obtain confidential data by listening to transmissions over neighboring wireless links. Therefore, privacy preservation assures data privacy that may be threatened by trusted sensor nodes and adversaries. Some aggregation functions, such as min and max, can also be used to breach data privacy. Therefore, the designed protocol must maintain data privacy while using aggregation functions [13].

b) Data Integrity and Data Tampering: One of the most common types of attack on data privacy is data tampering, in which the attacker tries to manipulate (with an intermediate result) sensor data at the aggregator level during the data aggregation phase. This type of attack leads to an incorrect aggregation result and eventually to an incorrect decision [2,13].

c) Efficiency: In WSNs, it is very difficult to avoid communication overhead, but it can be greatly minimized by reducing communication costs, computational costs, and memory and payload sizes. In WSNs, data aggregation must fulfill both bandwidth and energy efficiency requirements throughout network processing.

d) Accuracy and Dynamism: In WSNs, energy constraints must be properly managed. The data generated from all sensing nodes are important. Therefore, all nodes should have sufficient power to process the collected data [3,13].

2.2. Essential requirements for privacy preservation in an e-health cloud

Certain rules and regulations are defined to ensure the privacy of the data within an organization and are called the CIA model (confidentiality, integrity and availability) [5]. Nevertheless, the data managed by third-party vendors require more privacy measures than those existing in the CIA model. Abbas and Khan [7] stated that there are many threats to privacy in the cloud, such as spoofing, masquerading, tampering, replaying and denial of service. The following requirements must be fulfilled to achieve data privacy preservation:

- Confidentiality: The health information of patients must be protected not only in the cloud environment but also from external anomalies and unauthorized users [7].
- Integrity: The data must be protected from illegitimate actions while ensuring that the data have not been altered or tampered with by either authorized or unauthorized users [5,7].
- Anonymity: Health data contain vital information, such as the patient's diseases and name, and this information must be hidden [14,15]. The patient's identity must be protected from intruders, unauthorized users and other internal or external adversaries. Anonymity can be achieved by using a technique known as pseudonymity [5,7].

- Nonrepudiation: These threats are posed by a user who performs tasks and denies them later. In the medical health care area, neither a patient nor a doctor can deny modifying data [7].

3. Related work

Data aggregation, as a powerful technique for MHNWs, has attracted substantial attention in both academia and industrial fields. Recently, many privacy-preserving data aggregation schemes have been presented. In [11], ACS and Castelluccia proposed a privacy-preserving data aggregation scheme that applies the differential privacy concept by adding Laplace noise to aggregated data. However, the scheme increases network bandwidth and delay. Subsequently, they extended their scheme to support partial fault tolerance.

Lu et al. [15] introduced an efficient privacy-preserving scheme that reduces the computational overhead and delays in the network with its features, thereby providing fewer calculations, less traffic, higher accuracy and verifiable completeness. Khan et al. [16] proposed a fault-tolerant privacy-preserving data aggregation scheme in a fog-enabled Boneh-Goh-Nissam (BGN) cryptosystem used to preserve privacy. This scheme also reduces communication and computation costs. Zhang et al. [17] presented a privacy-preserving data aggregation scheme for health data monitoring in which the health data were stored and processed in the cloud and various strategies were applied based on the prioritization of the dataset. This scheme reduces the communication overhead but is not tolerant of node failures.

Won et al. [10] introduced a novel design for future ciphertext buffering to tolerate malfunctioning smart meters and achieved both differential privacy and error optimization. Chen et al. [12] also adopted the future ciphertext buffering mechanism that was proposed in [10] and proposed an aggregation scheme that supports fault tolerance, privacy preservation and data integrity. In addition, confidentiality was guaranteed by using Diffie-Hellman cryptography, while integrity was achieved by attaching a homomorphic message authentication code (HMAC) to each message.

Han et al. [8] addressed the fault tolerance issue within the health data monitoring framework. They proposed a cloud-based data aggregation scheme that supports additive and nonadditive data aggregation. A BGN cryptosystem was used to protect user privacy. Differential privacy was achieved by using multiple cloud servers. The scheme also guarantees data integrity. Chen et al. [18] presented another multifunctional data aggregation schema (MUDA) that takes advantage of the homomorphic property of the BGN cryptosystem and a bilinear map to provide confidentiality to user data. The MUDA was also extended to support differential privacy [19,20]. Zhu et al. [21] proposed a secure data integrity verification scheme based on a short signature algorithm. They introduced the use of cloud computing to augment computing and storage resources.

Since health data aggregation requires very high computational capabilities, the privacy of sensitive information can be guaranteed if it is encrypted [22] by the owner. Homomorphic encryption enables the cloud to compute the result of aggregation without knowing the raw data. Another way to provide security for health data is through the use of cryptographic storage [3,8,17].

Moreover, verification is an extremely crucial step in health data aggregation, as any tampering with the data results in an invalid aggregation, and such interference must therefore be detected and rejected. The message authentication code (MAC) is a protocol that is commonly used to detect false data and to protect and guarantee the integrity of the data [12, 23–25]. Zhang et al. [25] and Chen et al. [12] took advantage of the homomorphic properties of the MAC to guarantee data integrity. The hash-based MAC was used by Chen et al. [12] and Zhuo et al. [26] to verify user data.

4. Preliminaries

This section reviews the relevant definitions and terminologies. These definitions are necessary to understand the remainder of this work. The basic notations and symbols are listed in Table 1.

Table 1. Descriptions of symbols that are frequently used.

Variables	Description
$\mathbf{G_1, G_2, G_T}$	Cyclic multiplicative group
$\boldsymbol{g_1}$	Generator of G_1
$\boldsymbol{g_2}$	Generator of G_2
$\boldsymbol{p, q}$	Prime numbers
$\boldsymbol{Z_q^*}$	Multiplicative group of Z_q
$\boldsymbol{e(g_1, g_2)}$	The generator of group G_T
$\boldsymbol{A}$	A random mechanism
$\boldsymbol{\epsilon}$	A parameter expressing the privacy cost
$\boldsymbol{S}$	Subset of $Range(A)$
$\boldsymbol{Range(A)}$	A domain of the output under mechanism A
$\boldsymbol{\lambda}$	A security parameter
$\boldsymbol{pk}$	Public key
$\boldsymbol{sk}$	Secret key
$\mathbf{x}$	Message
$\boldsymbol{c}$	Ciphertext
MAC	Homomorphic MAC function
$\boldsymbol{h}$	Secure hash function

4.1. Bilinear pairing

A bilinear pairing e is a map $e: G_1 \times G_2 \rightarrow G_T$, where G_1, G_2 and G_T are cyclic multiplicative groups of the same prime order q, g_1 is a generator of G_1, and g_2 is a generator of G_2. The pairing e has the following properties:

Bilinearity: $e(g_1^a, g_2^b) = e(g_1, g_2)\ \forall\ g_1 \in G_1, g_2 \in G_2$ and $a, b \in Z_q^*$.

