Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 581426, 9 pages
http://dx.doi.org/10.1155/2014/581426

Research Article

Preserving Differential Privacy for Similarity Measurement in Smart Environments

Kok-Seng Wong and Myung Ho Kim

School of Computer Science and Engineering, Soongsil University, Information Science Building, Sangdo-dong, Dongjak-gu, Seoul 156-743, Republic of Korea

Correspondence should be addressed to Myung Ho Kim; kmh@ssu.ac.kr

Received 5 April 2014; Accepted 24 June 2014; Published 15 July 2014

Academic Editor: Jong-Hyuk Park

Advances in both sensor technologies and network infrastructures have encouraged the development of smart environments to enhance people's life and living styles. However, collecting and storing user's data in the smart environments pose severe privacy concerns because these data may contain sensitive information about the subject. Hence, privacy protection is now an emerging issue that we need to consider especially when data sharing is essential for analysis purpose. In this paper, we consider the case where two agents in the smart environment want to measure the similarity of their collected or stored data. We use similarity coefficient function ($\mathcal{F}_{SC}$) as the measurement metric for the comparison with differential privacy model. Unlike the existing solutions, our protocol can facilitate more than one request to compute $\mathcal{F}_{SC}$ without modifying the protocol. Our solution ensures privacy protection for both the inputs and the computed $\mathcal{F}_{SC}$ results.

1. Introduction

Advances in both sensor technologies and network infrastructures have encouraged the growth and the development of smart environments. The concept of smart environments is to promote the ideas of small world with great deal of different smart devices such as sensors, microcontrollers, handheld devices, and computers that connected via wired or wireless networks [1]. These smart devices can automatically collect real-time data from the users without human-to-human or human-to-computer interaction. Note that smart devices can collect large amounts of personal data when the users are operating and interacting with the environment. The organization and exploration of these heterogeneous personal data require intelligent software agents (hereafter we will refer to them as agents) to do the analysis in order to trigger actions for the environment. A study of the exploration of personal data has been conducted in [2].

There are many smart spaces (e.g., smart home, smart building, and smart office) which have been proposed and developed in the past few years to enhance a person's environment and way of life. For example, smart homes for ubiquitous healthcare [3] can support patients who live independently at home by providing health monitoring and remote assistance [4]. Smart office can adapt itself to the user needs and hence release the users from their routine tasks [5]. In such environment, office workers can communicate, collaborate, and work in a new and more efficient way.

Along with the potential benefits offered, the usage of smart environment also raises some security and privacy concerns to the data owners. Since a large amount of user's data is captured and possibly stored, issues arise relating to the storage and usage of sensitive data. In the existing implementations, there is no clear privacy protection in place. This may cause the users feel uncomfortable to work or stay in the smart environments. Therefore, data privacy is one of the main challenges for acceptance and adoption of smart environments.

The data privacy concern arising in the smart environments is mainly about the usage of the collected data. The intelligent software agents analyze the collected data to understand the changes of the environment and perform activity prediction. Some of the data collected from the users may be sensitive and, hence, the access control to share those data is becoming an important task. In a multiagent smart environment, two or more agents may concurrently (or

within a given period) collect data from the same user. A wide range of data analysis operations entails a similarity measurement between datasets collected. Based on the analysis results, the smart environments can improve the experience of their inhabitants by adapting the behavior of the users and other conditions in the environment.

When users (or agents) wish to compare datasets collected with other parties, a secure mechanism must be available to facilitate the computation in a secure manner. Assume that two parties would like to find the similarity between their collected datasets. We can utilize a measurement metric such as similarity coefficient for the comparison.

Similarity coefficient ($\mathcal{F}_{\mathcal{SC}}$) is a function used to study the coexistence of objects and the similarity of the objects. Finding similarities between two datasets is an important task in many research areas. The output from the comparison can be involved in such contexts as the study of the coexistence of species and the similarity of sampling sites [6, 7] (in the context of ecological and biogeographical research), as the matching of two given DNA sequences [8], or as the assignment of a set of observations into subsets called clusters [9] (in the clustering application). In the privacy preserving data mining (PPDM) applications such as clustering [9, 10], the similarity coefficient is used to assign a set of observations or data into subsets called clusters. Recently, similarity coefficient has also been applied in biometric areas to solve identification problems such as iris and fingerprint recognition [11].

1.1. Motivation. Advances in data collection technologies have led to an increasing number of data collected and stored in smart environments. In the early age, collected data were generally without considering security and privacy issues. Therefore, previously stored data may contain a vast amount of sensitive information. These data are important for the analysis purpose and for the comparison with the newly collected data in order to trigger accurate activity for the changing of the environment. Recent discussions about user's data privacy with respect to the data collected in the smart environment have shown that the public gradually realizes that this may have a long-term impact on their everyday life.

Let us consider a practical scenario where two agents (each embedded with a sensor) would like to analyze and extract useful information from the datasets they collected from the users. To improve the performance and accuracy of the changing condition in the environment, data from the same (or different) subject must be gathered and used for the analysis. These analyses require collaboration between agents and sharing of data collected by each sensor. However, the release and sharing of sensitive information raises some privacy concerns for the users.

In a context-sensitive environment, access to a resource requires the collection of confidential information. For instance, if the location of a person is used to grant access to resources such as printer and projector, the information about the acceptance or rejection of using a device will violate the person's privacy [12]. Consequently, privacy concerns arise in terms of how to control the sharing of sensitive information with other users or agents.

1.2. Problem Statement. In this paper, we will consider the comparison of both data types (old and newly collected data) for the similarity measurement. We define the problem in this paper as follows: let $X = \{x_1, x_2, \ldots, x_n\}$ and $Y = \{y_1, y_2, \ldots, y_n\}$ be two binary datasets belonging to two agents (a requestor and a supporter, resp.). We assume that the requestor wants to measure the similarity between X and Y without revealing X to the supporter. At the same time, the supporter is willing to participate if (1) Y will not be revealed to the requestor and (2) no extra information can be derived from the final output.

