sensors

MDPI

Article

Secure and Privacy-Preserving Body Sensor Data Collection and Query Scheme

Hui Zhu *, Lijuan Gao and Hui Li

Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China; gaolijuanxd@163.com (L.G.); lihui@mail.xidian.edu.cn (H.L.)

* Correspondence: zhuhui@xidian.edu.cn; Tel.: +86-29-8820-1014; Fax: +86-29-8820-1982

Academic Editor: Rongxing Lu

Received: 14 October 2015; Accepted: 27 January 2016; Published: 1 February 2016

Abstract: With the development of body sensor networks and the pervasiveness of smart phones, different types of personal data can be collected in real time by body sensors, and the potential value of massive personal data has attracted considerable interest recently. However, the privacy issues of sensitive personal data are still challenging today. Aiming at these challenges, in this paper, we focus on the threats from telemetry interface and present a secure and privacy-preserving body sensor data collection and query scheme, named SPCQ, for outsourced computing. In the proposed SPCQ scheme, users' personal information is collected by body sensors in different types and converted into multi-dimension data, and each dimension is converted into the form of a number and uploaded to the cloud server, which provides a secure, efficient and accurate data query service, while the privacy of sensitive personal information and users' query data is guaranteed. Specifically, based on an improved homomorphic encryption technology over composite order group, we propose a special weighted Euclidean distance contrast algorithm (WEDC) for multi-dimension vectors over encrypted data. With the SPCQ scheme, the confidentiality of sensitive personal data, the privacy of data users' queries and accurate query service can be achieved in the cloud server. Detailed analysis shows that SPCQ can resist various security threats from telemetry interface. In addition, we also implement SPCQ on an embedded device, smart phone and laptop with a real medical database, and extensive simulation results demonstrate that our proposed SPCQ scheme is highly efficient in terms of computation and communication costs.

Keywords: body sensor network; privacy-preserving; data query; outsourced computing

1. Introduction

In recent years, with the popularization of wearable sensors and telemedicine, body sensor networks (BSN), which comprise multiple sensor nodes and a coordinator worn on a human body, can collect the personal information of the human body (such as heart rate, blood glucose and electrocardiogram) by sensor nodes [1–6]. The collected information first is delivered to the coordinator, then is forwarded to a remote server through a network interface for further processing [7,8]. As shown in Figure 1, vast quantities of the sensor users' personal data are collected by body sensors and recorded by a data center per second. Since large-scale aggregate analysis of personal data can yield valuable results and insights, which can address public health challenges and provide new avenues for scientific discovery [9], data center trends toward providing on-demand data query service for users. However, this requires huge storage space and enormous computing resources, which are tremendous burdens on data centers. As the survey [10] shows that roughly 55 percent of respondents plan to use cloud services for analysis queries, cloud computing is a promising way to integrate personal data resources and to provide a uniform query service to researchers [11–15]. Since personal data are regarded as sensitive and private assets of sensor users, how to provide accurate data query services without revealing confidential personal data has attracted considerable interest recently.

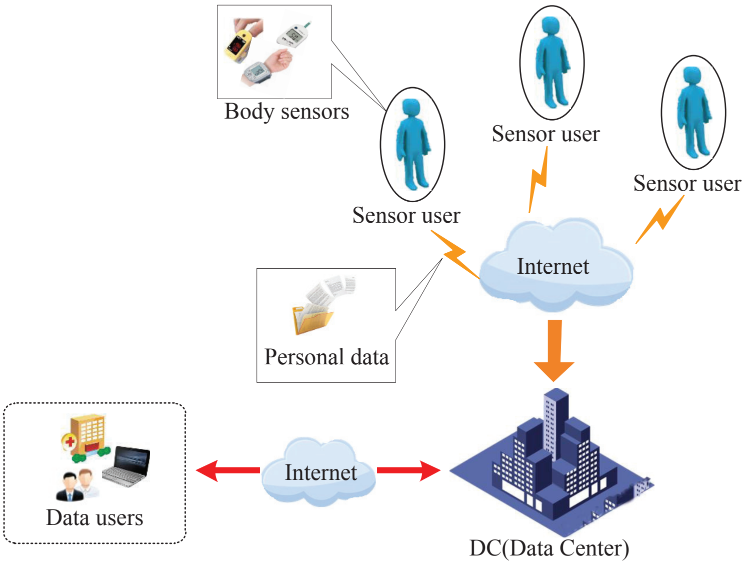


Figure 1. Body sensor data collection and query service scenario.

To address these security and privacy issues, differential privacy [16], homomorphic encryption [17] and searchable encryption [18] are widely used. However, differential privacy cannot provide accurate query results for users; traditional homomorphic encryption schemes are time consuming and resource consuming; searchable encryption cannot provide query services over encryption data. Therefore, the above methods are not suitable for multi-dimension personal data query services.

Different from the methods discussed above, in this paper, we focus on the threats from telemetry interface [19,20] and propose a new secure and privacy-preserving body sensor data collection and query scheme, called SPCQ, for outsourced computing. In the proposed scheme, users' personal information is collected by body sensors of different types and converted into multi-dimension data, and each dimension is converted into the form of a number and uploaded to the cloud server in ciphertext. After that, the cloud server provides query services to data users by the privacy-preserving weighted Euclidean distance contrast (WEDC) algorithm for multi-dimension vectors. Meanwhile, SPCQ can protect the confidentiality of sensor users' personal data and the privacy of data users' query, with low overheads in computation and communication.

The remainder of this paper is organized as follows. In Section 2, we review the related work. In Section 3, we define the system model and security model and identify our design goal. Additionally, in Section 4, we recall the bilinear pairing of the composite order, Euclidean distance and the 2DNF cryptosystem as the preliminaries. Then, we present our SPCQ scheme in Section 5, followed by the security analysis and performance evaluation in Sections 6 and 7, respectively. Finally, we draw our conclusions in Section 8.

2. Related Work

In recent years, how to achieve operations over encrypted data has attracted considerable interest, and most of the proposed schemes are based on differential privacy, homomorphic encryption and searchable encryption.

The differential privacy notion was first formulated by Dwork [16], which can provide information about the database while simultaneously ensuring very high levels of privacy. Barthe *et al.* [21]

presented CertiPriv, a machine-checked framework for reasoning about differential privacy built on top of the Coq proof assistant. The scheme provided a framework for fine-grained reasoning about an expressive class of confidentiality policies. Additionally, Tschantz *et al*. [22] presented the first results towards automated verification of source code for differentially-private interactive systems, which developed a formal probabilistic automaton model of differential privacy for systems by adapting prior work on differential privacy for functions. To achieve automated verification of distributed differential privacy, Eigner *et al*. [23] presented the framework by comprising a symbolic definition of differential privacy for distributed databases that takes into account Dolev–Yao intruders, and the scheme can overhear, intercept and synthesize the cryptographic messages exchanged on the network. However, the above differential privacy scheme cannot provide accurate query services because of added randomized noise.

Homomorphic encryption is a usual method to achieve data operations over encrypted data without decrypting it. Rivest *et al*. [17] first introduced homomorphism and presented four solutions to achieve homomorphic encryption. Then, Goldwasser *et al*. [24] proposed the first semantically-secure homomorphic encryption scheme, and many other additively homomorphic encryption schemes with proofs of semantic security [25–27] were presented. To achieve both additive and multiplicative homomorphisms, Gentry [28] designed a full homomorphic encryption scheme based on the mathematical object ideal lattices and uses the bootstrapping technique. It is semantically secure, and the security of the scheme is based on the split-key distinguishing problem. Then, other different full homomorphic encryption schemes were based on the elementary theory of algebraic number fields [29] and non-circuit [30]. However, most of these existing homomorphic encryption schemes have high time complexities, which is not suitable for practical use.

