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/TDSC.2015.2406699, IEEE Transactions on Dependable and Secure Computing

Privacy Protection for Wireless Medical Sensor Data

Xun Yi, Athman Bouguettaya, Dimitrios Georgakopoulos, Andy Song and Jan Willemson

Abstract—In recent years, wireless sensor networks have been widely used in healthcare applications, such as hospital and home patient monitoring. Wireless medical sensor networks are more vulnerable to eavesdropping, modification, impersonation and replaying attacks than the wired networks. A lot of work has been done to secure wireless medical sensor networks. The existing solutions can protect the patient data during transmission, but cannot stop the inside attack where the administrator of the patient database reveals the sensitive patient data. In this paper, we propose a practical approach to prevent the inside attack by using multiple data servers to store patient data. The main contribution of this paper is securely distributing the patient data in multiple data servers and employing the Paillier and ElGamal cryptosystems to perform statistic analysis on the patient data without compromising the patients' privacy.

Keywords—Wireless medical sensor network, patient data privacy, Paillier encryption, and ElGamal encryption

1 Introduction

A wireless sensor network (WSN) consists of spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on.

Healthcare applications are considered as promising fields for wireless sensor networks, where patients can be monitored in hospitals and even at home using wireless medical sensor networks (WMSNs). In recent years, many healthcare applications using WSNs have been developed, such as CodeBlue [20]. Alarm-Net [30], UbiMon [24], MEDiSN [14], and MobiCare [4]. A typical example of healthcare applications with WSNs is Alarm-Net [30] developed in University of Virginia for assisted-living and residential monitoring. The architecture of Alarm-Net is shown in Fig. 1.

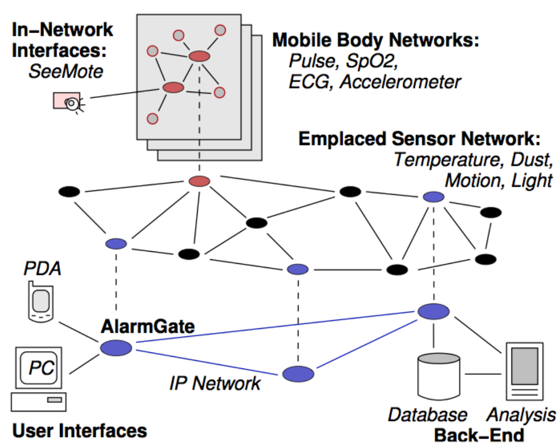

Fig. 1. Alarm-Net Architecture

Alarm-Net is composed of mobile body network, emplaced sensor network, AlarmGate applications, back-end systems, and user interfaces as follows:

- Mobile body network has wireless sensor devices worn by a patient which provide physiological sensing. Data from the mobile body network is transmitted through the emplaced sensors to user interfaces or back-end systems.
- Emplaced sensor network has devices deployed

- *Xun Yi, Athman Bouguettaya, Dimitrios Georgakopoulos and Andy Song are with the School of Computer Science and Information Technology, RMIT University, Melbourne, Victoria 3001, Australia.*
- *Jan Willemson is with Cybernetica, Ulikooli 2, Tartu 51003, Estonia*

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/TDSC.2015.2406699, IEEE Transactions on Dependable and Secure Computing

in the living space to sense environmental quality or conditions, such as temperature, dust, motion, and light. Emplaced sensors maintain connections with mobile body networks as they move through the living space.

- AlarmGate applications serve as application-level gateways between the wireless sensor networks and IP networks. These nodes allow user interfaces and a connection to a back-end database for long-term storage of data.
- Back-end systems provide online analysis of sensor data and long-term storage of data.
- User interfaces allow any legitimate user of the system to query sensor data.

Wireless medical sensor networks certainly improve patient's quality-of-care without disturbing their comfort. However, there exist many potential security threats to the patient sensitive physiological data transmitted over the public channels and stored in the back-end systems. Typical security threats to healthcare applications with WSNs can be summarized as follows.

Eavesdropping is a security threat to the patient data privacy. An eavesdropper, having a powerful receiver antenna, may be able to capture the patient data from the medical sensors and therefore knows the patient's health condition. He may even post the patient's health condition on social network, which can pose a serious threat to patient privacy.

Impersonation is a security threat to the patient data authenticity. In a home care application, an attacker may impersonate a wireless rely point while patient data is transmitting to the remote location. This may lead to false alarms to remote sites and an emergency team could start a rescue operation for a non-existent person. This can even defeat the purpose of wireless healthcare.

Modification is a security threat to the patient data integrity. While the patient data is transmitted to the physician, an adversary may capture the physiological data from the wireless channels and alter the physiological data. After the attacked data (i.e., altered data) is sent to the physician, it could endanger the patient.

Data breach is a security threat to the patient data privacy. A data breach is an incident in which sensitive, protected or confidential patient data has potentially been viewed, stolen or used by an individual unauthorized to do so. For example, a malicious patient database administrator may use the patient data (such as, patient identity) for their personal benefit, such as for medical fraud, fraudulent insurance claims, and sometimes this may even pose life-threatening risks.

To protect the wireless medical sensor networks against various attacks, a lot of work has been done. In 2012, a survey on the recently published literature on secure healthcare monitoring using wireless sensor networks was conducted by Kumar and Lee [16]. Current solutions are built on either secret-key encryption or public-key encryption as follows:

- Secret-key based solutions assume that the secret keys for encryption and authentication are deployed in the medical sensors and the servers in advance. A secret key cryptosystem, such as AES [1], is used for encryption, while the message authentication code (MAC) is used for authentication. Typical examples of secret-key based solutions include [7], [12], [15], [26], [29], [31], [32]. These solutions are usually efficient. However, the distribution of the secret keys are less efficient than the public-key based solutions.
- Public-key based solutions assume that a public-key cryptosystem, such as Diffie-Hellman key exchange protocol [8] or RSA [27], is used to establish a secret key for encryption on the basis of the public keys. Typical examples of public-key based solutions include [11], [13], [17], [19], [21], [22]. These solutions facilitate key distribution and update. However, they are usually inefficient and not directly applicable to the wireless medical sensor networks, where the sensors have limited computation and communication capabilities.

In addition, in order to protect patients' privacy, k-anonymity has been used to make each patient indistinguishable from other k-1 similar patients in wireless medical sensor data before releasing the data for medical research [2].

Most of current solutions focus on how to protect the wireless medical sensor networks against the outside attacks, where the attacker does not know any information about the secret keys. The outside attacks can be effectively prevented by encryption, authentication and access control.

In 2013, Yi et al. [31] gave a secret-key based solution to protect the wireless medical sensor networks against the inside attacks, where the attacker

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/TDSC.2015.2406699, IEEE Transactions on Dependable and Secure Computing

may be a malicious administrator of the patient database. In the solution, the Sharemind system [3] composed of three data servers is used to store the patient data and each sensor shares three different secret keys with the servers. When a medical sensor sends a patient data (e.g. temperature reading) to the Sharemind system, it splits the patient data into three numbers such that the sum of them is equal to the original data and submits them, respectively, to the three data servers via three secure channels. Sharemind is a data processing system capable of performing computations on input data without compromising its privacy. The three servers in Sharemind can cooperate to process some queries on the patient data from the users (e.g., doctors, nurses, medical professionals) without seeing the patient data. The solution can protect the patient data privacy as long as the number of the compromised data servers is at most one. If two of the three data servers are compromised by the inside attack, the solution becomes insecure.

In this paper, we further improve the security of the solution given by Yi et al. [31]. Like [31], we assume that the wireless medical sensor network is composed of some medical sensors, three data servers, and some users. Each sensor sends the patient data to the three data server in the same way as [31]. Unlike [31], the three data servers process the queries, such as statistical analysis on the patient data, from the users on the basis of the Paillier [25] and ElGamal [10] cryptosystems instead of the Sharemind system [3]. The patient data privacy can be preserved as long as at least one of three data servers is not compromised. Even if two data servers are compromised but one data server is not compromised, our solution is still secure.