Since the same datasets may be used for several similarity measurements, we design our protocol to facilitate more than one computation (without modifying the protocol). To support multiple similarity coefficients, we utilize a semitrusted anonymizer in our protocol to answer the requests from the requestor.

The execution of our protocol should preserve a number of fundamental security properties as described in [13]. In particular, all players must ensure that no extra information will be revealed other than the computed output (privacy is protected) and the output of the protocol is according to the prescribed functionality (correctness is guaranteed). We require all computations in our protocol to be performed in an encrypted form by utilizing a semantically secure homomorphic cryptosystem in our protocol design. The details of the homomorphic cryptosystem will be discussed in Section 3.1.

1.3. Organization of the Paper. This paper is organized as follows. Section 2 introduces the background for this research and discusses related works in the literature. Section 3 describes the technical preliminaries of our work, followed by the details of our private similarity coefficients computation protocol in Section 4. The analysis and discussion of our protocol are presented in Section 5 and our conclusion is presented in Section 6.

2. Background and Related Work

2.1. Similarity Coefficients. Binary data is a representation of presence or absence of an attribute in the given objects. The value "1" is used to show the presence of the attribute while "0" is used to represent the absence of the attribute. Hence, a binary dataset is composed of a series of strings with "1" and "0."

Let $X = \{x_i \mid i = 1, 2, \ldots, n\}$ and $Y = \{y_i \mid i = 1, 2, \ldots, n\}$ be two binary datasets, where $x_i, y_i \in \{0, 1\}$ and $C_{XY} = \{(x_i, y_i) \mid i = 1, 2, \ldots, n\}$. We further specify the following summation variables.

(i) a is the number of $(x_i, y_i) = (1, 1)$ in C_{XY}.

(ii) b is the number of $(x_i, y_i) = (1, 0)$ in C_{XY}.

(iii) c is the number of $(x_i, y_i) = (0, 1)$ in C_{XY}.

(iv) d is the number of $(x_i, y_i) = (0, 0)$ in C_{XY}.

In the relevant literature [14, 15], a is known as "positive match," b and c are known as "mismatch," and d is referred to as "negative match."

The computation of similarity coefficient $\mathcal{F}_{\mathcal{SC}}$ is based on the summation variables. A large number of $\mathcal{F}_{\mathcal{SC}}$ have been proposed in the literature. Similarity coefficient choice is based on some criterion. An important consideration is the inclusion or exclusion of negative match d in the computation. For some data, the absence of an element in both objects would indicate similarity, but, in certain cases, this might not be true. Hence, we can divide the similarity coefficients into two types.

The first type takes into consideration negative matches. For example, Russell and Rao [16] introduced the similarity coefficient of this type that can be expressed as follows:

$$\frac{a}{a+b+c+d}. \tag{1}$$

This similarity coefficient represents the proportion of positive matches in the dataset. Note that the denominator in (1) is actually the size of the dataset, n.

In the second type, we do not consider negative matches in the computation. For example, the Jaccard coefficient [17] can be calculated as follows:

$$\frac{a}{a+b+c}. \tag{2}$$

As shown in (2), the Jaccard coefficient is independent of the summation variable d.

In the asymmetric type of binary data, the positive matches are usually more significant than the negative matches [18, 19]. However, the inclusion or exclusion of negative matches in the similarity coefficients is still an ongoing issue in many research areas [14, 20]. We refer readers to [21] for a comprehensive similarity coefficients list (the authors compiled a list of 76 binary similarity coefficients).

In this paper, we particularly consider the similarity coefficients for binary data, but with the correct size of each summation variable in Section 2.1, the agent is able to compute dissimilarity coefficients of two datasets (i.e., X and Y). We do not discuss further dissimilarity coefficients in this paper, but we would like to stress that our protocol is also applied to dissimilarity coefficients computation.

2.2. Related Work. Data privacy protection is still a major concern in smart environments, although there have been efforts to protect personal information of the users by using mobile agents [22] and deploying security framework [23] and context-based solutions (e.g., context-sensitive services [12] and context-aware interface [24]). Context is often referred to as information used to identify activities or events that have occurred in the smart environment. Also, some security and privacy risk models have been proposed in the literature to help users (or designers) to identify and prioritize privacy risks for a specific application [25, 26]. Other specific solutions such as cloaking area creation schemes have been used to protect the location privacy of the users [27]. However, these solutions do not consider the privacy protection for data collected or stored in the environments. Instead, they try to prevent the leakage of sensitive information during the real-time data collection. Since our work in this paper is on the privacy protection for data analysis, we will focus our discussions on the existing solutions for the secure similarity measurement.

Various procedures and protocols for testing the similarity (or homogeneity) of two or more datasets have been proposed in the literature. Private matching is a practical problem to find common data from the joint databases without revealing any private information to any party [28]. The general approach was studied by Agrawal et al. in [29] which has motivated many researchers to find efficient solutions to address the private matching problem.

In 1982, Yao introduced the first two-party computation protocol (also known as millionaires' problem) in [30]. His idea is to allow two individuals to compare their richness without revealing their wealth to each other. The protocol is secure if no parties learn extra information from the protocol execution. Since then, many secure computation protocols have been proposed to solve problems such as secure multiparty computation [13] and cooperative computation [31]. As proved by Goldreich et al. in [32], there exists a secure solution for any functionality which can be represented as a combinatorial circuit. However, the generic construction of circuit evaluation is somehow inefficient for a large number of parties because the cost for large input can be very high.

The first secure protocol to evaluate $\mathcal{F}_{\mathcal{SC}}$ in the semihonest setting was proposed in [33]. As shown in [34], the solution in [33] is not secure due to its potential to leak the private input of one party. Hence, another protocol with the malicious model is proposed in [34].

