PPM-HDA: Privacy-preserving and multifunctional health data aggregation with fault tolerance

Song Han, Shuai Zhao, Qinghua Li, Chun-Hua Ju and Wanlei Zhou

***Abstract*—Wireless Body Area Networks (WBANs), as a promising health-care system, can provide tremendous benefits for timely and continuous patient care and remote health monitoring. Owing to the restriction of communication, computation and power in WBANs, cloud assisted WBANs, which offer more reliable, intelligent, and timely health-care services for mobile users and patients, are receiving increasing attention. However, how to aggregate the health data multifunctionally and efficiently is still an open issue to the cloud server (CS). In this paper, we propose a privacy-preserving and multifunctional health data aggregation mechanism (PPM-HDA) with fault tolerance for cloud assisted WBANs. With PPM-HDA, the CS can compute multiple statistical functions of users' health data in a privacy-preserving way to offer various services. Specifically, we first propose a multifunctional health data additive aggregation scheme (MHDA$^+$) to support additive aggregate functions such as average and variance. Then we put forward MHDA$^\oplus$ as an extension of MHDA$^+$ to support non-additive aggregations such as min/max, median, percentile and histogram. PPM-HDA can resist differential attacks, which most existing data aggregation schemes suffer from. The security analysis shows that PPM-HDA can protect users' privacy against many threats. Performance evaluations illustrate that the computational overhead of MHDA$^+$ is significantly reduced with the assistance of CSs. Our MHDA$^\oplus$ scheme is more efficient than previously reported min/max aggregation schemes in terms of communication overhead when the applications require large plaintext space and highly-accurate data.**

***Index Terms*—Multifunctional aggregation, Differential privacy, Spatial aggregation, Temporal aggregation, Fault tolerance, Privacy-preserving, Cloud assisted WBANs.**

I. Introduction

With the increasing number of elderly citizens and the demand for remote health monitoring in our daily life, wireless body area networks (WBANs), which can monitor patients or mobile users' health status in a timely manner, are going to play an important role in facilitating and maintaining heath-care systems [1]. WBANs provide various services in different areas such as remote health monitoring, sports, entertainment and the military. It can be used to collect different physiology parameters including blood pressure, electrocardiography (ECG) and temperature [2].

Nowadays, health data aggregation services are mainly applied for remote health monitoring of patients, who want to monitor their health status in a timely manner. However, in the near future, with the increase of elderly citizens and the improvement of people's living standards, more and more people will pay attention to their health, and health data aggregation services will be used on a large scale in the future. Spatial aggregate data (which is the aggregation of multiple users' data at the same time point, e.g., the average blood pressure of the people in an area) is needed by medicine research centers for pharmaceutical research and production. Temporal aggregate data (which is the aggregation of the same user's data at different time points, e.g., a user's highest blood pressure in the past 24 hours) is needed by certified hospitals to monitor the health condition of users and provide timely feedback. A more detailed discussion of applications will be given in the Motivation part that will follow. With the ever increasing demands from patients and mobile users, WBANs need to process the sensed data in a timely manner and store the doctors' feedback online. It is difficult to achieve these goals only relying on traditional WBANs, as real applications consume more resources, such as communication power, computation and storage resources [3]. It is also costly for hospitals to deploy the corresponding servers for storing and processing user's health data by themselves and they will outsource these services to a large data storage and processing company, such as Amazon Web Services (AWS) and Google. By taking advantage of its existing servers and resources, this large company can build a cloud server cluster to provide services for these hospitals. In this way, a hospital only needs to pay a certain amount of service fee for using the health data storage and processing services. Therefore, cloud server enabled WBANs, i.e. cloud assisted WBANs, are introduced to process and store health data.

Cloud assisted WBANs provide various services for mobile users and patients by making use of cloud servers to store large amounts of health data and process them for doctor's diagnosis [4], [5]. However, privacy and security are becoming significant issues, as mobile communications are deeply involved in cloud assisted WBANs [6]. Health data operations should be authenticated and resist malicious modifications in health-care applications. For example, network performance might

This work is partially supported by Zhejiang Province Qianjiang Distinguished Scholar Support Program, National Key Technology Support Project No.2014BAH24F06, Zhejiang Finance Department Fund No.1130JY3114020G, Zhejiang Xinmiao Project No.2014R408091, MOE Grant No.14JJD630011 and NSFC Grants No.61301142;61402406. The associate editor approving it for publication is Prof. Dr. S. Katzenbeisser.

S. Han, S. Zhao and C. Ju are with Zhejiang Gongshang University, Hangzhou, 310018, China. E-mail: hansongau@gmail.com;zjgsuzhaoshuai@gmail.com; jch@mail.zjsu.edu.cn.

Q. Li is with the Department of Computer Science and Computer Engineering, University of Arkansas, Fayetteville, AR 72701,US. Email: qinghual@uark.edu.

W. Zhou is with the School of Information Technology, Deakin University, Burwood, Victoria, 3125 Australia. Email: wanlei.zhou@deakin.edu.au.

be degraded as an adversary fabricates an false emergency call and makes it distributed in the network. Moreover, from the user's point of view, privacy is also a big concern as health data is highly relevant to users themselves. For example, some specific behaviors of a person, such as having meals, sleeping, etc., are reflected by their ECG. As a result, user's privacy will be violated if such health data is revealed. Therefore, users' health data needs to be protected from unauthorized entities.

Motivation: Consider a scenario where a cloud assisted WBAN can use privacy-preserving data aggregation to provide health-care services to the elderly users with hypertension in a community. Elderly users with hypertension will be equipped with some body area sensors to monitor their blood pressure. To provide health services for an individual elderly user, the certified hospital can monitor his blood pressure remotely through periodically collecting his maximum/minimum blood pressure in the past day (which is temporal aggregate data. If the maximum value of systolic pressure or the minimum value of diastolic pressure is abnormal, the hospital can trigger an alert and ask the user to come to the hospital for a thorough check. It is evident that temporal aggregation of an individual's data is needed by the hospital in this scenario so that it can provide better health-care services. Besides, the spatial aggregate statistics of multiple elderly users can be used by medicine research centers for pharmaceutical research and production, and for public agencies to provide better community services. In addition, preventing strong adversary from disclosing individual user's health data and making the system fault-tolerant are both important for the privacy of the elderly users with hypertension in this community. This needs a privacy-preserving multifunctional health data aggregation scheme to realize the above services. In addition, although the aggregation of n elderly users and that of $n-1$ elderly users are both protected, the cloud server might still conduct a differential attack and compromise the "differential" elderly user's privacy by facilitating the summation of n elderly users and that of $n-1$ elderly users. Several schemes are proposed to address this problem, such as [8], [10], [13]–[15]. However, these schemes can only preserve differential privacy for summation aggregations. Preserving differential privacy of additive aggregations, such as variance aggregations, and non-additive aggregations, such as min/max, median, percentile and histogram, is still an open problem.

Contributions: For mitigating differential attacks, dealing with malfunction of users and CSs, and assisting the CS to compute more complex statistics in a privacy-preserving way, in this paper, we propose a privacy-preserving and multifunctional health data aggregation mechanism (PPM-HDA) with fault tolerance for cloud assisted WBANs. With PPM-HDA, the cloud server can compute multiple statistical functions of users' health data in a privacy-preserving way while offering more services. Our proposed PPM-HDA mechanism can also resist differential attacks, which most existing data aggregation schemes suffer from. In addition, for supporting fault tolerance for both mobile users and CSs, this scheme provides privacy-preserving aggregation against a stronger adversary which may compromise a few CSs. Below are the major contributions of this paper:

- Firstly, since the CS may need to compute multiple statistical functions to provide multiple services, we present a PPM-HDA mechanism that supports multifunctional additive and non-additive aggregations for cloud assisted WBANs. In addition, inspired by the fact that individual user's health data may suffer from differential attacks, our proposed PPM-HDA mechanism is designed to support differential privacy upon multifunctional aggregations. Compared with existing data aggregation schemes that can only compute summation aggregation [8], [16], [17] and additive aggregation [22], our proposed scheme provides more diversity and security for the CS.
- Secondly, a main technical contribution of our proposed scheme is that it can achieve min/max aggregation with only one round of communication with the mobile users, and $O(log(M))$ (suppose M is the space of plaintexts) total message size from each user. In comparison, Li et al's work [21] has one round of communication but needs more than $O(log(M))$ message size, and Shi et al's INFOCOM work [19] has $O(log(M))$ total message size but needs $log(M)$ rounds of communications.
- Thirdly, we propose aggregation protocol in a more challenging adversary model in which the adversary could compromise some of CSs and obtain their private keys. Compared with [22] which only considers a single server, our scheme is securer in the more challenging security model. We take malfunction of both users and CSs into consideration. Our scheme can support fault tolerance for users and CSs failure. Furthermore, our scheme supports both spatial and temporal aggregation. Compared with existing works [10]–[12], our scheme is more reliable and practical in case user or CS malfunction occurs.
- Finally, we provide security and privacy analysis to show that the PPM-HDA can resist various security threats and preserve user privacy. In addition, performance evaluations show that the computational overhead of MHDA$^+$ is significantly reduced with the assistance of CSs. Our MHDA$^\oplus$ scheme is more efficient than PHADA [21], PriSense [19] and VPA [20] in terms of communication overhead when large plaintext spaces and highly-accurate data are needed in real applications, especially in health-care applications.

The remainder of this paper is organized as follows. Network model, adversary model and design goals are presented in Section II. Then, we briefly recall the Boneh-Goh-Nissim cryptosystem and differential privacy in Section III. In Section IV, we propose the detailed MHDA$^+$ for providing multifunctional additive aggregation. We propose the MHDA$^\oplus$ as an extension of MHDA$^+$ to support non-additive aggregation in Section V. Subsequently, security analysis and performance evaluations are reported in Sections VI and VII, respectively. The related works are investigated in Section VIII. Finally, Section IX concludes the paper.

II. Problem Formalization

In this section, we formalize our research problems in WBANs, including network model, adversary model, security requirements and design goal.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2015.2472369, IEEE Transactions on Information Forensics and Security

A. Network model

In our network model, we consider a cloud assisted WBAN for mobile users, which includes a trusted authority (TA), some social spots (SPs), a health-care cloud having a set of semi-trusted cloud servers $\mathbb{S} = \{S_1, S_2, ..., S_k\}$ and a lot of mobile users or patients $\mathbb{U} = \{U_1, U_2, ..., U_n\}$. Our model is depicted in Fig. 1.

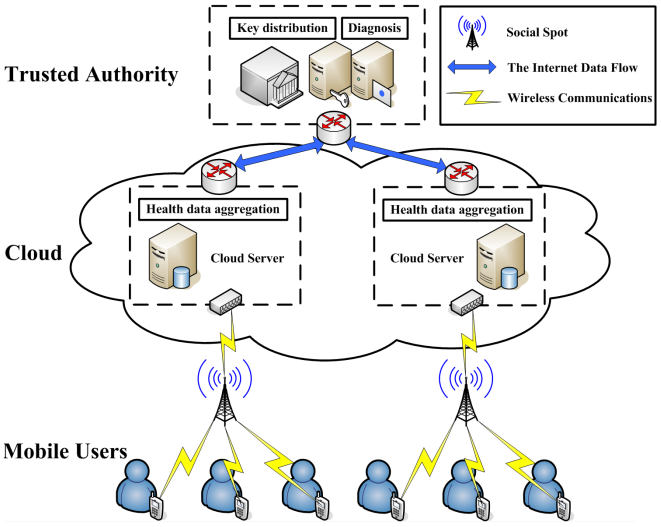

Fig. 1: Network model for cloud assisted WBAN

- The **Trusted Authority** is responsible for bootstrapping the whole system in the initialization phase. We assume that it is a trustable entity, and it could be a certified hospital which manages the users' health data. In the bootstrapping, the TA generates secret keys and users' certificates to each credible user. At the same time, it also generates secret keys for each cloud server. If TA wants to acquire the statistics of health data, $d+1$ working cloud servers will collaborate to decrypt the aggregated data, and one of these working CSs will send the statistics to this TA. On the other hand, if the TA wants to obtain each individual user's health data at one time point, the TA will receive $d+1$ decryption shares and decrypt the encrypted health data from each individual user by itself.
- The **Social Spot** is served as a local gateway to report the aggregated data to the cloud server. In real life, it could be a base station of Verizon. Thus we assume that it is an honest but curious entity, which will aggregate each user's health data and report the aggregated data to cloud servers honestly, but it is also curious about individual user's health data. We assume that it is also equipped with powerful and storage-rich communication devices. SPs are always deployed on intersections or hotspots where mobile users visit frequently. By wireless communication, SPs can collect health data from each user and report the aggregated data to cloud servers via the Internet.
- In this paper, the **Cloud Server** represents an individual server in the health-care cloud. An aggregation application may be deployed to multiple cloud servers managed by the same cloud service provider. Multiple cloud servers are needed for the purpose of workload sharing and fault tolerance. The health-care cloud as a whole is used to store and process the large volume of users' health data to provide information which can assist in a medical diagnosis. As each cloud server in the health-care cloud is a powerful entity, we assume that it is an honest but curious entity, which will store and process users' health data honestly, but it is also curious about individual user's health data. A strong adversary may compromise or paralyze some of the cloud servers $\mathbb{S} = \{S_1, S_2, ..., S_k\}$. It is costly for an adversary to compromise even a single cloud server, since each member of $\mathbb{S}$ is a powerful entity. Therefore, we assume that the strong adversary can only compromise a minority of the cloud servers, i.e., no more than $d = \lceil k/2 \rceil - 1$ of cloud servers. In order to protect users' privacy and data confidentiality, the health data stored in the CSs are in the form of ciphertexts.
- **Mobile users**, denoted by $\mathbb{U} = \{U_1, U_2, ..., U_n\}$, only need to report their health data to the SP. In order to monitor the individual user's health data and periodically report these health data to the SP, each mobile user is equipped with some body area sensors. After U_i obtains its health data, it only needs to upload these corresponding data to the SP via its PDA or smartphone [23].

