Improving the Accuracy and Storage Cost in Biometric Remote Authentication Schemes

Neyire Deniz Sarier

Bonn-Aachen International Center for Information Technology,
Computer Security Group, Dahlmannstr. 2, D-53113 Bonn, Germany

Abstract

Recently, Bringer et al. proposed a new approach for remote biometric based verification, which consists of a hybrid protocol that distributes the server side functionality in order to detach the biometric data storage from the service provider. Besides, a new security model is defined using the notions of Identity and Transaction Privacy, which guarantee the privacy of the identity-biometrics relationship under the assumption of non-colluding servers. However, due to the high communication and computational costs, the systems following this model cannot be implemented for large scale biometric systems.

In this paper, we describe an efficient multi-factor biometric verification system with improved accuracy and lower complexity by considering the range information of every component of the user biometrics separately. Also, the new scheme is provably secure based on the security model of Bringer et al and implements a different database storage that eliminates the disadvantages of encrypted biometric templates in terms of ciphertext expansion. Also, we evaluate different Private Information Retrieval (PIR) schemes applicable for this setting and propose a practical solution for our scheme that reduces the computation costs dramatically. Finally, we compare our results with existing provably secure schemes and achieve reduced computational cost and database storage cost due to the single storage of the common features of the users in the system and amortization of the time complexity of the PIR.

Key words: Remote authentication, Biometric template security, Identity privacy, Distributed systems, Private Information Retrieval

Preprint submitted to Network and Computer Applications *January 11, 2009*

1. Introduction

Biometric authentication could be categorized broadly as remote server or client end authentication, where in the first case, the remote server stores the reference biometric data and performs the matching. In a typical biometric based remote authentication scheme, the user registers his identity information and biometrics at the service provider. When the user wants to authenticate himself, the user provides a fresh biometric, which is compared to the previously stored biometric information and a decision is made based on a predefined threshold.

For remote biometric systems, it should not be easy to obtain the biometric data by compromising the central server, where the biometrics of each user is often associated with his personal information. This also affects the social acceptance of the biometric systems especially when biometric data is stored in a central database which can be vulnerable to internal or external attackers. Thus, the security and privacy protection of remote verification should be enhanced by implementing distributed biometric systems, where the goal is to detach the biometric data storage from the service provider and to guarantee the notions of identity and transaction privacy, which have been recently introduced as a new security model for biometric verification. In this model, the user U registers its biometric template in cleartext or in encrypted form at the database DB. Besides, U registers his personal information and the index of the database storage location of his biometrics at the service provider SP. To authenticate himself, U encrypts his (adjusted) biometric template using a homomorphic encryption scheme and sends this to SP, which retrieves the index of U to be used in a Private Information Retrieval (PIR) protocol between SP and DB. Finally, a decision is made after decryption or in the encryption domain by exploiting the homomorphic properties of the underlying encryption scheme. Current systems implementing this approach provide provable security in this new model, however, the biometric templates are stored as encrypted that leads to high database storage costs due to ciphertext expansion. Besides, the use of number-theory based PIR causes enormous computational cost at the DB end. Consequently, one has to design a secure and efficient remote biometric verification scheme for a distributed system, which minimizes the costs of storage, encryption and overall complexity and thus, the scheme also becomes applicable to large scale systems.

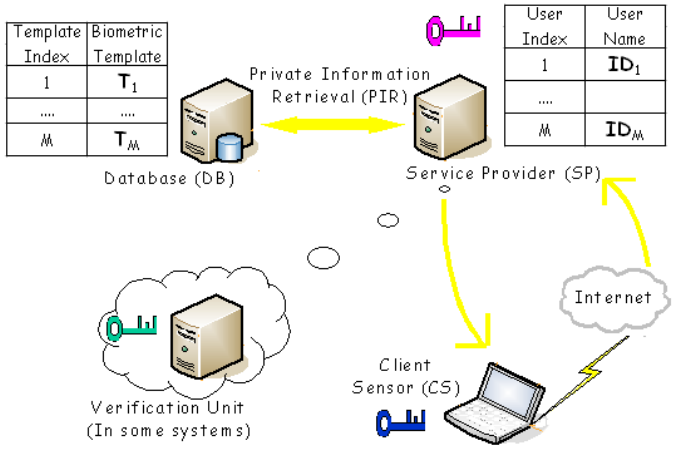
2. Related Work

The first provably secure remote biometric verification scheme for distributed environments is described in the paper of Bringer et al. [4], where the biometric template is assumed as a fixed binary string that is stored as a plaintext and a user sends the encryption of each single bit using Goldwasser-Micali scheme resulting in a high transmission and computation cost. Also, the relationship between the user's identity and his biometrics is kept private by employing a Private Information Retrieval (PIR) scheme with the communication cost linear in the size N of the database. Besides, an identification scheme using a Support Vector Machine and Paillier Public Key System is described in [3], where an attack against the scheme of [4] is presented that reveals the user's biometric data to SP. In [4, 3], a detached verification unit is additionally required for the matching operation and the final decision. Furthermore, the scheme of [4] is improved in terms of communication cost by combining a PIR, a secure sketch and a homomorphic encryption scheme [5, 6, 23]. An overview of these systems is presented in figure 1.