The most related work to our solution is the differential similarity computations proposed in [35]. Several two-party protocols have been proposed to compute exact and threshold similarities based on a specific $\mathcal{F}_{\mathcal{SC}}$ (e.g., scalar product and cosine similarity). In their designs, the same protocol cannot be used to facilitate another $\mathcal{F}_{\mathcal{SC}}$. A substantial modification is needed in order to use the same protocol to compute for other functions. Since there is no best $\mathcal{F}_{\mathcal{SC}}$ in the literature, we may need to consider the computation results from more than one $\mathcal{F}_{\mathcal{SC}}$. In this paper, we will design a solution that can be used to facilitate any $\mathcal{F}_{\mathcal{SC}}$ computation without modifying the existing protocol.

3. Technical Preliminaries

3.1. Homomorphic Encryption Scheme. In our protocol design, we utilize a multiplicative property from the homomorphic encryption scheme (i.e., ElGamal [36]) as our primary cryptographic tool. Let $\mathrm{Enc}_{\mathrm{pk}}(m)$ denote the encryption of m with the public key, pk. Given two ciphertexts $\mathrm{Enc}_{\mathrm{pk}}(m_1)$ and $\mathrm{Enc}_{\mathrm{pk}}(m_2)$, there exists an efficient algorithm $\cdot_h$ to compute $\mathrm{Enc}_{\mathrm{pk}}(m_1 \cdot m_2)$.

3.2. System Model. Our protocol consists of the following main players.

(i) Anonymizer $\mathcal{A}$: a semitrusted party who helps to facilitate the computation requests.

(ii) Requestor: a party who wants to learn the similarity between two binary datasets. The requestor will send a computation request to $\mathcal{A}$.

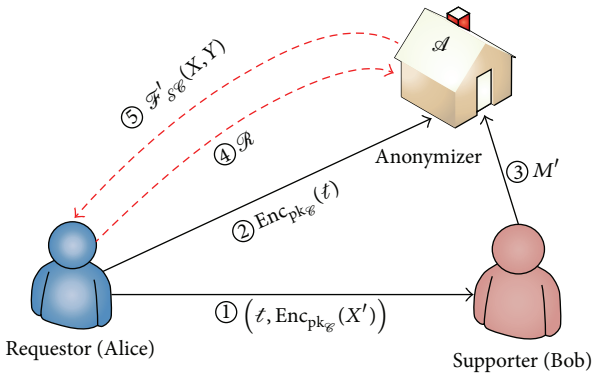

Figure 1: Overview of the proposed model.

(iii) Supporter: a party who collaborates with the requestor to perform the homomorphic operations.

Note that a supporter can also make a computation request to $\mathcal{A}$. We can assume that the players are intelligent software agents communicating with each other in the same or from different smart environments. We can select any agent as the anonymizer if it does not collect data to be used for the computation. The interactions of players in our proposed system are shown in Figure 1.

3.3. Adversary Model. In general, there are two types of adversary models that can be considered: (1) the semihonest model and (2) the malicious model. In the semihonest model, all parties follow the prescribed action in the protocol but might attempt to learn extra information from the intermediate computations.

In the malicious model, a malicious party might arbitrarily deviate from the protocol for their own gain, such as performing active steps to interrupt the execution of the protocol in order to gain access to private data. In this paper, we assume that all players are semihonest parties ("honest-but-curious"). They follow the prescribed actions in the protocol but might be interested to learn some extra information from the data they received during the protocol execution or from the final output.

3.4. Security Model. Generally, a two-party computation problem is cast by specifying a random process that maps pairs of inputs to pairs of outputs [37]. In the setting of a two-party computation, the requestor (with input X) and the supporter (with input Y) jointly compute for the function $f(X,Y)$ while preserving some security properties such as the correctness of the output and the data privacy [38].

Let Π be a protocol between the two players. Then, we can denote the requestor's output by $\Pi_r(X,Y)$ and the supporter's output by $\Pi_s(X,Y)$. Since only the client gets the output in our case, we can simply denote $\Pi(X,Y) = \Pi_C(X,Y)$. The perspective of the client and the server during the execution of protocol Π on input (X,Y) can be denoted as $\mathrm{VIEW}_C^{\Pi}(X,Y)$ and $\mathrm{VIEW}_S^{\Pi}(X,Y)$, respectively. Note that the view of each party includes their local input, their output, and their messages received from the other party. We now formally define our usage of the term privacy in our protocol (adapted from [39]) as follows.

Definition 1 (privacy with respect to semihonest behavior). Let $f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be a probabilistic polynomialtime function. One says that a two-party computation protocol Π securely computes f in the presence of semihonest adversaries if for every $X, Y \in \{0,1\}^* : \Pi(X,Y) = f(X,Y)$. Also, there exist probabilistic polynomial-time algorithms S_C and S_S, such that

$$\{S_C(X, f(X,Y))\}_{X,Y\in\{0,1\}^*} \overset{c}{\equiv} \left\{\mathrm{VIEW}_C^{\Pi}(X,Y)\right\}_{X,Y\in\{0,1\}^*} \quad (3)$$

$$\{S_S(Y)\}_{X,Y\in\{0,1\}^*} \overset{c}{\equiv} \left\{\mathrm{VIEW}_S^{\Pi}(X,Y),\right\}_{X,Y\in\{0,1\}^*}, \quad (4)$$

where $\overset{c}{\equiv}$ denotes computational indistinguishability according to the families of polynomial-size circuits. One refers the reader to [39] for the definition of computational indistinguishability.

Note that we can simulate each player's view by using a probabilistic polynomial-time algorithm, only given access to the party's input and output. Thus, we only need to show the existence of a simulator for each player that satisfies the requirements of (3) and (4).