B. Adversary model

A strong adversary $\mathscr{A}$ is considered in the adversary model. It not only can eavesdrop on communication flows but also can do the following attacks: i) In order to breach users' privacy, $\mathscr{A}$ is able to compromise a user directly. However, $\mathscr{A}$ would be unlikely to choose this approach, as there are a large number of mobile users. ii) $\mathscr{A}$ is able to install some malicious software on the SP to breach users' privacy; and iii) $\mathscr{A}$ is likely to compromise less than $d = \lceil k/2 \rceil - 1$ CSs to reveal users' privacy.

C. Security requirements

The strong adversary $\mathscr{A}$ is aiming to reveal as much of the mobile users' privacy as possible. In order to resist $\mathscr{A}$'s objective, the security requirements are as follows:

- $\mathscr{A}$ cannot expose users' private health data, even if it is able to eavesdrop on communication flows.
- $\mathscr{A}$ cannot expose users' private health data, even if it is able to compromise some users directly.
- $\mathscr{A}$ cannot expose users' private health data, even if it is able to install malicious software on the SP.
- $\mathscr{A}$ cannot expose users' private health data, even if it is able to compromise d CSs.

D. Design goal

Our design objective is to propose a privacy-preserving and multifunctional health data aggregation scheme with fault tolerance for cloud assisted WBANs. Specifically, we should realize the following three goals.

- *The proposed scheme should satisfy the security requirements.* As described in II(C), in order to resist $\mathscr{A}$'s

objective, the security requirements should be satisfied; otherwise, the users' privacy is breached.

- *The proposed scheme should achieve the multifunctional data aggregation.* Although the previous work [22] had realized some statistical additive aggregation, for example average, variance and one-way ANOVA, it cannot be directly applied to non-additive aggregation, for example min/max, median, percentile, histogram, etc. Therefore, the proposed scheme should achieve both additive and non-additive aggregation, so that the cloud server can compute multiple statistical functions of users' health data to provide various services.
- *The proposed scheme should guarantee fault tolerance and preserve differential privacy.* As the strong adversary $\mathscr{A}$ may compromise d CSs, the proposed scheme should guarantee that the remaining $k-d$ uncompromised CSs can decrypt the aggregated data from body area sensors. On the other hand, TA can still acquire the statistics of health data from working body area sensors, even if some body area sensors are malfunctioned. In addition, as individual user's health data may suffer from differential attacks, the proposed scheme should preserve differential privacy as well.

III. Preliminaries

We will recall the Boneh-Goh-Nissim cryptosystem [25] and differential privacy [13] briefly in this section, both of which serve as the basis of the proposed scheme.

A. Boneh-Goh-Nissim cryptosystem

In 2005 [25], Boneh, Goh and Nissim proposed the Boneh-Goh-Nissim cryptosystem which is a public key encryption scheme. Due to the homomorphic features, it has been widely used in many privacy-preserving aggregation schemes. Specifically, the Boneh-Goh-Nissim encryption includes three procedures: key generation, encryption and decryption (more details in [25]).

- Key Generation: In system initialization phase, the system first runs $Gen(\tau)$ to acquire the tuple $(p, q, \mathbb{G}, \mathbb{G}_1, e)$, in which $\tau \in \mathbb{Z}^+$ is a given security parameter, p, q are distinct primes with $|p| = |q| = \tau$, $\mathbb{G}$ and $\mathbb{G}_1$ are two cyclic groups of order $N = pq$, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ is a bilinear map [24]. The system chooses two generators $\mathrm{g}, x \in \mathbb{G}$ randomly and sets $h = x^q$. Then h is a random generator of the subgroup of $\mathbb{G}$ of order p. $SK = p$ is the private key. $PK = (N, \mathbb{G}, \mathbb{G}_1, e, \mathrm{g}, h)$ is the public key.
- Encryption: $m \in \{0, 1, ..., 2^w - 1\}$ represents a message, and the upper bound of this message space is $2^w - 1 \ll q$. $r \in \mathbb{Z}_N$ is a random number chosen by a user. Then, we can calculate the ciphertext as $C = \mathrm{g}^m \cdot h^r \in \mathbb{G}$.
- Decryption: After receiving the ciphertext $C \in \mathbb{G}$ and obtaining the private key $SK = p$, the system computes $C^p = (\mathrm{g}^m \cdot h^r)^p = (\mathrm{g}^p)^m$. Let $\mathrm{g}_p = \mathrm{g}^p$, then $C^p = \mathrm{g}_p^m$. By calculating the discrete logarithm of C^p base g_p, the system is able to get the message m in expected time $O(\sqrt{2^w - 1})$ by adopting Pollard's lambda method [26].

The Boneh-Goh-Nissim cryptosystem also has some homomorphic features. Firstly, it is additively homomorphic. For any ciphertexts $C_1, C_2 \in \mathbb{G}$ of messages $m_1, m_2 \in \{0, 1, ..., 2^w - 1\}$, one can simply calculate the product $C = C_1 C_2$ to acquire the ciphertexts of $m_1 + m_2$. Furthermore, utilizing the bilinear map by calculating $C = e(C_1, C_2) \in \mathbb{G}_1$, one can also acquire the product of two messages, and decrypt it similarly in the group $\mathbb{G}_1$. Note that there is only one multiplication which is involved in the Boneh-Goh-Nissim cryptosystem, and the result will be in $\mathbb{G}_1$. Therefore, the Boneh-Goh-Nissim cryptosystem does not have a total multiplicative homomorphism property, but it still supports additive homomorphism.

B. Differential privacy

In 2006 [13], Dwork first proposed differential privacy, which can preserve privacy by adding proper noise into the aggregation result. Subsequently, we review the formal definition of differential privacy as below.

Differential privacy (orginal definition from [13]): The aggregation function A gives ϵ-differential privacy if for any data set D_1 and D_2 differing by at most one element, and for any $S \subseteq \mathrm{Range(A)}$,

$$Pr[A(D_1) \in S] \leq exp(\epsilon) \cdot Pr[A(D_2) \in S]. \quad (1)$$

In this paper, we choose noises from the symmetric geometric distribution $Geom(\alpha)$, where $\alpha \in [0, 1]$. α could be seen as a discrete approximation of the Laplace distribution Lap(λ), i.e. $\alpha \approx exp(-\frac{1}{\lambda})$. In [15], Ghosh et al. pioneer the use of the geometric distribution to generate noise. The probability density function of the geometric distribution $Geom(\alpha)$ is

$$Pr[X = x] = \frac{1-\alpha}{1+\alpha}\alpha^{|x|}. \quad (2)$$

Formally, if the sensitivity of A is

$$\Delta A = \max_{D_1, D_2} ||A(D_1) - A(D_2)||_1, \quad (3)$$

for all D_1 and D_2 differing by at most one element, then the perturbed aggregation result can realize ϵ-differential privacy by means of adding geometric noise r randomly chosen from $Geom(exp(-\frac{\epsilon}{\Delta A}))$ to the aggregation data. Therefore, for any integer $k \in \mathrm{Range(A)}$,

$$Pr[A(D_1) + r = k] \leq exp(\epsilon) \cdot Pr[A(D_2) + r = k]. \quad (4)$$

IV. MHDA$^+$: Multifunctional health data additive aggregation

In this section, we propose a multifunctional health data additive aggregation scheme, called MHDA$^+$. It mainly consists of four phases: system initialization, user report generation, privacy-preserving report aggregation and secure report reading. In this part, we only illustrate average aggregation as an example. In [22], the authors have introduced some other aggregation, for example variance and one-way ANOVA aggregation. Although the procedure of additive aggregation is similar to previous works [22], our scheme has the following

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2015.2472369, IEEE Transactions on Information Forensics and Security

differences: i) In [22], the authors take a whole trusted control center into consideration, where the control center is in charge of the private key p. In our scheme, however, we introduce k semi-trusted CSs which assist in processing and storing the large volume of health data. The private key p is seen as a shared secret and each share of p is assigned to each CS as its private key in the system initialization phase. ii) We take the minority of CSs being compromised or malfunctioned into consideration. As a result, the more comprehensive and concrete consideration make our scheme more strong. In the next section, we present MHDA$^{\oplus}$ as an extension of MHDA^{+} to support non-additive aggregation, for example min/max, median, σ-percentile, histogram, etc.

A. System initialization

In the beginning, the TA is able to bootstrap the whole system. In particular, the TA runs $Gen(\tau)$ to acquire the bilinear map tuple $(p, q, \mathbb{G}, \mathbb{G}_1, e)$ in the system initialization phase. Subsequently, the TA utilizes the Boneh-Goh-Nissim cryptosystem to generate the tuple $(N, \mathbb{G}, \mathbb{G}_1, e, \mathrm{g}, h)$, where $h = \mathrm{g}^q$ is a random generator of the subgroup of $\mathbb{G}$ of order p and $\mathrm{g} \in \mathbb{G}$ is a random generator of $\mathbb{G}$. The TA also picks a one-way hash function $H : \{0,1\}^* \rightarrow \mathbb{G}$. Finally, the TA releases $(N, \mathbb{G}, \mathbb{G}_1, e, \mathrm{g}, h, H)$ as a public key in our system and randomly generates a secret polynomial function of degree d as $G(x) = p + a_1 x + ... + a_d x^d$, where $a_i \in \mathbb{Z}_N$, for $i = 1, 2, ..., d$. For each server $S_j \in \mathbb{S}$, TA first calculates the value of $G(j)$ and assigns it as S_j's private key.

B. Aggregation protocol for average

Average can be derived easily from sum. Therefore, in the followings steps, we first calculate sum aggregation and then derive the average aggregation.

1) User report generation: We assume that the mobile user will report its health data to the SP every 15 minutes. Mobile users can also assist the CS in monitoring the health status of users and offering a variety of services by means of reporting health data at each time point simultaneously. Specifically, each user $U_i \in \mathbb{U}$ gathers its health data $m_{i,\gamma} \in \{0, 1, ..., 2^w - 1\}$ at time point t_γ and carries out the following steps:

Step 1: U_i first calculates the hash value $\theta_\gamma = H(t_\gamma)$ at the current time point t_γ.

Step 2: U_i picks a random number $r_{i,\gamma} \in \mathbb{Z}_N^*$, and calculates $C_{i,\gamma} = \mathrm{g}^{m_{i,\gamma}} \cdot h^{\theta_\gamma \cdot r_{i,\gamma}}$.

Step 3: U_i reports $C_{i,\gamma}$ to the SP through WiFi.

2) Privacy-preserving report aggregation: According to the CS's requirements, the SP is able to aggregate users' health data in a privacy-preserving way, when n encrypted health data $C_{i,\gamma}$ for $i = 1, 2, ..., n$ are received. We define the aggregation of users' data as a function $A : \{0, 1, ..., 2^w - 1\}^n \rightarrow \mathbb{Z}_q$. The input is each user's health data $(m_1, m_2, ..., m_n)$ and the output is the aggregated result.

In addition, in order to achieve differential privacy, we allow the SP to perturb encrypted aggregations. Specifically, as the SP can obtain the random generator $\mathrm{g} \in \mathbb{G}$, after generating a noise from the geometric distribution, the SP can perturb encrypted aggregation by simply multiplying the encrypted noise to the aggregation. Moreover, the SP needs to add a noise proportional to the sensitivity of aggregation to ensure ϵ-differential privacy for a given ϵ. However, in order to ensure the desired privacy levels, the SP should pick the parameters of geometric distribution carefully, as different aggregations may have different sensitivities.

After receiving all users' encrypted data $C_{i,\gamma}$, the SP is able to calculate the aggregation with the following steps.

Step 1: The SP first utilizes users' encrypted health data $C_{i,\gamma}$ to calculate the encrypted aggregation A_γ as

$$\begin{aligned} A_\gamma &= \prod_{i=1}^{n} C_{i,\gamma} = \prod_{i=1}^{n} (\mathrm{g}^{m_{i,\gamma}} \cdot h^{\theta_\gamma \cdot r_{i,\gamma}}) \\ &= \mathrm{g}^{\sum_{i=1}^{n} m_{i,\gamma}} \cdot h^Z, \end{aligned} \tag{5}$$

where $Z = \sum_{i=1}^{n} (\theta_\gamma \cdot r_{i,\gamma}) \ mod \ p$.