Recently, Sarier [20] proposed a new multi-factor authentication scheme in the framework described in [4] that achieves improved computational costs due to the use of symmetric encryption, which is much more vulnerable to cryptographic attacks compared to asymmetric cryptography. The novel feature of this system is the storage of a random pool of features instead of the biometric templates of each user, which reduces the total database storage cost. However, the scheme is not robust against the variability of the same user's biometric data, i.e. white noise. Besides, the communication cost is higher than the above systems that follow the same security model and no formal security proof is presented. Finally, a survey of the distributed remote authentication systems with detached biometric databases is presented in [21].

3. Motivation and Contributions

Current biometric authentication systems designed in the framework of Bringer et al. work only for biometrics that is represented in discrete form since the biometrics (for instance iris) is assumed to be a binary string in the Hamming space that is either stored as plaintext [4] or input to a secure sketch scheme for discrete domain [5, 6, 23] . However, as it is noted in [22], it is impractical to apply a binary error correcting code on a 2048 bits iris code



Template Index	Biometric Template
1	T_1
....	....
M	T_M

User Index	User Name
1	ID_1
....	
M	ID_M

Private Information Retrieval (PIR)
Database (DB)
Service Provider (SP)
Internet
Client Sensor (CS)
Verification Unit (In some systems)

Figure 1: Current Biometric Based Remote Verification Systems [21]

with a large error correcting capacity. Also, many biometric templates consist of points that are elements of $\mathbb{R}$, hence to handle points in continuous domain, one should quantize the points to a discrete domain with a scalar quantizer Q_λ, where λ denotes the step size. After quantization, a codeword is assigned for each feature by considering a codebook for each quantized domain and by considering the range information of each feature separately to improve the accuracy. This property was not addressed in the systems of [4, 5, 6, 23] and [20], where the latter computes the hash of each biometric feature of arbitrary length using some collision-resistant hash function or maps each feature to an element of a finite field directly as in fuzzy encryption systems [2, 18].

In many biometric applications, the combination of different distance metrics is required, thus, a two-part sketch can be designed to correct the white noise and replacement noise since each point may be slightly perturbed (i.e. white noise) and a small number of points may be replaced (i.e. replacement noise) under noise. As in [22, 8], we focus on the correction of white noise and apply the white noise sketch of [22] although any secure sketch for set difference metric [14, 10] could be further employed to correct the replacement noise. Besides, the security of each quantized feature is provided by Elliptic Curve Discrete Logarithm (ECDL) problem, which guarantees that finding a given g and g^a is practically impossible although calculating g^a is easy. Next, we present a formal security analysis in the model of Bringer et al. and use bilinear pairings in equality testing for final decision as different from previous systems. Moreover, we evaluate different aspects of the used primitives and propose solutions for large biometric authentication systems that could be implemented for border control applications. Also, a detailed overview of current systems is presented and a typical value for each parameter is assigned to have a better understanding of these systems.

In this paper, we describe an efficient multi-factor biometric verification system with improved accuracy and lower complexity by considering the range information of every component of the user biometrics separately. Besides, an efficient and secure form of feature storage is introduced and the security of the system is formally analyzed. Particularly, we will consider the biometric data of a user as a set of quantized features, where each of these features takes some value in some range in the discrete domain. Furthermore, each quantized feature is randomly located as a separate entry in the database instead of storing the biometric template (in cleartext or in encrypted form) of a user, which is a different technique from all the existing schemes, since each feature is stored only once by detecting the common features that are already stored in the database. Specifically, each feature is stored in DB as exponentiations of the generator g of the ElGamal group $\mathbb{G}$ implemented on an elliptic curve. The security of each feature is provided due to the ECDL problem, whereas in [20] that was provided using a cryptographically secure hash function. Also, we try to solve the open problem stated in [20], namely reducing the communication and computational cost of the distributed systems with detached biometric databases. For this purpose, the techniques of hashing and batch codes are applied for amortizing the time complexity of PIR [17]. Furthermore, different PIR systems are evaluated and a practical solution is suggested for large scale biometric systems that could be used in border control applications. Based on this different approach for the database storage, we describe an efficient and accurate remote biometric-based verification system, compare our results with existing provably secure schemes in the framework of Bringer et al.'s model and achieve reduced computational cost and database storage cost due to the single storage of the common features and amortization of the time complexity of the PIR.