3.5. Differential Privacy. Differential privacy is a strong notion of privacy that guarantees the privacy protection in the presence of arbitrary auxiliary information. Intuitively, it aims to limit the information leakage from the output while a small change on the inputs. The formal definition is defined as follows.

Definition 2 (ϵ-differential privacy [40]). A randomized function $\mathcal{K}$ satisfies ϵ-differential privacy if, for any two neighboring datasets D_1 and D_2 differing on at most one element and all $S \subseteq \mathrm{Range}(\mathcal{K})$,

$$\Pr\left[\mathcal{K}(D_1) \in S\right] \le \exp(\epsilon) \times \Pr\left[\mathcal{K}(D_2) \in S\right]. \quad (5)$$

Definition 3 (global sensitivity [41]). The global sensitivity of a function $F : \mathcal{D} \to \mathbb{R}$ is

$$\Delta F = \max_{D_1,D_2} \left\| F(D_1) - F(D_2) \right\|_1 \quad (6)$$

over all pairs of neighboring datasets D_1 and D_2.

Theorem 4 (Laplacian mechanism [41]). *For $F : \mathcal{D} \to \mathbb{R}$, $\mathcal{K}$ achieves ϵ-differential if $\mathcal{K}(D) = F(D) + Lap(\Delta F/\epsilon)$.*

The parameter ϵ is a small positive value which is used to control the trade-off between data privacy and data utility. A smaller value of ϵ will guarantee a higher privacy, but the data utility can be affected.

For $\mathcal{F}_{\mathcal{SC}}$ computation, we can think of $D_1 = (X,Y)$ and $D_2 = (X,Y')$, where Y and Y' are only differing in one element. The change of one element in Y will increase (or decrease) the mismatch value (b or c) by 1 and also affects the value of a or d. Therefore, ΔF for each $\mathcal{F}_{\mathcal{SC}}$ can be different depending on the formula used.

3.6. Notations Used . We summarize the notations used hereafter in this paper in the Notations section.

4. Our Solution

In this section, we will explain the details of our computation protocol, in particular, the computation phases for each player.

4.1. Private Similarity Coefficients. At the preliminary phase, the semitrusted anonymizer ($\mathscr{A}$) generates an ElGamal cryptosystem key pair ($\text{pk}_{\mathscr{C}}, \text{pr}_{\mathscr{C}}$) and sends the public key $\text{pk}_{\mathscr{C}}$ to all the agents. For simplicity, let us assume that there are only two agents (Alice and Bob) in the protocol. We assume that there exists a secure channel for key exchange and data transmission.

Phase 1. Alice first randomly selects a prime number t to replace each $x_i \in X$ as follows:

$$x_i' = \begin{cases} t, & \text{if } x_i = 1 \\ t^{-1}, & \text{if } x_i = 0. \end{cases} \tag{7}$$

Next, Alice encrypts t and each $x_i' \in X'$ by using $\text{pk}_{\mathscr{C}}$ (e.g., $\text{Enc}_{\text{pk}_{\mathscr{C}}}(x_i')$). Alice sends $\text{Enc}_{\text{pk}_{\mathscr{C}}}(t)$ and $(t, \text{Enc}_{\text{pk}_{\mathscr{C}}}(X'))$ to $\mathscr{A}$ and Bob, respectively.

Phase 2. Bob replaces each $y_i \in Y$ as follows:

$$y_i' = \begin{cases} t^2, & \text{if } y_i = 1 \\ t, & \text{if } y_i = 0. \end{cases} \tag{8}$$

Next, Bob encrypts $y_i' \in Y'$ with $\text{pk}_{\mathscr{C}}$ to produce $\text{Enc}_{\text{pk}_{\mathscr{C}}}(Y') = \{\text{Enc}_{\text{pk}_{\mathscr{C}}}(y_i') \mid i = 1, 2, \ldots, n\}$. Note that the sequence of all encrypted data is the same as its sequence order in the original dataset.

Phase 3. In this phase, Bob computes $\text{Enc}_{\text{pk}_{\mathscr{C}}}(X'Y')$ by using the homomorphic multiplicative property. The multiplication is done in accordance with the sequence i of x_i' and y_i' (e.g., $\text{Enc}_{\text{pk}_{\mathscr{C}}}(x_i') \cdot_h \text{Enc}_{\text{pk}_{\mathscr{C}}}(y_i') = \text{Enc}_{\text{pk}_{\mathscr{C}}}(x_i' y_i')$). Next, Bob randomly permutes elements in $\text{Enc}_{\text{pk}_{\mathscr{C}}}(X'Y')$. We assume that there exists an efficient shuffle protocol π which randomly changes the sequence of elements in $\text{Enc}_{\text{pk}_{\mathscr{C}}}(X'Y')$. For simplicity, let $M = \text{Enc}_{\text{pk}_{\mathscr{C}}}(X'Y')$ and $M' = \pi(M)$. Bob transmits M' to $\mathscr{A}$ without revealing M to any party.

Phase 4. After receiving M' from Bob, $\mathscr{A}$ decrypts each $m_i' \in M'$ with its private key $\text{pr}_{\mathscr{C}}$:

$$\text{Dec}_{\text{pr}_{\mathscr{C}}}\left(\text{Enc}_{\text{pk}_{\mathscr{C}}}\left(x_i' y_i'\right)\right) = x_i' y_i'. \tag{9}$$

Next, $\mathscr{A}$ examines the decrypted values and computes the summation variables as follows:

$$x' y' = \begin{cases} t^3, & \text{increases } a \text{ by } 1 \\ t^2, & \text{increases } b \text{ by } 1 \\ t^1, & \text{increases } c \text{ by } 1 \\ t^0, & \text{increases } d \text{ by } 1. \end{cases} \tag{10}$$