Step 2: For average aggregation, A_γ is the encrypted sum aggregation of all users' data $M_{sum} = \sum_{i=1}^{n} m_{i,\gamma}$. Assume that $A_{sum}(D) = \sum_{U_i \in D} m_{i,\gamma}$, then $|A_{sum}(D_1) - A_{sum}(D_2)| \leq 2^w - 1$ holds for any two data sets D_1 and D_2 differing by at most one element. Therefore, we can obtain the sensitivity of A_{sum} by $\Delta A_{sum} = 2^w - 1$. The SP uses this sensitivity to randomly pick noises $\widetilde{m}$ from geometric distribution Geom($exp(-\frac{\epsilon}{2^w - 1})$).

Step 3: The SP calculates the final aggregations as

$$\begin{aligned} \widetilde{A}_\gamma &= A_\gamma \cdot \mathrm{g}^{\widetilde{m}} \\ &= \mathrm{g}^{\sum_{i=1}^{n} m_{i,\gamma} + \widetilde{m}} \cdot h^Z. \end{aligned} \tag{6}$$

The SP sends the aggregated data $\widetilde{A}_\gamma$ to the CS for further computation.

3) Secure report reading: CSs can calculate the needed statistics efficiently without exposing individual user's privacy, when the corresponding aggregations are received. Specifically, CSs will make the following calculations according to the TA's requirements.

Upon receiving $\widetilde{A}_\gamma$ at time point t_γ, $d+1$ working cloud servers $\wp \subset \mathbb{S}$ are randomly chosen to decrypt the aggregated data. Specifically, each cloud server $S_j \in \wp$ first computes

$$\beta_j = \prod_{i \in \wp, i \neq j} \frac{i}{i - j}, \tag{7}$$

then generates

$$D_{j,\gamma} = \widetilde{A}_\gamma^{\beta_j G(j)}. \tag{8}$$

After that, one of the $d+1$ working cloud servers collects all the $D_{j,\gamma}$ for each $S_j \in \wp$ and calculates

$$\begin{aligned} \widetilde{P}_\gamma &= \prod_{S_j \in \wp} D_{j,\gamma} = \prod_{S_j \in \wp} \widetilde{A}_\gamma^{\beta_j G(j)} \\ &= \widetilde{A}_\gamma^{\sum_{S_j \in \wp} \beta_j G(j)} = \widetilde{A}_\gamma^p \\ &= (\mathrm{g}^{\sum_{i=1}^{n} m_{i,\gamma} + \widetilde{m}})^p \cdot (h^Z)^p \\ &= (\mathrm{g}^p)^{\sum_{i=1}^{n} m_{i,\gamma} + \widetilde{m}} = \widehat{\mathrm{g}}^{\sum_{i=1}^{n} m_{i,\gamma} + \widetilde{m}}, \end{aligned} \tag{9}$$

where $\widehat{\mathrm{g}} = \mathrm{g}^p$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2015.2472369, IEEE Transactions on Information Forensics and Security

The polynomial function $G(x) = p + a_1 x + ... + a_d x^d$, where $a_i \in \mathbb{Z}_N$, for $i = 1, 2, ..., d$. According to the lagrange interpolation polynomial, we have

$$G(x) = \sum_{j=0}^{d}(\prod_{i=0, i\neq j}^{d} \frac{x_i - x}{x_i - x_j}) G(x_j). \quad (10)$$

Therefore,

$$\sum_{S_j \in \wp} \beta_j G(j) = \sum_{j=0}^{d}(\prod_{i=0, i\neq j}^{d} \frac{i-0}{i-j}) G(j) = G(0) = p. \quad (11)$$

We have $\sum_{i=1}^{n} m_{i,\gamma} + \widetilde{m} \leq (n+1)(2^w - 1)$, as $m_{i,\gamma}, \widetilde{m} \in \{0, 1, 2, \ldots, 2^w - 1\}$. By computing the discrete logarithm of $\widetilde{P}_\gamma$ with the base $\widehat{\mathrm{g}}$, the CS can get the sum of users' health data $M_{sum} = \sum_{i=1}^{n} m_{i,\gamma} + \widetilde{m}$ in expected time $O(\sqrt{(n+1)(2^w - 1)})$ using Pollard's lambda method [26]. Then, we can calculate the average of users' health data as $\overline{M} = \frac{1}{n} M_{sum}$. Finally, this CS sends $\overline{M}$ to the TA.

C. Fault tolerance

The proposed scheme can still work well even when cloud server malfunctions occur and user failures. In fact, our scheme can support fault tolerance and temporal aggregation. Moreover, it is very suitable for dynamic users. We will give the corresponding extensions below.

On the one hand, our proposed scheme can support fault tolerance of CS failures. Since a strong adversary may compromise or paralyze some of the cloud servers $\mathbb{S} = \{S_1, S_2, ..., S_k\}$ and each member of $\mathbb{S}$ is powerful entity, it is costly for an adversary to compromise even a single cloud server. So, we have assumed that the strong adversary can only compromise minority of the cloud servers, i.e., no more than $d = \lceil k/2 \rceil - 1$ of cloud servers. Therefore, so long as the failing CSs no more than d ones, there are still $k - d \geq d + 1$ working CSs can be used to keep the system processing normally.

On the other hand, our proposed scheme can also support fault tolerance of user failures. If some users $\widehat{U} \subset \mathbb{U}$ malfunction in the privacy-preserving report aggregation procedure, that is, $\widehat{U}$ cannot report their health data to the SP at time point t_γ, then the SP aggregates the received health data $\widehat{A}_\gamma$, and issues $\widehat{A}_\gamma$ to the CS.

$$\begin{aligned} \widehat{A}_\gamma &= \prod_{U_i \in \mathbb{U}/\widehat{U}} C_{i,\gamma} \cdot \mathrm{g}^{\widetilde{m}} \\ &= \prod_{U_i \in \mathbb{U}/\widehat{U}} (\mathrm{g}^{m_{i,\gamma}} \cdot h^{\theta_\gamma \cdot r_{i,\gamma}}) \cdot \mathrm{g}^{\widetilde{m}} \\ &= \mathrm{g}^{\sum_{U_i \in \mathbb{U}/\widehat{U}} m_{i,\gamma} + \widetilde{m}} \cdot h^{\sum_{U_i \in \mathbb{U}/\widehat{U}} (\theta_\gamma \cdot r_{i,\gamma})}. \end{aligned} \quad (12)$$

As described in the secure report reading section, $d+1$ working CSs can recover the aggregated data $\sum_{U_i \in \mathbb{U}/\widehat{U}} m_{i,\gamma} + \widetilde{m}$.

D. Temporal aggregation handling

In the WBAN, spacial aggregation is defined as the aggregation of different users' health data at the same time point, and has been applied to forecasting, fraud detection and so on [17]. Nevertheless, we sometimes also need to aggregate the same user's health data at different time points, i.e., monitoring personal blood pressure or physical condition. This kind of aggregation is named "temporal aggregation". Obviously, we should also preserve user's privacy and our proposed scheme can be easily extended to support privacy-preserving temporal aggregation.

We suppose that a temporal aggregation cycle including z time points, e.g. from t_1 to t_z, from t_{z+1} to t_{2z}, etc. For simplicity, we just take the first cycle as an example to illustrate the temporal aggregation below.

In our proposed scheme, the form of each user U_i's encrypted health data is $C_{i,\gamma} = \mathrm{g}^{m_{i,\gamma}} \cdot h^{\theta_\gamma \cdot r_{i,\gamma}}$ at time point t_γ. For temporal aggregation, the SP aggregates z ciphertexts as below.

$$\begin{aligned} \widetilde{C}_i &= \prod_{\gamma=1}^{z} C_{i,\gamma} = \prod_{\gamma=1}^{z} (\mathrm{g}^{m_{i,\gamma}} \cdot h^{\theta_\gamma \cdot r_{i,\gamma}}) \\ &= \mathrm{g}^{\sum_{\gamma=1}^{z} m_{i,\gamma}} \cdot h^{\sum_{\gamma=1}^{z} (\theta_\gamma \cdot r_{i,\gamma})}. \end{aligned} \quad (13)$$

Subsequently, the SP will issue $\widetilde{C}_i$ to CSs for further computation. After receiving $\widetilde{C}_i$, $d+1$ working CSs $\wp \subset \mathbb{S}$ will first calculate their decryption shares $D_{j,i}$ respectively, and then aggregate them to obtain

$$\begin{aligned} \widetilde{P}_i &= \prod_{S_j \in \wp} D_{j,i} = \prod_{S_j \in \wp} \widetilde{C}_i^{\beta_j F(j)} = \widetilde{C}_i^{p} \\ &= (\mathrm{g}^{\sum_{\gamma=1}^{z} m_{i,\gamma}})^p \cdot (h^{\sum_{\gamma=1}^{z} (\theta_\gamma \cdot r^{i,\gamma})})^p \\ &= (\mathrm{g}^p)^{\sum_{\gamma=1}^{z} m_{i,\gamma}} = \widehat{\mathrm{g}}^{\sum_{\gamma=1}^{z} m_{i,\gamma}}. \end{aligned} \quad (14)$$

we have $\sum_{\gamma=1}^{z} m_{i,\gamma} \leq z(2^w - 1)$, as $m_{i,\gamma} \in \{0, 1, 2, \ldots, 2^w - 1\}$. By computing the discrete logarithm of $\widetilde{P}_i$ base $\widehat{\mathrm{g}}$, the CS can get the sum of U_i' health data $M_{i,sum} = \sum_{\gamma=1}^{z} m_{i,\gamma}$ in expected time $O(\sqrt{z(2^w - 1)})$ using Pollard's lambda method. Therefore, our scheme can support spacial and temporal privacy-preserving aggregation simultaneously.

E. Adapting to dynamic users

In our proposed scheme, users may move frequently and change dynamically, i.e., mobile user may be a patient, who is equipped with some body area sensors and walks on the road, the mobile user set $\mathbb{U}$ sometimes may revoke an old user and add a new user. Then, we will show that our scheme is adaptive for dynamic users, especially, it is very suitable for user removal and addition.

If a set of old users $U_r \subset \mathbb{U}$ are removed from the system and a set of new users U_a are added in privacy-preserving report aggregation procedure, that is, U_r may died or recovered and U_a may register into the system at time point t_γ. Then, the SP aggregates the received health data $\widetilde{A}_\gamma$, and issues $\widetilde{A}_\gamma$ to the CS.

$$\begin{aligned} \widetilde{A}_\gamma &= \prod_{U_i \in (\mathbb{U}/U_r) \cup U_a} C_{i,\gamma} \cdot \mathrm{g}^{\widetilde{m}} \\ &= \prod_{U_i \in (\mathbb{U}/U_r) \cup U_a} (\mathrm{g}^{m_{i,\gamma}} \cdot h^{\theta_\gamma \cdot r_{i,\gamma}}) \cdot \mathrm{g}^{\widetilde{m}} \\ &= \mathrm{g}^{\sum_{U_i \in (\mathbb{U}/U_r) \cup U_a} m_{i,\gamma} + \widetilde{m}} \cdot h^{\sum_{U_i \in (\mathbb{U}/U_r) \cup U_a} (\theta_\gamma \cdot r_{i,\gamma})}. \end{aligned} \quad (15)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2015.2472369, IEEE Transactions on Information Forensics and Security

As described in secure report reading part, finally, $d+1$ working CSs can recover the aggregated data $\sum_{U_i \in (\mathbb{U}/U_r) \cup U_a} m_{i,\gamma} + \widetilde{m}$.

From above we can conclude that our proposed scheme can well aggregate the statistics of all users' health data at the current time point t_γ, no matter how frequently users revoke or add. In addition, after a new user registering into the system, he only need to report the health data to SP through WiFi, rather than doing some update operations [17]. Therefore, our scheme is adaptive for dynamic users, especially, it is very suitable for user removal and addition.

V. MHDA$^{\oplus}$: Multifunctional health data non-additive aggregation

MHDA^{+} is unable to be directly applied to non-additive aggregation, for example min/max, median, percentile, histogram, etc, which is widely applied in reality. In this section, we propose MHDA$^{\oplus}$ as an extension of MHDA^{+} to support non-additive aggregation.

A. Basic Idea

According to the observation that all the above non-additive aggregations are closely connected with COUNT aggregation which queries the number of users whose values fall within a certain range. Specially, let COUNT(Q)=COUNT([ran$_1$,ran$_2$]) be the number of users whose values fall within the range Q. TOP and BOT represent the upper bound and lower bound of all health data that a body area sensor may receive, respectively. We suppose that each health data value m_i is an integer between [BOT, TOP]=[0, 2^w-1]. We also denote the min, max, median, and σ-percentile of a data set by m_{min}, m_{max}, m_{med}, $m_{\sigma-per}$, respectively. It is easy to see that the following conditions hold.

