Secure computation scheme for private biometric data reconstruction

Bon K. Sy, *Member, IEEE,* Adam Ramirez, Arun Prakash K, Shing Ng

***Abstract*— We present a 2-party (client/server) secure computation scheme for private biometric data reconstruction. The secure computation scheme is referred to as SIPPA - Secure Information Processing with Privacy Assurance. SIPPA is an Eigen-based mechanism for discovering the similarity between the client sample data and the server source data without each party revealing their data to each other, nor to a third party. Furthermore, SIPPA allows the client to reconstruct the source data with a good approximation if the sample and source data are sufficiently similar. We show two biometric use cases of SIPPA: (1) reconstruction of digital face image, and (2) privacy preserving reconstruction of voice signature for person verification. We discuss the realization of SIPPA as a processing algorithm unit that is managed by a BFP, and the attachment of the BFP to a BSP in a BioAPI 2.0 framework. We also report the preliminary study on the application of SIPPA to the two biometric use cases.**

I. Introduction

Safeguarding privacy is important in any application requiring personal biometrics or involving the collection of biometric data of an individual. For example, biometric based authentication or credentialing requires an individual to enroll his/her biometric data (e.g., fingerprint) so that the biometric data can be stored away for future comparison purposes. For authentication, an individual will present his/her biometric data to be compared against the enrolled biometric template(s). For medical record or information retrieval, an individual will present his/her biometric data to a search engine in order for it to identify records with matching biometrics. In all these cases, the individual has to reveal his/her biometric data to another party.

Approaches such as cancellable biometrics [1,2] or fuzzy vault [3,4] have been advocated for privacy safeguard, which in general aim at providing confidentiality in the data level. In this research, we aim to tackle privacy safeguard by providing a control mechanism on biometric data sharing. In other words, we focus on issues that can be formulated as below:

There are two parties P1 (Client) and P2 (Server). Party P1 and P2 each have private data D1 and D2. Without the presence of a trusted third party, P1 and P2 would like to know whether D1 and D2 are sufficiently similar. And if so, P1 could derive D2 under the following two conditions:

1. P1 and P2 have to first find out whether D1 and D2 are sufficiently similar without any party sharing their private data to the another party.
2. If D1 and D2 are sufficiently similar, P1 can derive D2 without P2 ever sending D2. The only data that P2 will send to P1 is some helper data with negligible overhead.

SIPPA – Secure Information Processing with Privacy Assurance, is developed to achieve the comparison and retrieval tasks satisfying these two conditions. SIPPA is a kind of secure computation [5] grounded on privacy homomorphism [6].

The basic idea behind SIPPA is to model data (D1/D2) as a linearized vector, and to transform the vector into a symmetric matrix. To compare the two data sets, we take advantage of the approximately consistent relationship between the norm deviation of the data and the norm deviation of the corresponding eigenvectors weighted by the eigenvalues. For privacy protection, neither party will share data, the corresponding matrix and the eigenvalues/ eigenvectors with the other party. Instead, P1 and P2 will engage in a two-party secure computation to uncover the similarity between the weighted eigenvectors. Through the approximately consistent relationship, the similarity between the data can be derived, and used subsequently as a basis for the reconstruction of the source data D2. As we will discuss in the later section, SIPPA offers two very attractive properties:
(1) the private data D2 can be reconstructed by P1 only if P2 sends helper data to P1 after judging D1 and D2 sufficiently similar. In other words, P2 has the control over the sharing of his/her private data D2 – a desirable property often mentioned by the privacy advocates.
(2) the accuracy of the reconstruction depends on the level of closeness/similarity between D1 and D2. In other words, the ability of P1 to reconstruct accurately D2 decreases when P1 does not already know some D1 that bears similarity to D2 – typical scenario of biometrics where the biometric samples from the same individual are "sufficiently similar" but are seldom identical.

In the next section we will first survey the current research on secure computation, as well as its application as a lossless approach to protecting the privacy of personal biometrics. In section III the theoretical formulation and the mathematical properties of SIPPA will be presented and illustrated graphically. The algorithmic steps of SIPPA are then shown in section IV. The implementation of SIPPA as a processing algorithm unit and realized as a BFP in BioAPI 2.0 framework for interoperability will be discussed in section V. The experimental study and the results will be reported in section VI, followed by the conclusion in section VII.

This work was supported in part by a grant from PSC-CUNY Research Award and NSF DUE 0837535. We thank Lin (Leo) Peng for his assistance on running Linux scripts for pre-processing voice data on a batch mode.