Our contributions in this paper can be summarized as follows.

- To prevent the patient data from the inside attacks, we propose a new data collection protocol, where a sensor splits the sensitive patient data into three components according to a random number generator based on hash function and sends them to three servers, respective, via secure channels.
- To keep the privacy of the patient data in data access, we propose a new data access protocol on the basis of the Paillier cryptosystem. The protocol allows the user (e.g., physician) to access the patient data without revealing it to any data server.
- To preserve the privacy of the patient data in statistical analysis, we propose some new privacy-preserving statistical analysis protocols on the basis of the Paillier and ElGamal cryptosystems. These protocols allow the user (e.g., medical researcher) to perform statistical analysis on the patient data without compromising the patient data privacy.

These contributions are essentially different from the solution given in [31], which relies on the Sharemind system for data analysis without considering the collusion of data servers.

The rest of the paper is organized as follows. Section 2 introduces the basic building blocks by which our solution is constructed. Section 3 describes our solution. Security and performance analysis is carried out in Section 4. Conclusions are drawn in the last section.

2 PRELIMINARIES

Two basic building blocks of our solution are the Paillier and the ElGamal public key cryptosystems, which are described in this section.

2.1 Paillier Public-Key Cryptosystem

The Paillier encryption scheme [25], named after and invented by Pascal Paillier in 1999, is a probabilistic public key encryption algorithm. It is composed of key generation, encryption and decryption algorithms as follows.

2.1.1 Key generation

The key generation algorithm works as follows.

- Choose two large prime numbers p and q randomly and independently of each other such that
$$\gcd(pq, (p-1)(q-1)) = 1$$
- Compute
$$N = pq, \lambda = lcm(p-1, q-1)$$
where lcm stands for the least common multiple.
- Select random integer g where $g \in \mathbb{Z}^*_{N^2}$ and ensure N divides the order of g by checking the existence of the following modular multiplicative inverse:
$$\mu = (L(g^\lambda (mod N^2)))^{-1} (mod\ N)$$

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/TDSC.2015.2406699, IEEE Transactions on Dependable and Secure Computing

where function L is defined as

$$L(u) = \frac{u-1}{N}$$

Note that the notation a/b does not denote the modular multiplication of a times the modular multiplicative inverse of b but rather the quotient of a divided by b.

The public (encryption) key pk is (N, g).

The private (decryption) key sk is (λ, μ).

If using p, q of equivalent length, one can simply choose

$$g = N + 1, \lambda = \varphi(N), \mu = \varphi(N)^{-1}(mod\ N)$$

where $N = pq$ and $\varphi(N) = (p-1)(q-1)$.

2.1.2 Encryption

The encryption algorithm works as follows.

- Let m be a message to encrypt, where $m \in \mathbb{Z}_N$.
- Select random r where $r \in \mathbb{Z}_N^*$.
- Compute ciphertext as:

$$c = g^m \cdot r^N (mod\ N^2) \tag{1}$$

2.1.3 Decryption

The decryption algorithm works as follows.

- Let c be the ciphertext to decrypt, where the ciphertext $c \in \mathbb{Z}_{N^2}^*$.
- Compute the plaintext message as:

$$m = L(c^\lambda (mod\ N^2)) \cdot \mu (mod\ N) \tag{2}$$

2.1.4 Homomorphic Properties

A notable feature of the Paillier cryptosystem is its homomorphic properties. Given two ciphertexts

$$E(m_1, pk) = g^{m_1} r_1^N (mod\ N^2)$$

$$E(m_2, pk) = g^{m_2} r_2^N (mod\ N^2)$$

where r_1, r_2 are randomly chosen for $\mathbb{Z}_N^*$, we have the following homomorphic properties.

The product of two ciphertexts will decrypt to the sum of their corresponding plaintexts,

$$D(E(m_1, pk_1) \cdot E(m_2, pk_2)) = m_1 + m_2 (mod\ N)$$

The product of a ciphertext with a plaintext raising g will decrypt to the sum of the corresponding plaintexts,

$$D(E(m_1, pk_1) \cdot g^{m_2}) = m_1 + m_2 (mod\ N)$$

An encrypted plaintext raised to a constant k will decrypt to the product of the plaintext and the constant,

$$D(E(m_1, pk_1)^k) = km_1 (mod\ N)$$

However, given the Paillier encryptions of two messages, there is no known way to compute an encryption of the product of these messages without knowing the private key.

2.2 ElGamal Public-Key Cryptosystem

The ElGamal encryption scheme [10], named after and invented by Taher ElGamal in 1985, is a probabilistic public key algorithm. It is composed of key generation, encryption and decryption algorithms as follows.

2.2.1 Key Generation

The key generator works as follows.

- Generate a cyclic group G, of large prime order q, with generator g.
- Choose a random $x \in \{1, \ldots, q-1\}$ and compute

$$y = g^x \tag{3}$$

The public (encryption) key pk is (G, q, g, y).

The private (decryption) key sk is x.

2.2.2 Encryption

The encryption algorithm works as follows.

- Let m be a message to encrypt, where $m \in G$.
- Choose a random $r \in \{1, \ldots, q-1\}$.
- Compute the ciphertext $c = (A, B)$, where

$$A = g^r \tag{4}$$

$$B = m \cdot y^r \tag{5}$$

2.2.3 Decryption

The decryption algorithm works as follows.

- Let $c = (A, B)$ be a ciphertext to decrypt.
- Compute

$$m = B/A^x \tag{6}$$

The decryption algorithm produces the intended message, since

$$\begin{aligned} B/A^x &= m \cdot y^r / g^{rx} \\ &= m \cdot g^{xr} / g^{rx} \\ &= m \end{aligned}$$

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/TDSC.2015.2406699, IEEE Transactions on Dependable and Secure Computing

2.2.4 Homomorphic Property

ElGamal encryption scheme has homomorphic properties. Given two encryptions $(A_1, B_1) = (g^{r_1}, m_1 y^{r_1})$ and $(A_2, B_2) = (g^{r_2}, m_2 y^{r_2})$, where r_1, r_2 are randomly chosen from $\{1, 2, \cdots, q-1\}$ and $m_1, m_2 \in G$, one can compute

$$\begin{aligned}(A_1, B_1)(A_2, B_2) &= (A_1 A_2, B_1 B_2) \\ &= (g^{r_1} g^{r_2}, (m_1 y^{r_1})(m_2 y^{r_2})) \\ &= (g^{r_1+r_2}, (m_1 m_2) y^{r_1+r_2})\end{aligned}$$

which is the encryption of $m_1 m_2$.

3 Privacy-Preserving Wireless Medical Sensor Network

3.1 Our Model

Like most of healthcare applications with wireless medical sensor network, our architecture has four systems as follows.

- A wireless medical sensor network which senses the patient's body and transmits the patient data to a patient database system;
- A patient database system which stores the patient data from medical sensors and provides querying services to users (e.g., physicians and medical professionals);
- A patient data access control system which is used by the user (e.g., physician) to access the patient data and monitor the patient;
- A patient data analysis system which is used by the user (e.g., medical researcher) to query the patient database system and analyze the patient data statistically.

There may be a middleware between the wireless medical sensor network and the patient database system. If so, the role of the middleware is simply forwarding the encrypted patient data to the database system.

In our model, the patient database system is composed of multiple database servers. We assume that all data servers are semi-honest, often called "honest but curious". That is, all data servers run our protocol exactly as specified, but may try to learn as much as possible about the patient data from their views of the protocol. In addition, we assume that at least one data server is not compromised by attackers. For simplicity, we assume that the number of data servers is three. In fact, it can be any number

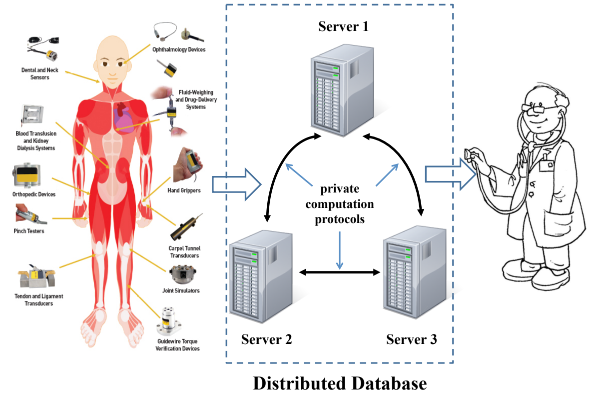

Fig. 2. Our Model

more than three. The architecture of our model with three data servers can be shown in Fig. 2.