Computation phases for Alice (Requestor)
Input: $X = \{x_i \mid i = 1, 2, \ldots, n\}$, $x_i \in \{0, 1\}$
Output: $\text{Enc}_{\text{pk}_{\mathscr{C}}}(X')$

/*Phase 1*/
Initialise: $I = 0$;
Randomly selects a prime numbers t;
for $i = 1$ **to** n **do**
  if $x_i = 1$ **then**
    $x_i' = t$;
  else
    $x_i' = t^{-1}$;
  end if
Encrypts x_i' with $\text{pk}_{\mathscr{C}}$ (e.g., $\text{Enc}_{\text{pk}_{\mathscr{C}}}(x_i')$);
end for
Encrypts t with $\text{pk}_{\mathscr{C}}$ (e.g., $\text{Enc}_{\text{pk}_{\mathscr{C}}}(t)$);
Let $\text{Enc}_{\text{pk}_{\mathscr{C}}}(X') = \{\text{Enc}_{\text{pk}_{\mathscr{C}}}(x_i') \mid i = 1, 2, \ldots, n\}$;
Sends $\text{Enc}_{\text{pk}_{\mathscr{C}}}(t)$ to $\mathscr{A}$;
Sends t and $\text{Enc}_{\text{pk}_{\mathscr{C}}}(X')$ to Bob;

Algorithm 1: Requestor's computation phases.

At the end of this phase, $\mathscr{A}$ obtains all the summation variables needed to compute $\mathscr{F}_{\mathscr{SC}}$.

Phase 5. Alice (or Bob) makes a request $\mathscr{R}$ to $\mathscr{A}$ to compute for a chosen $\mathscr{F}_{\mathscr{SC}}$ (i.e., Jaccard). The anonymizer computes $\mathscr{F}_{\mathscr{SC}}(X, Y)$ and adds Laplacian noise $\text{Lap}(\Delta F/\epsilon)$ to the computed result. At last, $\mathscr{A}$ sends $\mathscr{F}'_{\mathscr{SC}}(X, Y) = \mathscr{F}_{\mathscr{SC}}(X, Y) + \text{Lap}(\Delta F/\epsilon)$ to Alice (or Bob). Note that this phase can be used to compute any $\mathscr{F}_{\mathscr{SC}}$ in [21] without repeating Phases 1 to 4.

We show the pseudocode for requestor, supporter, and anonymizer in Algorithms 1, 2, and 3, respectively.

4.2. Computing Sensitivity. In Phase 5, the anonymizer adds a Laplace noise to the computed result of the requested similarity coefficient function before it releases the mixture to the requestor. The amount of noise to be added is proportional to the sensitivity ΔF of the requested function. For instance, the sensitivity of the requested function is the measurement of the changes of the output (i.e., $\mathscr{F}_{\mathscr{SC}}$) when a small change happens in the input (a, b, c, or d).

For simplicity, we use Jaccard's index to demonstrate how to compute the sensitivity of a similarity coefficient. We denote Jaccard's index between P and Q as $\mathscr{J}(P, Q)$. Let us consider X, Y, and Z to be three binary datasets such that Y and Z are the same except for one bit:

$$\begin{aligned} \Delta F\,(\text{Jaccard}) &= \|\mathscr{J}(X, Y) - \mathscr{J}(X, Z)\| \\ &= \left\| \frac{a}{a+b+c} - \frac{a \pm 1}{a+b+c} \right\| = \left\| \frac{\pm 1}{a+b+c} \right\|. \end{aligned} \tag{11}$$

As shown in (11), the difference between $\mathscr{J}(X, Y)$ and $\mathscr{J}(X, Z)$ is at most $1/(a+b+c)$. Therefore, the anonymizer can set $\Delta F(\text{Jaccard}) \leq 1/(a+b+c)$.

Since the anonymizer is designed to facilitate more than one request, it needs to ensure that the noise being added will

Computation phases for Bob (Supporter)
Input: $Y = \{y_i \mid i = 1, 2, \ldots, n\}$, $y_i \in \{0, 1\}$
Output: M'

/∗**Phase 2**∗/
Receives t and $\text{Enc}_{\text{pk}_{\mathscr{C}}}(X')$ from Alice (from Phase 1);
for $i = 1$ **to** n **do**
  if $y_i = 1$ **then**
    $y_i' = t^2$;
  else
    $y_i' = t$;
  end if
  Encrypts y_i' with $\text{pk}_{\mathscr{C}}$ (e.g., $\text{Enc}_{\text{pk}_{\mathscr{C}}}(y_i')$);
end for
Let $\text{Enc}_{\text{pk}_{\mathscr{C}}}(Y') = \{\text{Enc}_{\text{pk}_{\mathscr{C}}}(y_i') \mid i = 1, 2, \ldots, n\}$;

/∗**Phase 3**∗/
// homomorphic multiplicative property
for $i = 1$ **to** n **do**
  // computation of $\text{Enc}_{\text{pk}_{\mathscr{C}}}(x_i' y_i')$
  Computes $\text{Enc}_{\text{pk}_{\mathscr{C}}}(x_i') \cdot_h \text{Enc}_{\text{pk}_{\mathscr{C}}}(y_i')$;
end for
Let $M = \{\text{Enc}_{\text{pk}_{\mathscr{C}}}(x_i' y_i') \mid i = 1, 2, \ldots, n\}$;
Shuffles M such that $M' = \pi(M)$;
Sends M' to $\mathscr{A}$;

Algorithm 2: Supporter's computation phases.