Keyword searchable encryption schemes usually build an encrypted searchable index, such that its content is hidden from the server unless it is given appropriate trapdoors generated via secret keys [31]. Song *et al*. [18] firstly studied searchable encryption in the symmetric key setting, and Boneh *et al*. [32] presented the first searchable encryption construction, where anyone with a public key can write to the data stored on the server, but only authorized users with a private key can search. However, public key solutions are usually very computationally expensive, and the keyword privacy could not be protected in the public key setting. To solve the multi-keyword ranked search over encrypted data problem, Cao *et al*. [33] proposed a basic idea of MRSE using secure inner product computation, and two improved MRSE schemes were given to achieve various stringent privacy requirements in two different threat models. However, searchable encryption can only provide keyword query rather than accurate computation query, which is not suitable for multi-dimension personal data.

Different from the above works, our proposed SPCQ scheme aims at the efficiency, accurate and privacy issues, and based on an improved homomorphic encryption technology over composite order group, we develop an efficient and privacy-preserving body sensor data collection and query scheme for outsourced computing. In particular, the proposed SPCQ can be easily implemented on different terminals, and the processing of the query is just needed in the cloud server. The computational costs in both the terminal and cloud server are acceptable.

3. Models and Design Goals

In this section, we define the system model, security requirements and identify our design goal.

3.1. System Model

In our system model, we mainly focus on how to offer secure personal data collection and efficient query service over confidential personal data in the outsourced cloud server. Specifically, the system consists of four parts: *register center* (RC), *sensor user* (SU), *data user* (DU) and *cloud server*(CS), as shown in Figure 2.

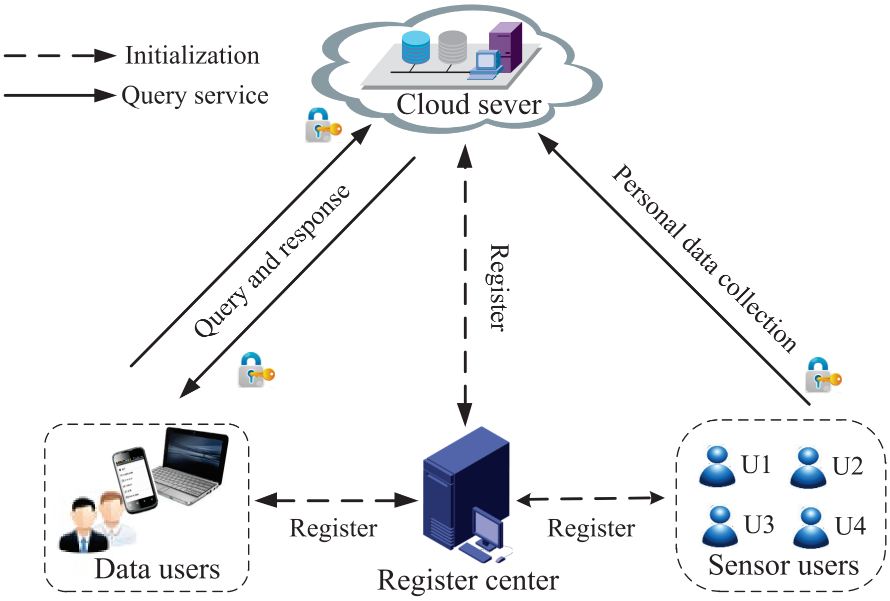


Figure 2. System model under consideration.

- RC is a trusted third party, which bootstraps the system initialization by generating system parameters and providing a registration system for SU, DU and CS.
- SU collects real-time personal data by body sensors, and all SUs' personal data will be uploaded to CS. To guarantee the data confidentiality, SU will perform some encryption operations before uploading data to CS.
- DU (e.g., the researcher), who is registered in RC, can send query request to CS for analysis of the accurate personal data items stored in it. To guarantee the privacy of DU's query information, DU will perform some encryption operations during the process of the query. Meanwhile, SUs' personal data should be kept secret from unauthorized users.
- CS is composed of many data storage nodes and computing nodes, stores more than a billion encrypted personal data items from SUs and provides accurate query services to DUs over encrypted personal data. CS mainly performs two functions: authentication and computing over encrypted data. The authentication component is used to check the identity of SUs and DUs, while the computing in encryption component is used to search and compute encrypted data items with DUs' encrypted query request.

3.2. Security Requirements

The confidentiality of personal data from SUs and the privacy of DU's query information are crucial for the success of a secure and privacy-preserving body sensor data collection and query scheme. In our security model, we consider CS is *honest-but-curious*. Specifically, CS faithfully executes the operations to search DUs' demanded information over the encrypted personal data from SUs, but it also tries to analyze the query information and encrypted data to obtain users' sensitive information. Therefore, in this paper, we focus on the threats from telemetry interface, and the following security requirements should be satisfied in a secure and privacy-preserving body sensor data collection and query scheme. Note that, in our current model, we do not consider that any two parties collude to disclose the third party's privacy, *i.e.*, the collusion attack on privacy is beyond the scope of this work and will be discussed in future research.

- *Confidentiality.* SUs' sensitive personal data should be kept secret from CS, *i.e.*, even if CS stores all personal data from SUs, it cannot identify any data item. In this circumstance, the confidentiality of the personal data can be guaranteed.
- *Privacy.* DU's query information should be secrete from CS, *i.e.*, even if CS obtains all DUs' queries and corresponding responses, it cannot identify DUs' query information accurately. Additionally, other users (e.g., SUs and other DUs) cannot get any information of DU. In this circumstance, the privacy-preserving requirements of DU's query information can be guaranteed. In addition, the privacy requirement also includes CS's responses, *i.e.*, only legal DU can decrypt the corresponding response.
- *Authentication.* Authenticating an encrypted query that is really sent by a legal DU and has not been altered during the transmission, *i.e.*, if an illegal DU forges a query, this malicious operation should be detected, and only correct queries can be received by CS. The responses from CS should also be authenticated so that DUs can receive authentic and reliable query results. Moreover, the encrypted personal data from SUs can be authenticated by CS.

3.3. Design Goals

Under the aforementioned system model and security requirements, our design goal is to develop a secure and privacy-preserving body sensor data collection and query scheme for outsourced computing, which will provide secure personal data collection and storage for SUs and privacy-preserving accurate Euclidean distance query service for DUs. Specifically, the following three objects should be achieved.

- *The Security Requirements Should be Guaranteed.* If the personal data collection and query scheme does not consider the security, SUs' data assets and DU's actual query information could be disclosed. Then, the data collection and query service cannot jump in popularity. Therefore, the proposed scheme should achieve the confidentiality, privacy and authentication simultaneously.
- *A Personal Data query Service with High Accuracy Should be Guaranteed.* The user experience is one of the most critical aspects of data query service, and it is important that the precision of Euclidean distance query service cannot be lowered when protecting DU's privacy. Therefore, the proposed scheme should also provide highly precise and reliable query service.
- *The Effectiveness in Computation and Communication Should be Achieved for Various Terminal Devices.* The personal data may be collected by different terminal devices, such as smart phone, embedded device, *etc.* Although the performance of terminal devices is continuously improved today, the battery is still limited. The proposed scheme should also consider the effectiveness in terms of computation and communication to reduce the power consumption of different terminals. Moreover, data users can access the data query service by mobile terminals, in order to lower the energy cost, the efficiency of the query service is very important. Furthermore, although CS is featured with high performance in storage and computation, since thousands of DUs will query the data at the same time, the efficiencies of computation and communication are still challenging.