4. Definitions and Preliminaries

Definition 4.1. Negligible Function: *$negl(k) : \mathbb{N} \rightarrow \mathbb{R}$ is a function such that for every constant c, there exists an integer k_c with $negl(k) < k^{-c}$ for all $k > k_c$.*

Definition 4.2. Bilinear Pairing: *Let $\mathbb{G}$ and $\mathbb{F}$ be multiplicative groups of prime order p and let g be a generator of $\mathbb{G}$, which is implemented on an elliptic curve. $\mathbb{Z}_p^*$ denotes $\mathbb{Z}_p \setminus \{0\}$ and $\mathbb{G}^*$ denotes $\mathbb{G} \setminus \{1\}$, where $\{0\}$ and $\{1\}$ are the identity elements of $\mathbb{Z}_p$ and $\mathbb{G}$, respectively. A bilinear pairing is denoted by $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{F}$ if the following two conditions hold.*

1. *$\forall\ a, b \in \mathbb{Z}_p$, we have $\hat{e}(g^a, g^b) = \hat{e}(g, g)^{ab}$*
2. *$\hat{e}(g, g) \neq 1_{\mathbb{F}}$, namely the pairing is non-degenerate.*

4.1. Distributed Systems with Detached Biometric Storage

In recent years, the privacy protection and the secure storage of the biometric templates were addressed in a number of papers. As it is noted in [23], privacy protection not only means the attackers inability to compromise the biometric template but also the protection of the sensitive relationship between the identity and the biometric information of the user. To achieve this property, the storage of personal identity information should be separated from the storage of biometrics using the distributed structure of [4, 5, 6, 23, 20, 3], which is composed of the user U_i, the sensor client SC, the service provider SP and the database DB. Some systems require the use of a smartcard for a multi-factor authentication [20] and/or a detached verification unit VU (or a $Matcher$) [4, 3]. The entities of the system (i.e. U_i, SC, SP, VU and DB) are independent (i.e. not colluding) of each other and they are all assumed to be malicious except for the sensor client. This way, SP cannot obtain the biometrics of the user and can have business agreements with different parties that make the sensor client available to users at different locations. Also, DB could function as a trusted storage for different SP's. Since SC captures the biometric data and performs the feature extraction, a biometric smartcard readers could be used as in [1] or SC could be installed as a Trusted Biometric Reader [19] where the smart card of the user for multi-factor authentication schemes verifies the correctness of integrity checking of the TBR. This way, leakage of the biometric data through CS is prevented. In our multi-factor verification scheme, no detached verification unit VU is required as opposed to [4, 3] thus the overall complexity is reduced.

4.2. Security Requirements

4.2.1. Identity Privacy

Informally, this notion guarantees the privacy of the sensitive relationship between the user identity and its biometrics against a malicious service provider or a malicious database even in case of multiple registrations of the same user with different personalized usernames. Briefly, it means that the service provider or the database (or an attacker that has compromised one of them) cannot recover the biometric template of the user [23].

4.2.2. Transaction Privacy

Informally, transaction anonymity means that a malicious database cannot learn anything about the personal identity of the user for any authentication request made to the service provider [23].

The formal definition of the notions Identity and Transaction privacy could be found in [4, 5, 6, 23, 3].

4.3. Private Information Retrieval (PIR)

In order to provide Transaction Privacy, the systems in [4, 5, 6, 23, 20] employ a number-theory based PIR system, which allows the SP to retrieve the i-th bit (more generally, the i-th item) from the DB consisting of m bits while keeping the value i private. The PIR of [11] has an additional benefit of retrieving more then one bit, and in particular many consecutive bits [17]. In this context, a Private Block Retrieval (PBR) protocol enables a user to retrieve a block from a block-database and the PIR/PBR setting of [5] consists of DB containing a list of N blocks $(R_1, ..., R_N)$ and the SP that runs a PBR protocol to retrieve R_i for any $i \in [1, N]$.

4.4. Homomorphic Encryption

To make an authentication decision in the encryption domain based on a certain metric or to construct a number-theory based PIR protocol, we need a secure cryptosystem that is homomorphic over an abelian group. For a given cryptosystem with $(Keygen, Enc, Dec)$ and the message and ciphertext spaces M, C that are groups $Dec(Enc(a) \star Enc(b)) = a * b$, where $a, b \in M$, and $*, \star$ represent the group operations of M, C respectively. The homomorphic encryption scheme that we employ for our setting is described as below.

4.5. ElGamal Encryption Scheme

- **Set up**: Let p be an l_p-bit prime and q an l_q-bit prime so that q divides $(p-1)$. Let $\mathbb{G}$ be the subgroup of $\mathbb{Z}_p^*$ of order q, and g be a generator of $\mathbb{G}$. Let Ω be a one-to-one encoding map from $\mathbb{Z}_q$ onto $\mathbb{G}$.

- **Key generation**: The private key is $x \leftarrow \mathbb{Z}_q$ corresponding public key is $y = g^x$.

- **Encryption**: To encrypt a message $m \in \mathbb{Z}_q$, one encodes m by computing $w = \Omega(m)$, randomly selects $r \leftarrow \mathbb{Z}_q$ and computes $(u, v) = (g^r, y^r w)$. The ciphertext is $c = (u, v)$.