Computation phases for Anonymizer $\mathscr{A}$
Input: Computation request $\mathscr{R}$
Output:$\mathscr{F}'_{\mathscr{S}\mathscr{C}}(X, Y)$

/∗**Phase 4**∗/
Initialise: $a = 0$, $b = 0$, $c = 0$ and $d = 0$;
Receives $\text{Enc}_{\text{pk}_{\mathscr{C}}}(t)$ from Alice (from Phase 1);
Receives M' from Bob (from Phase 3);
// decryption operation to obtain t
Computes $\text{Dec}_{\text{pr}_{\mathscr{C}}}(\text{Enc}_{\text{pk}_{\mathscr{C}}}(t))$;
for $i = 1$ **to** n **do**
  // decryption operation to obtain $x_i' y_i'$
  Computes $\text{Dec}_{\text{pr}_{\mathscr{C}}}(\text{Enc}_{\text{pk}_{\mathscr{C}}}(x_i' y_i'))$;
  if $x_i' y_i' = t^3$ **then**
    Increases a by 1;
  else if $x_i' y_i' = t^2$ **then**
    Increases b by 1;
  else if $x_i' y_i' = t^1$ **then**
    Increases c by 1;
  else if $x_i' y_i' = t^0$ **then**
    Increases d by 1;
  else
    stop (error)
  end if
end for

/∗**Phase 5**∗/
for each $\mathscr{R}$ **do**
  Computes $\mathscr{F}_{\mathscr{S}\mathscr{C}}(X, Y)$;
  Generates Laplacian noise $\text{Lap}(\Delta F/\epsilon)$;
  Returns $\mathscr{F}'_{\mathscr{S}\mathscr{C}}(X, Y) = \mathscr{F}_{\mathscr{S}\mathscr{C}}(X, Y) + \text{Lap}(\Delta F/\epsilon)$;
end for

Algorithm 3: Anonymizer's computation phases.

not affect the utility of the function. When the same request is received from the same (or different) requestor, a random noise should be used. This is to make sure that no party can learn the actual score for $\mathscr{F}_{\mathscr{S}\mathscr{C}}$.

5. Analysis and Discussion

5.1. Correctness and Utility Analysis. The output of our protocol is correct and accurate if all parties follow the protocol faithfully. Let us assume that both the requestor and the supporter are semihonest. At Phase 3, the multiplication of X' and Y' will give a correct result due to the multiplicative property of the ElGamal cryptosystem. Therefore, we can ensure that the anonymizer will receive the correct outputs (a, b, c, and d) after the decryption. Note that the outputs at Phase 4 can be viewed as $f(X, Y) = at^3 + bt^2 + ct^1 + dt^0$. The coefficients for variables in $f(X, Y)$ are the summation variables defined in Section 2.1.

In terms of utility, we can expect our protocol to achieve high accuracy. Our utility analysis is based on a set of similarity coefficients instead of specific function.

5.2. Security Analysis. To illustrate the efficacy of the security protection of our protocol in the presence of semihonest adversaries, we briefly explain how to simulate the view of each player using their respective inputs and outputs only (i.e., simulator $\mathcal{S}_r$ for the requestor and $\mathcal{S}_s$ for the supporter). If such simulation is indistinguishable from real world execution, it implies that the protocol does not reveal any extra information under semihonest model.

Let us assume that $\mathcal{S}_r$ simulates all internal coin flips of the requestor as described in our protocol. For instance, it simulates n ElGamal ciphertexts sent from the requestor to the supporter. Next, let us assume that $\mathcal{S}_s$ simulates all internal coin flips of the supporter as described in our protocol. This simulator simulates n ElGamal ciphertexts as the homomorphic multiplicative results. Based on the simulation for both parties, the computational indistinguishability for our protocol appears to hold on first inspection.

5.3. Privacy Analysis. In general, each player must ensure that it only releases the required data during the protocol execution. We assume that all communications in our protocol execution are via an authenticated channel, and the anonymizer will not reveal its private key to others as well. In order words, only the anonymizer can learn the summation variables after the decryption operation.

Based on the dataset $\text{Enc}_{\text{pk}\mathscr{C}}(X')$ computed by the requestor, the supporter is not able to distinguish which ciphertext is the encryption result of t or t^{-1}. This is because the ElGamal cryptosystem is semantically secure [42], such that the encryption of the same message will produce different ciphertexts due to randomization in the encryption

process. Hence, the supporter learns nothing about X by knowing $\text{Enc}_{\text{pk}_{\mathscr{C}}}(X')$ and t.

5.4. Comparisons with Existing Work. In this section, we will compare our protocol with the private similarity computations ($\mathscr{PSC}$) proposed in [35]. In $\mathscr{PSC}$, there are two types of settings that can be used to achieve the differential privacy: (1) data owners locally add noise to partially computed result (e.g., set intersection) and (2) anonymizer is responsible for inserting noise during the similarity computation. In both settings, all parties (data owners and anonymizer) must decide which $\mathscr{F}_{\mathscr{SC}}$ to be used in the computation. Unlike $\mathscr{PSC}$, our protocol does not require the data owners to specify $\mathscr{F}_{\mathscr{SC}}$ before the protocol begins. Instead, the anonymizer will compute any $\mathscr{F}_{\mathscr{SC}}$ requests by the requestor in Phase 5 and inserts noise into the computation result to preserve the differential privacy of the private inputs.

The main limitation of $\mathscr{PSC}$ is its protocol design which only can be used to compute a specific $\mathscr{F}_{\mathscr{SC}}$ in each round of its protocol execution. All parties are required to start the protocol again even though they use the same inputs and $\mathscr{F}_{\mathscr{SC}}$ in the new computation. In addition, the same protocol requires substantial modifications before it can be used to facilitate other $\mathscr{F}_{\mathscr{SC}}$. Unlike the solution in [35], our design can be used to facilitate more than one $\mathscr{F}_{\mathscr{SC}}$. In particular, we allow the requestor to send multiple requests (for distinct $\mathscr{F}_{\mathscr{SC}}$) to the anonymizer using the same protocol. The data owners do not need to repeat computation steps from Phase 1 to Phase 4 if they use the same datasets as the inputs for the computation.