- Min:
$$\begin{cases} COUNT([BOT, m_{min}-1]) = 0, \\ COUNT([m_{min}, m_{min}]) > 0. \end{cases} \tag{16}$$
- Max:
$$\begin{cases} COUNT([m_{max}+1, TOP]) = 0, \\ COUNT([m_{max}, m_{max}]) > 0. \end{cases} \tag{17}$$
- Median:
  - If n is odd, then
  $$\begin{cases} COUNT([BOT, m_{med}]) \geq (n+1)/2, \\ COUNT([m_{med}, TOP]) \geq (n+1)/2. \end{cases} \tag{18}$$
  - If n is even, then there exists $i, j \in \mathbb{U}$, such that $m_i \leq m_j$ and
  $$\begin{cases} COUNT([BOT, m_i]) \geq n/2, \\ COUNT([BOT, m_i-1]) < n/2, \\ COUNT([m_j, TOP]) \geq n/2, \\ COUNT([m_j+1, TOP]) < n/2, \end{cases} \tag{19}$$
  and $m_{med} = (m_i + m_j)/2$.
- $\sigma - percentile$: we only display the simplest case here
$$\begin{cases} COUNT([BOT, m_{\sigma-per}]) \geq \lfloor \sigma n/100 \rfloor, \\ COUNT([m_{\sigma-per}, TOP]) \geq \lfloor (100-\sigma) n/100 \rfloor. \end{cases} \tag{20}$$

As a result, we have $m^* = m_{min}$ (respectively, m_{max}, m_{med}, $m_{\sigma-per}$), if we can find m^* such that the conditions in Eq. (16) (respectively, (17), (18), (19) (20)) hold. Based on the above observation, MHDA$^{\oplus}$ combines the prefix membership verification scheme with binary search to realize non-additive aggregation functions.

We adopt the prefix membership verification scheme [27]–[30] as the building block of our MHDA$^{\oplus}$. The prefix membership verification scheme's basic idea is to transform the question of verifying whether a datum belongs to a range into a few issues of verifying whether two numerical value are equal. For a w-bit number m, its k-binary-prefix is defined as $\{0,1\}^k\{*\}^{w-k}$, which has 1s or 0s at the first $k \leq w$ bits, and after that is $w-k$ $*$s. Specially, $11**$ is a 2-prefix and it represents the range $[1100, 1111]$.

We denote the prefix family of m as $F(m)$. For a w-bit number $b_1 b_2 \dots b_w$, which can be converted it into a prefix family, and the prefix family including $w+1$ prefixes $\{b_1 b_2 \dots b_w, b_1 b_2 \dots b_{w-1}*, \dots, b_1 * \dots *, ** \dots *\}$, where $b_1 b_2 \dots b_{w-i+1} * \dots *$ is denoted the i-th prefix, each prefix represents a range including the number m and following the binary prefix format. Specially, the prefix family of datum 12 is defined as $F(12) = F(1100) = \{1100, 110*, 11**, 1***, ****\}$.

A range $[ran_1, ran_2]$ can also be converted into a minimum set of prefixes, which is represented as $R([ran_1, ran_2])$, and joining these prefixes together will recover $[ran_1, ran_2]$. Actually, each prefix indicates a subrange of $[ran_1, ran_2]$, and all subranges follow the binary prefix format. For example, $R([9,14]) = \{1001, 101*, 110*, 1110\}$.

If we want to verify whether m belongs to a range $[ran_1, ran_2]$, the following step is needed. At first, we transform the number m into the prefix family $F(m)$. Subsequently, we translate the range $[ran_1, ran_2]$ into a minimum set of prefixes $R([ran_1, ran_2])$. Finally, $m \in [ran_1, ran_2]$ if and only if $F(m) \cap R([ran_1, ran_2]) \neq \emptyset$.

Verifying whether $F(m) \cap R([ran_1, ran_2]) \neq \emptyset$ can be verified by comparing whether two numerical value are equal. The prefix numericalization scheme proposed in [32] is utilized to transform each prefix into an unique binary number. The process of prefix numericalization is that given a w-bit prefix $b_1 b_2 \dots b_k * \dots *$, a bit 1 is inserted after b_k, then each $*$ is replaced by 0. For a set of prefixes R, $\Gamma(R)$ represents the result set of numericalized prefixes. For instance, $\Gamma(F(12)) = \{11001, 11010, 11100, 11000, 10000\}$, and $\Gamma(R([9,14])) = \{10011, 10110, 11010, 11101\}$. Therefore, $m \in [ran_1, ran_2]$ if and only if $\Gamma(F(m)) \cap \Gamma(R([ran_1, ran_2])) \neq \emptyset$. For example, as $\Gamma(F(12)) \cap \Gamma(R([9,14])) = 11010$, we infer that $12 \in [9,14]$.

B. The Basic Scheme Without Differential Privacy

After receiving a non-additive aggregation request, the SP will translate it into several COUNT queries with ranges

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2015.2472369, IEEE Transactions on Information Forensics and Security

$Q_1, Q_2, \ldots$, until the desired m^* is found. Each COUNT query with range Q_x asks for the number of users whose data fall within the range Q_x and the result of the previous COUNT query with range Q_{x-1} will determine the next range Q_x. For user i possessing data m_i, if m_i falls within range Q_x, the SP will receive an answer "yes", which is denoted by a bit of value 1; otherwise, the SP will receive an answer "no" and it is represented by 0.

The processes of achieving privacy-preserving min/max, median, percentile and histogram aggregation are described below, under the assumption that m_i is an integer between [BOT, TOP]=[0, 2^w-1]. Obviously, our method can be extended to other non-additive aggregation easily.

1) Min/Max: As max is opposite to min, we only illustrate min for simplicity. Let $\mathbb{M} = \{m_1, m_2, \ldots, m_n\}$ denote all n users' private health data, and the values of BOT and TOP are known to the TA. The main idea of our privacy-preserving min aggregation protocol is as follows:

(1) As shown in the user report generation procedure, each user $U_i \in \mathbb{U}$ computes $\Gamma(F(m_i))$ and $E(\Gamma(F(m_i))) = \mathrm{g}^{\Gamma(F(m_i))} \cdot h^{\theta \cdot r_i}$. Then, U_i reports $E(\Gamma(F(m_i)))$ to the SP through WiFi. Eventually, $E(\Gamma(F(m_i)))$ is stored in the CS.

(2) As shown in the secure report reading procedure, upon receiving all n encrypted data $E(\Gamma(F(m_i)))$ for $i = 1, 2, \ldots, n$, $d+1$ working cloud servers $\wp \subset \mathbb{S}$ are randomly chosen to decrypt the encrypted data. One of the $d+1$ working cloud servers computes

$$P(E(\Gamma(F(m_i)))) = (\mathrm{g}^p)^{\Gamma(F(m_i))} = \widehat{\mathrm{g}}^{\Gamma(F(m_i))}, \qquad (21)$$

where $\widehat{\mathrm{g}} = \mathrm{g}^p$. Then, this cloud server could carry out the algorithm 1 to acquire m_{min}, where $m_{min} = min\{m_1, m_2, \ldots, m_n\}$. Eventually, this CS reports m_{min} to the TA.

Algorithm 1 The algorithm of basic min aggregation

Input: $P(E(\Gamma(F(m_i)))), i \in \{1, 2, \ldots, n\}$
Output: m_{min}
$BOT = 0, TOP = 2^w - 1, Mid = \lfloor \frac{BOT+TOP}{2} \rfloor$;
for $j = 1$ to w **do**
  $\xi_j = 0$;
  for $i = 1$ to n **do**
    if $P(E(\Gamma(F(m_i)))) \cap P(E(\Gamma(R([BOT, Mid])))) \neq \emptyset$ **then**
      $\theta_i = 1$;
    else
      $\theta_i = 0$;
    end if
    $\xi_j = \xi_j + \theta_i$;
  end for
  if $\xi_j \geq 1$ **then**
    $TOP = Mid, Mid = \lfloor \frac{BOT+TOP}{2} \rfloor$;
  else
    $BOT = Mid + 1, Mid = \lfloor \frac{BOT+TOP}{2} \rfloor$;
  end if
  if $BOT = TOP = Mid$ **or** $j = w$ **then**
    $m_{min} = BOT = TOP = Mid$;
    break;
  end if
end for
return m_{min};
End Procedure;

(3) The key idea of algorithm 1 is as follows. After receiving a min aggregation query, the CS first issues a COUNT query with $Q_1 = [BOT, 2^{w-1} - 1]$ and then counts the number of "yes" answers, represented by ξ_1. If $\xi_1 \geq 1$, the minimum number falls within $[BOT, 2^{w-1} - 1]$, thus the CS will present a new COUNT with $Q_2 = [BOT, 2^{w-2} - 1]$; otherwise, the minimum number falls within $[2^{w-1}, TOP]$, thus the CS will request a new COUNT with $Q_2 = [2^{w-1}, 2^{w-1} + 2^{w-2} - 1]$. For each additional COUNT query, the suspicion range where the minimum number falls is reduced by half. The above procedure runs until the suspicion range is reduced to one, then this minimum number is equal to suspicion range's lower bound or upper bound, and the number of users with the minimum number is equal to the last COUNT query result.

2) Median/Percentile: Due to the space limitation, we only describe median aggregation here. The method can be easily generalized to percentile. The method to achieve median aggregation is similar as min aggregation. Furthermore, for simplicity, we assume that n is odd, and the case of n being even can be easily extended from that of n being odd.

The main idea of median aggregation is similar to min aggregation. After receiving a median aggregation query, the CS first requests a COUNT query with $Q_1 = [BOT, 2^{w-1} - 1]$ and receives ξ_1. If $\xi_1 \geq (n+1)/2$, the median value falls within $[BOT, 2^{w-1} - 1]$, thus the CS will present a new COUNT with $Q_2 = [BOT, 2^{w-2} - 1]$; otherwise, the CS will issue the new COUNT with $Q_2 = [BOT, 2^{w-1} + 2^{w-2} - 1]$. The above procedure runs until the suspicion range of m_{med} is reduced to one, which takes w queries in total. We assume that $Q_{w-1} = [BOT, q_{w-1}]$ and $Q_w = [BOT, q_w]$ are the last two queries, and the CS obtains ξ_{w-1} and ξ_w accordingly. As a result, q_{w-1} and q_w differ by one, and there are four cases shown below.

- Case 1: if $q_{w-1} < q_w$ and $\xi_w \geq (n+1)/2$, we obtain $m_{med} = q_w$. The reason of Case 1 is shown below. Firstly, it must hold that $\xi_{w-1} < (n+1)/2$, as otherwise $m_{med} = q_{w-1}$, and $Q_w = [BOT, q_w]$ should not be queried. Secondly, according to the property of binary search, there must be a COUNT query $Q_x = [BOT, q_w + 1]$ with $x \in [1, w-2]$. Thirdly, we must have $\xi_x < (n+1)/2$, as otherwise $m_{med} < q_{w-1}$ and neither q_{w-1} nor q_w should be queried.
- Case 2: if $q_{w-1} > q_w$ and $\xi_w \geq (n+1)/2$, we obtain $m_{med} = q_w$.
- Case 3: if $q_{w-1} < q_w$ and $\xi_w < (n+1)/2$, we obtain $m_{med} = q_w + 1$.
- Case 4: if $q_{w-1} > q_w$ and $\xi_w < (n+1)/2$, we obtain $m_{med} = q_w + 1$.

The reasons of Cases 2∼4 are the same as that of Case 1. In addition, the algorithm of median aggregation is depicted in algorithm 2.

3) Histogram: A histogram is widely used to illustrate the distribution of data in statistics. It represents a frequency distribution, displayed as a bar chart, and describes the frequency of values falling into each class interval. By using some COUNT queries, we can achieve the histogram aggregation directly. In particular, after receiving a histogram aggregation query, the range $[BOT, TOP]$ can be divided into several class intervals based on the aggregation request. Then, for each class interval, the CS can issue a COUNT query. After that, the obtained

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2015.2472369, IEEE Transactions on Information Forensics and Security

Algorithm 2 The algorithm of basic median aggregation

Input: $P(E(\Gamma(F(m_i))))$, $i \in \{1,2,\ldots,n\}$, n is odd
Output: m_{med}
$BOT = 0, TOP = 2^w - 1, Mid_0 = \lfloor \frac{BOT+TOP}{2} \rfloor$;
for $j = 1$ to w **do**
  $\xi_j = 0$;
  for $i = 1$ to n **do**
    if $P(E(\Gamma(F(m_i)))) \cap P(E(\Gamma(R([BOT, Mid_{j-1}])))) \neq \emptyset$ **then**
      $\theta_i = 1$;
    else
      $\theta_i = 0$;
    end if
    $\xi_j = \xi_j + \theta_i$;
  end for
  if $\xi_j \geq \frac{n+1}{2}$ **then**
    if $Mid_{j-1} \leq \lfloor \frac{0+2^w-1}{2} \rfloor$ **then**
      $TOP = Mid_{j-1}, Mid_j = \lfloor \frac{BOT+TOP}{2} \rfloor$;
    else
      $BOT = 2Mid_{j-1} - TOP + 1, TOP = Mid_{j-1}$;
      $Mid_j = \lfloor \frac{BOT+TOP}{2} \rfloor, BOT = 0$;
    end if
  else
    $Mid_j = \lfloor \frac{Mid_{j-1}+1+TOP}{2} \rfloor$;
  end if
  if $j = w$ **then**
    if $Mid_{w-1} < Mid_w$ **and** $\xi_w \geq \frac{n+1}{2}$ **then**
      $m_{med} = Mid_{w-1}$; **break**;
    else if $Mid_{w-1} > Mid_w$ **and** $\xi_w \geq \frac{n+1}{2}$ **then**
      $m_{med} = Mid_{w-1}$; **break**;
    else if $Mid_{w-1} < Mid_w$ **and** $\xi_w < \frac{n+1}{2}$ **then**
      $m_{med} = Mid_w$; **break**;
    else if $Mid_{w-1} > Mid_w$ **and** $\xi_w < \frac{n+1}{2}$ **then**
      $m_{med} = Mid_w$; **break**;
    end if
  end if
end for
return m_{med};
End Procedure;

query results are equal to the number of users whose health data fall within the corresponding class interval.