Bon K. Sy is with the Computer Science Department, Queens College and University Graduate Center, City University of New York, 65-30 Kissena Blvd, Flushing NY 11367 USA (phone: 1-718-997-3566; email: bon@bunny.cs.qc.cuny.edu)

II. RELATED WORK

The privacy of personal biometrics is generally protected by means of two approaches; namely, lossy data processing and lossless data processing. In security surveillance, protecting privacy through lossy approach is not uncommon. Preserving privacy through the lossy approach is achieved by protecting the private content typically by perturbation, randomization or masking of the original data to the extent that it could still be useful for security purposes [7, 8].

Many face de-identification techniques for privacy protection are based on lossy approach [7, 8]. The basic idea is to conduct lossy anonymization to mask away granular biometric face information not essential to its end goal on security identification or surveillance. For example, Newton et al. [7] proposed a k-Same model based on the k-anonymity framework. The k-Same model takes the average of k face images in a face set and replaces these images with the average image; therefore, each face image presented to a face recognition/comparison system cannot be distinguished from at least k-1 faces in the gallery. In general, information leakage could be a significant risk when k is small and/or known unique aspect of an individual is not sufficiently anonymized. In other words, the degree of privacy protection based on lossy anonymization is data dependent and may not be extendable from one application to another that have different privacy requirements.

In this research we focus on lossless approach. An approach towards lossless privacy protection is Secure Multi-party Computation (SMC). SMC protects computational privacy while preserving content [9]. In other words, the original content on personal biometrics can be retrieved and used in some secure computation scheme for deriving event information of interest, but the scope of derivation is limited to what is allowed by the private computational mechanism of the process.

Generally speaking, SMC deals with the problem in which two or more parties with private inputs would like to compute jointly some function of their inputs, but no party wishes to reveal its private input to other participants. For example, a physician (party P1) wants to compare a medical image of a patient containing personal biometrics with the medical image of another patient stored in a hospital database. The data custodian (e.g., party P2 who is the database administrator) in the hospital and the physician (party P1) may participate in a SMC protocol to jointly compute the output of a matching function that compares the personal biometrics in the medical images. In another example, a user (party P1) and the authentication server (party P2) may jointly compute the distance function based on the user voice sample and the enrolled voice template withheld by the authentication server for biometric verification. The multi-party computation problem was first introduced by Yao [10] and extended by Goldreich et al. [9], and by many others.

Goldreich [9] pointed out that solutions to specific problems should be developed and customized for efficiency reasons. Du and Atallah [11, 12] presented a series of specific solutions to specific problems; e.g., privacy-preserving cooperative scientific computations, privacy-preserving database query, and privacy-preserving geometric computation. In their Privacy-Preserving Cooperative Scientific Computations (PPCSC) [11], they proposed a protocol between two parties to solve the problem (A1+A2)x = b1 +b2, where matrix A1 and vector b1 belong to party P1, matrix A2 and vector b2 belong to party P2. At the end of the protocol, both parties know the solution x while nobody knows the other party's private inputs.

In SIPPA, the private data exchange and information processing involves PPCSC. Specifically, biometric data is represented in the form of a matrix Ai (i=1, 2), and the (most significant) eigenvector weighted by the eigenvalue is represented by the vector bi (i=1, 2). We will show in the later section that the solution vector x satisfying (A1+A2)x = b1 +b2 for PPSCS offers the boundary information for each party to estimate the distance between the eigenvectors of the data matrices of the two parties. This distance estimate then forms the basis for comparing the biometric data of both parties as well as the generation of helper data for reconstructing the source data. Further details about this will be discussed in the later section.

In this research we tackle the problem of PPCSC by employing homomorphic encryption and singular value decomposition (SVD) on the matrices of P1 and P2 to achieve privacy protection. This approach was discussed in our BTAS 2009 paper [13]. For completeness, we will include the high level summary in section IV.

III. SECURE INFORMATION PROCESSING (SIPPA)

In this research the privacy model for biometrics can be formulated as below: Party P1 has some biometric data expressed in terms of a linearized vector D1. Party P2 has some linearized vector template D2 about a subject P3. The objective is for P1 to determine whether D1 and D2 are similar under the following conditions:

1. P1 and P2 do not reveal their private data to each other.
2. P1 and P2 both need to determine whether D1 and D2 are *sufficiently similar*.
3. If D1 and D2 are *sufficiently similar*, P2 will provide some *helper data* HD with a negligible overhead for P1 to reconstruct D2 using only D1 and HD.