Computability: $\forall\ g_1 \in G_1, g_2 \in G_2$, $e(g_1, g_2)$ can be computed by an efficient algorithm.

Nondegeneracy: $\forall\ g_1 \in G_1, g_2 \in G_2$, $e(g_1, g_2) \neq 1$.

4.2. Complexity assumptions

Definition 1. Discrete Logarithmic Problem [27] (DLP): Assume that G_1, G_2 are two cyclic multiplicative cyclic groups, G_1 is generated by g_1, and G_2 is generated by g_2. Suppose that

g_0, g_1 are two elements in G_1. It is computationally intractable to compute a such that

$$g_1 = g_0^a \tag{1}$$

Definition 2. Computational Diffie-Hellman Problem [28] (CDH): Assume that G_1, G_2 are two cyclic multiplicative cyclic groups, G_1 is generated by g_1, and G_2 is generated by g_2. Given $e(g_1, g_1^a, g_1^b)$ and $a, b \in Z_q^*$, it is intractable to derive g_1^{ab} from the given $e(g_1^a, g_1^b)$ in polynomial time.

Definition 3. Decisional Diffie-Hellman Problem [26] (DDH): Assume that G_1, G_2 are two cyclic multiplicative cyclic groups, G_1 is generated by g_1, and G_2 is generated by g_2. Given $e(g_1, g_1^a, g_1^b, g_1^c)$, where a, b, c $\in Z_q^*$, a DDH determines whether $c = ab \mod q$ by checking as follows:

$$e(\ g_1^a, g_1^b) \overset{?}{=} e(\ g_1^c, g) \tag{2}$$

Definition 4. Gap Diffie-Hellman [29] (GDH) Group: A group is Gap Diffie-Hellman if the computational Diffie-Hellman problem is hard but the Decisional Diffe-Hellman problem can be solved in a cyclic multiplicative group G_1, G_2.

4.3. Differential privacy

Definition 1. ($\epsilon-$ Differential Privacy) [30] A randomized mechanism A satisfies ϵ −differential privacy if for any two datasets $D_1$and D_2, where D_1 is obtained from D_2 by adding or removing a single element, and for all $S \subseteq \mathrm{Range}(A)$,

$$\Pr(A(D_1) \in S) \leq \ e^{\epsilon}. \Pr(A(D_2) \in S) \tag{3}$$

In the above definition, the parameter ϵ represents the privacy cost, which allows us to control the desired privacy level. A smaller value of ϵ denotes better privacy protection but implies that more noise is required and that the result will have lower accuracy. The most common mechanism for achieving ϵ-differential privacy is to add i.i.d Laplace noise sampled from the Laplace distribution to the aggregated result.

Definition 2. (2ϵ −Differential Privacy) [31] The noise $\mathrm{Lap}(\lambda)$ is sampled from the Laplace noise distribution with mean 0 and variance $2\lambda^2$. The probability density function of the distribution is given by

$$\mathrm{Lap}(\lambda) = \frac{1}{2\lambda} e^{|x|/\lambda} \tag{4}$$

In our scenario, each participant should generate random noise following a Laplace distribution. The Laplace distribution is infinity divisible, where each random variable is a summation of n other random variables as follows:

$$\mathrm{Lap}(\lambda) = \sum_{i=1}^{n} (G_i(n, \lambda) - G'_i(n, \lambda)) \tag{5}$$

where $G_i(n,\lambda)$ and $G_i{}'(n, \lambda)$ are gamma-distributed random variables with a gamma density given by

$$g(x, n, \lambda) = \frac{(1/\lambda)^{1/n}}{\Gamma(1/n)} x^{1/n-1} e^{-x/\lambda} \tag{6}$$

Additionally, $\Gamma(1/n)$ is the gamma function evaluated at $1/n$.

4.4. YASHE

Yet Another Somewhat Homomorphic Encryption (YASHE) is a scheme based on a modified version of n-th degree truncated polynomial ring units (NTRUs) and the multikey homomorphic encryption scheme [32]. It has become a trendy fully homomorphic encryption (FHE) scheme due to its superior performance with lightweight data compared with the performances of other homomorphic schemes [32,33].

The security of YASHE is based on the hardness of the decisional-ring learning with errors (RLWE) problem [34]: Given sample $a \leftarrow R_q$, error term $e \leftarrow \chi$, and a secret $s \leftarrow R_q$ where $a \leftarrow R_q$ is drawn uniformly at random, it is computationally hard for an adversary that does not know s and e to distinguish between the distribution of e $(sa + e, a)$ and that of (a, b) where $(b \leftarrow R_q)$.

YASHE. ParamGen(λ): Given a set of parameters λ, d, q, t, x_{key}, x_{err} and w, where λ is a security parameter, d is a fixed positive integer that determines R, and moduli q and t exist, with $1 < t < q$, x_{key} and x_{err} are distributions on R, and w is an integer base where $w > 1$. The algorithm generates $(d, q, t, \lambda, x_{key}, x_{err}, w)$.

YASHE. keyGen($d, q, t, \lambda, x_{key}, x_{err}, w$): $h, f \leftarrow x_{key}$ are computed; then, $f = [tf' + 1]_q$ and $h = [tgf']_q$ are set. $e, s \leftarrow x_{err}^{l_{w,q}}$ are sampled, and $\gamma = [P_{w,q}(f) + e + h.s]_q \in R^{l_w}$ is computed. Then, $(pk, sk, evk) = (h, f, \gamma)$ is generated.

YASHE. Encrypt (pk, x): $x \in R$ is encrypted, and ciphertext $c = [\Delta[x]_t + e + hs]_q \in R$ is generated.

YASHE. Decrypt (sk, c): A ciphertext c is decrypted by $x = \left[\left\lfloor \frac{t}{q} . [fc]_q \right\rceil\right]_t \in R$.

YASHE. Add (c_1, c_2): The ciphertext $c_{add} = [c_1 + c_2]_q$ is output.

4.5. Homomorphic MAC function

One of the basic methods for ensuring data integrity and preventing tampering attacks is to use a homomorphic MAC function. The homomorphic property means that for two messages x_1 and x_2, given two homomorphic MACs (MAC (x_1) and MAC (x_2)), anyone can compute MAC $(x_1 + x_2)$ without knowing x_1 or x_2. The MAC function can be constructed as follows:

$$\mathrm{MAC}\ (x_i) = g^x \mathrm{mod} q \tag{7}$$

where $x_i < q$. This MAC function satisfies the homomorphic property since it follows that

$$\mathrm{MAC}\,(x_1) \bmod q + \mathrm{MAC}\,(x_2) \bmod q = g^{x1+x2} \bmod q = \mathrm{MAC}\,(x_1 + x_2) \forall\, x_1, x_2 \in Z_q \quad (8)$$

4.6. Hash function

The cryptographic hash function is used to check the integrity and source of the given data. This function accepts an input of arbitrary length and maps it to a fixed length with a one-way, collision-resistant mapping. It is computationally infeasible to map two different input maps (a, b) to the same output such that $h(a) = h(b)$, where $a \neq b$. Additionally, it is impossible to infer a from $h(a)$ [35].