The security requirements for our model include:

- Data collection security: In the wireless medical sensor network, each medical sensor can securely send the patient data to the distributed database system.
- Data store security: In the distributed patient database system, the patient data cannot be revealed even if two of three data servers are compromised by the inside attackers.
- Data access security: In the patient access control system, only the authorized user can get access to the patient data. The patient data cannot be disclosed to any data server during the access.
- Data analysis security: In the patient data analysis system, the authorized user can get the statistical analysis results only. The patient data cannot be disclosed to any data server and even to the user during the statistical analysis.

Our model considers two types of attacks, the outside attack and the inside attack. The outside attacker does not know any secret key in our system, but attempts to learn the patient data from the views of our protocol, or modify the patient data, or impersonate a medical sensor. The inside attacker is a malicious data server or a coalition of two malicious data servers who know some secret keys in our system and attempt to learn the patient data.

3.2 Data Collection Protocol

There is an initial deployment phase between each medical sensor and each data server. For each

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/TDSC.2015.2406699, IEEE Transactions on Dependable and Secure Computing

medical sensor, three secret keys are pre-deployed and pre-shared with three data servers, respectively. Each secret key is used to create a secure channel between the sensor and one data server. In addition, one more secret key is pre-deployed in each sensor in order to generate random numbers. Note that different medical sensors are deployed with different secret keys.

When a medical sensor sends a sensitive numerical patient data ρ (e.g., temperature reading) to the distributed patient database, to prevent any data server from understanding the patient data and revealing the patient privacy (the inside attack), the medical sensor splits the patient data ρ (an integer) into three integers α, β, γ such that $\alpha+\beta+\gamma = \rho$ and sends them to the three data servers through three secure channels, respectively, as shown in Fig. 3.

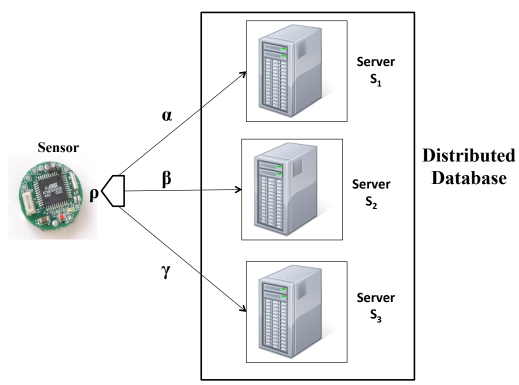

Fig. 3. Data Collection

Assume that the medical sensor sends a sequence of sensitive numerical patient data $\rho_1, \rho_2, \cdots$ (each has less than 32 bits) to the three data servers, it firstly generates a sequence of random numbers $a_1, b_1, a_2, b_2, \cdots$ (each has 40 bits) with SHA-3 [28] ($r = 40$ and $c = 160$) as shown in Fig. 4, where K is the random number generation secret key and the initial vector IV includes the current time stamp, the size of both K and IV is 80 bits.

Let $|\alpha_i|$ ($|\beta_i|$) be the first 32 bits of a_i (b_i). The sign of α_i (β_i) is positive if a_i (b_i) is even and otherwise negative. Then the medical sensor computes

$$\gamma_i = \rho_i - \alpha_i - \beta_i$$

for $i = 1, 2, \cdots$

Let A_i = {patient ID, data attribute, data unit}, the medical sensor sends $\{A_i, \alpha_i\}$ to S_1 through the secure channel for S_1, and $\{A_i, \beta_i\}$ to S_2 through the secure channel for S_2, and $\{A_i, \gamma_i\}$ to S_3 through the secure channel for S_3, for $i = 1, 2, \cdots$.

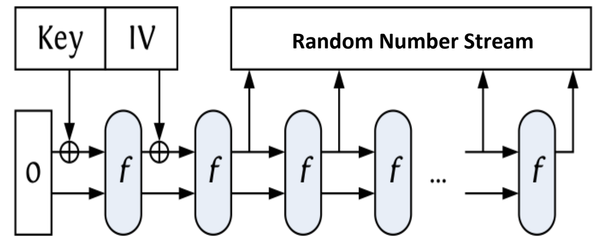

Fig. 4. Random Number Stream Generation

Each data server will create a database to store the patient data. The database structure looks like the patient's identity, the attribute of the data, the unit of the data, the share of the data and etc. For example, a record of the database in S_1 may look like {David Jones, temperature, celsius degree, $\alpha_i = 317481$, 12/12/2014, 9:31AM}.

As long as the three data server do not put their shares together, the privacy of the patient data can be protected. Note that our model assumes that at least one data server is not compromised.

Remark. The patient data may be decimal numbers with several digits after the point. In this case, the sensor should convert it to an integer and sends the shares of the data together with the unit of the data to three data servers, respectively.

3.3 Access Control Protocol

There is an initialization phase before any user (physician) can get access to the patient data. In this phase, the user generates a public and private key pair (pk, sk) for the Paillier cryptosystem [25] as described in Section 2.1 and a signature verification and signing key pair (pk^*, sk^*) for the Digital Signature Standard (DSS) [9]. For security reason, the size of N in the Paillier cryptosystem is required to be more than 1024 bits. Assume that there exists a Public Key Infrastructure (PKI), where there exists a Certificate Authority (CA) which certifies the public keys (pk, pk^*) for the user in a digital certificate. In addition, we assume that the user establishes three secure channel with three data servers, respectively.

To get access to the patient data, the user sends a request including the patient's identity, the data

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/TDSC.2015.2406699, IEEE Transactions on Dependable and Secure Computing

attribute, the signature of the user on the query, and the certificate of the user to the three data servers through the three secure channels, respectively.

Remark. We use the secure channels for the user to submit his queries because the patient's personal information in the queries needs to be protected against outside attackers.

If the user's request passes the signature verification and meets the access control policies, the three servers find the shares of the data α, β, γ according the patient's identity and the attribute of the data. Then the three data servers and the user run Algorithm 1.

Remark. We require each data server to verify the signature of the user and check the access control policies. The verification and check can be trusted because at least one data server is not compromised.

Algorithm 1 Patient Information Retrieval

Input: $\alpha, \beta, \gamma, pk, sk$

Output: $\rho = \alpha + \beta + \gamma$

1: The data server S_1 picks a random $r_1 \in \mathbb{Z}_N^*$ and computes

$$C_1 = Encrypt(\alpha, pk) = g^{\alpha} r_1^N (mod\ N^2)$$

and sends C_1 to the data server S_2.

2: The data server S_2 picks a random $r_2 \in \mathbb{Z}_N^*$ and computes

$$C_2 = Encrypt(\beta, pk) = g^{\beta} r_2^N (mod\ N^2)$$

and sends C_1C_2 to the data server S_3.

3: The data server S_3 picks a random $r_3 \in \mathbb{Z}_N^*$ and computes

$$C_3 = Encrypt(\gamma, pk) = g^{\gamma} r_3^N (mod\ N^2)$$

and replies $C_1C_2C_3$ to the user.