C. The Advanced Scheme With Differential Privacy

In this section, we provide additional protections to the user's health data privacy against differential attack [13], this is an advanced version of MHDA$^{\oplus}$. In our basic scheme, each user's encrypted data is aggregated at the SP, and then SP will report this aggregated data to CSs. Although each user's individual health data cannot be disclosed by CSs or malicious attackers, their private health data are still vulnerable to the differential attack which infringes users' privacy through analyzing the aggregated data. Especially, if an adversary obtains the min aggregations of two data sets D_1 and D_2 differing by at most one element, where $D_1, D_2 \subseteq \mathbb{U}$ and $D_2 = D_1 + U_x$. Let $A_{min}(D)$ denote the min aggregation on data set D, if $A_{min}(D_1) \neq A_{min}(D_2)$, then $A_{min}(D_2)$ is exactly the health data of user U_x. In order to against differential attack, we present an advanced version of MHDA$^{\oplus}$ with differential privacy as below.

1) Min/Max: The same as the basic scheme, we only illustrate min aggregation as an example. The main idea of our advanced min aggregation protocol is as follows:

(1) As shown in the user report generation procedure, each user $U_i \in \mathbb{U}$ computes $\Gamma(F(m_{i,\gamma}))$ and $E(\Gamma(F(m_{i,\gamma}))) = \mathrm{g}^{\Gamma(F(m_{i,\gamma}))} \cdot h^{\theta \cdot r_{i,\gamma}}$. Then, U_i reports $E(\Gamma(F(m_{i,\gamma})))$ to the SP through WiFi.

(2) As shown in the privacy-preserving report aggregation procedure, the SP is able to calculate the aggregation as follows. Assume that $|D|$ represents the number of data items in the data set D, $A_{min}(D) = min\{m_{1,\gamma}\, m_{2,\gamma}\, ..., m_{|D|,\gamma}\}$, then $|A_{min}(D_1) - A_{min}(D_2)| \leq 2^w - 1$ holds for any two data sets D_1 and D_2 differing by at most one element. Therefore, we can obtain the sensitivity of A_{min} by $\Delta A_{min} = 2^w - 1$. The SP uses this sensitivity to randomly pick noises $\widetilde{m}_{i,\gamma}$ from geometric distribution Geom($exp(-\frac{\epsilon}{\Delta A_{min}})$). It also picks a random number $\widetilde{r}_{i,\gamma} \in \mathbb{Z}_N^*$, and then calculates

$$\begin{aligned} \widetilde{E}(\Gamma(F(m_{i,\gamma}))) &= E(\Gamma(F(m_{i,\gamma}))) \cdot \mathrm{g}^{\widetilde{m}_{i,\gamma}} \\ &= \mathrm{g}^{\Gamma(F(m_{i,\gamma}))+\widetilde{m}_{i,\gamma}} \cdot h^{\theta \cdot r_{i,\gamma}}, \end{aligned} \tag{22}$$

and

$$\widetilde{C}_{i,\gamma} = \mathrm{g}^{\widetilde{m}_{i,\gamma}} \cdot h^{\widetilde{r}_{i,\gamma}}. \tag{23}$$

The SP sends $\widetilde{E}(\Gamma(F(m_{i,\gamma})))$ and $\widetilde{C}_{i,\gamma}$ to the CS for further computation.

(3) As shown in the secure report reading procedure, upon receiving all n encrypted data $\widetilde{E}(\Gamma(F(m_{i,\gamma})))$ and $\widetilde{C}_{i,\gamma}$ for $i = 1, 2, \ldots, n$, $d+1$ working cloud servers $\wp \subset \mathbb{S}$ are randomly chosen to decrypt the encrypted data. One of the $d+1$ working cloud servers computes

$$P(\widetilde{E}(\Gamma(F(m_{i,\gamma})))) = (\mathrm{g}^p)^{\Gamma(F(m_{i,\gamma}))+\widetilde{m}_{i,\gamma}} = \widehat{\mathrm{g}}^{\Gamma(F(m_{i,\gamma}))+\widetilde{m}_{i,\gamma}}, \tag{24}$$

$$P(\widetilde{C}_{i,\gamma}) = (\mathrm{g}^p)^{\widetilde{m}_{i,\gamma}} = \widehat{\mathrm{g}}^{\widetilde{m}_{i,\gamma}}, \tag{25}$$

where $\widehat{\mathrm{g}} = \mathrm{g}^p$. Then, this cloud server could carry out the algorithm 3 to acquire $\widetilde{m}_{min,\gamma}$, where $\widetilde{m}_{min,\gamma} = min\{\widetilde{m}_{1,\gamma}, \widetilde{m}_{2,\gamma}, \ldots, \widetilde{m}_{n,\gamma}\}$. Eventually, this CS reports $\widetilde{m}_{min,\gamma}$ to the TA.

2) Median/Percentile: The same as the basic scheme, we only illustrate median aggregation as an example and we also assume that n is odd. The main idea of our advanced median aggregation is similar to our advanced min aggregation.

The procedure of health data aggregation in our advanced median aggregation is almost the same as that in our advanced min aggregation. The only difference is shown below: i) In the privacy-preserving report aggregation procedure, we assume that $|D|$ represents the number of data items in the data set D, $A_{median}(D) = median\{m_{1,\gamma}\, m_{2,\gamma}\, ..., m_{|D|,\gamma}\}$, then $|A_{median}(D_1) - A_{median}(D_2)| \leq 2^w - 1$ holds for any two data sets D_1 and D_2 differing by at most one element. Therefore, we can obtain the sensitivity of A_{median} by $\Delta A_{median} = 2^w - 1$. The SP uses this sensitivity to randomly pick noises $\widetilde{m}_{i,\gamma}$ from geometric distribution Geom($exp(-\frac{\epsilon}{\Delta A_{median}})$). ii) In the secure report reading procedure, one of the $d+1$ working cloud servers carry out the algorithm 4 to acquire $\widetilde{m}_{median,\gamma}$, where $\widetilde{m}_{median,\gamma} = median\{\widetilde{m}_{1,\gamma}, \widetilde{m}_{2,\gamma}, \ldots, \widetilde{m}_{n,\gamma}\}$. Eventually, this CS reports $\widetilde{m}_{median,\gamma}$ to the TA.

3) Histogram: The same as our basic scheme, we can also use some COUNT queries to achieve the histogram aggregation directly. In particular, after receiving a histogram aggregation query, the range $[BOT, TOP]$ can be divided into several class intervals based on the aggregation request. In the privacy-preserving report aggregation procedure, we can

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2015.2472369, IEEE Transactions on Information Forensics and Security

Algorithm 3 The algorithm of advanced min aggregation

Input: $P(\widetilde{E}(\Gamma(F(m_{i,\gamma}))))$, $P(\widetilde{C}_{i,\gamma})$, $i \in \{1,2,\ldots,n\}$
Output: $\widetilde{m}_{min,\gamma}$
$BOT = 0, TOP = 2^w - 1, Mid = \lfloor \frac{BOT+TOP}{2} \rfloor$;
for $j = 1$ to w **do**
  $\xi_j = 0$;
  for $i = 1$ to n **do**
    if $P(\widetilde{E}(\Gamma(F(m_{i,\gamma})))) \cap P(E(\Gamma(R([BOT, Mid])))) \cdot P(\widetilde{C}_{i,\gamma}) \neq \emptyset$ **then**
      $\theta_i = 1$;
    else
      $\theta_i = 0$;
    end if
    $\xi_j = \xi_j + \theta_i$;
  end for
  if $\xi_j \geq 1$ **then**
    $TOP = Mid, Mid = \lfloor \frac{BOT+TOP}{2} \rfloor$;
  else
    $BOT = Mid + 1, Mid = \lfloor \frac{BOT+TOP}{2} \rfloor$;
  end if
  if $BOT = TOP = Mid$ **or** $j = w$ **then**
    $m_{min,\gamma} = BOT = TOP = Mid$;
    break;
  end if
end for
for $i = 1$ to n **do**
  if $P(\widetilde{E}(\Gamma(F(m_{i,\gamma})))) = P(E(\Gamma(F(m_{min,\gamma})))) \cdot P(\widetilde{C}_{i,\gamma})$ **then**
    $\widetilde{m}_{min,\gamma} = m_{min,\gamma} + \widetilde{m}_{i,\gamma}$;
    break;
  end if
end for
return $\widetilde{m}_{min,\gamma}$;
End Procedure;

Algorithm 4 The algorithm of advanced median aggregation

Input: $P(\widetilde{E}(\Gamma(F(m_{i,\gamma}))))$, $P(\widetilde{C}_{i,\gamma})$, $i \in \{1,2,\ldots,n\}$, n is odd
Output: $\widetilde{m}_{median,\gamma}$
$BOT = 0, TOP = 2^w - 1, Mid_0 = \lfloor \frac{BOT+TOP}{2} \rfloor$;
for $j = 1$ to w **do**
  $\xi_j = 0$;
  for $i = 1$ to n **do**
    if $P(\widetilde{E}(\Gamma(F(m_{i,\gamma})))) \cap P(E(\Gamma(R([BOT, Mid_{j-1}])))) \cdot P(\widetilde{C}_{i,\gamma}) \neq \emptyset$ **then**
      $\theta_i = 1$;
    else
      $\theta_i = 0$;
    end if
    $\xi_j = \xi_j + \theta_i$;
  end for
  if $\xi_j \geq \frac{n+1}{2}$ **then**
    if $Mid_{j-1} \leq \lfloor \frac{0+2^w-1}{2} \rfloor$ **then**
      $TOP = Mid_{j-1}, Mid_j = \lfloor \frac{BOT+TOP}{2} \rfloor$;
    else
      $BOT = 2Mid_{j-1} - TOP + 1, TOP = Mid_{j-1}$;
      $Mid_j = \lfloor \frac{BOT+TOP}{2} \rfloor, BOT = 0$;
    end if
  else
    $Mid_j = \lfloor \frac{Mid_{j-1}+1+TOP}{2} \rfloor$;
  end if
  if $j = w$ **then**
    if $Mid_{w-1} < Mid_w$ **and** $\xi_w \geq \frac{n+1}{2}$ **then**
      $m_{median,\gamma} = Mid_{w-1}$; **break**;
    else if $Mid_{w-1} > Mid_w$ **and** $\xi_w \geq \frac{n+1}{2}$ **then**
      $m_{median,\gamma} = Mid_{w-1}$; **break**;
    else if $Mid_{w-1} < Mid_w$ **and** $\xi_w < \frac{n+1}{2}$ **then**
      $m_{median,\gamma} = Mid_w$; **break**;
    else if $Mid_{w-1} > Mid_w$ **and** $\xi_w < \frac{n+1}{2}$ **then**
      $m_{median,\gamma} = Mid_w$; **break**;
    end if
  end if
end for
for $i = 1$ to n **do**
  if $P(\widetilde{E}(\Gamma(F(m_{i,\gamma})))) = P(E(\Gamma(F(m_{median,\gamma})))) \cdot P(\widetilde{C}_{i,\gamma})$ **then**
    $\widetilde{m}_{median,\gamma} = m_{median,\gamma} + \widetilde{m}_{i,\gamma}$;
    break;
  end if
end for
return $\widetilde{m}_{median,\gamma}$;
End Procedure;

obtain the sensitivity of $A_{Histogram}$ by $\Delta A_{Histogram} = n$. The SP uses this sensitivity to randomly pick noises $\widetilde{m}_{i,\gamma}$ from geometric distribution Geom($exp(-\frac{\epsilon}{\Delta A_{Histogram}})$). Then, for each class interval $[r_1, r_2]$, the CS can issue a COUNT query, if $P(\widetilde{E}(\Gamma(F(m_{i,\gamma})))) \cap P(E(\Gamma(R([r_1, r_2])))) \cdot P(\widetilde{C}_{i,\gamma}) \neq \emptyset$ is held, we can learn that $m_{i,\gamma} \in [r_1, r_2]$. After that, the obtained query results are equal to the number of users whose health data fall within the corresponding class interval.

VI. Security analysis

As stated in II(C), in order to resist $\mathscr{A}$'s objective, the security requirements should be satisfied. In this section, some security issues involved in our proposed scheme will be analyzed. Particularly, to preserve users' private data from a strong adversary $\mathscr{A}$.