4. Preliminaries

In this section, we recall the bilinear pairing technique, Euclidean distance and 2DNF cryptosystem, which serve as the basis of our proposed SPCQ scheme.

4.1. Bilinear Pairing of Composite Order

Let $\mathbb{G}$ and $\mathbb{G}_t$ be two multiplicative cyclic groups of the same composite order $N = p_1 \cdot p_2$ (where p_1 and p_2 are big primes), and g is a generator of $\mathbb{G}$. We suppose $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_t$ denotes the bilinear map (also referred to as a paring), which has the following properties.

(1) Bilinearity. $e(u^a, v^b) = e(u, v)^{ab}$ holds for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_N$;
(2) Non-degeneracy. $e(g, g) \neq 1_{\mathbb{G}_t}$;

(3) Computability. For all $u, v \in \mathbb{G}$, $e(u, v)$ can be computed efficiently.

4.2. Euclidean Distance

Euclidean distance is a common definition of distance, it corresponds to the true distance of two points in n-dimensional space. The Euclidean distance between two points P and Q is the length of the line segment connecting them in Euclidean space. In Cartesian coordinates, if $\mathrm{P} = (p_1, p_2, ..., p_n)$ and $\mathrm{Q} = (q_1, q_2, ..., q_n)$ are two different n-dimensional vectors, then the Euclidean distance from P to Q or from Q to P is given by the Pythagorean formula:

$$d(\mathrm{P}, \mathrm{Q}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

To satisfy the practical circumstance and provide DUs with an accurate query service, we set different weight numbers $(w_1, w_2, ..., w_n)$ for each dimension to form a weighted Euclidean distance:

$$d(\mathrm{P}, \mathrm{Q}) = \sqrt{w_1(q_1 - p_1)^2 + w_2(q_2 - p_2)^2 + \cdots + w_n(q_n - p_n)^2}$$

4.3. 2DNF Cryptosystem

The 2DNF [34] cryptosystem is a public-key system that can achieve the homomorphic properties, which resembles the Paillier [27] and Okamoto-Uchiyama [26] encryption schemes. Specially, the 2DNF cryptosystem consists of three sections: key generation, encryption and decryption.

- Key generation $Gen(\mu)$. Given a security parameter $\mu \in \mathbb{Z}^+$, two $\mu-$bit prime numbers p_1 and p_2 are first chosen, and $N = p_1 \cdot p_2$ is computed. Two groups of the same order N are generated, g and u are two generators of $\mathbb{G}$. Then, $h = u^{p_2}$ is computed as a random generator of $\mathbb{G}$'s subgroup with order p_1. Finally, the public key $PK = (n, \mathbb{G}, \mathbb{G}_t, e, g, h)$ and private key $SK = p_1$ are generated.
- Encryption. We assume the message space consists of integers in the set $\{0, 1, \cdots, T\}$ with $T < p_2$. Then, to encrypt a message m with public key PK, a random number r is selected from $\{0, 1, \cdots, N-1\}$, and the ciphertext $C = g^m h^r \in \mathbb{G}$ is computed.
- Decryption. To decrypt ciphertext C with private key $SK = p_1$, notice that $C^{p_1} = (g^m h^r)^{p_1} = (g^{p_1})^m$; let $\hat{g} = g^{p_1}$. To recover the corresponding message m, we need to compute the discrete log of $C^{p_1} = \hat{g}^m$, where $\hat{g} = g^{p_1}$. Since $0 \le m \le T$, it only takes expected time $\hat{O}(\sqrt{T})$ using Pollard's lambda method [35] to get the message m.

Note that the decryption time in the system would be the polynomial time in the size of the message space T_s. Hence, it is obvious that the cryptosystem is efficiently suitable for short messages.

5. Proposed SPCQ

In this section, we present a secure and privacy-preserving body sensor data collection and query scheme for outsourced computing, which mainly consists of three phases: *system initialization*, *secure data collection* and *privacy-preserving query service*. For an easier expression, the definition of notations to be used in the proposed SPCQ scheme are shown in Table 1.

Table 1. Definition of notations in the proposed secure and privacy-preserving body sensor data collection and query (SPCQ) scheme.

Notation	Definition
μ	the system security parameter
p_1, p_2	two big prime numbers
$N = p_1 \cdot p_2$	the product of p_1 and p_2
$\mathbb{G}, \mathbb{G}_t$	the bilinear groups with order N
e, g ,h	the parameters of bilinear groups
B_1	$B_1 = g^{p_1}$
B_2	$B_2 = e(g,g)^{p_1}$
$E()$	the asymmetric encryption algorithm, *i.e.*, ECC
$H()$	the secure cryptographic hash function
HPS	the evaluation dataset
$(x_{i1}, x_{i2}, ..., x_{in})$	the feature parameters of a data item
$W = (w_1, w_2, ..., w_n)$	the weighted number of different dimensions
d	the weighted Euclidean search range of DU's query
$F_i = (f_{i1}, f'_{i1}, f_{i2}, f'_{i2}, ..., f_{in}, f'_{in})$	the encrypted search index of a data item
$\langle q_1, q'_1, ..., q_n, q'_n \rangle$	DU's encrypted query parameters

5.1. System Initialization

We consider RC as the trusted third party, which bootstraps the system. In the system initialization phase, RC first selects a security parameter μ, generates system parameters $(\mathbb{G}, \mathbb{G}_t, p_1, p_2, e, g, h, N = p_1 \cdot p_2)$ by executing $Gen(\mu)$ and calculates two secret bases $B_1 = g^{p_1}$ and $B_2 = e(g,g)^{p_1}$. Next, RC decides a multi-dimension weight vector $W = (w_1, w_2, ..., w_n)$ that each number denotes the weight value of the corresponding dimension. Then, RC picks a random number $r_{RC} \in \mathbb{Z}_N^*$ as its private key SK_{RC} and computes the corresponding public key $PK_{RC} = g^{SK_{RC}}$. In addition, RC determines an asymmetric cryptographic algorithm $E()$, *i.e.*, ECC, and a secure cryptographic hash function $H()$, where $H : \{0,1\}^* \to \mathbb{Z}_N^*$ and $\mathbb{Z}_N^*$ is a nonzero group of integer modulo N. Finally, RC publishes the system parameters as $\langle N, \mathbb{G}, \mathbb{G}_T, e, g, h, PK_{RC}, E(), H() \rangle$ and keeps $\langle p_1, SK_{RC} \rangle$ secretly.