4: The user computes

$$\rho = Decrypt(C_1C_2C_3, sk)$$

5: **return** ρ

Due to the homomorphic properties of the Paillier cryptosystem, we have

$$\begin{aligned} C_1C_2C_3 &= E(\alpha, pk)E(\beta, pk)E(\gamma, pk) \\ &= (g^{\alpha} r_1^N)(g^{\beta} r_2^N)(g^{\gamma} r_3^N)(mod\ N^2) \\ &= g^{\alpha+\beta+\gamma}(r_1 r_2 r_3)^N (mod\ N^2) \\ &= E(\alpha + \beta + \gamma, pk) \end{aligned}$$

Therefore,

$$\rho = Decrypt(C_1C_2C_3, sk) = \alpha + \beta + \gamma$$

3.4 Statistical Analysis Protocols

Our system supports not only access control to the patient data but also privacy-preserving statistical analysis on the patient data for medical research, where the three data servers cooperate to help the medical researcher analyze the patient data without revealing the patient privacy.

3.4.1 Average Analysis Protocol

When a user queries the average of n patient data $x_1, x_2, \cdots, x_n$, where $x_i = \alpha_i + \beta_i + \gamma_i$ for $i = 1, 2, \cdots, n$ and $\alpha_i, \beta_i, \gamma_i$ are stored in the three data servers, respectively, he submits his query with his signature and certificate to the three servers. If the signature of the user is genuine and the access control policies permit the user to access the average of n patient data, the three servers and the user run Algorithm 2.

Algorithm 2 Average Computation

Input: $(\alpha_i, \beta_i, \gamma_i)$ for $i = 1, 2, \cdots, n,\ pk, sk$

Output: $\overline{x} = \sum_{i=1}^n x_i/n$

1: The data server S_1 picks a random $r_1 \in \mathbb{Z}_N^*$ and computes

$$C_1 = Encrypt(\sum_{i=1}^n \alpha_i, pk) = g^{\sum_{i=1}^n \alpha_i} r_1^N$$

and sends C_1 to the data server S_2.

2: The data server S_2 picks a random $r_2 \in \mathbb{Z}_N^*$ and computes

$$C_2 = Encrypt(\sum_{i=1}^n \beta_i, pk) = g^{\sum_{i=1}^n \beta_i} r_2^N$$

and sends C_1C_2 to the data server S_3.

3: The data server S_3 picks a random $r_3 \in \mathbb{Z}_N^*$ and computes

$$C_3 = Encrypt(\sum_{i=1}^n \gamma_i, pk) = g^{\sum_{i=1}^n \gamma_i} r_3^N$$

and replies $C_1C_2C_3$ to the user.

4: The user computes

$$\overline{x} = Decrypt(C_1C_2C_3, sk)/n$$

5: **return** $\overline{x}$

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/TDSC.2015.2406699, IEEE Transactions on Dependable and Secure Computing

Due to the homomorphic properties of the Paillier cryptosystem, we have

$$\begin{aligned} C_1C_2C_3 &= (g^{\sum \alpha_i} r_1^N)(g^{\sum \beta_i} r_2^N)(g^{\sum \gamma_i} r_3^N)(mod\ N^2) \\ &= g^{\sum(\alpha_i+\beta_i+\gamma_i)}(r_1r_2r_3)^N(mod\ N^2) \\ &= E(\sum x_i, pk) \end{aligned}$$

Therefore,

$$\overline{x} = Decrypt(C_1C_2C_3, sk)/n = \sum_{i=1}^{n} x_i/n$$

3.4.2 Correlation Analysis Protocol

When a user queries the correlation of two measures of patient data, $X = (x_1, x_2, \cdots, x_n)$ and $Y = (y_1, y_2, \cdots, y_n)$, where (x_i, y_i) belongs to one patient and $x_i = \alpha_i + \beta_i + \gamma_i$ and $y_i = \alpha'_i + \beta'_i + \gamma'_i$ for $i = 1, 2, \cdots, n$, and $(\alpha_i, \alpha'_i), (\beta_i, \beta'_i), (\gamma_i, \gamma'_i)$ are stored in the three data servers, respectively, he submits his query with his signature and certificate to the three servers. If the signature of the user is genuine and the access control policies permit the user to access the correlation of two measures X and Y for the n patient data, the three servers and the user run Algorithm 3.

In Algorithm 3, r_i, r'_i are randomly chosen by the data server S_i from $\mathbb{Z}_N$.

Due to the homomorphic properties of the Paillier cryptosystem, we have

$$\begin{aligned} C'_{i1}C'_{i2}C'_{i3} &= (C_{i1}C_{i2}C_{i3})^{\alpha'_i+\beta'_i+\gamma'_i}(r'_1r'_2r'_3)^N \\ &= (g^{\alpha_i+\beta_i+\gamma_i}(r_1r_2r_3)^N)^{y_i}(r'_1r'_2r'_3)^N \\ &= g^{x_iy_i}((r_1r_2r_3)^{y_i}r'_1r'_2r'_3)^N \\ &= E(x_iy_i, pk) \end{aligned}$$

Therefore,

$$C = E(\sum_{i=1}^{n} x_iy_i, pk)$$

$$s_{xy} = Decrypt(C, pk) = \sum_{i=1}^{n} x_iy_i$$

In Algorithm 3, let $\alpha'_i = \alpha_i$, $\beta'_i = \beta_i$ and $\gamma'_i = \gamma_i$, the user can obtain $s_{x^2} = \sum_{i=1}^n x_i^2$. Let $\alpha_i = \alpha'_i$, $\beta_i = \beta'_i$ and $\gamma_i = \gamma'_i$, the user can obtain $s_{y^2} = \sum_{i=1}^n y_i^2$. In addition, by Algorithm 2, the user can obtain $s_x = \sum_{i=1}^n x_i$ and $s_y = \sum_{i=1}^n y_i$.

Finally, the user can compute the correlation of the two measures X and Y, namely,

$$r_{xy} = \frac{n\sum x_iy_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}}$$

Algorithm 3 Product Computation

Input: $(\alpha_i, \beta_i, \gamma_i), (\alpha'_i, \beta'_i, \gamma'_i)$ for $i = 1, 2, \cdots, n$ pk, sk

Output: $s_{xy} = \sum_{i=1}^n x_iy_i$

1: Let $C = 1$
2: For $i = 1$ to n
3: S_1 computes and sends C_{i1} to S_2.
$$C_{i1} = Encrypt(\alpha_i, pk) = g^{\alpha_i}r_1^N(mod\ N^2)$$
4: S_2 computes and sends $C_{i1}C_{i2}$ to S_3.
$$C_{i2} = Encrypt(\beta_i, pk) = g^{\beta_i}r_2^N(mod\ N^2)$$
5: S_3 computes and sends $C_{i1}C_{i2}C_{i3}$ to S_1, S_2.
$$C_{i3} = Encrypt(\gamma_i, pk) = g^{\gamma_i}r_3^N(mod\ N^2)$$
6: S_1 computes and sends C'_{i1} to S_2.
$$C'_{i1} = (C_{i1}C_{i2}C_{i3})^{\alpha'_i}{r'_1}^N(mod\ N^2)$$
7: S_2 computes and sends $C'_{i1}C'_{i2}$ to S_3.
$$C'_{i2} = (C_{i1}C_{i2}C_{i3})^{\beta'_i}{r'_2}^N(mod\ N^2)$$
8: S_3 computes
$$C'_{i3} = (C_{i1}C_{i2}C_{i3})^{\gamma'_i}{r'_3}^N(mod\ N^2)$$
$$C \leftarrow C'_{i1}C'_{i2}C'_{i3}C(mod\ N^2)$$
9: End For i
10: S_3 replies C to the user.
11: The user computes
$$s_{xy} = Decrypt(C, sk)$$
12: **return** s_{xy}

In Algorithm 3, if the user can get the intermediate encryption results, e.g., C_{i1}, C_{i2} and C_{i3}, he can obtain the individual patient data, e.g., x_i, because he has the decryption key. To prevent the user from learning the individual patient data, we provide an improved solution on the basis of a combination of the Paillier and the ElGamal cryptosystems.