- **Decryption**: To decrypt a ciphertext $c = (u, v)$, one computes $w = vu^{-x}$ and recovers the original plaintext $m = \Omega^{-1}(w)$.

This cryptosystem is one-way under the CDH assumption, and indistinguishability holds under the DDH assumption.

4.6. Variability of Biometrics

As in [22, 8], we focus on the white noise, which means that each point in the continuous domain can be perturbed by a distance less than δ. Particularly, we assume that each biometrics can be written as a sequence $b = (\nu_1, ..., \nu_k)$, where a feature ν is an element of the universe U such that $\nu \in \mathbb{R}$ and $0 \leq \nu < 1$. R denotes a similarity relation on U, $R \subset U \times U$. For each pair of biometrics (b, b'), one can write $(b, b') \in R$, if there exists a set $S \subset b \cap b'$ with $|S| \geq t$ for some threshold t, and for every $\nu \in S$, $|\nu - \nu'| < \delta$ for some threshold δ. The quantizer Q_λ is a member of a family of quantizers Q parameterized by the step size λ, which is defined as $Q_\lambda : U \rightarrow M$, where M denotes the set of finite points. In other words, a quantization is applied to transform the points in the continuous domain to a discrete domain and the step size $\lambda \in \mathbb{R}$ as a measure of the precision of the quantized biometrics. We assume that $0 < \lambda \leq \delta$ [22].

For example, for a feature $\nu \in \mathbb{R}$ we employ a scalar quantizer Q_λ with step size $\lambda = 0.001$ to map the feature to an integer in [0,1000], such that $Q_\lambda(\nu) = w$. The quantization of b is defined as $Q_\lambda(b) = \langle Q_\lambda(\nu_1), ..., Q_\lambda(\nu_k)\rangle$ and the corresponding quantized domain is $M_\lambda = [0, \lceil\frac{1}{\lambda}\rceil]$.

Similar to the case in the continuous domain, we have $|w - w'| < \delta_\lambda$ in the quantized domain , where $\delta_\lambda = \lceil\frac{\delta}{\lambda}\rceil$.

Furthermore, for each quantized domain M_λ we consider a codebook C_λ, where every codeword $c \in C_\lambda$ has the form $c = z(2\delta_\lambda + 1)$ for some non-negative integer z. We use $C_\lambda(\cdot)$ to denote the function such that given a quantized feature w, it returns a value $c = C_\lambda(w)$ such that $|w - c| \leq \delta_\lambda$. That is, the function finds the unique codeword c that is nearest to w in the codebook [22].

4.7. Secure Sketches

In many systems, biometrics is assumed as a fixed binary string, which is obtained by quantizing the features to generate multiple bits per feature, coding per feature and concatenating the output codes to be used in error correction coding (ECC) [9]. The main purpose of Secure Sketches is to correct the noise in the biometric measurement by using some public information PAR, which is derived from the original biometric template b as follows [10].

- The **Gen** function takes the biometrics b as input and returns the public parameter PAR,

- The **Rep** function takes a biometric b' and PAR as input and computes b if and only if $\mathbf{dis}(b, b') \leq t$, where $dis()$ is the distance metric used to measure the variation in the biometric reading and t is the error tolerance parameter.

An important requirement for such a scheme is that the value PAR should not reveal too much information about the biometric template b [7]. The first scheme of [5] and the scheme of [23] implement a secure sketch protocol to test for equality in the encryption domain using the homomorphic property of the encryption system.

For our setting, we implement the white noise sketch of [22] that corrects the white noise on each component of the biometric vector is as follows:

- The **Gen** function takes the quantized biometrics $Q_\lambda(b) = (w_1, ..., w_k) \in M_\lambda$ as input and computes for each w_i, $c_i = C_\lambda(w_i)$ and outputs the public parameter $PAR = (\Delta_1, ..., \Delta_k) = (w_1 - c_1, ..., w_k - c_k)$,

- The **Rep** function takes a quantized fresh biometric $Q_\lambda(b')$ and PAR as input and computes $c_i = C_\lambda(w'_i - \Delta_i)$ for $i \in [1, k]$ and outputs $Q_\lambda(b) = (c_1 + \Delta_1, ..., c_k + \Delta_k)$.

5. A New Biometric Authentication Scheme

In this section, we present a new multi-factor biometric verification scheme using a different approach for storing the biometric features resulting in a secure and more efficient protocol compared to the existing protocols. For this purpose, we use ElGamal encryption scheme and a suitable signature scheme. Also, an efficient PIR protocol is required, which allows SP to retrieve an item from the DB without revealing which item SP is retrieving.

5.1. Authentication Workflow

Our system consists of four independent entities: A human user U with a smart card, the client sensor CS, the service provider SP and the database DB. Similar to existing authentication schemes, our system is composed of two phases: the registration and the verification phase, where the registration phase has a different structure compared to existing schemes.