When an SU or DU registers itself to RC, it picks a random number $r \in \mathbb{Z}_N^*$ as the private key SK and computes and submits the corresponding public key $PK = g^{SK}$ to RC for the signature. Then, RC sends $\langle B_1, B_2, W \rangle$ to the registered SU and DU through a secure channel. Similarly, when CS registers itself, it generates the private and public key pair as $SK_{CS} \in \mathbb{Z}_N^*$, $PK_{CS} = g^{SK_{CS}}$, and submits PK_{CS} to RC for the signature. After that, RC calculates $HP_j = H({B_2}^{j^2})$, where $0 \leq j \leq \eta$ and η is a big integer whose length is much less than 256 bits, and structures the set of data values $HPS = \{HP_0, HP_1, ..., HP_\eta\}$. Then, RC ranks the dataset from the smallest to the largest and sends the ordered dataset HPS to CS. It is noteworthy that $\langle B_1, B_2 \rangle$ is not given to CS. After providing the registration function for SU, DU and CS, RC goes offline or suffers slowdowns against the single point of attack, since it has many secret parameters.

5.2. Secure Data Collection

SUs collect their real-time personal data through body sensors, and the data can be described by n-dimensional vectors $(x_{i1}, x_{i2}, ..., x_{in})$. Before uploading to CS, each data item in SU should be processed as follows.

- SU computes $x'_{i1} = x_{i1} + H(B_1)$, $x'_{i2} = x_{i2} + H(B_1)$,..., $x'_{in} = x_{in} + H(B_1)$, where B_1 is only known by registered SUs and DUs; this operation can resist the exhaustive attack.
- SU chooses n random numbers $r_1, r_2, ..., r_n \in Z_N^*$ and computes the encrypted search index $F_i = (f_{i1}, f'_{i1}, f_{i2}, f'_{i2}, ..., f_{in}, f'_{in})$, which can be implicitly formed as follows.

$$\begin{cases} f_{i1} = B_2^{w_1 \cdot {x'_{i1}}^2} & f'_{i1} = g^{x'_{i1}} \cdot h^{r_1} \\ f_{i2} = B_2^{w_2 \cdot {x'_{i2}}^2} & f'_{i2} = g^{x'_{i2}} \cdot h^{r_2} \\ \vdots & \vdots \\ f_{in} = B_2^{w_n \cdot {x'_{in}}^2} & f'_{in} = g^{x'_{in}} \cdot h^{r_n} \end{cases}$$

- SU makes a signature $Sig = H(F_i \| ID \| TS_1)^{SK}$ using the private key SK, where TS_1 is the current timestamp to resist potential replay attack, and ID is the identify number of SU. Then, SU sends the signed data item $\langle F_i \| ID \| TS_1 \| Sig \rangle$ to CS.
- After receiving the signed data item from SU, CS first checks the timestamp TS_1 and verifies the signature Sig by computing whether $e(g, Sig) = e(PK, H(F_i \| ID \| TS_1))$. If it does hold, the signature is accepted, since $e(g, Sig) = e(g, H(F_i \| ID \| TS_1))^{SK} = e(PK, H(F_i \| ID \| TS_1))$. Then, CS stores the data item F_i.

5.3. Privacy-Preserving Query Service

5.3.1. User Query Generation

Registered DU U_j is able to send a query request to CS without revealing his or her query information by the following steps.

- U_j first decides a data item with n feature parameters $\{y_1, y_2, ..., y_n\}$ that he or she is willing to query and computes $y'_1 = y_1 + H(B_1), y'_2 = y_2 + H(B_1), ..., y'_n = y_n + H(B_1)$ to increase the sample space.
- U_j determines the weighted Euclidean distance search range d from the data item that he or she wants to query and computes encrypted query $(q_1, q'_1, q_2, q'_2, ..., q_n, q'_n)$ as follows.

$$\begin{cases} q_1 = B_2^{w_1 \cdot y'^2_1 - d^2} & q'_1 = B_1^{2w_1 \cdot y'_1} \\ q_2 = B_2^{w_2 \cdot y'^2_2} & q'_2 = B_1^{2w_2 \cdot y'_2} \\ \vdots & \vdots \\ q_n = B_2^{w_n \cdot y'^2_n} & q'_n = B_1^{2w_n \cdot y'_n} \end{cases}$$

- U_j uses the public key of CS PK_{CS} to compute $Q = E_{PK_{CS}}(q_1||q'_1||q_2||q'_2||...||q_n||q'_n)$.
- U_j makes a signature $Sig_j = (H(Q \| U_j \| TS_2))^{SK_{U_j}}$ using his or her private key SK_{U_j}, where TS_2 is the current timestamp to resist potential replay attack. Then, U_j sends the encrypted data query request $\langle Q \| U_j \| TS_2 \| Sig_j \rangle$ to CS.

5.3.2. Search and Response

After receiving encrypted data query request $\langle Q \| U_j \| TS_2 \| Sig_j \rangle$ from U_j, CS executes the following procedures to provide personal data query service.

- CS first checks the timestamp TS_2 and verifies the signature Sig_j by computing whether $e(g, Sig_j) = e(PK_{DU_j}, H(Q \| U_j \| TS_2))$. If it does hold, the signature is accepted, since $e(g, Sig_j) = e(g, H(Q \| U_j \| TS_2))^{SK_{U_j}} = e(PK_{U_j}, H(Q \| U_j \| TS_2))$.
- CS uses its secret key SK_{CS} to decrypt Q and obtain $\langle q_1, q'_1, q_2, q'_2, ..., q_n, q'_n \rangle$. Then, CS executes the proposed WEDC algorithm as follows.

- For each data item F_i stored in it, CS computes the search criteria D_i as follows.

$$\begin{aligned}
D_i &= \frac{e(f'_{i1}, q'_1) \cdot e(f'_{i2}, q'_2) \cdot ... \cdot e(f'_{in}, q'_n)}{f_{i1} \cdot f_{i2} \cdot ... \cdot f_{in} \cdot q_1 \cdot q_2 \cdot ... \cdot q_n} \\
&= \frac{e(g^{x'_{i1}} h^{r_1}, B_1^{2w_1 \cdot y'_1}) \cdot ... \cdot e(g^{x'_{in}} h^{r_n}, B_1^{2w_n \cdot y'_n})}{B_2^{w_1 {x'_{i1}}^2} \cdot ... \cdot B_2^{w_n {x'_{in}}^2} \cdot B_2^{w_1 y'^2_1 - d^2} \cdot ... \cdot B_2^{w_n y'^2_n}} \\
&= \frac{e(g^{x'_{i1}} h^{r_1}, g^{p_1 \cdot 2w_1 \cdot y'_1}) \cdot ... \cdot e(g^{x'_{in}} h^{r_n}, g^{p_1 \cdot 2w_n \cdot y'_n})}{B_2^{w_1 {x'_{i1}}^2} \cdot ... \cdot B_2^{w_n {x'_{in}}^2} \cdot B_2^{w_1 y'^2_1 - d^2} \cdot ... \cdot B_2^{w_n y'^2_n}} \\
&= \frac{e(g, g)^{p_1 2w_1 \cdot x'_{i1} y'_1} \cdot ... \cdot e(g, g)^{p_1 \cdot 2w_n \cdot x'_{in} y'_n}}{B_2^{w_1 {x'_{i1}}^2} \cdot ... \cdot B_2^{w_n {x'_{in}}^2} \cdot B_2^{w_1 y'^2_1 - d^2} \cdot ... \cdot B_2^{w_n y'^2_n}} \\
&= B_2^{d^2 - (w_1 (x'_{i1} - y'_1)^2 + ... + w_n (x'_{in} - y'_n)^2)} \\
&= B_2^{d^2 - (w_1 (x_{i1} - y_1)^2 + ... + w_n (x_{in} - y_n)^2)}
\end{aligned}$$

- CS computes $HD_i = H(D_i)$ and searches HD_i within the evaluation dataset HPS by binary search algorithm to confirm whether HD_i belongs to it. If HD_i belongs to HPS, it means that data item F_i satisfies DU's query condition; add one to M_{num}, where M_{num} denotes the number of data items that meets the query condition; otherwise, data item F_i does not meet DU's query condition.
- After traversing through all data items, CS gets the number of data items that satisfy DU's query condition M_{num} and the number of all data items N_{num}, which can help DU to achieve the statistical query of the personal data. Then, CS encrypts N_{num} and M_{num} with the asymmetric encryption algorithm $E()$ and the public key of U_j PK_{U_j} and uses its private key to make a signature $Sig_{CS} = H(E_{PK_{U_j}}(N_{num} \| M_{num}) \| TS_3)^{SK_{CS}}$.
- Finally, CS sends $\left\langle E_{PK_{U_j}}(N_{num} \| M_{num}) \| TS_3 \| Sig_{CS} \right\rangle$ to U_j.