- *The user's privacy is protected against eavesdropping.* As described in II(B), $\mathscr{A}$ may eavesdrop on communication flows from users to the SP. However, this method is impractical for $\mathscr{A}$, as the mobile users in WBANs are dynamic. In addition, even if $\mathscr{A}$ can eavesdrop on user U_i's ciphertext at the time point t_γ, such as $g^{m_{i,\gamma}} \cdot h^{\theta_\gamma \cdot r_{i,\gamma}}$ or $g^{\Gamma(F(m_{i,\gamma}))} \cdot h^{\theta_\gamma \cdot r_{i,\gamma}}$, as $h^{\theta_\gamma \cdot r_{i,\gamma}}$ is transparent to $\mathscr{A}$, it is impossible for $\mathscr{A}$ to obtain the private data $m_{i,\gamma}$. Therefore, $\mathscr{A}$ cannot expose users' private data, even if it can eavesdrop on communication flows.
- *The uncompromised users' privacy will not be exposed.* In our adversary model, $\mathscr{A}$ can compromise some of the mobile users and it can breach mobile users' privacy. However, $\mathscr{A}$ would be unlikely to choose this approach, as there are a large number of users in WBANs. In addition, by using the private data it acquired from the compromised users, such as the private health data and their private keys, $\mathscr{A}$ may try to expose the uncompromised users' private data. Nevertheless, since each user's private key is generated independently and learning one user's private key exposes nothing about others, this attack won't succeed. Especially, as the sum of all users' private keys is transparent to $\mathscr{A}$, even though $\mathscr{A}$ can learn $n-1$ users' private keys and health data, the last user's private key and health data still cannot be exposed. Therefore, the privacy of uncompromised users is guaranteed.
- *The users' private data and aggregated data will not be revealed in the SP.* For the additive aggregation scheme, the SP can collect all users' ciphertext and obtain the aggregated ciphertext at each time point t_γ. Even though $\mathscr{A}$ can install malicious software on the SP, it can only gain all users' ciphertexts and the aggregated ciphertext. As $h^{\theta_\gamma \cdot r_{i,\gamma}}$ is unknown to $\mathscr{A}$ and the SP, any user's private data won't be revealed. Furthermore, the form of aggre-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2015.2472369, IEEE Transactions on Information Forensics and Security

gated ciphertext is $g^{\sum_{i=1}^{n} m_{i,\gamma}} \cdot h^{\sum_{i=1}^{n}(\theta_\gamma \cdot r_{i,\gamma})}$, $\mathscr{A}$ cannot expose all users' sum statistics, since $\sum_{i=1}^{n}(\theta_\gamma \cdot r_{i,\gamma})$ is transparent to $\mathscr{A}$. For the non-additive aggregation scheme, $\mathscr{A}$ can also learn all n users' ciphertexts, i.e., $g^{\Gamma(F(m_i))} \cdot h^{\theta \cdot r_i}$. Then, it may try to utilize algorithm 1 to launch a brute-force attack by exhaustively testing each possible value of m_i. However, $\mathscr{A}$ needs to know $h^{\theta \cdot r_i}$ first, which is impractical as r_i is transparent to $\mathscr{A}$. Therefore, the privacy of each user's data and aggregated data can be guaranteed even though $\mathscr{A}$ can install malicious software on the SP.

- *Even though d CSs are compromised, the adversary cannot acquire users' private data and aggregated data.* In our scheme, $k \geq 3$ CSs are considered to work synergistically, and the TA will issue different private keys $G(j)$, $j = 1, 2, \ldots, k$ to each CS in the system initialization phase. The system can protect users' private data against the adversary, even though no more than $d = \lceil k/2 \rceil - 1$ CSs have malfunctioned or been compromised. In particular, we assume that $\mathscr{A}$ has compromised d CSs and gained their private keys $G(j)$, $j = 1, 2, \ldots, d$. However, $\mathscr{A}$ still cannot obtain the private secret p, since at least $d+1$ CSs are needed to recover p according to the "all or nothing" property of secret sharing [33]. Similarly, in order to decrypt the aggregated data of users, $d+1$ working CSs are necessary to calculate $d+1$ decryption shares $D_{\gamma,j} = A_\gamma^{\beta_j G(j)}$. $\mathscr{A}$ can only obtain d decryption shares, which are insufficient to get $\widetilde{P}_\gamma = \widetilde{A}_\gamma^p$, thus $\mathscr{A}$ cannot obtain users' aggregated data. On the other hand, our system can support fault tolerant of CS failure, so long as the number of compromised CSs is less than d, there are still $k - d \geq d + 1$ working CSs can be used to keep the system working normally. According to the above discussion, even though d CSs have been malfunctioned, the strong adversary still cannot reveal users' private data and aggregated data.

VII. Performance evaluation

In this section, the performance of our proposed PPM-HDA mechanism will be evaluated in comparison to the computational and communication overhead in the wireless body area communication. For the multifunctional health data additive aggregation scheme MHDA$^+$, we compare it with PDAFT [17] and MuDA [22] in the computational and communication overhead. Then, for the multifunctional health data non-additive aggregation scheme MHDA$^\oplus$, we will compare it with EPADA [21], PriSense [19] and VPA [20] in the communication overhead.

A. Computational overhead comparison

We compare the computational overhead of the PPM-HDA with that of PDAFT [17] and MuDA [22]. As PDAFT can only support sum aggregation, we only compare the computational overhead of sum aggregation among them. We consider the computational overhead of these schemes in four aspects, such as that of the individual user, the aggregator (i.e., social spot and gateway), the CSs and the TA. Let T_{exp} represent the time of modular exponential operation in $\mathbb{Z}_{N^2}$, T_{mul} denote the time of modular multiplication and T_{plm} represent the time of using Pollard's lambda method to compute the discrete logarithm.

In the PPM-HDA, each individual user encrypts the health data with 2 modular exponential operations in $\mathbb{Z}_{N^2}$ and one modular multiplication. The computational overhead of each individual user is $2T_{exp} + T_{mul}$ in total. In the privacy-preserving report aggregation phase, the SP calculates the encrypted aggregation A_γ with $n-1$ modular multiplications, which takes $(n-1)T_{mul}$ in total. In addition, we use k CSs to assist in processing and storing the large volume of health data, which is different from PDAFT and MuDA. The computational overhead of CSs will take place in two situations: when the TA wants to acquire the statistics of health data and when it wants to obtain each individual user's health data at one time point. In the former case, each CS $S_j \in \wp$ first computes its decryption share $D_{j,\gamma}$ with one modular exponential operation. Then each CS reports its decryption share to one of these CSs to decrypt the aggregated data by using Pollard's lambda method. Finally, this CS will send the statistics to TA. As $d+1$ working CSs are collaborating to decrypt the aggregated data, thus it will take $(d+1)T_{exp} + dT_{mul} + T_{plm}$. In the latter case, each CS is doing the same as above at first, which takes $(d+1)T_{exp}$. Therefore, the computational overhead of CSs is $(d+1)T_{exp} + dT_{mul} + T_{plm} + (d+1)T_{exp} = 2(d+1)T_{exp} + dT_{mul} + T_{plm}$ in total. Then each CS $S_j \in \wp$ reports its decryption share to TA, and the TA will costs $dT_{mul} + T_{plm}$ to decrypt the aggregated data.

Next, we consider the computational overhead of PDAFT and MuDA. For each individual user and aggregator (i.e., gateway), the computational overhead of both PDAFT and MuDA are the same as our proposed PPM-HDA mechanism, which takes $2T_{exp} + T_{mul}$ and $(n-1)T_{mul}$ respectively. As there are no cloud server in PDAFT and MuDA, we only consider the computational overhead of TA. In the PDAFT, each server $S_j \in \wp$ first calculates its decryption share $D_{j,\gamma}$ with one modular multiplication and 2 modular exponential operations in the control center. Then each server reports its decryption share to one of these servers to decrypt the aggregated data by Paillier decryption. As $d+1$ working servers are collaborating to decrypt the aggregated data, it will take $(T_{mul}+2T_{exp})(d+1)+dT_{mul}+2T_{mul} = 2(d+1)T_{exp}+(2d+3)T_{mul}$. As we take the TA wants to acquire the statistics of health data and obtain each individual user's health data at one time point into consideration, the computational overhead of TA is $4(d+1)T_{exp}+2(2d+3)T_{mul}$ in total. In the MuDA, the TA possesses the private key p. In the secure report reading phase, the TA only need to compute $(A_{1,\gamma})^p$ to decrypt the aggregated data by using Pollard's lambda method. Therefore, the computational overhead of TA is $2T_{exp} + 2T_{plm}$ in total.

The computational overhead of PDAFT, MuDA and PPM-HDA is depicted in Table I. Furthermore, we adopt OpenSSL Library [35] running on a 3.0GHz-processor 2GB-memory computing machine to evaluate the computational overhead of operations. The experiments are based on some assumptions, such as we choose the security parameter $\tau = 512$, the bit length $w = 13$ and each CS can provide services for 2500

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2015.2472369, IEEE Transactions on Information Forensics and Security

TABLE I: The comparison of computational overhead

	PDAFT [17]	**MuDA [22]**	**Our scheme**
Individual user	$2T_{exp} + T_{mul}$	$2T_{exp} + T_{mul}$	$2T_{exp} + T_{mul}$
Aggregator	$(n-1)T_{mul}$	$(n-1)T_{mul}$	$(n-1)T_{mul}$
CSs	N/A	N/A	$2(d+1)T_{exp} + dT_{mul} + T_{plm}$
TA	$4(d+1)T_{exp} + 2(2d+3)T_{mul}$	$2T_{exp} + 2T_{plm}$	$dT_{mul} + T_{plm}$

TABLE II: The experimental results of T_{plm} for different n when $\tau = 512$, $w = 13$

n	10000	20000	30000	40000	50000	60000	70000	80000	90000	100000
d	2	4	6	8	10	12	14	16	18	20
$\boldsymbol{T_{plm}}$ **(μs)**	43539.0	61207.5	75135.7	86843.0	96805.8	105663.6	115128.4	122052.8	130337.6	137211.7
$\frac{T_{plm}}{\sqrt{n(2^w-1)}}$	4.81	4.78	4.79	4.80	4.78	4.77	4.81	4.77	4.80	4.79

mobile users which means $d = \lceil n/2500/2 \rceil$. The experimental results indicate that $T_{exp} = 9387.18us$, $T_{mul} = 16.48us$ and T_{plm} is depicted in Table II. From Table II, it's easy to see that the expected time of Pollard's lambda algorithm in proportion to $\sqrt{n(2^w - 1)}$. It agrees with the fact that CS can get the sum of users' health data in expected time $O(\sqrt{n(2^w - 1)})$ using Pollard's lambda method in the secure report reading procedure. After obtaining the computational overheads of T_{exp}, T_{mul} and T_{plm}, we depict the variation of computational overheads in terms of n in Fig. 2. As shown in Fig. 2, the computational overhead of CSs+TA in our scheme is higher than that in MuDA. It is slightly higher than that in PDAFT when $n < 40000$, but it is lower than that in PDAFT when $n > 40000$. However, the computational overhead of TA in our scheme is lower than that in PDAFT and MuDA. Therefore, it is obvious that the computational overhead of TA in our scheme is significantly reduced with the assistance of CSs.

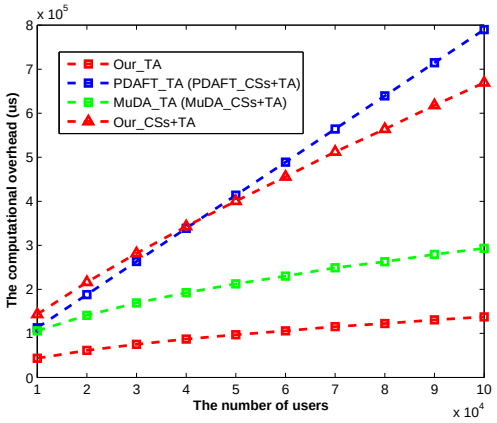

Fig. 2: The variation of the computational overhead of TA and CSs+TA in PDAFT, MuDA and ours in terms of n, when $\tau = 512$, $w = 13$.

B. Communication overhead comparison

The PDAFT and MuDA can only support additive aggregation, which has been researched and compared in many existing papers [16]–[20], [22], [34]. In addition, for PDAFT and MuDA, the communication overhead of additive aggregation are intuitively the same as the PPM-HDA. Especially, each user will report its encrypted data to the aggregator in the user report generation phase. After the aggregator obtains the aggregated data, it will send this aggregated data to the control center or CSs in the privacy-preserving report aggregation phase. As the ciphertext size of PPM-HDA is equal to that of PDAFT and MuDA, the communication overhead of PPM-HDA is the same as that of PDAFT and MuDA for additive aggregation. Therefore, in this subsection, we focus on the communication overhead of the non-additive aggregation scheme.

The communication overhead of the proposed non-additive aggregation scheme MHDA$^{\oplus}$ can be considered in the communication of individual users, where each user generates its own health data report and delivers it to the SP. As the EPADA scheme proposed in [21] can only support min/max aggregation, we only compare the communication overhead of min/max aggregation among EPADA, PriSense, VPA and ours. Our min/max aggregation without differential privacy is regarded as the basic scheme, and our min/max aggregation with differential privacy is regarded as the advanced scheme in Table III. In our basic and advanced scheme, the size of U_i's report is $|E(\Gamma(F(m_i)))| = 2\tau(w+1)$ bits if we choose the security parameter τ. Note that we do not consider other payloads such as user ID and timestamp, which are relatively short compared to the report. In EPADA basic and advanced scheme [21], the ciphertext size of min/max aggregation is $(\Delta+1)\lceil log(n+1) \rceil$ and $2^{\varepsilon-1}(log\Delta + 2)$, respectively, in which n denotes the number of mobile users, Δ represents the maximum value of any user's health data and $\frac{1}{2^{\varepsilon}}$ is defined as the upper bound of the relative error in [21]. Since $\Delta = 2^w$ in our proposed non-additive aggregation scheme, the ciphertext size of min/max aggregation in EPADA basic and advanced scheme is $(2^w+1)\lceil log(n+1) \rceil$ and $2^{\varepsilon-1}(w+2)$, respectively. In PriSense and VPA, the size of U_i's report is the same as ours, but they need $log(\Delta) = w$ rounds of communications, thus the ciphertext size of min/max aggregation is $2\tau w(w+1)$.