1. In the registration phase, the human user U presents its biometrics b to CS, which computes the public parameter $PAR = (\Delta_1, ..., \Delta_k)$ using the codebook C_{λ_i} that is selected according to the range information δ_{λ_i} of each quantized feature w_i in the discrete domain. The parameters of the transformations (λ_i, Δ_i) are stored in the smart card of U. Next, U registers each quantized feature after some transformation at a randomly selected storage location i_j in DB and registers his personalized username ID at the SP. Finally, U stores the index list $Index = (i_1, ..., i_k)$ as encrypted with the public key of SP in the smart card. Here, the size of the database is denoted as N and the dimension of the user's feature vector is denoted as k.
2. In the verification phase, the user U presents its biometrics to CS, which computes the feature vector b' in the continuous domain. Using the parameters stored in the smart card, CS computes $w_i = C_{\lambda_i}(w'_i - \Delta_i) + \Delta_i$ via the PAR and the codebook C_{λ_i} for $i = 1, ..., k$. In practice, $\lambda_i = \delta_i$ as in [22]. Using cryptographic techniques, SP communicates with CS and DB to accept or reject the user U using the set overlap as the distance metric, where the threshold t represents the error tolerance in terms of minimal set overlap.

5.2. Assumptions on the system

- Sampling Assumption : In the registration phase, enough number of samples (biometric features) is obtained from each user to assign a codeword $c_i \in C_{\lambda_i}$ for the computation of PAR by considering the corresponding range information of each feature separately. The features are always ordered and in continuous domain. The parameters of this transformation (i.e. λ_i, Δ_i) are determined and stored in the user's smart card.

- Liveliness Assumption: This is an indispensable assumption for any biometric system as it guarantees with high probability that the biometrics is coming from a live human user.
- Security link Assumption: To provide the confidentiality and integrity of sensitive information, the communication channel between U, CS, SP, DB should be encrypted using standard protocols.
- Collusion Assumption: Due to the distributed system structure, we assume that the user U, SP and DB are malicious but they do not collude. Additionally, the CS is always honest.

5.3. Registration Phase

The registration phase consists of the following initialization of the components.

1. The parameters of the ElGamal encryption scheme are initialized by choosing the groups $\mathbb{G}$ and $\mathbb{F}$ of prime order p and g as a generator of $\mathbb{G}$. To avoid the encoding problem, we use the same group $\mathbb{G}$ and the generator g as the ElGamal public parameters for the DB, CS and SP as in [23]. Also, we denote a bilinear pairing as $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{F}$ and use a hash function $H : \{0, 1\}^* \rightarrow \mathbb{Z}_p^*$.
2. DB generates an ElGamal key pair (pk_{DB}, sk_{DB}), where $pk_{DB} = (\mathbb{G}, g, g^{db})$ and $sk_{DB} = db$.
3. SP generates an ElGamal key pair (pk_{SP}, sk_{SP}), where $pk_{SP} = (\mathbb{G}, g, g^{sp})$ and $sk_{SP} = sp$.
4. CS generates an ElGamal key pair (pk_{CS}, sk_{CS}), where $pk_{CS} = (\mathbb{G}, g, g^{cs})$ and $sk_{CS} = cs$.
5. CS and SP generates two key pairs for a signature scheme.
6. The user U presents its biometrics to CS which extracts the feature vector b, quantize it to $w_i \in M_{\lambda_i}$ and computes the PAR as described in section 4.7 .

7. The user picks some random $i_j \in \mathbb{Z}$ where $1 \le j \le k$ and registers g^{μ_j} at the locations i_j of the database, where $\mu_i = H(w_i)$ for $i \in [1, k]$.

   Remark 5.1. *If some of the locations i_j's are already occupied by other features, then the user selects other random indices. Also, if some of the features of the user are already stored in DB, then DB returns the indices of the common features. Thus, common features are not stored more than once, which decreases the total storage cost of DB.*

8. The user U registers its personalized username ID at the SP and stores the index list $Index = (i_1, ..., i_k)$ as encrypted with the public key of the SP together with the parameters in his smart card.

5.4. Verification Phase

The verification phase has the following workflow.

1. The user U presents its biometrics b' to the client sensor CS, which extracts the feature set, quantize it to discrete domain using the parameters stored in the smart card, applies the secure sketch scheme to error correct each quantized feature w'_j as described in section 5.1. Next, CS computes for each $\mu'_j = H(w'_j)$ with $1 \le j \le k$,

   $$V_j = Enc([Enc((g^{\mu'_j}), pk_{DB})], pk_{SP})$$

   CS also signs each V_j and sends for $1 \le j \le k$, (V_j, σ_j) and the encrypted index list to the service provider, where σ_j is the signature of CS on V_j.
2. SP verifies the signature and obtains the index list $Index$ after decryption. SP also decrypts V_j to obtain $Enc((g^{\mu'_j}), pk_{DB}) = (C_j^1, C_j^2)$.
3. For $1 \le t \le N$, DB computes $Enc((g^{\mu_t}), pk_{DB})$ after an authentication request from the SP.
4. SP runs a PIR protocol using the index list of the user and obtains $Enc((g^{\mu_j}), pk_{DB}) = (C_j^3, C_j^4)$ for each $j \in [1, k]$. Next, SP selects randomly $s_j \in \mathbb{Z}_p^*$ to compute for $1 \le j \le k$

   $$R_j = (R_j^1, R_j^2) = \left(\left(\frac{C_j^1}{C_j^3}\right)^{s_j}, \left(\frac{C_j^2}{C_j^4}\right)^{s_j}\right)$$