Eigen-based approach has been developed in the early 90s for people identification based on face biometrics [19]. An important property of eigenface is to represent a linearized image as a covariance matrix with its corresponding eigenvectors as a set of linearly independent basis for capturing the variation of the image.

Covariance matrix of an image representation is symmetric. In fact, any linearized data vector D will yield a symmetric matrix out of $D \cdot D^T$. Let A1 and A2 be the proper transformation of some linearized data D1 and D2 through $D1 \cdot D1^T$ and $D2 \cdot D2^T$ respectively. Let $\{(\lambda^1_i, V^1_i) | i= 1,2\ldots\}$ and $\{(\lambda^2_i, V^2_i) | i= 1,2\ldots\}$ be the corresponding sets of eigenvalues and unity normalized eigenvectors for A1 and A2 respectively. For the sake of discussion and without the lost of generality, we will focus on only the largest Eigen components (λ^1_1, V^1_1) and (λ^2_1, V^2_1). Let x be a vector such that $(A1+A2)x = \lambda^1_1 \cdot V^1_1 + \lambda^2_1 \cdot V^2_1$. By definition, $A1V^1_1 = \lambda^1_1 \cdot V^1_1$ and $A2V^2_1 = \lambda^2_1 \cdot V^2_1$. These relationships manifest an

endomorphism mapping with the following three observations:

Observation 1: $\lambda^1_1 \cdot V^1_1$ and $\lambda^2_1 \cdot V^2_1$ are the transformation of the eigenvectors V^1_1 and V^2_1 through A1 and A2 respectively.

Observation 2: The resultant sum of the vectors $\lambda^1_1 \cdot V^1_1$ and $\lambda^2_1 \cdot V^2_1$ are the transformation of the vector x through (A1+A2).

Observation 3: Vector x can be decomposed into components with V^i_1 (i=1,2) as basis; i.e., $x = V^1_1 + \varepsilon_1$ and $x = V^2_1 - \varepsilon_2$; whereas ε_i (i=1,2) can be considered as an error/offset term accounting for the deviation of x from V^i_1 (i=1,2).

The graphical interpretation of the observations just mentioned above is illustrated below:

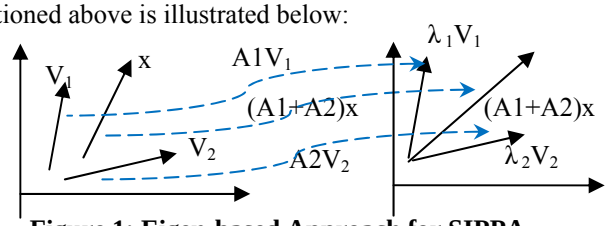

Figure 1: Eigen-based Approach for SIPPA

As one could notice from above, x converges to the eigenvector as (λ^1_1, V^1_1) and (λ^2_1, V^2_1) converge to each other. This, together with observation 3, lead to the following trivial, yet important property:

Property 1: As (λ^1_1, V^1_1) and (λ^2_1, V^2_1) converge to each other, x converges to the eigenvector V^i_1 (i=1,2) and ε_is (i=1,2) converge to zero.

The key mathematical structure of SIPPA is the algebraic system of linear equations defined by $(A1+A2)x = \lambda^1_1 \cdot V^1_1 + \lambda^2_1 \cdot V^2_1$. These linear equations define the constraint relationship between a separator boundary and the eigenvectors of the two parties; whereas the solution to the algebraic system reveals the information needed to derive the lower bound distance of the norm deviation of the eigenvectors. If the lower bound deems acceptable, the eigenvalue and some scalar value would be sent by the server (P2) to the client (P1) as the helper data. Based on the helper data and D1, P1 can derive the norm deviation between the eigenvectors of P1 and P2 under the assumption of equi-distance. Subsequently, P1 can derive a sufficiently good approximation of D2.

As one could note in property 1, it essentially states that one could infer the closeness between V^1_1 and V^2_1 through the V^i_1 and x without knowing V^j_1– the basis of SIPPA; where (i j) = (1 2) or (2 1). Furthermore, it can easily be shown that $A1\varepsilon_1 = A2\varepsilon_2$, or $D1 \cdot D1^T \cdot \varepsilon_1 = D2 \cdot D2^T \cdot \varepsilon_2$, which provides a convenient way in the SIPPA scheme to derive Dj when the scalar $Dj^T \cdot \varepsilon_j$ is known. In other words, SIPPA can easily facilitate "separation of duty" in the sense that data exchange/processing is only possible when both parties collaborate. Similarly, SIPPA separates the step for similarity comparison and that for retrieval. Therefore, the security principles "need-to-know" and "least privilege" can be implemented in the SIPPA environment.