In EPADA scheme [21], the upper bound of relative error is $\frac{1}{2^{\varepsilon}}$. In our advanced scheme, we assume that a noise $\widetilde{m}_{i,\gamma}$ is introduced to the exact aggregation $A_{i,\gamma}$, and we will obtain the perturbed aggregation $\widetilde{A}_{i,\gamma}$. Therefore, the relative error

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2015.2472369, IEEE Transactions on Information Forensics and Security

TABLE III: The comparison of relative error and communication overhead

	EPADA basic scheme [21]	EPADA advanced scheme [21]	PriSense [19]	VPA [20]	Our basic scheme	Our advanced scheme
Relative error	0	$\leq \frac{1}{2^{\varepsilon}}$	0	0	0	$\frac{2exp(-\frac{\epsilon}{\Delta A_{i,\gamma}})}{A_{i,\gamma}(1-exp(-\frac{2\epsilon}{\Delta A_{i,\gamma}}))}$
Ciphertext size (bit)	$(2^w+1)\lceil log(n+1)\rceil$	$2^{\varepsilon-1}(w+2)$	$2\tau w(w+1)$	$2\tau w(w+1)$	$2\tau(w+1)$	$2\tau(w+1)$

$\zeta_{i,\gamma} = \frac{|\widetilde{A}_{i,\gamma} - A_{i,\gamma}|}{A_{i,\gamma}}$ is calculated as follows.

$$\begin{aligned}\mathbb{E}(\zeta_{i,\gamma}) &= \frac{\mathbb{E}|\widetilde{A}_{i,\gamma} - A_{i,\gamma}|}{A_{i,\gamma}} \\ &= \frac{\mathbb{E}|\widetilde{m}_{i,\gamma}|}{A_{i,\gamma}},\end{aligned} \tag{26}$$

where

$$\begin{aligned}\mathbb{E}|\widetilde{m}_{i,\gamma}| &= \sum_{x=-\infty}^{\infty} |x| \cdot Pr[X=x] \\ &= \sum_{x=-\infty}^{\infty} |x| \cdot \frac{1-\alpha}{1+\alpha}\alpha^{|x|} \\ &= \frac{2}{1+\alpha} \cdot \sum_{x=1}^{\infty} x(1-\alpha)\cdot\alpha^x \\ &= \frac{2}{1+\alpha} \cdot (\sum_{x=1}^{\infty} x\cdot\alpha^x - \sum_{x=1}^{\infty} x\cdot\alpha^{x+1}) \\ &= \frac{2}{1+\alpha} \cdot \sum_{x=1}^{\infty} \alpha^x \\ &= \frac{2}{1+\alpha} \cdot \frac{\alpha}{1-\alpha} \qquad (\because 0<\alpha<1) \\ &= \frac{2\alpha}{1-\alpha^2}.\end{aligned} \tag{27}$$

As shown in III(B), $\alpha = exp(-\frac{\epsilon}{\Delta A_{i,\gamma}})$, therefore, the mean relative error of our advanced scheme is $\mathbb{E}(\zeta_{i,\gamma}) = \frac{\mathbb{E}|\widetilde{m}_{i,\gamma}|}{A_{i,\gamma}} = \frac{2exp(-\frac{\epsilon}{\Delta A_{i,\gamma}})}{A_{i,\gamma}(1-exp(-\frac{2\epsilon}{\Delta A_{i,\gamma}}))}$. The comparison of EPADA, PriSense, VPA and ours is described in Table III. We first compare the individual communication overhead of EPADA, PriSense, VPA and ours. We choose the security parameter $\tau = 512$. At first, we assume that the maximum value of any user's health data is required to be lower than 2^{14} in the WBAN and $\varepsilon = 11$. We vary n from $\{5000, 10000, 15000, 20000, 25000, 30000, 35000, 40000\}$ and depict the variation of computational overhead of EPADA, PriSense, VPA and ours in terms of n. Secondly, we assume that there are 20000 users in the WBAN and $\varepsilon = 11$. We vary w from $\{12, 13, 14, 15, 16, 17\}$ and depict the variation of computational overhead of EPADA, PriSense, VPA and ours in terms of w. Finally, we assume that $w = 17$, $n = 20000$ and we vary ε from $\{7, 8, 9, 10, 11, 12, 13, 14, 15\}$. We depict the variation of the communication overhead of EPADA advanced scheme, PriSense, VPA and ours in terms of ε and depict the variation of the relative error of EPADA advanced scheme in terms of ε.

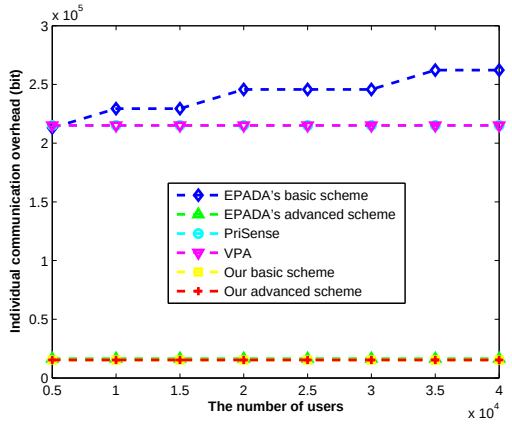

Fig. 3: The variation of the individual communication overhead of EPADA, PriSense, VPA and ours in terms of n, when $\tau = 512$, $w = 10$, $\varepsilon = 11$.

Fig. 3 demonstrates that the individual communication overhead of ours is significantly lower than that of EPADA basic scheme, PriSense and VPA and it is slightly lower than that of EPADA advanced scheme when $\tau = 512$, $w = 10$, $\varepsilon = 11$. In addition, as n increases, the gap of communication overhead between EPADA basic scheme and ours increases, since our scheme's communication overhead is not related with n. With the increase of n, our scheme's communication overhead is invariable, which means that our scheme is scalable and can be applied to scenarios with large number of users.

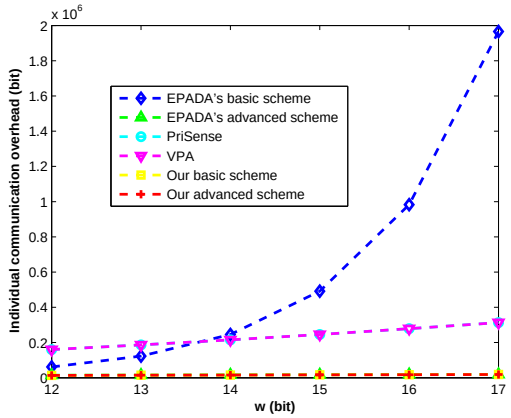

Fig. 4: The variation of the individual communication overhead of EPADA, PriSense, VPA and ours in terms of w, when $\tau = 512$, $n = 20000$, $\varepsilon = 11$.

Fig. 4 shows that the individual communication overhead of ours is lower than that of EPADA basic scheme, PriSense and VPA and it is slightly lower than that of EPADA advanced scheme when $\tau = 512$, $n = 20000$, $\varepsilon = 11$. With the increase of w, the communication overhead of EPADA basic scheme

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2015.2472369, IEEE Transactions on Information Forensics and Security

increases exponentially and it will exceed that of PriSense and VPA when the plaintext space is greater than 2^{14} bits. In addition, as the plaintext space continues to increase, the gap of communication overhead between EPADA basic scheme and ours increases exponentially, which implies that EPADA basic scheme cannot support large plaintext space. In our scheme, the communication overhead increases steadily with the increase of w. Therefore, our scheme is suitable for large plaintext space.

Fig. 5 shows the variation of the individual communication overhead of EPADA advanced scheme, PriSense, VPA and ours, and the relative error of EPADA advanced scheme (green dashed line) all in terms of ε, when $\tau = 512$, $w = 17$, and $n = 20000$. Fig. 5 reveals that as ε increases, the communication overhead of EPADA advanced scheme increases exponentially because its communication overhead is of exponential relation with ε as described in Table III. The communication overhead of other schemes are stable. It can be see that our scheme has much lower communication overhead than PriSense and VPA. The communication overhead of our scheme is also much lower than the EPADA advanced scheme when $\varepsilon > 11$. When $\varepsilon < 11$, our communication overhead is a little higher than it. However, when ε is small, the relative error of EPADA advanced scheme is high as shown by the green dashed line, which is unacceptable for applications requiring highly-accurate data.

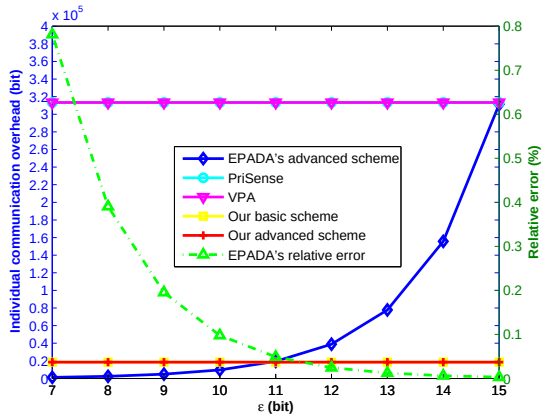

Fig. 5: The variation of the individual communication overhead of EPADA advanced scheme, PriSense, VPA and ours, the relative error of EPADA advanced scheme, all in terms of ε, when $\tau = 512$, $w = 17$, $n = 20000$.

Next, we compare the relative error of EPADA advanced scheme and our advanced scheme. As EPADA scheme can only support min/max aggregation, we only compare the relative error of min aggregation between EPADA advanced scheme and our advanced scheme. In the simulation, we also choose the security parameter $\tau = 512$. We assume that the privacy parameter ϵ is selected from $\{0.1, 0.2, 0.3, 0.4\}$ and $\varepsilon = \lfloor \frac{2}{5} w \rfloor$. We vary w from $\{12, 13, 14, 15, 16, 17, 18, 19\}$ and depict the variation of relative error of EPADA advanced scheme and our advanced scheme in terms of w.

Fig. 6 shows that as ϵ increases, the relative error of our advanced scheme will decrease and the relative error of our advanced scheme is approximately 0.125% when $\epsilon = 0.4$.

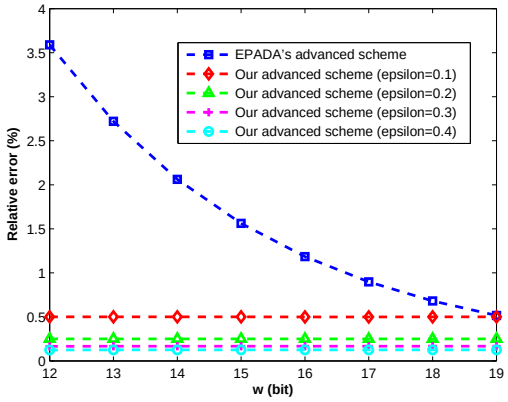

Fig. 6: The variation of the relative error of EPADA advanced scheme and our advanced scheme in terms of w, when $\tau = 512$, $\varepsilon = \lfloor \frac{2}{5} w \rfloor$.

In addition, with the increase of w, the relative error of EPADA advanced scheme decreases exponentially and the relative error of our advanced scheme is lower than that of EPADA advanced scheme when $w \leq 19$. In reality, the plaintext space $2^{19} = 524288$ is sufficient for ordinary health-care applications. Therefore, it is easy to find that our scheme is more appropriate for applications requiring highly-accurate data, especially in health-care applications.

VIII. Related Work

In recent years, privacy-preserving data aggregation schemes have been widely investigated in academia. Data aggregation techniques can significantly reduce communication and computational overhead of CSs and help CSs to offer low-latency and effective services including leakage forecasting and detection. Therefore, in this section, we briefly discuss some other research works [16], [17], [19]–[22], [34] closely related to our scheme. The result of the comparison of closely related schemes' features is demonstrated in Table IV. In [16], Lu et al. present an efficient and privacy-preserving aggregation scheme called EPPA, which is adopted to support multi-dimensional data aggregation. In order to defend a semi-honest aggregator, the homomorphic Paillier encryption technique is utilized to realize privacy-preserving aggregation. It structures multi-dimensional data into a ciphertext by adopting a super-increasing sequence. EPPA can obviously decrease the system's communication and computational overhead by using the batch verification technique. However, this scheme is not tolerant to user failures and it does not preserve differential privacy either. In [36], Fan et al. propose a protocol which allows the server to detect invalid data. However, the aforementioned two schemes cannot be applied to compute multifunctional aggregations.