Another distinction between our protocol and $\mathscr{PSC}$ is the roles of each participating party. In $\mathscr{PSC}$, the data owners must cooperate with each other to decrypt the homomorphically encrypted value in order to learn the computation result while the anonymizer takes part in the protocol by computing the intermediate results (e.g., set intersection). In our solution, the data owners only cooperate to compute the multiplicative operation while the anonymizer is responsible to perform the decryption operation and noise generation for each $\mathscr{F}_{\mathscr{SC}}$ request.

In terms of complexity, our protocol achieves a significant lower computational overhead as the $\mathscr{PSC}$. Practically, running two or more protocols (using same datasets) for different $\mathscr{F}_{\mathscr{SC}}$ will incur high computation costs. Although the second setting in $\mathscr{PSC}$ can achieve the same complexity as our protocol, however, it only can be used to compute one $\mathscr{F}_{\mathscr{SC}}$. Note that both the basic construction of $\mathscr{PSC}$ and our protocol are based on the homomorphic cryptosystem.

6. Conclusion

Due to the advances in ubiquitous technologies and the demands of data privacy protection, a secure mechanism is required to increase the confidence of the users in the smart environments. In this paper, we have proposed a secure protocol to compute $\mathscr{F}_{\mathscr{SC}}$ within differential privacy model for data privacy protection in smart environments. Although our target area is a smart environment, the same solution can be applied to other related areas such as pervasive or ubiquitous computing [43] and intelligent environments [44].

In order to preserve differential privacy, the anonymizer needs to compute distinct noise for each request especially when the requestor sends the same $\mathscr{F}_{\mathscr{SC}}$ request. This is because the identical output may allow the adversary from learning the private dataset of the owner or noise added to the computation result. Since the same request for a specific $\mathscr{F}_{\mathscr{SC}}$ may output two slightly different results (due to the noise added by $\mathscr{A}$), we can ensure that the result from our protocol execution will not compromise the privacy of any data owner. Hence, agents in smart environments can utilize our protocol to compare datasets with other entities without compromising the data privacy of the users.

Notations

X:	Private dataset from the requestor
Y:	Private dataset from the supporter
x_i:	ith element of X
y_i:	ith element of Y
n:	Size of the private input
t:	Prime number chosen by anonymizer
$\mathscr{R}$:	Computation request from requestor
$\text{pk}_{\mathscr{C}}$:	Public key of the anonymizer
$\text{pr}_{\mathscr{C}}$:	Private key of the anonymizer
$\text{Enc}_{\text{pk}_{\mathscr{C}}}(\cdot)$:	Encryption operation using $\text{pk}_{\mathscr{C}}$
$\text{Dec}_{\text{pr}_{\mathscr{C}}}(\cdot)$:	Decryption operation using $\text{pr}_{\mathscr{C}}$
π:	Shuffling protocol
$\mathscr{F}_{\mathscr{SC}}$:	Similarity coefficient function
ΔF:	Sensitivity of $\mathscr{F}_{\mathscr{SC}}$
$\mathscr{F}'_{\mathscr{SC}}$:	Similarity coefficient with noise.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] P. Nixon, G. Lacey, and S. Dobson, *Managing Interactions in Smart Environments*, Springer, London, UK, 2000.

[2] T. Teraoka, "Organization and exploration of heterogeneous personal data collected in daily life," *Human-Centric Computation and Information Sciences*, vol. 2, pp. 1–15, 2012.

[3] J. K.-Y. Ng, "Ubiquitous healthcare: healthcare systems and applications enabled by mobile and wireless technologies," *Journal of Convergence*, vol. 3, no. 2, pp. 15–20, 2012.

[4] S. Szewcyzk, K. Dwan, B. Minor, B. Swedlove, and D. Cook, "Annotating smart environment sensor data for activity learning," *Technology and Health Care*, vol. 17, no. 3, pp. 161–169, 2009.

[5] C. Ramos, G. Marreiros, R. Santos, and C. Filipe Freitas, "Smart offices and intelligent decision rooms," in *Handbook of Ambient Intelligence and Smart Environments*, pp. 851–880, Springer, Berlin, Germany, 2009.

[6] E. F. Connor and D. Simberloff, "Interspecific competition and species co-occurrence patterns on islands: null models and the evaluation of evidence," *Oikos*, vol. 41, no. 3, pp. 455–465, 1983.

[7] M. E. Gilpin and J. M. Diamond, "Factors contributing to non-randomness in species Co-occurrences on Islands," *Oecologia*, vol. 52, no. 1, pp. 75–84, 1982.

[8] M. H. Pandi, O. Kashefi, and B. Minaei, "A novel similarity measure for sequence data," *Journal of Information Processing Systems*, vol. 7, no. 3, pp. 413–424, 2011.

[9] D. H. Fisher, "Knowledge acquisition via incremental conceptual clustering," *Machine Learning*, vol. 2, no. 2, pp. 139–172, 1987.

[10] H. Wang, W. Wang, J. Yang, and P. S. Yu, "Clustering by pattern similarity in large data sets," in *Proceedings of the ACM SIGMOD International Conference on Managment of Data*, pp. 394–405, Madison, Wis, USA, June 2002.

[11] P. Willett, "Similarity-based approaches to virtual screening," *Biochemical Society Transactions*, vol. 31, no. 3, pp. 603–606, 2003.

[12] U. Hengartner and P. Steenkiste, "Avoiding privacy violations caused by context-sensitive services," in *Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom '06)*, pp. 222–233, March 2006.

[13] Y. Lindell and B. Pinkas, "Secure multiparty computation for privacy-preserving data mining," *Journal of Privacy and Confidentiality*, vol. 1, pp. 59–98, 2009.

[14] P. H. A. Sneath and R. R. Sokal, *Numerical Taxonomy: The Principles and Practice of Numerical Classification*, W. H. Freeman, San Francisco, Calif, USA, 1973.