Like the Paillier cryptosystem [25], the three data servers randomly choose large primes p, q and compute $N = pq$. Like the ElGamal cryptosystem [10], the three data servers choose a generator $g = a^{p(p-1)(q-1)}(mod\ N^2)$ with order of q, where a is a random integer and $g \neq 1$. Each of the three data server and the user randomly chooses the private key $sk_i \in \mathbb{Z}_q^*$ and computes the public key

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/TDSC.2015.2406699, IEEE Transactions on Dependable and Secure Computing

$pk_i = g^{sk_i}(mod\ N^2)$, where $i = 1, 2, 3, 4$. Then the three data servers and the user run Algorithm 4.

Algorithm 4 Improved Product Computation

Input: $(\alpha_i, \beta_i, \gamma_i), (\alpha'_i, \beta'_i, \gamma'_i)$ for $i = 1, 2, \cdots, n$ $N, g, pk_i, sk_i, i = 1, 2, 3, 4$

Output: $s_{xy} = \sum_{i=1}^{n} x_i y_i$

Let $A = B = 1, g_1 = (N+1)^p, pk = \prod_{i=1}^{4} pk_i$

For $i = 1$ to n

S_1 computes and sends (A_{i1}, B_{i1}) to S_2.

$$A_{i1} = g^{r_1}(mod\ N^2), B_{i1} = g_1^{\alpha_i} pk^{r_1}(mod\ N^2)$$

S_2 computes and sends $(A_{i1}A_{i2}, B_{i1}B_{i2})$ to S_3.

$$A_{i2} = g^{r_2}(mod\ N^2), B_{i2} = g_1^{\beta_i} pk^{r_2}(mod\ N^2)$$

S_3 computes and sends (A_i, B_i) to S_1, S_2.

$$A_{i3} = g^{r_3}(mod\ N^2), B_{i3} = g_1^{\gamma_i} pk^{r_3}(mod\ N^2)$$

$$A_i = A_{i1}A_{i2}A_{i3}(mod\ N^2)$$

$$B_i = B_{i1}B_{i2}B_{i3}(mod\ N^2)$$

S_1 computes and sends (A'_{i1}, B'_{i1}) to S_2.

$$A'_{i1} = A_i^{\alpha'_i} g^{r'_1}(mod\ N^2), B'_{i1} = B_i^{\alpha'_i} pk^{r'_1}(mod\ N^2)$$

S_2 computes and sends $(A'_{i1}A'_{i2}, B'_{i1}B'_{i2})$ to S_3.

$$A'_{i2} = A_i^{\beta'_i} g^{r'_2}(mod\ N^2), B'_{i2} = B_i^{\beta'_i} pk^{r'_2}(mod\ N^2)$$

S_3 computes

$$A'_{i3} = A_i^{\gamma'_i} g^{r'_3}(mod\ N^2), B'_{i3} = B_i^{\gamma'_i} pk^{r'_3}(mod\ N^2)$$

$$A \leftarrow A'_{i1}A'_{i2}A'_{i3}A(mod\ N^2)$$

$$B \leftarrow B'_{i1}B'_{i2}B'_{i3}B(mod\ N^2)$$

End For i

S_3 sends A to S_1, S_2

S_1 sends $D_1 = A^{sk_1}(mod\ N^2)$ to S_3.

S_2 sends $D_2 = A^{sk_2}(mod\ N^2)$ to S_3.

S_3 computes $D_3 = A^{sk_3}(mod\ N^2)$ and replies the user with (A, C), where

$$C = B/(D_1D_2D_3)(mod\ N^2)$$

The user computes

$$D = A^{sk_4}(mod\ N^2), s_{xy} = L(C/D(mod\ N^2))/p$$

return s_{xy}

In Algorithm 4, r_i, r'_i are randomly chosen by the data server S_i from $\mathbb{Z}_q^* = \{1, 2, \cdots, q-1\}$.

Due to the homomorphic properties of the ElGamal cryptosystem, we have

$$\begin{aligned} A'_{i1}A'_{i2}A'_{i3} &= (A_i^{\alpha'_i} g^{r'_1})(A_i^{\beta'_i} g^{r'_2})(A_i^{\gamma'_i} g^{r'_2}) \\ &= (A_{i1}A_{i2}A_{i3})^{\alpha'_i+\beta'_i+\gamma'_i} g^{r'_1+r'_2+r'_3} \\ &= (g^{r_1}g^{r_2}g^{r_3})^{y_i} g^{r'_1+r'_2+r'_3} \\ &= g^{(r_1+r_2+r_3)y_i+(r'_1+r'_2+r'_3)} \end{aligned}$$

$$\begin{aligned} B'_{i1}B'_{i2}B'_{i3} &= (B_i^{\alpha'_i} pk^{r'_1})(B_i^{\beta'_i} pk^{r'_2})(B_i^{\gamma'_i} pk^{r'_2}) \\ &= (B_{i1}B_{i2}B_{i3})^{\alpha'_i+\beta'_i+\gamma'_i} pk^{r'_1+r'_2+r'_3} \\ &= (g_1^{\alpha_i} pk^{r_1} g_1^{\beta_i} pk^{r_2} g_1^{\gamma_i} pk^{r_3})^{y_i} pk^{r'_1+r'_2+r'_3} \\ &= g_1^{x_i y_i} pk^{(r_1+r_2+r_3)y_i+(r'_1+r'_2+r'_3)} \end{aligned}$$

Therefore,

$$(A, B) = (g^r, g_1^{\sum_{i=1}^{n} x_i y_i} pk^r)$$

for some r, which is an ElGamal encryption of $\sum_{i=1}^{n} x_i y_i$. Furthermore, we have

$$\begin{aligned} C/D &= B/(D_1D_2D_3D) \\ &= g_1^{\sum_{i=1}^{n} x_i y_i} pk^r / (\prod_{i=1}^{4} (g^r)^{sk_1}) \\ &= (1+N)^{p\sum_{i=1}^{n} x_i y_i} \\ &= 1 + (p\sum_{i=1}^{n} x_i y_i)N(mod\ N^2) \end{aligned}$$

Therefore,

$$\begin{aligned} s_{xy} &= L(C/D(mod\ N^2))/p \\ &= \frac{1 + (p\sum_{i=1}^{n} x_i y_i)N - 1}{Np} \\ &= \sum_{i=1}^{n} x_i y_i \end{aligned}$$

Note that $\sum_{i=1}^{n} x_i y_i$ is usually much less than q even for large n because $x_i y_i$ is about 64 bits, but q is required to be at least 512 bits. Therefore, $p\sum_{i=1}^{n} x_i y_i$ is much less than $N = pq$.

3.4.3 Variance Analysis Protocol

When a user queries the variance of n patient data $x_1, x_2, \cdots, x_n$, where $x_i = \alpha_i + \beta_i + \gamma_i$ for $i = 1, 2, \cdots, n$ and $\alpha_i, \beta_i, \gamma_i$ are stored in the three data servers, respectively, he submits his query with his signature and certificate to the three servers. If the signature of the user is genuine and the access control policies permit the user to access the variance of n patient data, the user runs Algorithm 2 and Algorithm 3 or 4 with the three data servers to get $\overline{x} = \sum_{i=1}^{n} x_i/n$ and $s_{x^2} = \sum_{i=1}^{n} x_i^2$, respectively. Then the user computes the variance

$$
\begin{aligned}
v &= \sqrt{\frac{\sum_{i=1}(x_i - \overline{x})^2}{n-1}} \\
&= \sqrt{\frac{\sum_{i=1} x_i^2 + (1-2n)\overline{x}^2}{n-1}}
\end{aligned}
$$

3.4.4 Regression Analysis Protocol

When a user queries the linear relationship $y = kx + b$ of two measures of patient data, $X = (x_1, x_2, \cdots, x_n)$ and $Y = (y_1, y_2, \cdots, y_n)$, where (x_i, y_i) belongs to one patient and $x_i = \alpha_i + \beta_i + \gamma_i$ and $y_i = \alpha'_i + \beta'_i + \gamma'_i$ for $i = 1, 2, \cdots, n$, and $(\alpha_i, \alpha'_i), (\beta_i, \beta'_i), (\gamma_i, \gamma'_i)$ are stored in the three data servers, respectively, he submits his query with his signature and certificate to the three servers. If the signature of the user is genuine and the access control policies permit the user to access the linear relationship of two measures X and Y for the n patient data, the user runs Algorithm 2 and Algorithm 3 or 4 with the three data servers to obtain $s_x = \sum_{i=1}^{n} x_i$, $s_y = \sum_{i=1}^{n} y_i$, $s_{x^2} = \sum_{i=1}^{n} x_i^2$, $s_{xy} = \sum_{i=1}^{n} x_i y_i$, respectively. Then the user computes

$$
\begin{aligned}
k &= \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2} \\
b &= \frac{\sum y_i - k \sum x_i}{n}
\end{aligned}
$$

4 Security and Privacy Analysis

4.1 Security Analysis

In our architecture as shown in Fig. 2, there are three parts of communications as follows.