5. Proposed approach

Data aggregation is an important tool in MHNWs, in which a vast amount of sensitive data is transmitted, processed, and analyzed. Therefore, fault tolerance and privacy have become critical issues for health data aggregation. Without appropriate privacy protection, users may not be willing to share their data. Therefore, we introduce a fault-tolerant privacy-preserving data aggregation scheme for health data.

In our scheme, the computational overhead is reduced. Privacy is provided by the fully homomorphic YASHE in addition to embedded noise for differential privacy. Fault tolerance is achieved by applying the future message mechanism to properly sustain network operability even in the presence of failures. To enhance the efficiency of the proposed scheme, a health institution can control malfunctioning nodes. The basic notations and symbols of the scheme are listed in Table 2.

Table 2. Basic notations for our proposed scheme.

Variables	Description
$\boldsymbol{U_i}$	Participant
$\mathbf{sk_i}$	Private key for the participant
$\mathbf{pk_i}$	Public key for the participant
$\mathbf{sk_c}$	Private key for the health care institution
$\mathbf{pk_c}$	Public key for the health care institution
$\mathbf{x_{i,t}}$	Health data
$\hat{\mathbf{r}}$	Random noise
$\hat{\mathbf{x}}_{\mathbf{i,t}}$	Noisy data
$\mathbf{c_{i,t}}$	Ciphertext
$\boldsymbol{H}$	Secure hash function
$\boldsymbol{\sigma}_{\mathbf{i,t}}$	Signature generated by the secure hash function for participant U_i
$\hat{\mathbf{c}}_{\mathbf{i,t+B}}$	Future ciphertext for time $t + B$
$\boldsymbol{MAC}$	Homomorphic message authentication code
$\widehat{\mathbf{MAC}}$	Future homomorphic message authentication code
$\boldsymbol{\mu}$	Encrypted aggregation result
$\boldsymbol{\sigma}$	Verification of the correctness of the obtained aggregation result

5.1. System Model

Our system model consists of four main entities, as shown in Figure 1: mobile workers (MWs), the health care institution (HC), the cloud (C), and the trusted authority (TA).

- Trusted Authority (TA): The primary responsibility of the TA is the initialization of the entire system, which includes registering the participants, the HCs and the cloud; generating the required public parameters; and distributing the keys.
- Health Care Institution (HC): The HC is the requester that seeks aggregation statistics from patients' data. Due to limited storage and computation capabilities, the HC delegates computations to the cloud.
- Cloud: The cloud server receives encrypted data from MWs and computes the desired statistical results. The cloud server encrypts the computation results and forwards them to the HC.
- Participant (U): Participants refers to users or MWs who have smartphones and contribute their data to an HC. MWs are randomly chosen and encrypt and send their sensing data to the cloud.

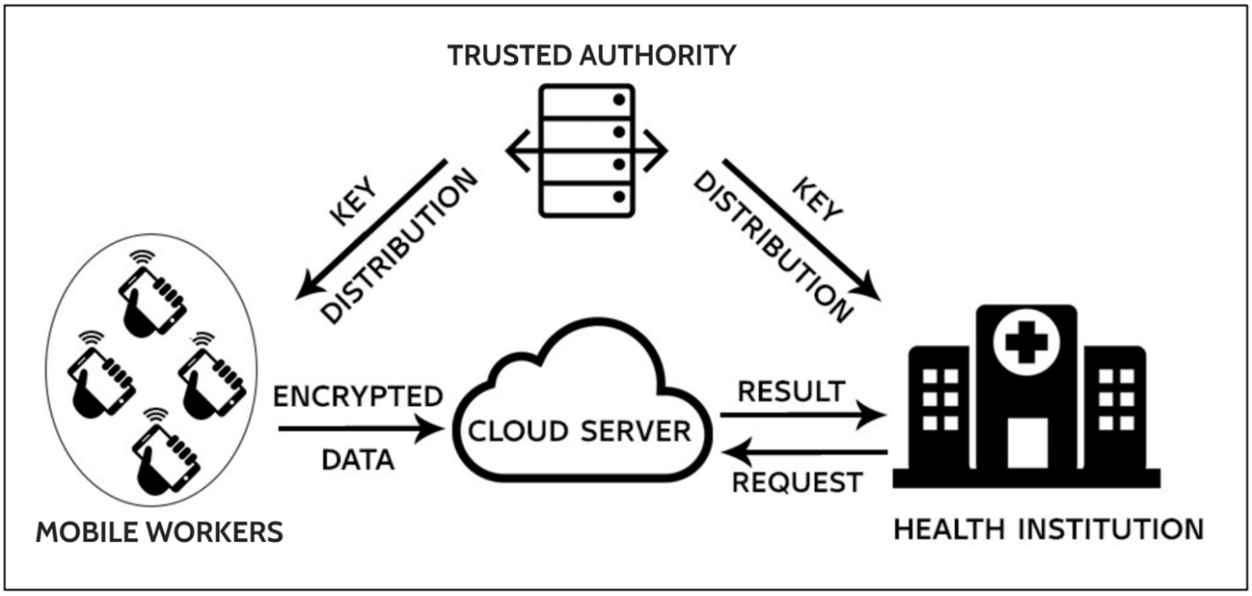

Figure 1. System model.

Figure 2 depicts the framework of our proposed scheme, which contains three main entities: the client, the cloud, and the health institution. The cloud is the most prominent of these entities in our proposed scheme and contains three main modules: the data integrity verification module, the fault tolerance module, and the data aggregation module.

The workflow of our framework is as follows: First, the user's encrypted data are sent with two parameters: the first is the future ciphertext, and the second is the verification code. Then, the cloud server will verify the data integrity and calculate the aggregation result. If the aggregator fails to receive the data from one or more users up to m, the aggregator will use the future ciphertext from the buffer memory to calculate the aggregation result and then send the result to the HC. Finally, the HC will decrypt the result.