5. SP checks for $1 \leq j \leq k$ whether

$$\hat{e}(g^{db}, R_j^1) = \hat{e}(g, R_j^2)$$

by computing $2k$ bilinear pairings. Finally, SP counts the number of the equations satisfying the above condition and based on the threshold t, SP decides to authenticate or reject U.

6. Analysis of the Protocol

In the first part, we evaluate the major security criteria that should be satisfied in a biometric authentication system.

6.1. Security Proof for Identity Privacy

Since DB does not have access to any information about the user's identities, DB cannot have any advantage in the Identity Privacy game described in [4]. Even if both the SP and DB are compromised, the adversary will not find a link between the identity data stored in SP and the biometric features stored in DB since the SP does not store any index values of the DB locations as opposed to the systems [5, 23, 4].

Lemma 6.1. *The proposed scheme satisfies Identity Privacy against a malicious service provider under the semantic security of the ElGamal scheme and the existential unforgeability of the signature scheme.*

6.2. Security Proof for Transaction Anonymity

At the registration phase, a user selects a random DB-index for each feature of his biometrics and each feature is stored as a separate entry using this index value. Hence, even if the database is compromised, the attacker would not be able to find an index that points to a biometric template stored as cleartext or encrypted. This also provides security against the database since it only stores a randomly ordered pool of quantized features from different users, where each quantized feature is hashed using a specific cryptographic hash function and stored as exponentiations of the generator g of $\mathbb{G}$ in the database.

Lemma 6.2. *The proposed scheme satisfies Transaction Anonymity against a malicious database under the semantic security of the ElGamal scheme.*

6.3. Efficiency of the Protocol

Our new design has the following advantages in terms of computation and storage costs.

- **Efficient memory storage**: Since each feature is stored as a separate entry in the database, there could be common features belonging to different users. Thus, during registration phase, the database could check for this situation and could return the index of the previously stored feature. This way, the size of the registered feature set and the total storage in the database could be smaller, which could be observed by referring to the experiment in [15], which measures minutiae pair matches for fingerprint verification on a small fingerprint database of 100 users with 8 prints of the same finger as shown in Table 1. In this experiment, the total number of pairs of matched minutiae (i.e. fingerprint feature) is counted for $\binom{50}{2} = 1225$ comparisons of fingerprints belonging to 50 different users. Since a fingerprint is represented by 30-50 minutiae [15], one can easily compute that when our system is applied even on such a small database, we can reduce the storage cost approximately by 10% (i.e, $3991/(50 \cdot 1225 - 3991)$). In case of large identification systems, the storage cost will decrease much more.

Table 1: The number of common features [15]

	No.of Users	No. of Fingerprint Pairs Compared	Total Matched Point Pairs
Same User	50	1400	37705
Diff. User	50	1225	3991

Besides, since no biometric template is stored as an entry, there is no need to apply a homomorphic encryption scheme to store the biometric template as encrypted, where the ciphertext size is twice the plaintext size as in [23, 5] and the storage cost of each user in [6] is given as 128kbytes. Finally, the choice of the system parameters of [6, 4] results in a constraint on the size of DB. However, the database storage cost of our system is $(k-c) \cdot P$ for each user due to the c common features that are not stored twice, where P is the size of an element of the elliptic curve group $\mathbb{G}$. For instance, $P = 171$ bits for a 160-bit ECC curve.

- **Computational cost**: In [6, 4], the database performs $O(N)$ exponentiations modulo n^2 [6] and modulo n [4], where n is an RSA modulus with $|n|$=2048 bits. Similarly, the schemes of [23, 5] require $O(N)$ exponentiations in the group on which the ElGamal public key scheme is defined. The computational cost of our scheme is dominated by the $O(N)$ exponentiations in group $\mathbb{G}$. Finally, PIR protocol also causes high computational cost requiring $\Omega(m)$ operations on the m-bit DB [12] since if the DB does not process some of its entries, it will learn that the user is not interested in them, therefore the PIR system will not provide full privacy.

  In the following table, we summarize various remote biometric-based authentication schemes that satisfy the security model described in section 4.2. When we take typical values for the parameters in Table 2, we obtain the following relations. For biometric modalities with M=512 bytes template sizes [13] and for 160-bit ECC curves, $M \approx kP$, if $20 \leq k \leq 30$ as implemented in [22, 15]. Also, for current PIR systems with communication cost PIR, we have $PIR << O(N)$.