- The communications between the medical sensors and the three servers;
- The communications between the user (e.g., physicians or medical professional) and three servers;
- The communications among the three servers.

In our solution, the communication between each medical sensor and each data server is through a secure channel, which is implemented by a secret-key cryptosystem. The patient data over the secure channel is encrypted with the secret key pre-shared between the sensor and the data server. Without the secret key, the attacker cannot eavesdrop the patient data.

Because the medical sensors are usually low-power and low-cost, we can choose the lightweight encryption scheme and the message authentication code (MAC) generation scheme proposed in [31] for the secure channel. Both schemes are built on the smallest version of the SHA-3 with $r = 40, c = 160$, which can provide a security level sufficient for many applications. In addition, the random numbers in our data collection protocol are also generated with SHA-3 as shown in Fig. 4.

By the lightweight encryption scheme and the MAC generation scheme [31], we can achieve data confidentiality, authenticity and integrity between each medical sensor and each data server.

In our solution, the communication between the user and each data server is also through a secure channel. Because the three data servers and the user's computing device are usually much more powerful in computation and communication than the medical sensors, we choose the Advanced Encryption Standard (AES) [1] for the secure channel. The secret key can be established by a public key cryptosystem, such as the Diffie-Hellman key exchange protocol [8] or RSA [27]. The public keys of users and three data servers are certified by a Certificate Authority (CA) in a Public Key Infrastructure (PKI). In addition, we choose the Digital Signature Standard (DSS) [9] for data authentication and integrity.

By AES and DSS, we can achieve data confidentiality, authenticity and integrity between the user and each data server.

In our solution, the communications among three data servers can be also through secure channels. Like the secure communication between the user and the data servers, any two of the three data servers can establish a secret key with a public key cryptosystem. Then the communication between the two data servers can be encrypted with AES based on the secret key.

In our model, the three data servers are assumed to be semi-honest. Otherwise, the user can never obtain correct patient data and statistical analysis results. To ensure data authenticity and integrity in the communications among the three data servers, we choose the Digital Signature Standard (DSS) [9].

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/TDSC.2015.2406699, IEEE Transactions on Dependable and Secure Computing

4.2 Privacy Analysis

In the data collection protocol, the medical sensor splits the patient data into three numbers and sends them to the three data servers, respectively, through secure channels. Two of the three numbers are generated by SHA-3 with a secret key K and an initial vector IV as shown in Fig. 4. The key is pre-deployed and known to the medical sensor only. Any inside attacker, including each data server, cannot guess the two random numbers without the secret key. As long as at least one data server is not compromised by the inside attack, none can reveal the patient data during data collection.

In the access control protocol (Algorithm 1) and the statistical analysis protocols (Algorithms 2 and 3), the patient data is always encrypted by the public key of the user. Without the private key of the user, even if two data servers are compromised by the inside attacks, the attacker can never obtain the patient data. Algorithms 1-3 are useful when the user is permitted to get access the patient data, but does not have a secure environment to protect patient data in local statistical analysis.

In Algorithm 4, all intermediate statistical data are encrypted by the common public key $pk = \prod_{i=1}^{4} pk_i$. Because p and q are public, the user may attempt to decrypt the encrypted intermediate data by the decryption manner of the Paillier cryptosystem, e.g., raising B_{ij}, B_i, B'_{ij}, B to the power of q to remove the effect of pk^r. However,

$$\begin{aligned} B_i^q &= (g_1^{x_i} pk^{r_1+r_2+r_3})^q (mod\ N^2) \\ &= (N+1)^{pqx_i} (mod\ N^2) \\ &= 1 + x_i pqN (mod\ N^2) \\ &= 1\ (mod\ N^2) \end{aligned}$$

In the same way, we can see that $B_{ij}^q = {B'_{ij}}^q = B^q = 1(mod\ N^2)$. Therefore, this attack cannot get any information about the patient data. In addition, because $A_{ij}^q = A_i^q = {A'_{ij}}^q = A^q = 1$, any attacker cannot determine the random exponents in A_{ij}, A_i, A'_{ij}, A like the Paillier decryption.

Even if the user can get the encrypted intermediate data, he cannot decrypt it without cooperation with all three data servers. Note that we assume that at least one data server is not compromised by the inside attack. Until the end of the algorithm, the user is not allowed to decrypt the final statistical result. Therefore, Algorithm 4 can be used when the user is not permitted to know the individual patient data in the statistical analysis.

5 Performance Analysis

In our data collection protocol, we can use the lightweight encryption scheme and MAC generation scheme proposed in [31]. In addition, our random number stream generation scheme is also based on SHA-3. All security mechanisms in the sensor can be implemented with the same SHA-3. This design is suitable for wireless sensor networks where area is particularly important since it determines the cost of the sensors.

Our access control protocol is built on the Paillier cryptosystem [25], where the dominated computation is the modular exponentiation, i.e., $a^x (mod\ N^2)$ where $x \in \mathbb{Z}_N^*$. In Algorithm 1, each data server computes two modular exponentiations and exchange $|N^2| = 2|N|$ bits, where $|N|$ is the length of N. The user computes one modular exponentiation and exchanges $2|N|$ bits.

Our average analysis protocol is also built on the Paillier cryptosystem. In Algorithm 2, the computation and communication complexities for each data server and the user are the same as those in Algorithm 1.

In our correlation analysis protocol, with the help of the three data servers, the user compute s_x, s_y by Algorithm 2 and compute s_{xy}, s_{x^2}, s_{y^2} by Algorithm 3 or Algorithm 4 and then computes the correlation r_{xy}. Algorithm 3 is based on the Paillier cryptosystem and can be used when the user is permitted to access the individual patient data. Algorithm 4 is based on the combination of the ElGamal and Paillier cryptosystems and can be used when the user is not permitted to access any individual patient data.

In Algorithm 3, each data server computes $4n$ modular exponentiations and exchanges $8|N|n$ bits in average, where n is the number of patients. The user computes one modular exponentiation and exchanges $2|N|$ bits. In Algorithm 4, each data server computes $8n$ modular exponentiations and exchanges $20|N|n$ bits in average. The user computes one modular exponentiation and exchange $4|N|$ bits. The computations of modular exponentiation in Algorithms 3 and 4 are different. In Algorithm 3, we compute $a^x (mod\ N^2)$ where $x \in \mathbb{Z}_N$, denoted as Exp. In Algorithm 4, we compute

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/TDSC.2015.2406699, IEEE Transactions on Dependable and Secure Computing

Protocols	Each Data Server			User		
	Comp.	Comm.	Time	Comp.	Comm.	Time
Access Control	2 Exp.	$2\ \lvert N\rvert$	2.68 ms	1 Exp.	$2\ \lvert N\rvert$	1.34 ms
Average Analysis	2 Exp.	$2\ \lvert N\rvert$	2.68 ms	1 Exp.	$2\ \lvert N\rvert$	1.34 ms
Correlation Analysis 1	$(12n+4)$ Exp.	$(24n+4)\ \lvert N\rvert$	2.68 m	5 Exp.	$10\ \lvert N\rvert$	6.7 ms
Variance Analysis 1	$(4n+2)$ Exp.	$(8n+2)\ \lvert N\rvert$	0.89 m	2 Exp.	$4\ \lvert N\rvert$	2.68 ms
Regression Analysis 1	$(8n+4)$ Exp.	$(16n+2)\lvert N\rvert$	1.79 m	4 exp.	$8\ \lvert N\rvert$	2.68 ms
Correlation Analysis 2	$24n$ exp. + 4 Exp	$(60n+4)\ \lvert N\rvert$	2.68 m	3 exp.+ 2 Exp.	$16\ \lvert N\rvert$	4.69 ms
Variance Analysis 2	$8n$ exp. + 2 Exp.	$(20n+2)\ \lvert N\rvert$	0.89 m	1 exp. +1 Exp.	$6\ \lvert N\rvert$	2.01 ms
Regression Analysis 2	$16n$ exp. + 4 Exp.	$(40n+4)\lvert N\rvert$	1.79 m	2 exp. +2 Exp.	$12\ \lvert N\rvert$	4.02 ms

TABLE 1
Performance Analysis

$a^x(mod\ N^2)$ where $x \in \mathbb{Z}_p$, denoted as exp. It is estimated that Exp.$\approx$ 2$\cdot$exp.