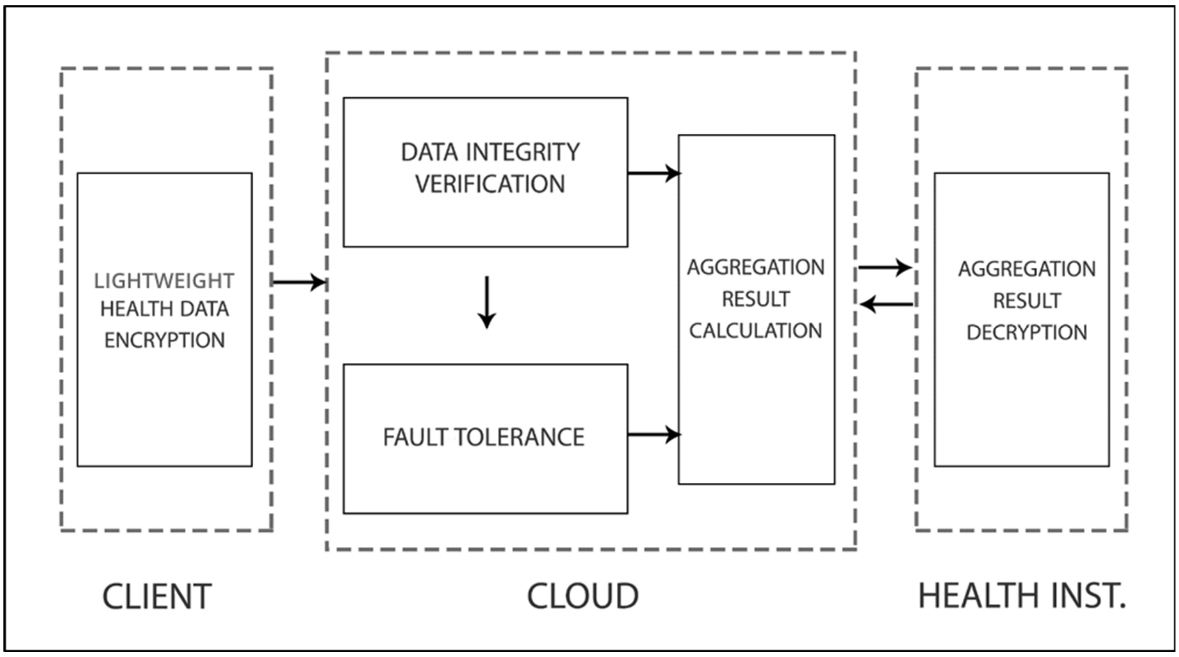
LIGHTWEIGHT HEALTH DATA ENCRYPTION
DATA INTEGRITY VERIFICATION
FAULT TOLERANCE
AGGREGATION RESULT CALCULATION
AGGREGATION RESULT DECRYPTION
CLIENT
CLOUD
HEALTH INST.

Figure 2. Our proposed framework.

5.2. A novel fault-tolerant privacy-preserving cloud-based data aggregation scheme for lightweight health data

Step 1: Setup and key management

The TA generates the necessary parameters and keys for the system, generates the bilinear parameters $(q, g, h, e, G_1, G_2, G_T)$ and encryption parameters for YASHE $(d, q, t, \lambda, x_{key}, x_{err}, w)$ and chooses a secure hash function $H(x)$. The TA registers all mobile users, the requester and the cloud in the system by sending them a private/public key pair (sk_c, pk_c). The TA selects N mobile users and registers them. Each registered MW is assigned private/public key pairs (sk_i, pk_i). Both the requester and workers are assigned encryption keys (α, β) for the homomorphic MAC.

Step 2: Sensing and reporting

During each time period t, each participant U_i reports his/her sensing data $x_{i,t}$ as follows. First, U_i computes

$$\hat{x}_{i,t} = x_{i,t} + \hat{r}_{i,t} \tag{9}$$

$$\hat{r}_{i,t} = G_{i,t}(n, \lambda) - G'_{i,t}(n, \lambda) = \widehat{G}_{i,t}(n, \lambda) \tag{10}$$

where $\hat{r}_{i,t}$ represents random noise variables with gamma densities. The sum of all random noise from all participants guarantees differential privacy due to the divisibility of the Laplace distribution, as described in Section 4.1.

However, adding random noise $\hat{r}$ is not adequate for ensuring the privacy of the data. As a result, the noisy data $\hat{x}_{i,t}$ should be encrypted using the public key pk_c of the requester to obtain

$$\mathrm{c_{i,t}} = \mathrm{Enc_{pk_c}}\ (\hat{\mathrm{x}}_{\mathrm{i,t}}) \tag{11}$$

Each ciphertext $\mathrm{c_{i,t}}$ is signed with its corresponding signature $\sigma_{\mathrm{i,t}}$ (generated by the secure hash function H() using participants' private keys $\mathrm{sk_i}$) to prevent tampering attacks and ensure data integrity as follows:

$$\sigma_{\mathrm{i,t}} = \mathrm{H(t||c_{i,t})^{sk_i}} \tag{12}$$

To address fault tolerance, we use the proactive aggregation protocol based on the future ciphertext mechanism. Each participant $\mathrm{U_i}$ computes two kinds of ciphertext—$\mathrm{c_{i,t}}$ for $\hat{\mathrm{x}}_{\mathrm{i,t}}$ and a future ciphertext $\hat{\mathrm{c}}_{\mathrm{i,t}}$ adapted from $\mathrm{c_{i,t}}$ as follows:

$$\hat{\mathrm{c}}_{\mathrm{i,t+B}} = \mathrm{Enc_{pk_c}}\ (\hat{\mathrm{r}}_{\mathrm{i,t+B}} + \mathrm{Lap_{i,t+B}}(\lambda)) \tag{13}$$

We assume that the aggregator has a buffer memory (B) to store future ciphertexts for each node. In our design, the aggregator is the cloud, which has intensive storage. Each node i sends its ciphertext $\mathrm{c_{i,t}}$ at time t and B future ciphertexts $\hat{\mathrm{c}}_{\mathrm{i,t}}$, $\hat{\mathrm{c}}_{\mathrm{i,t+1}}$, $\hat{\mathrm{c}}_{\mathrm{i,t+2}}$... $\hat{\mathrm{c}}_{\mathrm{i,t+B-1}}$, as shown in Figure 3. In the next iteration, each node sends two ciphertexts: The first ciphertext is the current ciphertext $\mathrm{c_{it}}$, and the second ciphertext is the future ciphertext $\hat{\mathrm{c}}_{\mathrm{i,t+B}}$ and the corresponding signature $\sigma_{\mathrm{i,t}}$. The purpose of a future ciphertext is to replace a given ciphertext if the cloud is unable to receive ciphertexts from the corresponding participant node. For increased efficiency, the HC controls the number of malfunctioning nodes using the parameter factor M.

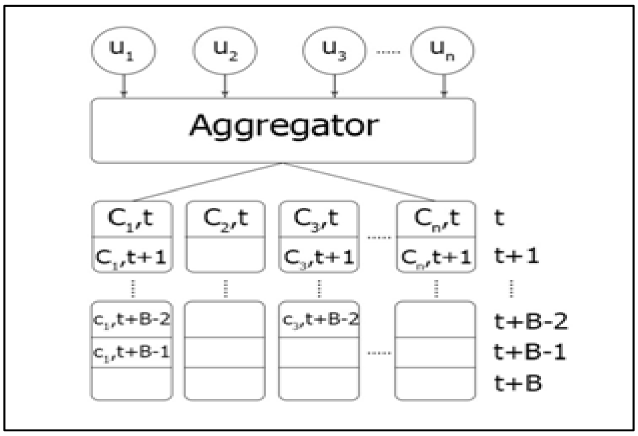

Figure 3. Future ciphertext mechanism.