Correctness of WEDC Algorithm. Here, we prove that CS can provide the correct statistical query service for DUs by executing the WEDC algorithm. Specifically, taking a look at the exponential of search criteria $D_i = {B_2}^{d^2 - (w_1 (x_{i1} - y_1)^2 + ... + w_n (x_{in} - y_n)^2)}$, we know B_2 is the generator of a cyclic group with order p_2, which is selected larger than 512 bits, and $w_1(x_{i1} - y_1)^2 + ... + w_n(x_{in} - y_n)^2$ is the square of the weighted Euclidean distance between DU's query data and data item F_i. In addition, since search range d is usually less than 10,000, we can define $\eta = 10,000$ and $\eta^2 = 100,000,000$. Therefore, if F_i meets DU's query condition, *i.e.*, $0 \leq d^2 - (w_1(x_{i1} - y_1)^2 + ... + w_n(x_{in} - y_n)^2) \leq d^2 \leq 100,000,000$, the corresponding $H(D_i)$ must be in HPS, and F_i will be counted as eligible data item; otherwise, $d^2 - (w_1(x_{i1} - y_1)^2 + ... + w_n(x_{in} - y_n)^2) \leq 0$, and $D_i = {B_2}^{d^2 - (w_1 (x_{i1} - y_1)^2 + ... + w_n (x_{in} - y_n)^2)} = {B_2}^{p_2 + d^2 - (w_1 (x_{i1} - y_1)^2 + ... + w_n (x_{in} - y_n)^2)}$; then the corresponding $H(D_i)$ will not be in HPS since $p_2 + d^2 - (w_1(x_{i1} - y_1)^2 + ... + w_n(x_{in} - y_n)^2) \gg 100,000,000$; meanwhile, F_i will not be counted as an eligible data item. Through the method we have stated, CS can correctly judge whether F_i satisfies DUs' query condition by utilizing WEDC algorithm.

5.3.3. Query Result Reading

After receiving $\left\langle E_{PK_{U_j}}(N_{num} \| M_{num}) \| TS_3 \| Sig_{CS} \right\rangle$ from CS, U_j checks TS_3 and the signature Sig_{CS} by verifying whether $e(g, Sig_{CS}) = e(PK_{CS}, H(E_{PK_{U_j}}(N_{num} \| M_{num}) \| TS_3)^{SK_{CS}})$ holds. Then, U_j decrypts $E_{PK_{U_j}}(N_{num} \| M_{num})$ with SK_{U_j} to obtain the query result.

6. Security Analysis

In this section, we analyze the security properties of the proposed SPCQ scheme. Specifically, following the security requirements discussed earlier, our analysis will focus on how the proposed SPCQ scheme can achieve personal data confidentiality, DU's query information privacy and source authentication of the personal data, query request and response.

- *The Proposed SPCQ Can Achieve Confidential Personal Data.* In our proposed SPCQ, personal data are secret from CS and DUs, although CS stores all encrypted data items and receives all query requests. First, since feature parameters $(x_{i1}, x_{i2}, ..., x_{in})$ collected by body sensors usually cover a smaller scope, to avoid the exhaustive attack against $(x_{i1}, x_{i2}, ..., x_{in})$ by Pollard's lambda method, feature parameters are disturbed by calculating $x'_{i1} = x_{i1} + H(B_1)$, $x'_{i2} = x_{i2} + H(B_1), ..., x'_{in} = x_{in} + H(B_1)$. In this way, the sample space is increased to more than 512 bits, which can prevent exhaustive attack efficiently. Before uploading to CS, feature parameters are encrypted to corresponding search index $(f_{i1}, f'_{i1}, f_{i2}, f'_{i2}, ..., f_{in}, f'_{in})$ by computing $f_{i1} = B_2^{w_1 \cdot x'^2_{i1}}, f'_{i1} = g^{x'_{i1}} \cdot h^{r_1}$, *etc.*, where r_1 is a random number to guarantee that for the same feature parameter, different SUs can obtain different search indexes. The above operations can achieve data perturbation and substitution and prevent CS from directly accessing SUs' personal data. Moreover, to avoid the guessing attacks for B_2 in the evaluation dataset HPS, the relationship between B_2 and HP_j is hidden by a secure hash function $H()$, where $HP_j = H(B_2^{j^2})$. Therefore, from the above three aspects, CS cannot obtain the feature parameters of personal data according to uploaded data items. In addition, since DUs only can get the query statistic result from CS, SUs' personal data are secret from DUs.
- *DU's Query Information is Privacy-Preserving in the Proposed SPCQ.* In our proposed SPCQ, similarly, DU's query condition is encrypted before being sent to CS. Specifically, DU's query information $y_1, y_2, ..., y_n$ is disturbed by calculating $y'_1 = y_1 + H(B_1), y'_2 = y_2 + H(B_1), ...,$ $y'_n = y_n + H(B_1)$, which can resist the exhaustive attack by Pollard's lambda method. Then, the query condition is encrypted by calculating $q_1 = B_2^{w_1 \cdot y'^2_1 - d^2}, q'_1 = B_1{}^{2w_1 \cdot y'_1}, q_2 = B_2^{w_2 \cdot y'^2_2}$, $q'_2 = B_1{}^{2w_2 \cdot y'_2}, ..., q_n = B_2^{w_n \cdot y'^2_n}, q'_n = B_1{}^{2w_n \cdot y'_n}$, which can prevent CS from directly accessing the query data item and search range d. Since B_1 and B_2 are only known by SUs and DUs, and the collusion attack is not considered in the current security model, CS cannot obtain query information from the query request during the query process. Specifically, encrypted request and encrypted data are computed in CS to obtain the result, which will be sent back to DU, and CS also cannot obtain any useful information of DUs' queries, even in the continuous search queries environment. Meanwhile, CS still can provide accurate query service to DUs by the proposed WEDC algorithm. Concretely, CS traverses all stored data items to compute the search criteria D_i and find out all data items that satisfy the query condition, then it achieves the query statistic result and sends it to DU. It is notable that the result does not have particular meaning without any other useful information of DUs' queries. Moreover, SUs are not involved in the query process, and DU's query request is encrypted by CS's public key PK_{CS} before being sent to CS, so SUs cannot get DU's query information even if they steal the request by eavesdropping. In addition, the response is encrypted by DU's public key before being sent by CS, and thus, SUs and other registered DUs cannot decrypt the response. Therefore, from the above four aspects, DU's query information is privacy-preserving in the proposed SPCQ.
- *The Authentication of the Personal Data, Query Request and Response are Achieved in the Proposed SPCQ.* In the proposed SPCQ, personal data from SUs, registered DU's query request and the response of CS are signed by the BLS [36] short signature. Since the BLS short signature is provably secure under the CDH problem in the random oracle model, the source authentication can be guaranteed. Specifically, personal data from SU is signed by computing $Sig = H(F_i \| ID \| TS_1)^{SK}$, where TS_1 is the current timestamp to resist potential replay attack and

SK is SU's private key to make sure only itself can make the signature. After receiving the signed data item, CS computes whether $e(g, Sig) = e(PK, H(F_i \| ID \| TS_1))$ to verify the source of the signature. Similarly, the registered DU's query request and the response of CS are signed by the above operations. Moreover, since the unregistered user (such as SU and DU) does not have secret keys B_1 and B_2, he or she cannot upload personal data item or submit valid query request to CS. Therefore, personal data and the query request from the unregistered user and the response from the mendacious CS can be detected in the proposed SPCQ.