Table 2: Comparison of various biometric authentication systems

Scheme	Computation Cost	Storage at DB-index	Storage per user	Communication cost
Sys. 1 [4]	M exp + $(MN)/2$ mult	M bits	M bits	$O(N)$
Sys. 2 [3]	$O(N)$ exp	$\|n\|k$ bits	$\|n\|k$ bits	$O(N)$
Sys.*3 [5]	$O(N)$ exp	$2M$ bits	$2M$ bits	PIR
Sys. 4 [6]	$O(N)$ exp	$\|n\| \cdot M$ bits	$\|n\| \cdot M$ bits	PIR
Sys. 5 [23]	$O(N)$ exp	$2M$ bits	$2M$ bits	PIR
New Sys.	$O(N)$ exp	P bits	$(k-c)P$ bits	PIR

*The first biometric scheme

Abbreviations: N= number of entries in DB; k=dimension of the feature vector; M= size of the biometric template; P=size of a single stored feature; c= number of common features of a user; $|n|$=size of an RSA modulus

6.4. Complexity of the PIR

Except for the systems [3, 4] with communication complexity $O(N)$, the communication cost of the systems evaluated in Table 2 is dominated by the PIR, which is usually instantiated using the number-theory based PIR systems such as [11], which has currently the best bound for communication complexity of $O(\log(m) + d)$, where d is the size of the block to be retrieved from an m-bit DB. However, the computational cost of a number-theory based PIR is roughly a modular multiplication per bit of DB, which limits the usability of these schemes except for small DBs. In [12], the authors suggest to use batch codes to amortize the computational cost of PIR with a moderate increase on the communication cost, which is already very low. When the SP wants to retrieve k-bits (not necessarily consecutive) out of m-bit DB, batch code constructions can achieve $k^{1+o(1)}$ communication and $m^{1+o(1)}$ computation.

Since our system has to retrieve k non-consecutive blocks of size P, a naive solution is to just run the PIR solution of [11] with complexity PIR independently k times, which results in the complexity of $k \cdot PIR$. However, in [17], the solution to the problem of retrieving k items that are not necessarily consecutive is presented using hashing. This way, the complexity is much smaller than the naive solution, namely $s \cdot PIR$, where $s = \sigma \log(kP)$. Furthermore, better performance is derived via explicit batch codes instead of hashing, since small values of k do not work with hashing. The reader is referred to [17] for a more detailed discussion of application of batch codes for amortizing the time complexity of PIR. Recently, [16] introduced an efficient noise-based PIR scheme, which is 100 times faster than all of the number-theory based PIR systems and has reasonable communication. The communication cost of [16] is not optimal as the cost of [11], however, communication cost is not the main performance measurement of PIR due to the enormous computational cost at the DB-end for number-theory based PIR schemes [16].

6.5. A Practical Solution

As it is noted in [16], the number-theory based PIR systems are impractical except for small DBs. Besides, the additional homomorphic encryptions performed for each entry of DB causes the systems to be unimplementable for large DBs, even if the number-theory based PIR is replaced by the noise-based PIR of [16]. A practical solution for large scale biometric identification systems could be masking the index (or the index list for our scheme) of the user with additional random indices instead of using a PIR scheme. This approach leaks partial information on the item(s) that the user is interested in, especially for the systems that store each biometric template as a single entry of DB. Thus, the probability of the DB to guess which user is actually authenticating is Pr=$\frac{1}{S+1}$, where S is the number of additional random indices. However, in our system, each biometric feature of a user is stored separately at a random entry of DB, hence the above probability is Pr=$1/\binom{S+k}{k}$, which becomes negligible for suitable values of S. This way, our system does not require a PIR and the total computational cost is $S+k$ exponentiations in $\mathbb{G}$ instead of $O(N)$, which results in a suitable system for border control application, which requires biometric databases with millions of users.

7. Conclusion

In this paper, we present a new design for remote biometric verification that follows the state-of-the-art security model for biometric authentication systems. Due to the correction of the white noise, our system is robust against the variability of the user biometrics. Besides, a different storage mechanism for the biometric data is introduced, which results in decreased storage costs even in small databases due to the elimination of the ciphertext expansion problem caused by the encrypted template storage and due to the single storage of the common features of different users. Thus, the size of the stored biometric data is much smaller than in existing systems that store biometrics as encrypted with public key encryption. The system could be applied to a variety of biometrics that could be represented by a feature vector, where each feature point could be an element of $\mathbb{R}$ as in the case of most biometric modalities and there is no need for binarization of the feature vector to generate a standard biometric template for each user necessary for secure sketches working in discrete domain. As a final point, we note that the compromise of the database (namely, a random pool of features) would not help an attacker in the recovery of a user's template, which could otherwise only be guaranteed by storing the biometric templates as encrypted.

Acknowledgement

The author is grateful to her supervisor Prof. Dr. Joachim von zur Gathen for his valuable support, encouragement and guidance.