Step 3: Verifying the correctness of the health data aggregation

To ensure end-to-end verification, we use the HMAC function *MAC*. Each participant $\mathrm{U_i}$ signs the reported data with the corresponding homomorphic *MAC* value $MAC\ (\hat{x}_{i,t})$ and calculates the homomorphic MAC value for the future ciphertext $\widehat{MAC}\ (\hat{r}_{i,t+B})$. Participant $\mathrm{U_i}$ sends $c_{i,t}$, $\hat{c}_{i,t+B}$, $\sigma_{i,t}$, $MAC\ (\hat{x}_{i,t})$ and $\widehat{MAC}\ (\hat{r}_{i,t+B})$ to the cloud.

Step 4: Data aggregation and verification

After receiving all reports, the cloud verifies whether the received reports were obtained from the chosen participants for each ciphertext $c_{i,t}$ using participants' public keys pk_i by checking

$$e(\sigma_{i,t}, g_2) = e(H(t||c_{i,t}), pk_i) \tag{14}$$

If the above equation is valid, then data integrity is guaranteed, and the cloud proceeds to compute the aggregation result μ. If not, a breach has occurred.

$$\mu = \sum_{i=1, i \notin U_F}^{n} c_{i,t} \tag{15}$$

However, this equation does not consider fault tolerance. If some reports were not received by the cloud, the cloud cannot verify the received reports or obtain the aggregation results. For a more efficient and reliable schema, we modify the future ciphertext mechanism to enable users to set a preference configuration parameter *M* and resist the failure of a maximum of *M* participants out of *N* total participants.

$$\mu = \sum_{i=1, i \notin U_F}^{n} c_{i,t} + \sum_{j=1, j \in U_F}^{n} \hat{c}_{j,t} \tag{16}$$

If the cloud does not receive the ciphertext c from between one and M nodes, where HC can specify M, the cloud uses the future ciphertext ĉ , which corresponds to the malfunctioning node from the buffer memory. If the number of malfunctioning nodes exceeds M, then the system is reinitialized to choose new medical health care nodes.

To verify the correctness of the aggregation result, the cloud computes the corresponding homomorphic message authentication code MAC as follows:

$$\sigma = \sum_{i=1}^{n} MAC\ (\hat{x}_{i,t}) \tag{17}$$

If the number of participants who fail to send their data is less than M, the cloud verifies the correctness of the aggregation result as follows:

$$\sigma = \sum_{i=1, i \notin U_F}^{n} MAC\ (\hat{x}_{i,t}) + \sum_{j=1, j \in U_F}^{n} \widehat{MAC}\ (\hat{r}_{j,t}) \tag{18}$$

The cloud forwards the results and the corresponding homomorphic MAC values $\{\mu, \sigma\}$ to the requester (the HC).

Step 5: Decryption and verification of the results

When HC receives $\{\mu, \sigma\}$ from the cloud, it derives the aggregation result $\sum_{i=1}^{n} c_{i,t}$ by decrypting μ as follows:

$$\sum_{i=1}^{n} Dec_{sk_c}(c_{i,t}) = Dec_{sk_c}(\mu) \tag{19}$$

The HC verifies the correctness of the aggregation result obtained using the homomorphic MAC algorithm by checking

$$MAC(Dec_{sk_c}(\mu)) \overset{?}{=} \prod_{i=1}^{n} MAC(\hat{x}_{i,t}) + \prod_{j=1}^{n} \widehat{MAC}\ (\hat{r}_{j,t}) \tag{20}$$

If the verification fails, the HC rejects the results. Otherwise, the HC accepts the results.

6. Security analysis

This section analyzes the security and privacy requirements satisfied by our proposed scheme. Moreover, we demonstrate how our proposed scheme resists different types of adversary models.

6.1. Data privacy

In our scheme, health data $\mathrm{x_i}$ are encrypted using YASHE, which is indistinguishable under the chosen ciphertext attack (IND-CPA) and secure under the decisional-RLWE assumption [34]. It is impossible for any time-bounded adversary to decrypt the ciphertext and obtain the health data without the knowledge of the private key, which is known only by the HC.

- *Resilience against external attacks:*

Proof: The external adversary cannot eavesdrop on the ciphertext $c_{i,t}$ and extract $x_{i,t}$ successfully since he/she has no knowledge of t, q, f or e, h. Such knowledge is impossible because f is held securely by participant U_i and e, h is privately held by the HC.

6.2. Differential privacy

During each time period t, the cloud can perform one of the above two types of queries. Both queries provide 2ϵ-differential privacy, where $\lambda = GS/\epsilon$ and GS is the global sensitivity of the aggregation result. Although the cloud uses the current and future ciphertexts to infer the sensing data $x_{i,t}$,, i.e., $c_{i,t}$ - $\hat{c}_{i,t} = x_{i,t} - Lap_{i,t}(\lambda)$, it also provides ϵ -differential privacy for the data $x_{i,t}$ [12], as the Laplace distribution has a symmetric shape around its mean of zero. Therefore, during each time period, from the participants' perspective, our scheme provides 2ϵ-differential privacy based on its parallel composition and sequential composition properties. Furthermore, our scheme provides protection against human factor-aware differential aggregation (HAD) [36]. This type of attack aims to break individual privacy. Suppose there are three MWs MW_1, MW_2 and MW_3, and the sensing data x_1, x_2 of MW_1,MW_2, respectively, are stable at time slots t_1 and t_2. MW_3 does not report any data at time slot t_2. From Eqs (5), (9), (10) and (13), the aggregated results for t_1 and t_2 are M_1=$\sum_{i=1}^{3} x_{i,1} + Lap_1(\lambda)$ and M_2=$\sum_{i=1}^{2} x_{i,2} + Lap_2(\lambda) + Lap_{3,2}(\lambda)$, respectively. It is infeasible for the adversary to derive the sensing data x_3 of MW_3 at time slot t_1 by comparing the aggregated result of t_1 and t_2 since $M_1 - M_2 = \ x_{3,1} - Lap_{3,1}(\lambda)$.

6.3. Data integrity

In our scheme, the cloud can easily detect if a report has been modified or interrupted by any adversary. Each report will be signed by a secure hash function at each time t.

- *Resilience against modification attacks:*

Proof: Assume that the adversary modifies $c_{i,t}$ and $\sigma_{i,t}$ into $c_{i,t}'$ and $\sigma_{i,t}'$, respectively. The modified message passes the verification step if and only if $\sigma_{i,t}'$ is guessed correctly. However, GDH group theory posits that it is infeasible for the adversary to determine $\sigma'_{i,t}$ from $e(\sigma'_{i,t}, g_2) = e(H(t||c'_{i,t})), pk_i)$ since G_1 is a GDH group. Additionally, for the given $\sigma'_{i,t}$, it is impossible to extract $c'_{i,t}$ from $e(\sigma'_{i,t}, g_2) = e(H(t\ ||c'_{i,t}), pk_i)$ due to the features of the secure hash function and GDH group.