From the above analyses, we can conclude that SPCQ is secure and privacy-preserving and can achieve our security design goal.

7. Performance Evaluation

In this section, we evaluate the performance of our proposed SPCQ scheme in terms of the computation complexity of SU, DU and CS. In order to measure the integrated performance of SPCQ in a real environment, we also implement SPCQ on an embedded device, smart phone and laptop with a real medical database in a wireless network, by using a custom simulator built in JAVA. Specifically, an embedded device with a 650-MHz dual-core processor, a smart phone with a 1.4-GHz quad-core processor, 2 GB RAM, Android 4.0, and a laptop with a 2.0-GHz 4-core processor, 8 GB RAM, are chosen to simulate SU, DU and CS. Based on our proposed SPCQ scheme, a personal data gathering application is installed on the embedded device to simulate SU; a personal data query application built by JAVA, named SPCQ.apk, is installed on the smart phone to simulate DU; and simulators of CS are deployed in a laptop. In order to evaluate SPCQ in a real environment, a diabetes database [37], which has 100,000 items with 55 attributes, is selected as the data source, and the corresponding storage space is 692.37 MB. Meanwhile, an evaluation dataset *HPS* with 10,000 preprocessed SHA-256 values is constructed, which just needs a 312.5-KB storage space. In addition, we define p_1 and p_2 as 512-bit prime numbers, and η^2 as 100,000,000.

7.1. Computation and Communication Costs

The proposed SPCQ scheme can achieve effective personal data query service for CS and DUs. Specifically, we assume the dimension of each personal data item is n, and SU needs n multiplication operations and $3n$ exponentiation operations for each personal data item. When a DU U_j generates an encrypted query $(q_1, q_1', q_2, q_2', ..., q_n, q_n')$, it requires $2n$ exponentiation operations in $\mathbb{Z}_{p_2}$. After receiving the query from U_j, CS firstly computes the search criteria D_i for each data item F_i stored in it, which takes $n * N$ pairing operations and $2n * N$ multiplication operations for checking N resource items. After receiving the response from CS, U_j decrypts the query statistic result with asymmetrical encryption algorithm, which is considered negligible compared to exponentiation and pairing operations. Denote the computational costs of an exponentiation operation in $\mathbb{Z}_{p_2}/\mathbb{Z}_{N^2}$, a multiplication operation in $\mathbb{G}/\mathbb{G}_t/\mathbb{Z}_{N^2}$ and a pairing operation by C_e, C_m and C_p, respectively. Then, for SU, DU and CS, the computational costs are $3n * C_e + n * C_m$, $2n * C_e$ and $nN * C_p + 2nN * C_m$ in the proposed SPCQ.

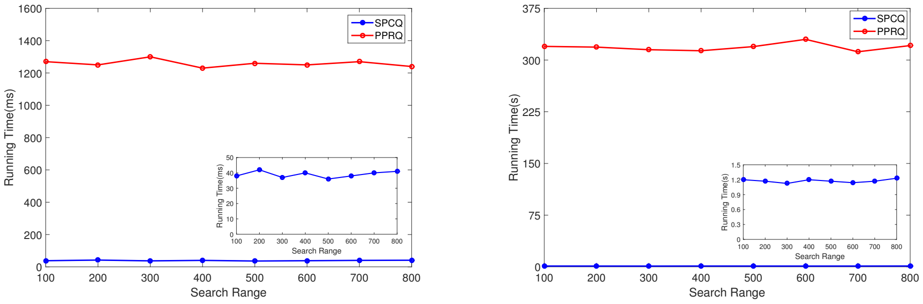
Different from other time-consuming encryption techniques, the proposed SPCQ uses improved homomorphic encryption technology over a composite order group, which can provide accurate personal data query service and largely reduce the encryption time for the smart phone. In the following, for the comparison with SPCQ, we selected a privacy-preserving range query scheme (PPRQ) [38], which can only provide a one-dimensional range query service to DUs. Let m be the bit length of the attribute values, and the computational costs of SU, DU and CS are $2n * C_e + n * C_m$, $4m * C_e + 2m * C_m$ and $23mnN * C_e + 23mnN * C_m$, respectively.

Due to the factor of our proposed SPCQ, we take an n-dimension query into consideration. Then, we present the computation complexity comparison of the proposed SPCQ and PPRQ in Table 2, and it is obvious that our proposed SPCQ can achieve a privacy-preserving personal data query service with low computation complexity in DU and CS.

Table 2. Comparison of computation complexity.

Phase of Scheme	SPCQ	PPRQ
SU	$3n * C_e + n{*}C_m$	$2n * C_e + n * C_m$
DU	$2n * C_e$	$4m * C_e + 2m * C_m$
CS	$nN * C_p$+$2nN * C_m$	$23mnN * C_e$+$23mnN * C_m$

For better comparison, we have implemented SPCQ and PPRQ in JAVA. In Figure 3a,b, we have plotted the computational overheads of SPCQ and PPRQ varying with different search ranges in DU and CS. From the two figures, we can see that in both DU and CS, the computational overheads of SPCQ and PPRQ vary slightly by increasing the search range, while the overheads of PPRQ are much higher than those of our proposed SPCQ scheme. It can be obviously shown that the SPCQ scheme largely reduces the computational complexity in DU and CS.



(a)

(b)

Figure 3. Computational overheads of SPCQ and PPRQ. (**a**) Average running time in DU with different search ranges; (**b**) average running time in CS with different search ranges.

In addition, we have made the comparison of communication costs between SPCQ and PPRQ, as shown in Table 3. In SPCQ, the communication length in DU is $164 * n$ bytes, which is much less than that of PPRQ, whose communication length in DU is $512 * n$ bytes; the communication length in CS is 256 bytes, while that of PPRQ in CS is $1024 * N$ bytes; in addition, two times of communications between DU and CS are needed in both SPCQ and PPRQ. As we mentioned above, SPCQ is more efficient than PPRQ in terms of communication costs.

Table 3. Comparison of communication costs.

Phase of Scheme	SPCQ	PPRQ
Communication length in DU	$164 * n$ bytes	$512 * n$ bytes
Communication length in CS	256 bytes	$1024 * N$ bytes
Communication times	2	2

7.2. Simulation and Evaluation

To have a better evaluation of our proposed SPCQ, we analyze the factors that affect the computational costs of SU, DU and CS in detail. In addition, we evaluate the integrated performance of SPCQ.