References

[1] Atallah, M.J., Frikken, K.B., Goodrich, M.T., Tamassia, R.: Secure biometric authentication for weak computational devices. In Patrick, A.S., Yung, M. (eds.) FC 2005. LNCS, vol. 3570, pp. 357–371. Springer (2005)

[2] Baek, J., Susilo, W., Zhou, J.: New constructions of fuzzy identity-based encryption. In ASIACCS 2007, pp. 368–370. ACM (2007)

[3] Barbosa, M., Brouard, T., Cauchie, S., de Sousa, S.M.: Secure biometric authentication with improved accuracy. In Mu, Y., Susilo, W., Seberry, J. (eds.) ACISP 2008. LNCS, vol. 5107,pp. 21–36. Springer (2008)

[4] Bringer, J., Chabanne, H., Izabachène, M., Pointcheval, D., Tang, Q., Zimmer, S.: An application of the goldwasser-micali cryptosystem to biometric authentication. In Pieprzyk, J., Ghodosi, H., Dawson, E. (eds.) ACISP 2007. LNCS, vol. 4586, pp. 96–106. Springer (2007)

[5] Bringer, J., Chabanne, H., Pointcheval, D., Tang, Q.: Extended private information retrieval and its application in biometrics authentications. In Bao, F., Ling, S., Okamoto, T., Wang, H., Xing, C. (eds.) CANS 2007. LNCS, vol. 4856, pp. 175-193. Springer (2007)

[6] Bringer, J., Chabanne, H.: An authentication protocol with encrypted biometric data. In Vaudenay, S. (eds.) AFRICACRYPT 2008. LNCS, vol. 5023, pp. 109–124. Springer (2008)

[7] Bringer, J. Chabanne, H. Cohen, G. Kindarji, B. Zemor, G.: Optimal Iris Fuzzy Sketches. BTAS 2007. pp. 1–6. IEEE (2007)

[8] Chang, E., Li, Q.: Hiding secret points amidst chaff. In Vaudenay, S. (eds.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 59–72. Springer (2006)

[9] Chen, C., Veldhuis, R. N. J., Kevenaar, T. A. M., Akkermans, A. H. M.: Multi-bits biometric string generation based on the likelyhood ratio BTAS 2007. pp. 1–6. IEEE (2007)

[10] Dodis, Y., Reyzin, L., Smith, A.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 523–540. Springer (2004)

[11] Gentry, C., Ramzan, Z.: Single-database private information retrieval with constant communication rate. In Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 803–815. Springer (2005)

[12] Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Batch codes and their applications. STOC 2004. pp. 262–271. ACM (2004)

[13] Itakura, Y., Tsujii, S.: Proposal on a multifactor biometric authentication method based on cryptosystem keys containing biometric signatures. Int. J. Inf. Sec. **4**(4), 288–296 (2005)

[14] Juels, A., Sudan, M.: A Fuzzy Vault Scheme. Des. Codes Cryptography. **38**(2), 237–257 (2006)

[15] Mansukhani, P., Tulyakov, S., Govindaraju, V.: Using support vector machines to eliminate false minutiae matches during fingerprint verification. In Prabhakar, S., Ross, A.A. (eds.) Biometric Technology for Human Identification IV. SPIE, vol. 6539, pp. 65390B. SPIE (2007)

[16] Melchor, C.A., Gaborit, P.: A fast private information retrieval protocol. ISIT 2008. pp. 1848 – 1852. IEEE (2008)

[17] Ostrovsky, R., Skeith, W.E.: A Survey of Single-Database Private Information Retrieval: Techniques and Applications In Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 393–411. Springer (2007)

[18] Sahai, A., Waters, B.: Fuzzy identity-based encryption. In Cramer, R. (eds.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer (2005)

[19] Salaiwarakul, A., Ryan, M.D.: Verification of integrity and secrecy properties of a biometric authentication protocol. In Chen, L., Mu, Y., Susilo, W. (eds.) ISPEC 2008. LNCS, vol. 4991, pp. 1–13. Springer (2008)

[20] Sarier, N.D.: A new approach for biometric template security and remote authentication. In Tistarelli, M., Nixon, M. (eds.) Advances in Biometrics - ICB 2009. LNCS, vol. 5558, pp. 916–925. Springer (2009)

[21] Sarier, N.D.: A Survey of Distributed Biometric Authentication Systems In Biometrics and Electronic Signatures-Research and Applications, BIOSIG 2009. LNI, vol. P-155, pp. 129–140. GI (2009).

[22] Sutcu, Y., Li, Q., Memon, N.: Secure Sketch for Biometric Templates. In Chen, K., Lai, X. (eds) Advances in Cryptology - ASIACRYPT 2006. LNCS, vol. 4284, pp. 99–113. Springer (2006).

[23] Tang, Q., Bringer, J., Chabanne, H., Pointcheval, D.: A formal study of the privacy concerns in biometric-based remote authentication schemes. In Chen, L., Mu, Y., Susilo, W. (eds.) ISPEC 2008. LNCS, vol. 4991, pp. 56–70. Springer (2008)