Therefore, when the adversary tries to transmit a modified message $c'_{i,t}$ to the cloud, the modification can be detected by the cloud. As a result, our proposed scheme is resilient against modification attacks.

- *Resilience against impersonation attacks:*

Proof: To impersonate U_1, the adversary must know the private key sk_i. Using the public key pk_i and the signature $\sigma_{i,t} = H\ (t||c_{i,t})^{sk_i}$, it is intractable to find sk_i in polynomial time due to the discrete logarithmic assumption in G_1.

- *Resilience against reply attacks:*

Proof: The adversary launches a reply attack by sending ciphertext c_i with the signature $\sigma_{i,1}$ at time t_2, which has been used at time t_1, where $(t_1 < t_2)$. This can be detected by the cloud since $e(\sigma_{i,1}, g_2) = e\ (H\ (t_1||c_{i,1}), pk_i)$.

6.4. Robustness

To achieve robustness and node failure resistance in our scheme, we utilize a future ciphertext mechanism that requires low memory expenses. In the case of node failure, the cloud can still compute the aggregation and allows the HC to verify the correctness of aggregation. This in turn guarantees fault tolerance and robustness.

Table 3. Comparison of the security features of the proposed approach and related works.

Features	Our scheme	Won et al. [10]	Zhuo et al. [26]	Chen et al. [12]
PPR	Yes	Yes	Yes	Yes
REX	Yes	No	Yes	No
DPR	Yes	Yes	No	Yes
RIM	Yes	No	Yes	-
RMO	Yes	No	Yes	-
RRE	Yes	No	-	-
ROU	Yes	Yes	No	Yes
Correctness of Verification	Yes	No	Yes	Yes
Computation Delegation	Yes	No	Yes	No

PPR:	Privacy preservation.	RMO:	Resilience against modification attacks.
REX:	Resilience against external attacks.	RRE:	Resilience against reply attacks.
DPR:	Differential privacy.	ROU:	Robustness.
RIM:	Resilience against impersonation attacks.		

6.5. Correctness of the verification process

We use HMAC to ensure the correctness of the obtained aggregation result. First, during each time period t, the cloud computes the summation of the HMACs' σ for all received data and forwards the sum to the HC with the aggregation result μ. Then, the HC computes the HMAC for the aggregation result μ and checks whether the equation below holds:

$$MAC\left(Dec_{sk_c}(\mu)\right) \overset{?}{=} \sigma$$

Therefore, if the adversary tampers with the aggregation result, this tampering can be detected by the HC. Moreover, Table 3 demonstrates a comparison between the security features of our proposed scheme and those of other works [10,12,26].

7. Performance analysis

Our proposed scheme is implemented based on the homomorphic scheme developed by Lepoint and Naehrig [32] using the Fast Library for Number Theory (FLINT) arithmetic library and the GNU Multiple Precision (GMP) math library. Our simulation experiments and benchmark tests are executed on a laptop with an Intel core i5 processor, 6 GB of RAM and the Windows 7 (64-bit) operating system. We also implement the scheme of Won et al. [10] for comparison. The performance results are stated in terms of milliseconds.

Additionally, we consider that the health data are manipulated by the patient's mobile phone (MW). Encryption is performed by the MW before sending the data to the cloud, and decryption is performed by the HC after sending the data to the cloud. The cloud receives the encrypted data, computes the summation of these encrypted data, and forwards the encrypted results to the HC. The size of the encrypted dataset is relatively small, as our scheme focuses on lightweight health data. Our simulation dataset is randomly generated from 35 to 42 human body temperature readings.

Table 4. Key generation cost for different security levels.

Security Bit	Our Scheme				Scheme of Won et al. [10]	
	n	q	CPU Time (ms)	Real Time (ms)	CPU Time (ms)	Real Time (ms)
128	2048	54	0.115	0.346	0.960	1.917
192	4096	75	0.151	0.395	0.985	1.982
256	8192	118	0.185	0.469	0.995	2.078

7.1. Cost of key generation

First, we compare the key generation costs of an MW in our scheme with those in the scheme of Won et al. [10] by changing the security bit to examine the key generation costs at different security levels. Table 4 shows the parameter sets used in our benchmarks. We choose these parameters based on [37]. The comparison is shown in Figures 4 and 5 for the real-time and CPU time flags yielded by benchmark testing. For this comparison, we calculate the cost based on a group of 100 MWs. The graphs plotted in Figures 4 and 5 indicate that the required time for key generation in our scheme is lower than that in the scheme of Won et al. [10] in terms of both real time and CPU time. Note that the real time required for key generation in the scheme of Won et al. [10] is three times higher than that required for key generation in our scheme. Thus, our scheme is four times faster than that of Won et al. [10] based on CPU time flags. The key generation cost is critical to the MW, as a lower cost for key generation leads to a longer battery life.

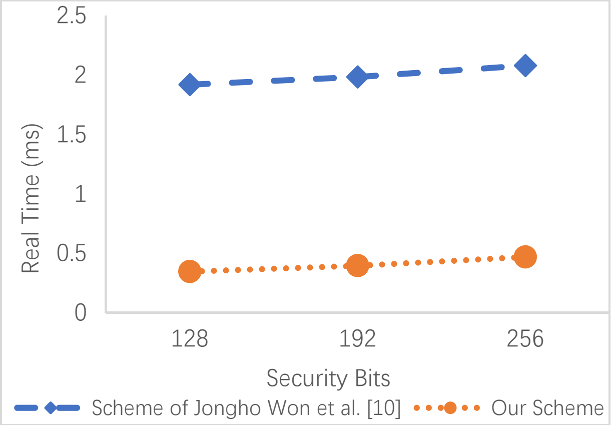

Figure 4. Comparison of key generation costs in terms of real time.

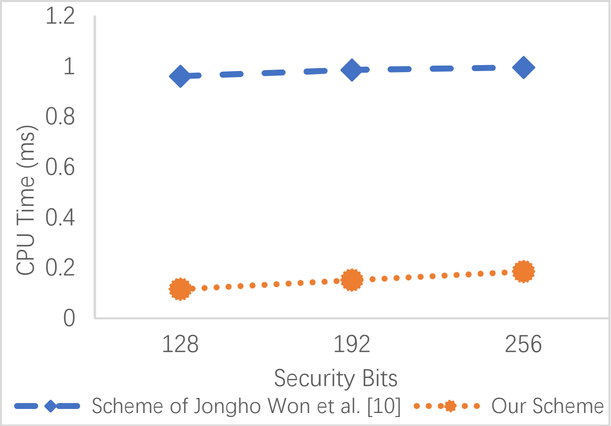

Figure 5. Comparison of key generation costs in terms of CPU time.