7.2.1. SU

In our proposed SPCQ scheme, SUs collect their real-time personal data from body sensors and upload these data to CS per certain period. Before being sent to CS, the gathered personal data should be operated to obtain $(f_{i1}, f'_{i1}, f_{i2}, f'_{i2}, ..., f_{in}, f'_{in})$. Therefore, we have chosen different dimensions of personal data to illustrate the SU's computational cost on the embedded device. As shown in Figure 4a, the dimensions of collected personal data are chosen from 5 to 40, and the average computational cost increases linearly with the increase of the dimension. Meanwhile, the computation on the embedded device is less than 150 milliseconds, which is acceptable for SU.

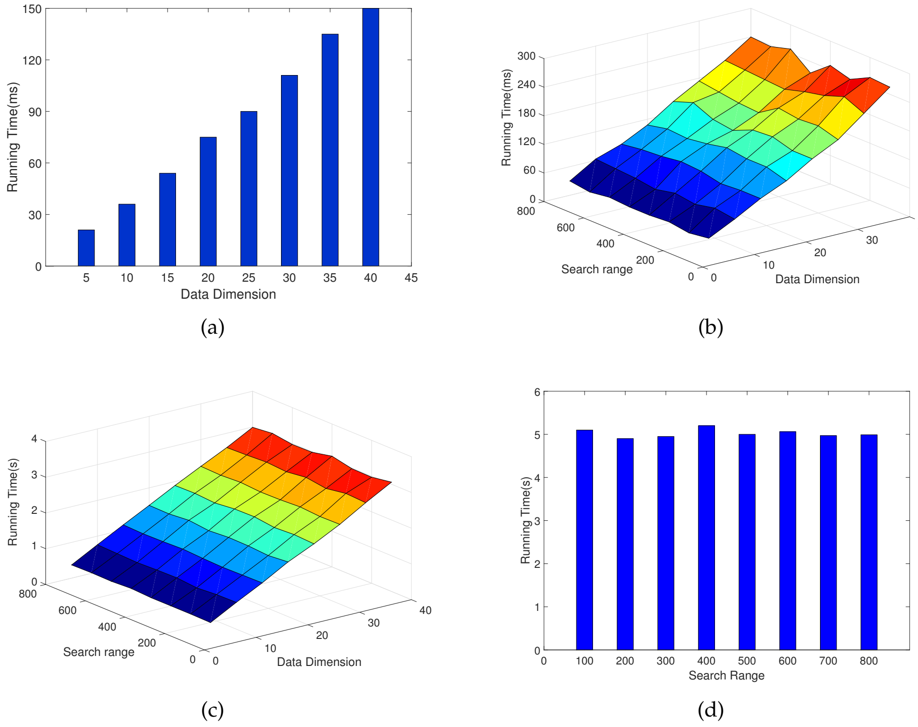


Figure 4. Computational cost of SPCQ. (**a**) Computational cost of SU in data collection; (**b**) computational cost of DU in query generation; (**c**) computational cost of CS with different search ranges and dimensions; (**d**) query response time in a real environment.

7.2.2. DU

The query response time of DU (*i.e.*, smart phone) is an important aspect for our proposed SPCQ scheme, and the computational operations in the smart phone are query generation and result reading. Since the result reading only requires DU to decrypt the query result, which is negligible, therefore we have chosen different dimensions of the query request and different search ranges to illustrate DU's computational cost. To observe the computational cost of the smart phone, the dimensions of each query are chosen from 5 to 40, and the search ranges are chosen from 100 to 800. Each condition is executed 100 times, and we have calculated the average time for different dimensions of the query request and search range. As shown in Figure 4b, the average computational cost increases linearly with the increase of the dimension, and it is nearly the same with different search ranges. The reason is that, when the smart phone generates a query request, it computes encrypted query parameters with the query condition. For the query condition with a high dimension, it takes more time to get the request, while different search ranges do not affect the computational cost.

7.2.3. CS

In our proposed SPCQ scheme, after receiving a query request from DU, CS will compute the search criteria D_i for each data item it stored, by using bilinear pairing over the composite order group, which is the main computation overhead of CS, *i.e.*, the efficiency of CS is impacted by the number of encrypted data resources, the dimension of each data item and the search range d. It is obvious that the computational cost in CS is increased with the number of encrypted data resources. Therefore, we have chosen different dimensions of data resources and different search ranges to illustrate the computational cost. As shown in Figure 4c, the dimensions of data resources are selected from 5 to 40, and eight search ranges of DU's requests are selected from 100 to 800. We can learn from the figure that the computational cost of CS is nearly the same with different search ranges; meanwhile, the computational cost increases linearly with the increase of the data resource's dimension.

7.2.4. Integrated Performance in a Real Environment

In order to evaluate the integrated performance of our proposed scheme, SPCQ is deployed in a real environment with the real medical database mentioned above. Specifically, we have chosen 1000 items from the diabetes database, and the information of resources and corresponding encryption information is stored in CS, respectively. In addition, the smart phone and CS are connected through an 802.11g WLAN, and when DUs input the query data item and search range by SPCQ.apk, the smart phone will send a query request to CS and get the response through WLAN. We have run 100 times to evaluate the performance of SPCQ with eight search ranges (from 100 to 800), as shown in Figure 4d, the runtime is about five seconds, which is acceptable in a real environment.

8. Conclusions

In this paper, we have proposed a secure and privacy-preserving body sensor data collection and query scheme, called SPCQ, for outsourced computing. Based on an improved homomorphic encryption technology over composite order group, the proposed SPCQ scheme can achieve the confidentiality of SUs' personal data and privacy-preserving of DU's query information. Specifically, SUs' personal data are collected in the form of multi-dimension vectors and uploaded to CS in ciphertext, and the data query request from the registered DU can be directly performed over ciphertext in CS, then the query result only can be decrypted by the registered DU. Therefore, DU can get an accurate query result without divulging his or her query information. Detailed security analysis shows its security strength and privacy-preserving ability, and extensive experiments are conducted to demonstrate its efficiency.

Acknowledgments: The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the paper. This work was supported by the National Natural Science Foundation of China (No.61303218, U1401251), the Huawei Innovation Research Program and the 111 Project (B08038).

Author Contributions: Hui Zhu contributed to the original ideas, scheme designing, experiment supervision, data analysis guidelines and manuscript drafting. Lijuan Gao contributed to scheme designing, model simulations, data analysis and manuscript drafting. Hui li contributed to the original ideas, scheme designing and revision of the manuscript.

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

References

1. Yang, G.Z.; Yacoub, M. *Body Sensor Networks*; Springer London: London, UK, 2006.
2. Ullah, S.; Higgins, H.; Braem, B.; Latre, B.; Blondia, C.; Moerman, I.; Saleem, S.; Rahman, Z.; Kwak, K.S. A comprehensive survey of wireless body area networks. *J. Med. Syst.* **2012**, *36*, 1065–1094.
3. Sun, W.; Zhang, Z.; Ji, L.; Wong, W.C. An optimized handover scheme with movement trend awareness for body sensor networks. *Sensors* **2013**, *13*, 7308–7322.
4. Yu, R.; Yang, G.Z.; Lo, B.P.L. Autonomic body sensor networks. In the Proceedings of the 2014 IEEE MTT-S International Microwave Workshop Series on RF and Wireless Technologies for Biomedical and Healthcare Applications (IMWS-Bio), London, UK, 8–10 December 2014; pp. 1–3.
5. Moshaddique, A.A.; Jingwei, L.; Kyungsup, K. Security and Privacy Issues in Wireless Sensor Networks for Healthcare Applications. *J. Med. Syst.* **2012**, *36*, 93–101.
6. Ren, Y.; Oleshchuk, V.; Li, F.Y.; Ge, X. Security in Mobile Wireless Sensor Networks—A Survey. *J. Commun.* **2011**, *34*, 1302–1325.
7. Movassaghi, S.; Abolhasan, M.; Lipman, J.; Smith, D.; Jamalipour, A. Wireless body area networks: A survey. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1658–1686.