For simplicity, our statistical analysis protocols based on Algorithms 3 and 4 are denoted as statistical analysis 1 and 2, respectively. In our correlation analysis 1, each data server computes $3 \cdot 4n + 2 \cdot 2 = 12n + 4$ Exp. and exchanges $3 \cdot 8|N|n + 2 \cdot 2|N| = (24n + 4)|N|$ bits in average. The user computes 5 Exp. and exchange $10|N|$ bits. In our correlation analysis 2, each data server computes $3 \cdot 8n$ exp. +$2 \cdot 2$ Exp. =$24n$ exp. + 4 Exp. and exchanges $3 \cdot 20|N|n + 2 \cdot 2|N| = (60n+4)|N|$ bits in average. The user computes 3 exp. +2 Exp. and exchange $16|N|$ bits.

In our variance analysis protocol, the user computes $\overline{x}$ by Algorithm 2 and s_{x^2} by Algorithm 3 or 4 and then computes the variance. In the variance analysis 1, each data server computes $4n + 2$ Exp. and exchanges $(8n + 2)|N|$ in average. The user computes 2 Exp. and exchange $4|N|$ bits. In the variance analysis 2, each data server computes $8n$ exp. + 2 Exp. and exchanges $(20n + 2)|N|$ in average. The user computes 1 exp. + 1 Exp. and exchange $6|N|$ bits.

In our regression analysis protocol, the user computes s_x, s_y by Algorithm 2 and s_{x^2}, s_{xy} by Algorithm 3 or 4 and then determines the line. In the regression analysis 1, each data server computes $8n+4$ Exp. and exchanges $(16n+4)|N|$ in average. The user computes 4 Exp. and exchange $8|N|$ bits. In the regression analysis 2, each data server computes $16n$ exp. + 4 Exp. and exchanges $(40n+4)|N|$ in average. The user computes 2 exp. + 2 Exp. and exchange $12|N|$ bits.

The performance of all of our protocols are summarized in TABLE 1.

With reference to Crypto++ 5.6.0 Benchmarks [5], a modular exponentiation with a 1024-bit modulus takes about 0.67 milliseconds. Note that it is coded in C++, compiled with Microsoft Visual C++ 2005 SP1 (whole program optimization, optimize for speed), and runs on an AMD Opteron 8354 2.2 GHz processor under Linux. Based on this result, in the Paillier cryptosystem with a 1024-bit modulus, one modular exponentiation takes about 1.34 milliseconds if we use the Chinese remainder theorem to compute $a^x(mod\ N^2) = a^x(mod\ p^2q^2)$. Taking the most expensive correlation analysis 1 for $n = 10,000$ as an example, assuming that the three data servers are connected by a 100-gigabit network, the total computation and communication times are estimated to be 2.68 minutes and 1 second, respectively. Our algorithms support parallel computation. If each data server runs 10 computers in parallel, the total running time of our correlation analysis 1 for $n = 10,000$ can be reduced to 16 seconds. The estimated time for access control and other data analyses for $n = 10,000$ are listed in TABLE 1 as well.

6 Conclusion

In this paper, we have investigated the security and privacy issues in the medical sensor data collection, storage and queries and presented a complete solution for privacy-preserving medical sensor network. To secure the communication between medical sensors and data servers, we used the lightweight encryption scheme and MAC generation scheme based on SHA-3 proposed in [31]. To keep the privacy of the patient data, we proposed a new data collection protocol which splits the patient data into three numbers and stores them in three data servers, respectively. As long as one data server is not

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/TDSC.2015.2406699, IEEE Transactions on Dependable and Secure Computing

compromised, the privacy of the patient data can be preserved. For the legitimate user (e.g., physician) to access the patient data, we proposed an access control protocol, where three data servers cooperate to provide the user with the patient data, but do not know what it is. For the legitimate user (e.g., medical researcher) to perform statistical analysis on the patient data, we proposed some new protocols for average, correlation, variance and regression analysis, where the three data servers cooperate to process the patient data without disclosing the patient privacy and then provide the user with the statistical analysis results. Security and privacy analysis has shown that our protocols are secure against both outside and inside attacks as long as one data server is not compromised. Performance analysis has shown that our protocols are practical as well.

Unlike [31], our solution can preserve the patient data privacy as long as one of three data server is not compromised. [31] requires that the number of the compromised data servers is at most one.

REFERENCES

[1] Advanced Encryption Standard (AES). FIPS PUB 197, November 26, 2001. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[2] P. Belsis and G. Pantziou. A k-anonymity privacy-preserving approach in wireless medical monitoring environments. Journal Personal and Ubiquitous Computing, 18(1): 61-74, 2014.

[3] D. Bogdanov, S. Laur, J. Willemson. Sharemind: a Framework for Fast Privacy-Preserving Computations. In Proc. ESORICS'08, pages 192-206, 2008.

[4] R. Chakravorty. A Programmable Service Architecture for Mobile Medical Care. In Proc. 4th Annual IEEE International Conference on Pervasive Computing and Communication Workshop (PERSOMW'06), Pisa, Italy, 13-17 March 2006.

[5] Crypto++ 5.6.0 Benchmarks. http://www.cryptopp.com / benchmarks.html.

[6] J. Daemen, G. Bertoni, M. Peeters, G. V. Assche, Permutation-based Encryption, Authentication and Authenticated Encryption, DIAC'12, Stockholm, 6 July 2012. Available at http://www.hyperelliptic.org/DIAC/slides /Permutation-DIAC2012.pdf

[7] S. Dagtas, G. Pekhteryev, Z. Sahinoglu, H. Cam, N. Challa. Real-Time and Secure Wireless Health Monitoring. Int. J. Telemed. Appl. 2008, doi: 10.1155/2008/135808.

[8] W. Diffie and M. Hellman. New Directions in Cryptography. IEEE Transactions on Information Theory, 22 (6): 644-654, 1976.

[9] Digital Signature Standard (DSS). FIPS PUB 186-4, July 2013, http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf

[10] T. ElGamal. A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. IEEE Transactions on Information Theory, 31 (4): 469-472, 1985.

[11] D. He, S. Chan and S. Tang. A Novel and Lightweight System to Secure Wireless Medical Sensor Networks. IEEE Journal of Biomedical and Health Informatics, 18 (1): 316-326, 2014.

[12] F. Hu, M. Jiang, M. Wagner, D. C. Dong. Privacy-Preserving Telecardiology Sensor Networks: Toward a Low-Cost Portable Wireless Hardware/Software Codesign. IEEE Trans. Inform. Tech. Biomed, 11: 619-627, 2007.

[13] Y. M. Huang, M. Y. Hsieh, H. C. Hung, J. H. Park. Pervasive, Secure Access to a Hierarchical Sensor-Based Healthcare Monitoring Architecture in Wireless Heterogeneous Networks. IEEE J. Select. Areas Commun. 27: 400-411, 2009.