7.2. Cost of encryption

We also simulate the costs of encryption incurred by an MW when each group of our scheme has 100 participants (U) and compare the calculated costs with those of the scheme of Won et al. [10] at different security levels by changing the security bit. The simulation results are shown in Figures 6 and 7 for the real-time and CPU time flags yielded by benchmark testing. As shown in Figures 6 and 7, the encryption time of our scheme is superior to that of the scheme developed by Won et al. [10]. In terms of both real time and CPU time, the encryption cost of our scheme is six times lower than that of the scheme of Won et al. [10]. The low efficiency of the Won et al. [10] scheme is attributed to its encryption mechanism, where each participant in each time period t must communicate with all partners from the same group to exchange the secret keys $\boldsymbol{sk_{i,t}}$ to be used as the encryption key. To reduce the encryption cost in the scheme of Won et al. [10], we need to reduce the number of participants (U) in each group. This reduction would cause a decrease in the privacy level of the data and result in a reduced security level. The opportunity for the adversary to attack and disclose the data would then increase.

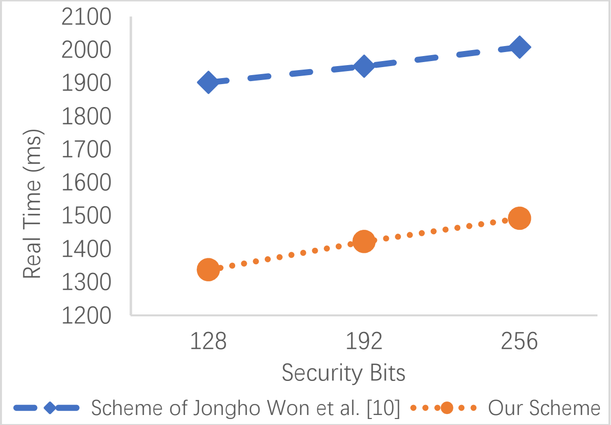

Figure 6. Comparison of encryption costs in real time for different security levels.

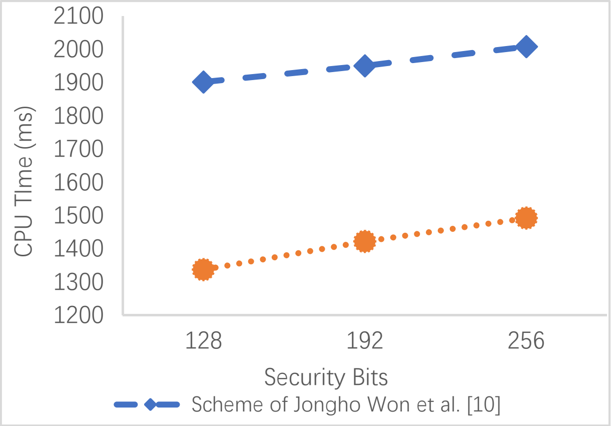

Figure 7. Comparison of encryption costs in CPU time for different security levels.

7.3. Low aggregation error

To make our scheme more practical, we utilize the future ciphertext mechanism proposed by Won et al. [10] to guarantee fault tolerance at the expense of two main requirements. If failures occur, the cloud can still calculate the aggregation result and the corresponding data integrity verification value. To evaluate our fault tolerance protocol, we measure the closeness between the actual summation of the sequence of data and the noisy sum calculated using the root mean square error (RMSE). Figure 8 shows the simulation result of our proposed scheme, where p is the probability of failure for MWs. The error in our scheme is significantly lower than that in the scheme developed by Won et al. [10].

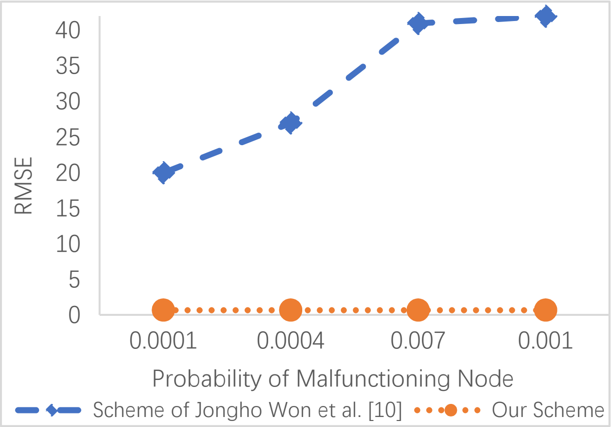

Figure 8. Comparison of encryption costs in real time yielded by changing the number of participants.

8. Conclusions and future work

We propose a fault-tolerant privacy-preserving cloud-based data aggregation scheme for lightweight health data. Our proposed scheme takes advantage of the numerous capabilities of the cloud by enabling an HC to delegate data aggregation tasks to the cloud. In our proposed scheme, we implement YASHE to protect the patient's identity and privacy, which enables the cloud to calculate the aggregation result with encrypted data. For differential privacy, we distribute noise among the MWs. Although our scheme enables the HC to verify the correctness of the aggregation result, our fault tolerance scheme is proactive and based on a future ciphertext mechanism. For increased efficiency, we enable the HC to control the number of acceptable malfunctioning nodes.

Compared with the aggregation process in the scheme of Won et al. [10], that in our scheme has a lower aggregation error and is not affected by the number of malfunctioning nodes. In addition, the performance evaluation shows that the computational overhead is significantly reduced. Unlike the encryption time in the scheme of Won et al. [10], that in our scheme is not affected by the number of participants utilized. The simulation results demonstrate the efficiency and feasibility of our scheme. In future work, we will improve our scheme to support multifunctional health data aggregation. Additionally, we will apply batch verification instead of individually verifying the reported data, which will improve the performance of the scheme.