8. Sun, W.; Ge, Y.; Zhang, Z.; Wong, W.C. Performance Evaluation of Wearable Sensor Systems: A Case Study in Moderate-Scale Deployment in Hospital Environment. *Sensors* **2015**, *15*, 24977–24995.
9. Horvitz, E.; Mulligan, D. Data, privacy, and the greater good. *Science* **2015**, *349*, 253–255.
10. Barlow, M. *Migrating Big Data Analytics into the Cloud*; O'Reilly: Fairfield, CT, USA, 2014.
11. Lien, I.T.; Lin, Y.H.; Shieh, J.R.; Wu, J.L. A Novel Privacy Preserving Location-Based Service Protocol With Secret Circular Shift for K-NN Search. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 863–873.
12. Shao, J.; Lu, R.; Lin, X. FINE: A Fine-Grained Privacy-Preserving Location-Based Service Framework for Mobile Devices. In Proceedings of the 2014 Proceedings IEEE INFOCOM, Toronto, ON, Canada, 27 April–2 May 2014; pp. 1452–1461.
13. Liu, J.K.; Au, M.H.; Susilo, W.; Liang, K.; Lu, R.; Srinivasan, B. Secure sharing and searching for real-time video data in mobile cloud. *IEEE Netw.* **2015**, *29*, 46–50.
14. Juliadotter, N.V.; Choo, K.K.R. Cloud Attack and Risk Assessment Taxonomy. *IEEE Cloud Comput.* **2015**, *2*, 14–20.
15. Ab Rahman, N.H.; Choo, K.K.R. A survey of information security incident handling in the cloud. *Comput. Secur.* **2015**, *49*, 45–69.
16. Dwork, C. *Differential Privacy*; Springer US: Venice, Italy, 2006; pp. 1–12.
17. Rivest, R.L.; Adleman, L.; Dertouzos, M.L. On data banks and privacy homomorphisms. *Found. Secur. Comput.* **1978**, *4*, 169–180.
18. Song, D.X.; Wagner, D.; Perrig, A. Practical techniques for searches on encrypted data. In Proceedings of the 2000 IEEE Symposium on Security and Privacy, Berkeley, CA, USA, 14–17 May 2000; pp. 44–55.
19. Rushanan, M.; Rubin, A.D.; Kune, D.F.; Swanson, C.M. SoK: Security and Privacy in Implantable Medical Devices and Body Area Networks. In Proceedings of the 2014 IEEE Symposium on Security and Privacy, San Jose, CA, USA, 18–21 May 2014; pp. 524–539.
20. Li, C.; Raghunathan, A.; Jha, N.K. Hijacking an insulin pump: Security attacks and defenses for a diabetes therapy system. In Proceedings of the 2011 13th IEEE International Conference on E-Health Networking Applications and Services (Healthcom), Columbia, MO, USA, 13–15 June 2011; pp. 150–156.
21. Barthe, G.; Köpf, B.; Olmedo, F.; Zanella Béguelin, S. Probabilistic relational reasoning for differential privacy. In Proceedings of the 39th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Philadelphia, PA, USA, 25–27 January 2012; Volome 47, pp. 97–110.
22. Tschantz, M.C.; Kaynar, D.; Datta, A. Formal Verification of Differential Privacy for Interactive Systems. *Electron. Notes Theor. Comput. Sci.* **2011**, *276*, 61–79.
23. Eigner, F.; Maffei, M. Differential Privacy by Typing in Security Protocols. In Proceedings of the 2013 IEEE 26th Computer Security Foundations Symposium (CSF), New Orleans, LA, USA, 26–28 June 2013; pp. 272–286.
24. Goldwasser, S.; Micali, S. Probabilistic encryption and how to play mental poker keeping secret all private information. In Proceedings of the 14th Annual ACM Symposium on Theory of Computing, San Francisco, CA, USA, 5–7 May 1982; pp. 365–377.
25. Benaloh, J.D.C. *Verifiable Secret-Ballot Elections*; Department of Computer Science, Yale University: New Haven, CT, USA, 1987.
26. Okamoto, T.; Uchiyama, S. A new public-key cryptosystem as secure as factoring. In *Advances in Cryptology—EUROCRYPT'98*; Springer Heidelberg: Berlin, Germany, 1998; pp. 308–318.
27. Paillier, P. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology–EUROCRYPT'99*; Springer Heidelberg: Berlin, Germany, 1999; pp. 223–238.
28. Gentry, C. Fully homomorphic encryption using ideal lattices. In Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing, Bethesda, Maryland, 31 May–2 June 2009; Volume 9, pp. 169–178.
29. Smart, N.P.; Vercauteren, F. Fully homomorphic encryption with relatively small key and ciphertext sizes. In *Public Key Cryptography–PKC 2010*; Springer Heidelberg: Berlin, Germany, 2010; pp. 420–443.
30. Xiao, L.; Bastani, O.; Yen, I.L. An Efficient Homomorphic Encryption Protocol for Multi-User Systems. *IACR Cryptol. ePrint Archive* **2012**, *2012*, 193.
31. Kamara, S.; Lauter, K. Cryptographic cloud storage. In *Financial Cryptography and Data Security*; Springer Heidelberg: Berlin, Germany, 2010; pp. 136–149.

32. Boneh, D.; di Crescenzo, G.; Ostrovsky, R.; Persiano, G. Public key encryption with keyword search. In *Advances in Cryptology-Eurocrypt 2004*; Springer Heidelberg: Berlin, Germany, 2004, pp. 506–522.
33. Cao, N.; Wang, C.; Li, M.; Ren, K.; Lou, W. Privacy-preserving multi-keyword ranked search over encrypted cloud data. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 222–233.
34. Boneh, D.; Goh, E.J.; Nissim, K. Evaluating 2-DNF formulas on ciphertexts. In *Theory of Cryptography*; Springer Heidelberg: Berlin, Germany, 2005; pp. 325–341.
35. Menezes, A.J.; van Oorschot, P.C.; Vanstone, S.A. *Handbook of Applied Cryptography*; CRC Press: Boca Raton, FL, USA, 2010.
36. Boneh, D.; Lynn, B.; Shacham, H. Short signatures from the Weil pairing. In *Advances in Cryptology—ASIACRYPT 2001*; Springer Heidelberg: Berlin, Germany, 2001; pp. 514–532.
37. Clore, J.; Cios, K.J.; DeShazo, J.; Strack, B. Diabetes 130-US hospitals for years 1999-2008 Data Set. Available onine: http://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008 (accessed on 3 September 2015).
38. Samanthula, B.K.; Jiang, W. Efficient privacy-preserving range queries over encrypted data in cloud computing. In Proceedings of the 2013 IEEE Sixth International Conference on Cloud Computing (CLOUD), Santa Clara, CA, USA, 28 June–3 July 2013; pp. 51–58.