In [17], Chen et al. propose a scheme named PDAFT, which supports fault tolerance and privacy-preserving data aggregation. PDAFT takes advantage of homomorphic Paillier Encryption technique to encrypt private user data so that it can prevent the control center from knowing personal user data while acquiring the aggregated data. In addition, a strong adversary who aims to breach user privacy can reveal nothing

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2015.2472369, IEEE Transactions on Information Forensics and Security

TABLE IV: Comparison of closely related schemes' features

Features	Schemes							
	EPPA [16]	PDAFT [17]	PriSense [19]	VPA [20]	EPADA [21]	PHDA [34]	MuDA [22]	Ours
Privacy-preserving	√	√	√	√	√	√	√	√
Considering strong adversary	×	√	×	×	×	×	×	√
Spatial and temporal aggregation	×	√	×	×	×	×	×	√
Fault tolerance	×	√	×	×	×	×	×	√
Differential privacy	×	×	×	×	×	×	√	√
Multifunctional additive aggregations	×	×	√	√	×	×	√	√
Supporting non-additive aggregation	×	×	√	√	√	×	×	√

even though he has already compromised a few servers at the control center. Moreover, PDAFT also supports both spatial and temporal aggregation and it supports fault tolerant. Furthermore, it reduces the communication overhead of previously reported competitive approaches [18]. However, Chen et al.'s scheme does not provide multifunctional aggregations and it does not preserve differential privacy either.

In [19], Shi et al. propose a scheme named PriSense. By using the idea of data mixing and slicing, it supports various statistical additive and non-additive aggregation. Furthermore, it can also resist the collusion attack during the aggregation. In order to achieve user privacy and data integrity for people-centric urban sensing systems (PC-USSs), Zhang et al. [20] present a new method to verify privacy-preserving data aggregation in PC-USSs, named VPA. VPA can support multifunctional additive and non-additive aggregation, for example count, average, variance, min/max, median, histogram, percentile, etc. However, differential attacks may occur and their scheme can not support fault tolerant in the case of users or cloud servers may fail.

In [21], Li et al. present an efficient and privacy-aware data aggregation in mobile sensing, called EPADA, which studies how an untrusted aggregator can obtain desired statistics without knowing individual user's private data. In EPADA, the authors utilize a novel key management technique and an additive homomorphic encryption to support large plaintext space. In addition, EPADA can also support min aggregate of time-series data, which needs only one round of communication between user and aggregator. However, their scheme can not support fault tolerant and it cannot be used to compute multifunctional aggregations. Although their min aggregate protocol can be easily extended to max aggregate protocol, it is difficult to support the other non-additive aggregate statistics, such as median and percentile.

In [34], Zhang et al. present a scheme named PHDA, which is a priority based health data aggregation scheme. It is used to improve the aggregation efficiency among different types of health data. By utilizing social spots, PHDA can forward health data and enable users to select the optimal relay according to their social ties. According to different data priorities, the adjustable forwarding strategies can be selected to forward the user's health data to the cloud servers with the reasonable communication overheads. In addition, PHDA can resist the forgery attacks and achieve the desirable delivery ratio with reasonable communication costs and lower delay for the data in different priorities. At the same time, it reduces the communication overheads. However, their scheme was not fault-tolerant in the case of users or cloud servers may fail and it does not resist differential attacks either. Furthermore, it cannot be used to compute multifunctional aggregations.

In [22], Chen et al. propose a scheme named MuDA, which supports multifunctional aggregations. By utilizing it, the control center can compute multiple statistical functions of users' data in a privacy-preserving way to provide various services. Except for average aggregation, their scheme can also achieve differential privacy for other more complex aggregations such as variance aggregation and one-way ANOVA aggregation. In addition, differential privacy is adopted to resist differential attacks that most data aggregation schemes may suffer and it just introduces acceptable noise. Furthermore, MuDA deceases the communication overhead of a popular aggregation scheme [10]. However, their scheme does not consider the non-additive aggregation, for example min/max, median, percentile, histogram, etc. Furthermore, their scheme does not support temporal aggregation and it cannot support fault tolerant in the case of users or cloud servers may fail.

Although our presented PPM-HDA mechanism deals with the similar issues as the above works, such as providing multifunctional and privacy-preserving data aggregate, supporting both spatial and temporal aggregation, supporting fault tolerant and resisting differential attacks, our research focuses are different: i) In our PPM-HDA mechanism, the CS can compute statistic additive and non-additive functions. Moreover, according to the requirement of CSs, the SP can compute multifunctional aggregations without requiring the mobile users to transmit any other reports; ii) Our proposed PPM-HDA mechanism can not only preserve differential privacy for additive aggregations, such as summation and variance aggregations, but also non-additive aggregations, such as min/max, median, percentile and histogram. iii) We propose our aggregation protocol in a more challenging threat model in which the adversary can compromise some of the CSs and obtain their private keys. In addition, we also prevent individual health data from being disclosed to the adversary; iv) We take malfunction of both users and CSs into consideration. By supporting fault tolerance, the reliability and feasibility of our aggregation scheme is improved. Furthermore, our scheme supports both spatial and temporal aggregation.

IX. Conclusion

In this paper, for cloud assisted WBANs, we have proposed a PPM-HDA mechanism that supports multifunctional additive aggregation, such as average and variance, and non-additive aggregations, for example min/max, median, σ-percentile and

histogram. In addition, our proposed PPM-HDA mechanism can also resist differential attacks, which most existing data aggregation schemes suffered from. Compared with existing data aggregation schemes that can only compute summation aggregation [8], [16], [17] and additive aggregation [22], our proposed scheme provides more diversity and security for cloud servers in the health data aggregation framework.

Our multifunctional health data aggregation framework considered a stronger adversary model where an attacker is allowed to compromise some CSs and obtain their private keys. The significance of considering such a stronger attacker model is to realize user health data privacy protection although an attacker could get some CSs' private keys.

Fault tolerance is drastically critical to health data aggregation domain because timely and reliable aggregation reporting is significant to user health-care. Our health data aggregation framework can deal with malfunction of users and CSs that makes our aggregation scheme more reliable in WBANs.

In addition, the performance evaluation illustrates that the computational overhead of MHDA^{+} is significantly reduced with the assistance of CSs. Our MHDA$^{\oplus}$ scheme is more efficient than some previously reported min/max aggregation scheme in terms of communication overheads when the applications require large plaintext spaces and highly-accurate data, especially in health-care applications.

For our future work, we will also consider fault tolerance for SPs in the framework of health data aggregation.

ACKNOWLEDGMENT

The authors sincerely thank the anonymous reviewers for their precious comments and suggestions.

REFERENCES

[1] H. Viswanathan, B. Chen, D. Pompili, "Research challenges in computation, communication, and context awareness for ubiquitous healthcare," *Communications Magazine, IEEE,* vol. 50, pp. 92-99, 2012.

[2] U. Mitra, B. A. Emken, S. Lee, et al, "KNOWME: A case study in wireless body area sensor network design," *Communications Magazine, IEEE,* vol. 50, pp. 116-125, 2012.

[3] J. M. L. P. Caldeira, J. J. P. C. Rodrigues, P. Lorenz, "Toward ubiquitous mobility solutions for body sensor networks on healthcare," *Communications Magazine, IEEE,* vol. 50, pp. 108-115, 2012.

[4] A. Azadeh, I. M. Fam, M. Khoshnoud, et al, "Design and implementation of a fuzzy expert system for performance assessment of an integrated health, safety, environment (HSE) and ergonomics system: The case of a gas refinery," *Information Sciences,* vol. 178, pp. 4280-4300, 2008.

[5] N. Botts, B. Thoms, A. Noamani, et al, "Cloud computing architectures for the underserved: Public health cyberinfrastructures through a network of healthatms," *System Sciences (HICSS), 2010 43rd Hawaii International Conference on. IEEE,* 2010, pp. 1-10.

[6] M. Valero, S. S. Jung, A. S. Uluagac, Y. Li & R. Beyah, "Di-sec: A distributed security framework for heterogeneous wireless sensor networks," *INFOCOM, 2012 Proceedings IEEE,* 2012, pp. 585-593.

[7] K. Alharbi, X. Lin, "Lpda: a lightweight privacy-preserving data aggregation scheme for smart grid," *Wireless Communications & Signal Processing (WCSP), 2012 International Conference on. IEEE,* 2012, pp. 1-6.

[8] W. Jia, H. Zhu, Z. Cao, et al, "Human-Factor-Aware Privacy-Preserving Aggregation in Smart Grid," *Systems Journal, IEEE,* vol. 8, pp. 598-607, 2014.

[9] F. Li, B. Luo, P. Liu, "Secure information aggregation for smart grids using homomorphic encryption," *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on. IEEE,* 2010, pp. 327-332.

[10] E. Shi, T. H. H. Chan, E. G. Rieffel, R. Chow, D. Song, "Privacy-preserving aggregation of time-series data," *NDSS,* vol. 2, p. 4, 2011.

[11] F. D. Garcia, B. Jacobs, "Privacy-friendly energy-metering via homomorphic encryption," *Security and Trust Management. Springer Berlin Heidelberg,* vol. 6710, pp. 226-238, 2011.

[12] V. Rastogi, S. Nath, "Differentially private aggregation of distributed time-series with transformation and encryption," *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data. ACM,* 2010, pp. 735-746.

[13] C. Dwork, "Differential privacy," *Automata, languages and programming. Springer Berlin Heidelberg,* vol. 4052, pp. 1-12, 2006.

[14] C. Dwork, "Differential privacy: A survey of results," *Theory and applications of models of computation. Springer Berlin Heidelberg,* vol.4978, pp. 1C19, 2008.

[15] A. Ghosh, T. Roughgarden, M. Sundararajan, "Universally utility-maximizing privacy mechanisms," *SIAM Journal on Computing,* vol. 41, pp. 1673-1693, 2012.

[16] R. Lu, X. Liang, X. Li, et al, "Eppa: An efficient and privacy-preserving aggregation scheme for secure smart grid communications," *Parallel and Distributed Systems, IEEE Transactions on,* vol. 23, pp. 1621-1631, 2012.

[17] L. Chen, R. Lu, Z. Cao, "PDAFT: A privacy-preserving data aggregation scheme with fault tolerance for smart grid communications," *Peer-to-Peer Networking and Applications,* 2014, pp. 1-11.

[18] Z. Erkin, G. Tsudik, "Private computation of spatial and temporal power consumption with smart meters," *Applied Cryptography and Network Security. Springer Berlin Heidelberg,* 2012, pp. 561-577.

[19] J. Shi, Y. Zhang, Y. Liu, "Prisense: privacy-preserving data aggregation in people-centric urban sensing systems," *INFOCOM, 2010 Proceedings IEEE,* 2010, pp. 1-9.

[20] R. Zhang, J. Shi, Y. Zhang, et al, "Verifiable Privacy-Preserving Aggregation in People-Centric Urban Sensing Systems," *Selected Areas in Communications, IEEE Journal on,* vol. 31, pp. 268-278, 2013.

[21] Q. Li, G. Cao, T. La Porta, "Efficient and privacy-aware data aggregation in mobile sensing," *Dependable and Secure Computing, IEEE Transactions on,* vol. 11, pp. 115-129, 2013.

[22] L. Chen, R. Lu, Z. Cao, et al, "MuDA: Multifunctional data aggregation in privacy-preserving smart grid communications," *Peer-to-Peer Networking and Applications,* 2014, pp. 1-16.

[23] Medical Body Area Networks First Report and Order, 2009 [Online]. Available: http://www.fcc.gov/document/medical-body-area-networks-first-report-and-order.

[24] D. Boneh, K. Rubin, A. Silverberg, "Finding composite order ordinary elliptic curves using the cockscpinch method," *Journal of Number Theory* vol. 131, pp. 832C841, 2011.

[25] D. Boneh, E. J. Goh, K. Nissim, "Evaluating 2-dnf formulas on ciphertexts," *Theory of cryptography. Springer Berlin Heidelberg,* vol. 3378, pp. 325C341, 2005.

[26] A. J. Menezes, P. C. V. Oorschot, S. A. Vanstone, *Handbook of applied cryptography* : CRC press, 1996.

[27] J. Cheng, H. Yang, S. H. Wong, S. Lu, "Design and implementation of cross-domain cooperative firewall," *Network Protocols, 2007. ICNP 2007. IEEE International Conference on,* 2007, pp. 284-293.

[28] A. X. Liu, F. Chen, "Collaborative enforcement of firewall policies in virtual private networks," *Proceedings of the twenty-seventh ACM symposium on Principles of distributed computing. ACM,* 2008, pp. 95-104.

[29] F. Chen, A. X. Liu, "SafeQ: Secure and efficient query processing in sensor networks," *INFOCOM, 2010 Proceedings IEEE,* pp. 1-9, 2010.

[30] Y. Yao, N. Xiong, J. H. Park, et al, "Privacy-preserving max/min query in two-tiered wireless sensor networks," *Computers & Mathematics with Applications,* vol. 65, pp. 1318-1325, 2013.

[31] P. Gupta, N. McKeown, "Algorithms for packet classification," *Network, IEEE,* vol. 15, pp. 24C32, 2001.

[32] Y. -K. Chang, "Fast binary and multiway prefix searches for packet forwarding," *Computer Networks,* vol. 51, pp. 588-605, 2007.

[33] A. Shamir, "How to share a secret," *Communications of the ACM,* vol. 22, pp. 612-613, 1979.

[34] K. Zhang, X. Liang, M. Baura, et al, "PHDA: A priority based health data aggregation with privacy preservation for cloud assisted WBANs," *Information Sciences,* vol. 284, pp. 130-141, 2014.

[35] OpenSSL 1.0.2 [Online]. Available: http://www.openssl.org/source/.

[36] J. Fan, Q. Li and G. Cao, "Privacy-Aware and Trustworthy Data Aggregation in Mobile Sensing," *to appear in the IEEE Conference on Communications and Network Security (CNS), 2015*.