Acknowledgments

This research project was supported by a grant from the Research Center of the Female Scientific and Medical Colleges, Deanship of Scientific Research, King Saud University.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. G. Dhand, S. S. Tyagi, Data aggregation techniques in WSN: survey, *Proc. Comput. Sci.*, **92** (2016), 378–384.
2. K. Zhang, K. Yang, X. Liang, Z. Su, X. Shen, H. H. Luo, Security and privacy for mobile healthcare networks: from a quality of protection perspective, *IEEE Wirel. Commun.*, **22** (2015), 104–112.
3. C. Castelluccia, A. C. F. Chan, E. Mykletun, G. Tsudik, Efficient and provably secure aggregation of encrypted data in wireless sensor networks, *ACM Trans. Sens. Netw. (TOSN)*, **5** (2009), 1–36.
4. N. Dong, H. Jonker, J. Pang, Challenges in eHealth: from enabling to enforcing privacy, in *International Symposium on Foundations of Health Informatics Engineering and Systems*, Springer, Berlin, (2012), 195–206.
5. A. Abbas, S. U. Khan, E-health cloud: privacy concerns and mitigation strategies, in *Medical Data Privacy Handbook*, Springer International Publishing, (2015), 389–421.
6. S. P. Ahuja, S. Mani, J. Zambrano, A survey of the state of cloud computing in healthcare, *Netw. Commun. Technol.*, **1** (2012), 12.
7. A. Abbas, S. U. Khan, A review on the state-of-the-art privacy-preserving approaches in the e-health clouds, *IEEE J. Biomed. Health Inf.*, **18** (2014), 1431–1441.
8. S. Han, S. Zhao, Q. Li, C. Ju, W. Zhou, PPM-HDA: privacy-preserving and multifunctional health data aggregation with fault tolerance, *IEEE Trans. Inf. Forensics Secur.*, **11** (2016), 1940–1955.
9. L. Bergamini, L. Becchetti, A. Vitaletti, Privacy-preserving environment monitoring in networks of mobile devices, in *NETWORKING 2011 Workshops*, Springer, (2011), 179–191.
10. J. Won, C. Y. T. Ma, D. K. Y. Yau, N. S. V. Rao, Proactive fault-tolerant aggregation protocol for privacy-assured smart metering, in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, IEEE, (2014), 2804–2812.
11. G. Ács, C. Castelluccia, I have a DREAM! (differentially private smart metering), in *International Workshop on Information Hiding*, Springer, (2011), 118–132.
12. J. Chen, H. Ma, D. Zhao, Private data aggregation with integrity assurance and fault tolerance for mobile crowd-sensing, *Wirel. Netw.*, **23** (2017), 131–144.
13. R. Bista, J. W. Chang, Privacy-preserving data aggregation protocols for wireless sensor networks: a survey, *Sensors (Basel)*, **10** (2010), 4577–4601.
14. B. Fabian, T. Ermakova, P. Junghanns, Collaborative and secure sharing of healthcare data in multi-clouds, *Inf. Syst.*, **48** (2015), 132–150.
15. R. Lu, X. Liang, X. Li, X. Lin, X. Shen, EPPA: an efficient and privacy-preserving aggregation scheme for secure smart grid communications, *IEEE Trans. Parallel Distrib. Syst.*, **23** (2012), 1621–1631.
16. H. M. Khan, A. Khan, F. Jabeen, A. U. Rahman, Privacy preserving data aggregation with fault tolerance in fog-enabled smart grids, *Sustainable Cities Soc.*, **64** (2021), 102522.
17. K. Zhang, X. Liang, M. Baura, R. Lu, X. Shen, PHDA: a priority based health data aggregation with privacy preservation for cloud assisted WBANs, *Inf. Sci.*, **284** (2014), 130–141.
18. L. Chen, R. Lu, Z. Cao, K. AlHarbi, X. Lin, MuDA: multifunctional data aggregation in privacy-preserving smart grid communications, *Peer Peer Netw. Appl.*, **8** (2015), 777–792.

19. J. Won, C. Y. T. Ma, D. K. Y. Yau, N. S. V. Rao, Privacy-assured aggregation protocol for smart metering: a proactive fault-tolerant approach, *IEEE/ACM Trans. Netw.*, **24** (2016), 1661–1674.
20. K. Grining, M. Klonowski, P. Syga, Practical fault-tolerant data aggregation, in *International Conference on Applied Cryptography and Network Security*, Springer, Cham, (2016), 386–404.
21. H. Zhu, Y. Yuan, Y. Chen, Y. Zha, W. Xi, B. Jia, et al., A secure and efficient data integrity verification scheme for cloud-IoT based on short signature, *IEEE Access*, **7** (2019), 90036–90044.
22. Benaloh, M. Chase, E. Horvitz, K. Lauter, Patient controlled encryption: ensuring privacy of electronic medical records, in *Proceedings of the 2009 ACM Workshop on Cloud Computing Security*, Association for Computing Machinery, (2009), 103–114.
23. M. M. Fouda, Z. M. Fadlullah, N. Kato, R. Lu, X. S. Shen, A lightweight message authentication scheme for smart grid communications, *IEEE Trans. Smart Grid*, **2** (2011), 675–685.
24. H. Bao, L. Chen, A lightweight privacy-preserving scheme with data integrity for smart grid communications, *Concurr. Comput. Pract. Exp.*, **28** (2016), 1094–1110.J.
25. R. Zhang, J. Shi, Y. Zhang, C. Zhang, Verifiable privacy-preserving aggregation in people-centric urban sensing systems, *IEEE J. Sel. Areas Commun.*, **31** (2013), 268–278.
26. G. Zhuo, Q. Jia, L. Guo, M. Li, P. Li, Privacy-preserving verifiable data aggregation and analysis for cloud-assisted mobile crowdsourcing, in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, IEEE, (2016), 1–9.
27. C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, M. Naor, Our data, ourselves: privacy via distributed noise generation, in *Advances in Cryptology-EUROCRYPT 2006*, Springer, (2006), 486–503.
28. C. Meshram, An efficient ID-based cryptographic encryption based on discrete logarithm problem and integer factorization problem, *Inf. Process. Lett.*, **115** (2015), 351–358.
29. D. Boneh, B. Lynn, H. Shacham, Short signatures from the weil pairing, in *International conference on the theory and application of cryptology and information security*, Springer, (2001), 514–532.
30. C. Dwork, Differential privacy, in *International Colloquium on Automata, Languages, and Programming*, Springer, 2006.
31. J. He, L. Cai, Differential private noise adding mechanism: basic conditions and its application, in *2017 American Control Conference (ACC)*, IEEE, (2017), 1673–1678.
32. T. Lepoint, M. Naehrig, A comparison of the homomorphic encryption schemes FV and YASHE, in *International Conference on Cryptology in Africa*, Springer, Cham, (2014), 318–335.
33. A. Costache, N. P. Smart, Which ring based somewhat homomorphic encryption scheme is best?, in *Cryptographers' Track at the RSA Conference* , Springer, (2016), 325–340.
34. J. W. Bos, K. Lauter, J. Loftus, M. Naehrig, Improved security for a ring-based fully homomorphic encryption scheme, in *IMA International Conference on Cryptography and Coding*, Springer, Berlin, Heidelberg, (2013), 45–64.
35. J. Shao, Efficient verifiable multi-secret sharing scheme based on hash function, *Inf. Sci.*, **278** (2014), 104–109.
36. W. Jia, H. Zhu, Z. Cao, X. Dong, C. Xiao, Human-factor-aware privacy-preserving aggregation in smart grid, *IEEE Syst. J.*, **8** (2014), 598–607.

37. K. Laine, Simple encrypted arithmetic library 2.3. 1, Microsoft Research, 2017. Available from: https://www. microsoft. com/en-us/research/uploads/prod/2017/11/sealmanual-2-3-1. Pdf.
38. H. Liu, T. Gu, Y. Liu, J. Song, Z. Zeng, Fault-tolerant privacy-preserving data aggregation for smart grid, *Wirel. Commun. Mobile Comput.*, **2020** (2020).