IV. SIPPA ALGORITHMIC DETAILS

There are three major aspects of SIPPA: (1) derivation of the eigenvalues and the corresponding unity normalized eigenvectors of the symmetric matrix representation of the data; (2) derivation of a boundary vector separating the eigenvectors of the two parties, which is formulated as a two-party PPCSC secure computation, and (3) reconstruction of the source data based on the helper data composed of the eigenvalue and a scalar derived from the vector product between the transpose of the linearized source data vector and the boundary vector. We will first outline the key steps of SIPPA, and then the secure computation protocol for PPCSC.

Let Dv and De be the sample and source linearized data respectively. The key steps of SIPPA are summarized below:

Step 1: Derive symmetric matrix representation of the data in the form of $Dv \bullet Dv^T$ (= A1) and $De \bullet De^T$ (= A2).

Step 2: Derive the eigenvalues and the corresponding unity normalized eigenvectors $\{(\lambda^1_i, V^1_i) | i = 1,2\ldots\}$ of $Dv \bullet Dv^T$ and $\{(\lambda^2_i, V^2_i) | i = 1,2\ldots\}$ of $De \bullet De^T$.

Step 3: Compute x such that $(A1+A2)x = \lambda^1_1 \cdot V^1_1 + \lambda^2_1 \cdot V^2_1$.

Step 4: Derive the closeness between De and Dv via the min. distance between V^1_1 and V^2_1. The min. distance between V^1_1 and V^2_1 is estimated via $\|V^2_1 - x\|$.

Step 5: If De and Dv are sufficiently close as measured by $\|V^2_1 - x\| <$ some pre-defined threshold, proceed to send the following helper data: λ^2_1 and $De^T \cdot x$.

Step 6: Derive $\underline{V}^2_1 = V^1_1 + 2(x - V^1_1)$, and then the estimated source data $De' = \lambda^2_1 \cdot \underline{V}^2_1 / De^T \cdot x$

The boundary vector x in step 3 will be derived under PPCSC. The basic idea behind the secure computation protocol for PPCSC required in step 3 for solving $(A_1+A_2)x = b_1 + b_2$ is to solve $P_1(A_1+A_2)P_2 y = P_1(b_1+b_2)$; where (P_1, A_1, b_1) are private to P1, and (P_2, A_2, b_2) are private to P2. Note that even if P2 knows $P_1(A_1+A_2)P_2$ and $P_1(b_1+b_2)$, P2 can only derive y but could not know A_1, b_1, and P_1; thus the privacy of A_1, b_1, and P_1 for P1 is preserved. Once y is solved, P2 can derive $x = P_2 y$ and sends it to P1. Note that any adversary intercepting or sniffing from the network the value of x cannot derive De or Dv unless the adversary also has either (A_1, b_1, V^2_1) or (A_2, b_2, V^1_1).

During the process of secure computation, A_1, b_1, A_2, b_2 are never exposed individually except $P_1(A_1+A_2)P_2$ and $P_1(b_1+b_2)$ for P2. Therefore, it is information-theoretic secure; i.e., the privacy of (A_1, b_1) for P1, and the privacy of (A_2, b_2) for P2, is guaranteed. Readers interested in the step-by-step details on the secure computation mechanism for solving x in $(A_1+A_2)x = b_1+b_2$ are referred to our paper elsewhere [14].

V. BIOAPI STANDARD BASED SIPPA IMPLEMENTATION

In this section we describe the implementation of SIPPA and its design for deployment. The implementation objective is to realize SIPPA as a standard based service component for its primary use case on biometric data. As such, our focus is to implement SIPPA as a service component in compliance with the ISO standard for biometrics – BioAPI 2.0 [15]. The advantage of a standard based service component is interoperability between SIPPA and other application services within the BioAPI framework.

Although the deployment goal specific to this project is to expose SIPPA as a service component for BioAPI framework, we also aim for an easy adaptation of the service

component for the standards developed for other application domains; e.g., ISO 19092:2008. Our strategy towards staging the SIPPA service component for flexibility and adaptation is to employ Java RMI technology. Specifically, the "core" of the SIPPA server and SIPPA client are wrapped as two RMI servers that provide services reachable from the corresponding RMI clients. These RMI clients are in turn wrapped as a *processing algorithm unit* managed by a BFP (Biometric Function Provider), which is attached to a BSP (Biometric Service Provider) and made available to other BSPs. For an end user to access the service provided by SIPPA, one must use the "gateway" provided by the BSP. By providing a gateway, we allow the safe exposure of SIPPA as a service component that is also scalable to further improve performance through parallel processing.