[14] J. Ko, J. H. Lim, Y. Chen, R. Musaloiu-E., A. Terzis, G. M. Masson. MEDiSN: Medical Emergency Detection in Sensor Networks. ACM Trans. Embed. Comput. Syst. 10: 1-29, 2010.

[15] P. Kumar, Y. D. Lee, H. J. Lee. Secure Health Monitoring Using Medical Wireless Sensor Networks. In Proc. 6th International Conference on Networked Computing and Advanced Information Management, pages 491-494, Seoul, Korea, 16-18 August 2010.

[16] P. Kumar and H. J. Lee. Security Issues in Healthcare Applications Using Wireless Medical Sensor Networks: A Survey. Sensors 12: 55-91, 2012.

[17] X. H. Le, M. Khalid, R. Sankar, S. Lee. An Efficient Mutual Authentication and Access Control Scheme for Wireless Sensor Network in Healthcare. J. Networks 27: 355-364, 2011.

[18] H. J. Lee and K. Chen. A New Stream Cipher for Ubiquitous Application. In Proc. ICCIT'07, South Korea, 2007.

[19] X. Lin, R. Lu, X. Shen, Y. Nemoto, N. Kato. SAGE: A Strong Privacy-Preserving Scheme Against Global Eavesdropping for eHealth System. IEEE J. Select. Area Commun. 27: 365-378, 2009.

[20] D. Malan, T. F. Jones, M. Welsh, S. Moulton. CodeBlue: An Ad-Hoc Sensor Network Infrastructure for Emergency Medical Care. In Proc. MobiSys 2004 Workshop on Applications of Mobile Embedded Systems (WAMES'04), Boston, MA, USA, 6-9 June 2004.

[21] K. Malasri, L. Wang. Design and Implementation of Secure Wireless Mote-Based Medical Sensor Network. Sensors 9: 6273-6297, 2009.

[22] J. Misic, V. Misic. Enforcing Patient Privacy in Healthcare WSNs Through Key Distribution Algorithms. Secur. Commun. Network 1: 417-429, 2008.

[23] K. Montgomery, C. Mundt, G. Thonier, A. Tellier, U. Udoh, V. Barker, R. Ricks, L. Giovangrandi, P. Davies, Y. Cagle, J. Swain, J. Hines, G. Kovacs. Lifeguard - A Personal Physiological Monitor for Extreme Environments. In Proc. 26th Annual International Conference of the IEEE EMBS, pages 2192-2195, San Francisco, CA, USA, 1-5 September 2004.

[24] J. Ng, B. Lo, O. Wells, M. Sloman, N. Peters, A. Darzi, C. Toumazou, G. Z. Yang. Ubiquitous Monitoring Environment for Wearable and Implantable Sensors (UbiMon). In Proc. 6th International Conference on Ubiquitous Computing (UbiComp'04), Nottingham, UK, 7-14 September 2004.

[25] P. Paillier. Public-key Cryptosystems Based on Composite Degree Residuosity Classes. In Proc. EUROCRYPT'99, pages 223-238, 1999.

[26] S. Raazi, H. Lee, S. Lee, Y. K. Lee. BARI+: A Biometric Based Distributed Key Management Approach for Wireless Body Area Networks. Sensors 10: 3911-3933, 2010.

[27] R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM, 21 (2): 120-126, 1978.

[28] SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. Draft FIPS PUB 202, May 2014. http:csrc.nist.govpublicationsdraftsfips202fips_202_draft.pdf

[29] A. B. Waluyo, I. Pek, X. Chen, W.-S.Yeoh. Design and Evaluation of Lightweight Middleware for Personal Wireless Body Area Network. Pers. Ubiquit. Comput, 13: 509-525, 2009.

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/TDSC.2015.2406699, IEEE Transactions on Dependable and Secure Computing

[30] A. Wood, G. Virone, T. Doan, Q. Cao, L. Selavo, Y. Wu, L. Fang, Z. He, S. Lin, J. Stankovic. ALARM-NET: Wireless Sensor Networks for Assisted-Living and Residential Monitoring. Technical Report CS-2006-01; Department of Computer Science, University of Virginia: Charlottesville, VA, USA, 2006.

[31] X. Yi, J. Willemson, F. Nat-Abdesselam. Privacy-Preserving Wireless Medical Sensor Network. In Proc. TrustCom'13, pages 118-125, 2013.

[32] H. Zhao, J. Qin, and J. Hu. An Energy Efficient Key Management Scheme for Body Sensor Networks. IEEE Transactions on Parallel and Distributed Systems, 24 (11): 2202-2210, 2013.

[33] B. Zoltak. An Efficient Message Authentication Scheme for Stream Cipher. Cryptology ePrint Archive 2004.

PLACE PHOTO HERE

Xun Yi is currently a Professor with School of Computer Science and IT, RMIT University, Australia. His research interests include applied cryptography, computer and network security, mobile and wireless communication security, and privacy-preserving data mining. He has published more than 150 research papers in international journals, such as IEEE Trans. Knowledge and Data Engineering, IEEE Trans. Wireless Communication, IEEE Trans. Dependable and Secure Computing, IEEE Trans. Circuit and Systems, and conference proceedings. He has ever undertaken program committee members for more than 20 international conferences. Since 2014, he has been an Associate Editor for IEEE Trans. Dependable and Secure Computing.

PLACE PHOTO HERE

Athman Bouguettaya received the PhD degree in computer science from the University of Colorado at Boulder in 1992. He is the head of the School of Computer Science and Information Technology at RMIT, Melbourne, Australia. He was a science leader at the CSIRO ICT Centre, Canberra, Australia. He was also previously a tenured faculty member in the Computer Science Department at the Virginia Polytechnic Institute and State University (commonly known as Virginia Tech). He currently holds adjunct professorships at the Australian National University, Canberra, the University of Queensland, Brisbane, Australia, and Macquarie University, Sydney, Australia. He is on the editorial boards of several journals, including the VLDB Journal, the Distributed and Parallel Databases Journal, the International Journal of Cooperative Information Systems, and the IEEE Transactions on Services Computing. He has published more than 130 articles in journals and conferences in the area of databases and service computing (e.g., IEEE Transactions on Knowledge and Data Engineering, the ACM Transactions on the Web, the International Journal on Very Large Data Bases, SIGMOD, ICDE, VLDB, and EDBT). His current research interests include the foundations of web service management systems. He is a fellow of the IEEE.

PLACE PHOTO HERE

Dimitrios Georgakopoulos is currently a Professor with School of Computer Science and IT, RMIT University, Australia. Before joining RMIT, he was a Research Director at the CSIRO ICT Centre where he headed the Information Engineering Laboratory that is based in Canberra and Sydney. Dimitrios is also an Adjunct Professor at the Australian National University. Before coming to CSIRO in October 2008, Dimitrios held research and management positions in several industrial laboratories in the US. From 2000 to 2008, he was a Senior Scientist with Telcordia, where he helped found Telcordia's Research Centers in Austin, Texas, and Poznan, Poland. From 1997 to 2000, Dimitrios was a Technical Manager in the Information Technology organization of Microelectronics and Computer Corporation (MCC), and the Chief Architect of MCC's Collaboration Management Infrastructure (CMI) consortial project. From 1990-1997, Dimitrios was a Principal Scientist at GTE (currently Verizon) Laboratories Inc. Dimitrios has received a GTE (Verizon) Excellence Award, two IEEE Computer Society Outstanding Paper Awards, and was nominated for the Computerworld Smithsonian Award in Science. He has published more than one hundred journal and conference papers.

PLACE PHOTO HERE

Andy Song is currently a Lecturer with School of Computer Science and IT, RMIT University, Australia. His research interests include machine vision and texture analysis, supervised learning and classification, genetic programming, evolutionary computation, prediction and learning agents.

PLACE PHOTO HERE

Jan Wilimson is currently a senior researcher with Cybernetica, Tartu, 51003, Estonia. His research interests include practical data security, secure multi-party computations, security issues of digital document formats, and e-voting.