[15] L. A. Goodman and W. H. Kruskal, "Measures of association for cross classifications., II: further discussion and references," *Journal of the American Statistical Association*, vol. 54, no. 285, pp. 123–163, 1959.

[16] P. F. Russel and T. R. Rao, "On habitat and association of species of Anophe-line larvae in South-eastern Madras," *Journal of Malaria Institute India*, vol. 3, pp. 153–178, 1940.

[17] P. Jaccard, "Nouvelles recherches sur la distribution florale," *Bulletin de la Sociète Vaudense des Sciences Naturelles*, vol. 44, pp. 223–270, 1908.

[18] C. Baroni-Urbani and M. W. Buser, "Similarity of binary data," *Systematic Zoology*, vol. 25, pp. 251–259, 1976.

[19] D. P. Faith, "Asymmetric binary similarity measures," *Oecologia*, vol. 57, no. 3, pp. 287–290, 1983.

[20] R. R. Sokal and P. H. A. Sneath, *Principles of Numeric Taxonomy*, W.H. Freeman, San Francisco, Calif, USA, 1963.

[21] S. S. Choi, *Correlation analysis of binary similarity and dissimilarity measures [Ph.D. dissertation]*, Pace University, 2008.

[22] S.-H. Bae and J. Kim, "Development of personal information protection model using a mobile agent," *Journal of Information Processing Systems*, vol. 6, pp. 185–196, 2010.

[23] C.-E. Pigeot, Y. Gripay, M. Scuturici, and J. Pierson, "Context-sensitive security framework for pervasive environments," in *Proceedings of the 4th European Conference on Universal Multiservice Networks (ECUMN '07)*, pp. 391–400, Toulouse, France, February 2007.

[24] J. McNaull, J. C. Augusto, M. Mulvenna, and P. McCullagh, "Flexible context aware interface for ambient assisted living," *Human-Centric Computation and Information Sciences*, vol. 4, pp. 1–41, 2014.

[25] J. I. Hong, J. D. Ng, S. Lederer, and J. A. Landay, "Privacy risk models for designing privacy-sensitive ubiquitous computing systems," in *Proceedings of the 5th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques (DIS '04)*, pp. 91–100, Cambridge, Mass, USA, 2004.

[26] A. P. A. Ling and M. Masao, "Selection of model in developing information security criteria for smart grid security system," *Journal of Convergence*, vol. 2, no. 1, pp. 39–46, 2011.

[27] H.-I. Kim, Y.-K. Kim, and J.-W. Chang, "A grid-based cloaking area creation scheme for continuous LBS queries in distributed systems," *Journal of Convergence*, vol. 4, pp. 23–30, 2013.

[28] K.-S. Wong and M. H. Kim, "Privacy-preserving frequent itemsets mining via secure collaborative framework," *Security and Communication Networks*, vol. 5, no. 3, pp. 263–272, 2012.

[29] R. Agrawal, A. Evfimievski, and R. Srikant, "Information sharing across private databases," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 86–97, San Diego, Calif, USA, June 2003.

[30] A. C. Yao, "Protocols for secure computations," in *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, pp. 160–164, 1982.

[31] J.-S. Kang and D. Hong, "A practical privacy-preserving cooperative computation protocol without oblivious transfer for linear systems of equations," *Journal of Information Processing Systems*, vol. 3, pp. 21–25, 2007.

[32] O. Goldreich, S. Micali, and A. Wigderson, "How to play ANY mental game," in *Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC '87)*, New York, NY, USA, 1987.

[33] K.-S. Wong and M. H. Kim, "Privacy-preserving similarity coefficients for binary data," *Computers and Mathematics with Applications*, vol. 65, no. 9, pp. 1280–1290, 2013.

[34] B. Zhang and F. Zhang, "Secure similarity coefficients computation with malicious adversaries," in *Proceedings of the 4th International Conference on Intelligent Networking and Collaborative Systems (INCoS '12)*, pp. 303–310, Bucharest, Romania, September 2012.

[35] M. Alaggan, S. Gambs, and A.-M. Kermarrec, "Private similarity computation in distributed systems: from cryptography to differential privacy," in *Principles of Distributed Systems*, A. Fernàndez Anta, G. Lipari, and M. Roy, Eds., vol. 7109 of *Lecture Notes in Computer Science*, pp. 357–377, Springer, Berlin, Germany, 2011.

[36] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[37] O. Goldreich, "Foundations of cryptography—a primer," *Foundations and Trends in Theoretical Computer Science*, vol. 1, no. 1, pp. 1–116, 2005.

[38] C. Hazay and Y. Lindell, "Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries," in *Proceedings of the 5th Conference on Theory of Cryptography*, New York, NY, USA, 2008.

[39] O. Goldreich, *Foundations of Cryptography*, vol. 2, Cambridge University Press, 2004.

[40] C. Dwork, "Differential privacy: a survey of results," in *Proceedings of the 5th International Conference on Theory and Applications of Models of Computation*, Xi'an, China, 2008.

[41] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proceedings of the 3rd Conference on Theory of Cryptography*, New York, NY, USA, 2006.

[42] S. Goldwasser and S. Micali, "Probabilistic encryption & how to play mental poker keeping secret all partial informatio," in *Proceedings of the 14th Annual ACM Symposium on Theory of Computing*, San Francisco, Calif, USA, 1982.

[43] J. Ma, "Smart u-things and ubiquitous intelligence," in *Proceeding of the 2nd International Conference on Embedded Software and Systems (ICESS '05)*, pp. 776–776, Xi'an, China, December 2005.

[44] J. C. Augusto, V. Callaghan, D. Cook, A. Kameas, and I. Satoh, "Intelligent environments: a manifesto," *Human-Centric Computation and Information Sciences*, vol. 3, pp. 1–18, 2013.