Scaling the BioAPI system to processing among multiple SIPPA client and server "cores" is an attempt to increase the overall performance of the system as well as reliability. By sending an end user's job to be processed among *n* SIPPA clients and servers, we achieve parallel processing assisted by multithreaded Java RMI servers. To maintain the integrity of the SIPPA process over multiple clients and servers, we designed "controller" application to manage the communication between clients and servers, to develop efficient I/O and to optimize the performance of each client server pair based on the size of the end user's input. The deployment of SIPPA as a service component for the BioAPI framework is shown below:

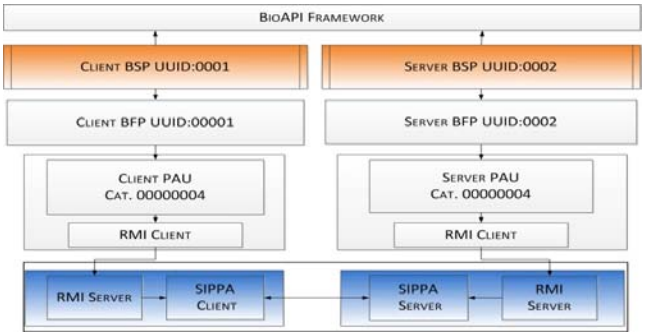

Figure 2: SIPPA Deployment for BioAPI Framework

VI. EXPERIMENTAL STUDIES AND RESULTS

For proof-of-concept, we conducted experimental studies to better understand the potential of SIPPA for biometric applications. This experimental study consists of three parts. The first part is a simulation study targeting at the specific parameters to investigate their inter-relationship. This experimental study aimed at tackling the following two questions: 1. How is the quality of the estimate on the closeness between De and Dv affected by the dimension of x? 2. How is the closeness between De and De' (estimated De) affected by (a) $||V^2_1 - x||$, and (b) dimension of x? The second part is an application of SIPPA to the private reconstruction of digital images, and the third part is the private reconstruction of the voiceprint.

Experimental study part 1

In the simulation study, we generated 5 test data sets categorized by different dimensions. The vector dimensions in these five data sets are 5, 10, 20, 40, and 60 respectively. In each data set, 10 pairs of client (Dv)/server (De) vectors are generated, thus totaling 50 pairs for all dimensions. Figure 3 shows the normalized data difference |De-Dv|/Dim in y-axis and the normalized eigenvector difference $|V^2_1 - V^1_1|$/Dim in x-axis. Figure 4 shows graphically the relationship between the normalized deviation measure $|V^2_1 - V^1_1|$/Dim and $|V^2_1 - x|$/Dim.

Figure 3 shows an approximately linear relationship between the deviation of the client (Dv)/server (De) and that of the corresponding eigenvectors. The degree of closeness between two sources of data is reflected by the closeness between the corresponding eigenvectors. In other words, the client with Dv and the corresponding eigenvector V^1_1 will be able to deduce the degree of closeness between Dv and De if the similarity distance as measured by 2-norm $|V^2_1 - V^1_1|$ is known. And if the degree of closeness between Dv and De is known, it provides a basis for reconstructing De.

Figure 4 also shows an approximately linear relationship in relatively lower vector dimension (≤ 20) between the degree of closeness of the eigenvectors (of the client and server), and that of the server (thus the client in an inversely proportional sense) and the boundary vector x. Consequently, if the dimension of the data vector is relatively low (i.e., ≤ 20), then the degree of closeness as measured by the 2-norm $|V^1_1 - x|$ or $|V^2_1 - x|$ can act as a predictor for $|V^2_1 - V^1_1|$.

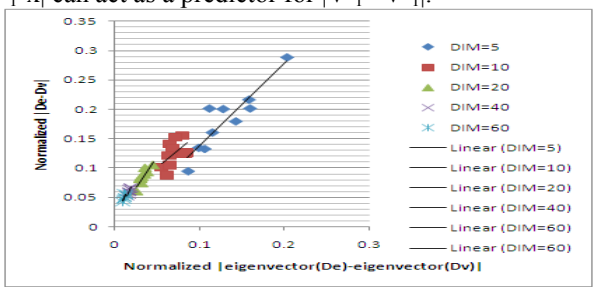

Figure 3: Data deviation vs eigenvector deviation

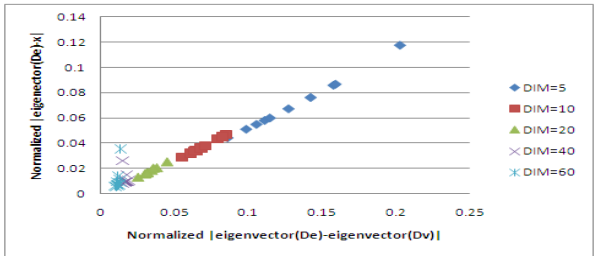

Figure 4: Goodness of fit of the difference estimator

Based on the result of the simulation study, we chose the data dimension to be 10 in the second and third part of our experimental studies.

Experimental study part 2

In the SIPPA experimentation with digital images, three doll faces and one real person image containing biometric face information were used. All images are stored in PGM (Portable Gray Map) format. In the experimentation using doll faces, Figure 5 shows the result of the experimentation. The first row shows two black-and-white original images of two Barbies. The first column shows the reference sample images consisting of two other Barbies, and a generic blonde

hair doll. The resolution of each image is 64x85 with 255 gray-level. Altogether 6 doll faces are reconstructed from every combination pair of a reference image and a sample image. In the magnified version of Figure 5, the best reconstruction is from Barbie-to-Barbie, while the worst reconstructions are the two on the last rows.

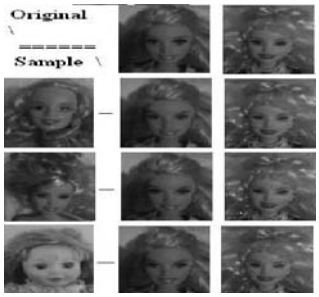

Figure 5: SIPPA application to doll's face

When using the real human image, the server side is a black-and-white image with 255 gray level (Figure 6). The resolution of the image is 256x192. The linearization of the image results in a 49152 (=256x192) x 1 vector. This is about 9 times larger in comparing to the linearization of the doll images, which have a vector size of 5440 (=64x85) x 1. Obviously the dimensions of the linearized vector in both cases are beyond the optimal performance range of SIPPA. As such, the 49152x1 (or 5440x1) linearized image vector is split into multiple 10x1 vectors. SIPPA is then applied iteratively to reconstruct the original image.

During the reconstruction, two parties − referred to as client and server − will participate in a SIPPA session. The client has an image, referred to as a sample image. The server has an image, referred to as a source image. In the SIPPA session, each party takes a portion of the linearized image sequentially in the form of a 10x1 vector to construct a symmetric matrix, and derives the eigenvalue/eigenvector of the matrix. Then both parties participate in a secure computation protocol for PPCSC as described in step 3 of SIPPA in the previous section. The outcome of PPCSC is the boundary vector x. The client party then uses the boundary vector x and the client side 10x1 linearized vector of the sample image to reconstruct the server side 10x1 linearized vector of the source image. The only information that the server provides is the helper data composed of the eigenvalue of the server side 10x1 linearized source image and a scalar. The original 10x1 linearized source image is never shared with the client party. Once the reconstruction of the 10x1 server side linearized source image is completed, the SIPPA process repeats for the next block of the 10x1 linearized vector until all 10x1 image blocks are processed. The entire source image is then reconstructed by the client party using the estimated 10x1 source image blocks.

The SIPPA reconstruction of the entire human source image is conducted twice in this experiment. In the first trial, the client side provides a sample image shown in Figure 7 that is sufficiently similar to the server side source image as shown in Figure 6. All images have a gray scale of 255 and have the same resolution 256x192.

The outcome of the reconstruction by applying SIPPA based on the server side source image (Figure 6) and the client side sample image (Figure 7) is shown in Figure 8. During the reconstruction, the source image is never shared with any party except the corresponding eigenvector information used in the SIPPA processing.

During the second trial, a sample image of a different human subject (not Figure 6 or 7) is used on the client side for SIPPA. The image of the other human subject is not shown in this paper due to the restriction on the human subject clearance. The outcome of the reconstruction is shown in Fig. 8, which shows a poorer quality in comparison to Figure 9.

Figure 6: Source image

Fig. 7: Similar sample image

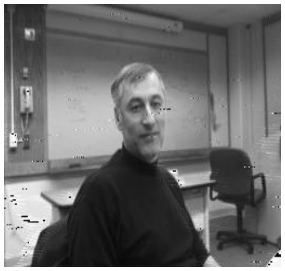

Fig. 8: Reconstruction using a similar image

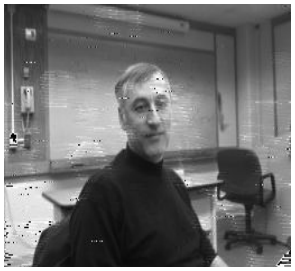

Fig. 9: Reconstruction using a dissimilar image

By visual inspection and using Figure 6 as a reference, the face biometrics in Figure 8 is preserved better than that in Figure 9 – when the client side sample image is closer to that of the server side source image (Figure 6).

Experimental study part 3

The experimental study of SIPPA using voiceprint is conducted using the speaker verification system reported in our paper last year [13]. The objective of the experimental study is to study SIPPA in terms of its ability to reconstruct voiceprints that can cause the speaker verification system to behave in the same way as if the original voiceprints of its users are applied to the system. Mathematically, this is equivalent to examine the difference in the distance function scores. The distance function score is computed when the enrolled voice template is compared against the voiceprint of a speaker. Then the distance function score is computed when the enrolled voice template is compared against the voiceprint privately reconstructed by SIPPA. And finally the two scores are compared against each other.

In this study, the distance function is the (symmetric version of the) KL-distance as we discussed elsewhere [14], while the similarity measure computed by SIPPA is based on 2-norm Euclidean distance. Both KL-distance function and 2-norm Euclidean distance function converge to zero when the sample voiceprint and enrolled template are identical. However, the divergence rates are different when the two are different; thus the possible differences in the distance scores.

Twelve speakers of different native languages participated in the experimental study. Each speaker enrolled three times. The best one judged by the speaker himself/herself is used as

the enrolled voice template for comparison during a verification process. Altogether 391 voiceprints were collected during the calibration and verification phases over a period of seven days.

After filtering the instances of FTE (Failure to Enroll) and FTA (Failure to Acquire), each voiceprint for verification and the enrolled voice template are used by SIPPA to reconstruct the voiceprint of a speaker for the speaker verification system; i.e., the speaker does not present his/her voiceprint to the speaker verification system. The distance between the enrolled voice template and the voiceprint reconstructed by SIPPA - abbreviated by KL-dist(enroll, sippa), is computed. For the control experiment, the distance between the enrolled voice template and the original voiceprint – abbreviated by KL-dist(enroll,original) is also computed.

The system behavior characterized by RoC using original speaker voiceprints and SIPPA reconstructed voiceprints are shown in Figure 10 for a comparison purpose. In this experimentation we deliberately set the *pre-defined threshold* as stated in step 5 of the SIPPA algorithm in the previous section to be infinity. In doing so, the "usability" of SIPPA is highest, while the performance is expected to be the lowest when compared with the cases where pre-defined threshold is used to filter the cases where the source and sample data are significantly different.

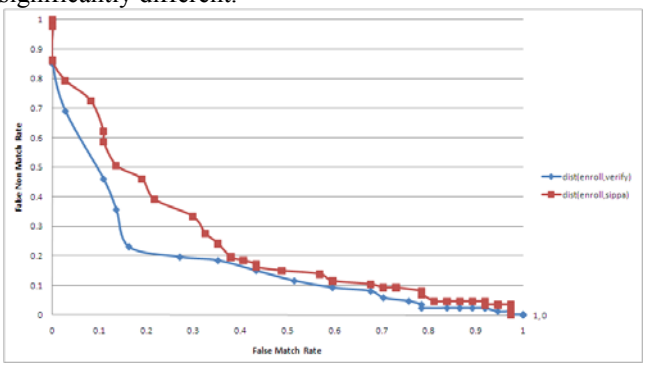

Fig. 10: System behavior characterized by ROC curves

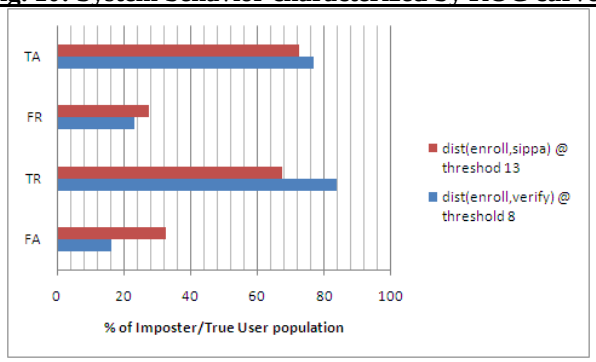

Fig. 11: System behavior as examined by TA,TR,FA,FR

As shown in Figure 10, the control ROC curve using KL-dist(enroll,original) has an EER of approximately 0.2, whereas the ROC curve using the SIPPA reconstruction (i.e KL-dist(enroll,sippa)) has an ERR of approximately 0.3. The optimal threshold setting as defined by the distance closest to (0,0) is approximately 8 for KL-dist(enroll,verify) and is approximately 13 for KL-dist(enroll,sippa). We then investigated the system behavior when the threshold is set at 8 for KL-dist(enroll,verify), and at 13 for KL-dist(enroll,sippa). Figure 11 shows the breakdown on the cases for true acceptance/rejection, and false acceptance/rejection.

VII. CONCLUSION

This paper presents a secure computation technique SIPPA that guarantees security and privacy for the reconstruction of images useful for medical biometrics. SIPPA aims for sufficient privacy homomorphism and computational efficiency. Although it is not complete privacy homomorphism, SIPPA can be applied to the class of secure computation problems that is linearly decomposable. For proof of concept, we conducted both simulation study and the application of SIPPA to reconstructing images with face biometrics and voiceprints. Although our experimental study showed that the results were consistent with the theory behind SIPPA, SIPPA is only operated on a batch mode and has not yet met the real time performance requirements. Specifically the computational load increases in (1) the quadratic order of the image resolution, and (2) the choice of color quality and gray scale level. In the application of SIPPA to voice biometrics, we need to further investigate the mathematical relationship between different distance functions as used by SIPPA and a biometric system. Our future research will focus on these aspects and a practical slice-based architecture for realizing the full potential of SIPPA for real time performance in a BioAPI 2.0 environment.

REFERENCES

[1] N.K. Ratha, J.H. Connell, R.M. Bolle, "Enhancing Security and Privacy in Biometric-based Authentication Systems," IBM System Journal, 40(3), 2001.

[2] N.K. Ratha, S. Chikkerur, J.H. Connell, R.M. Bolle, "Generating Cancelable Fignerprint Templates," IEEE Trans. On PAMI, 29(4) :561-572, 2007.

[3] http://cs-www.bu.edu/faculty/reyzin/papers/fuzzysurvey.pdf Y. Dodis, L. Reyzin, A. Smith, "Fuzzy Extractors: A brief Survey of Results from 2004 to 2006."

[4] K. Nandakumar, A.K. Jain, S. Pankanti, "Fingerprint-based Fuzzy Vault : Implementation and Performance," IEEE Trans. On Information Forensics and Security, 2 :744-757, 2007.

[5] A.C.Yao., "Protocols for secure computations," In *23rd IEEE Sym. on Foundations of Computer Science* (1982).

[6] C. Gentry. "Fully Homomorphic Encryption Using Ideal Lattices," In the 41st ACM Symposium on Theory of Computing (STOC), 2009.

[7] E. Newton, L. Sweeney, B. Mali, "Preserving Privacy by De-identifying Facial Images," IEEE Transactions on Knowledge and Data Engineering, 17 (2) February 2005, pp. 232-243.

[8] R. Gross, E. Airoldi, B. Malin, L. Sweeney, Integrating Utility into Face De-Identification. 2005

[9] Goldreich, O.: Secure Multi-Party Computation (working draft),

[10] A.C. Yao, « Protocols for Secure Computation, » in 23rd IEEE Sym. On Foundations of Computer Science (1982).

[11] W. Du, M.J. Atallah, "Privacy-Preserving Cooperative Scientific Computations." In *14th IEEE Computer Security Foundations Workshop*, pp. 273-282 (2001)

[12] W. Du, M.J. Atallah, "Secure Multi-Party Computation Problems and Their Applications: A Review and Open Problems." *In New Security Paradigms Workshop*, pp. 11-20 (2001)

[13] B.K. Sy, "Slice-based Architecture for Biometrics: Prototype Illustration on Privacy Preserving Voice Verification," *Proc. of the IEEE 3rd Inter Conf on BTAS*, D.C., Sept 2009.

[14] B.K. Sy, "Secure Computation for Biometric Data Security – Application to Speaker Verification," Special Issue on Biometric Systems, IEEE Systems Journal, Dec 2009.

[15] BioAPI 2.0, ISO/IEC 19784-1, *Information Technology – BioAPI – Biometric Application Programming Interface – Part 1 Specification.*