NEYIRE DENIZ SARIER (2010). Practical Multi-factor Biometric Remote Authentication. In *Fourth IEEE International Conference on Biometrics : Theory Applications and Systems (BTAS)*, 1–6.

Practical Multi-factor Biometric Remote Authentication

Neyire Deniz Sarier
B-IT Cosec, Dahlmannstr. 2, 53113 Bonn, Germany
denizsarier@yahoo.com

***Abstract*— In this paper, we evaluate the security properties of multi-factor biometric authentication (MFBA) and define the notion of User Privacy, where the biometrics is assumed as a set of features that can be either ordered or unordered depending on the biometric modality. We propose efficient schemes for MFBA, which do not incorporate secure sketches, thus, different template extraction methods are applicable. We formally describe the security model for MFBA, where the computations are performed in the encrypted domain but without requiring a decryption key for the authentication decision. Thus, leakage of the secret key of any system component does not affect the security of the scheme as opposed to the current biometric systems involving cryptographic techniques.**

Keywords: Cancelable Biometrics, Remote Authentication, User Privacy, Homomorphic encryption, Zero knowledge proof

I. Introduction

Combining biometrics and cryptography is a promising research topic that aims to protect the privacy of biometrics using cryptographic techniques, where biometrics can be stored on a central database or on a tamper-proof smart card. There exists different types of biometric cryptosystems such as fuzzy extractors, fuzzy vault and recently introduced bipartite biotokens, which could be used for biometric key generation, key binding and key release, respectively. Also, cancelable biometrics perform a distortion of the biometric image or features before matching. The variability in the distortion parameters provides the cancelable nature of the scheme.

However, the implementation of these systems come along with various attacks that question the security of them. Firstly, remote biometric authentication systems are vulnerable to four classes of attacks: Attacks to the sensor via spoofing or compromising the sensor, attacks to the database (tampering with the templates, substitution attacks), attacks to the matcher and intercepting/eavesdropping to the communication channel. The first and second classes of attacks can be avoided by additional security factors (password, smart cards) and by storing the (cancelable) templates as encrypted. Also, if a decryption is performed during the matching stage a Trojan horse type attack can lead to the disclosure of the raw biometric. Thus, the comparison should be made in the encrypted domain without any decryption operation. Finally, the communication between the entities should be encrypted with session keys to prevent the last classes of attacks. Clearly, all of these countermeasures assumes the secrecy of the system's private keys and session keys.

Besides, multi-factor biometric authentication (MFBA) protocols that store the user's biometric template in a smart card combine a local authentication on the card by a remote authentication at the server. Cancelable biometrics is another approach that stores the masked biometrics at the server, where the masking is performed using a one-way transformation or a high entropy randomness that is stored in the user's smart card to be used later for authentication in the transformed space. This way, biometric data stored at the server is protected through this transformation and biometrics can be updated by changing the transformation function or the randomness.

A. Related Work

Juels and Wattenberg [1] introduced the fuzzy commitment scheme, which assumes biometrics as a binary string (for instance a 2048 bit Iris code) and replaces biometric matching algorithms by error-correction techniques. Also, Juels and Sudan have developed the *fuzzy vault* [2], which assumes biometrics as an unordered set of features and is designed for the set difference metric in order to hide a secret key (i.e. an AES key) using biometrics.

Also, Scheirer and Boult proposed revocable biotokens [3] and its implementation in [4], where their system is an example of cancelable biometrics that was introduced by Ratha et al. in [5].

Besides, Bringer et al. [6], [7] defined the security notions for biometric remote authentication and described a new architecture applicable for small-scale biometric systems requiring highest security. A survey of these systems could be found in [7]. Additionally, [8], [9], [10] perform biometric authentication in the encrypted domain.Although a simple client server biometric authentication system is proposed in [11], the decision can be computed after a decryption operation as in the schemes summarized in [7], thus the leakage of the system's secret keys endangers the security of the system. Besides, MFBA systems are proposed in [12], [13], [14], [15], [16], where the last two schemes performs the matching on card for a local authentication followed by a remote authentication. Also, cancelable biometrics is combined with a smart card for storing only the helper information [17], [18] for a MFBA.

B. Motivation and Contributions

When we analyse different biometric authentication systems, we see that the most dangerous event is the

*This work was supported in part by NRW State within the B-IT Research School. The author is grateful to her supervisor Prof. Dr. Joachim von zur Gathen for his valuable support and encouragement.

leakage of the session keys encrypting the communication channel and the system's secret keys that are used for decryption of the stored templates at the matching stage or for decrypting the final decision when homomorphic encryption is used. However, is it possible to have user's privacy even if these keys are compromised. In other words, is there a way to store the biometrics as encrypted and perform the matching in encrypted domain without any decryption operation. Partially, current systems achieve this using homomorphic encryption schemes, however, for the final decision, the system's secret key is still needed. Thus, we need a different encryption method that also determines the final decision without using any secret key. Besides, an attacker that compromised the session key between the server and client could eavesdrop to the communication channel and later perform a replay attack by sending the same ciphertext (i.e. encrypted biometrics) without even knowing the true biometrics of the user. How do we prevent replay attacks? A solution could be attaching a proof of knowledge of the plaintext (i.e. biometrics) to the ciphertext, which proves that the user knows the biometrics without revealing it to the server. However, this zero knowledge proof (ZPK) must include a time stamp and additional data such as user specific information to become non-malleable and to avoid the replay attack that sends the ciphertext and the corresponding ZPK obtained previously. Another point one should consider is the compromise of the sensor. In this case, can we have still privacy?

In this paper, we try to answer these questions and design a new biometric verification protocol that does not require additional detached components at the server end and strong assumptions on the system. Instead, we propose a simple client server architecture for a MFBA by combining cancelable biometrics and cryptographic techniques, where the complete biometric template of the user is not stored in any system component. We formally design the security model for MFBA based on the privacy/security issues summarized as in figure 1, where we allow an adversary trying to impersonate a user against a honest-but-curious server to access different oracles. Basically, these oracles model the adversaries capabilities such as eavesdropping on the communication channel - even in the case of a compromised session key that is used to build a secure communication link before the start of the protocol execution- and compromise of either the sensor (namely biometrics of the user) or the smart card of the user through side channel analysis. The security of our design is based on the security of the underlying homomorphic encryption scheme and the associated zero knowledge proof (ZPK).

Firstly, we follow the biometric template extraction method of [3], where the biometrics is transformed using a scaling and a translation in order to separate the stable and non stable part of each biometric feature. The encrypted stable parts together with the corresponding ZPK's are stored at the service provider and the non-stable parts are stored in clear together with the separation (i.e. transformation) parameters in the tamper-proof smart card of the user. This operation results in a cancelable biometric template as changing the parameters and/or the public key for encrypting the stable parts will lead to a different template. In addition, security against the honest-but-curious server is guaranteed by storing the stable parts as encrypted with the user's public key, where the corresponding secret key is not needed and thus not stored anywhere. Also, this storage mechanism at the server avoids substitution/masquerade attacks due to the secret transformation parameters and encrypted storage, and prevents tampering with the templates due to the ZPK proofs. The main difference to the previously defined systems is that we do not need to use any decryption key at any stage of the protocol and the authentication is performed in the encrypted domain. Currently, the systems perform authentication in the encrypted domain using the homomorphic properties of the encryption scheme but later require a decryption for the final decision as in [17], [6], [7]. However, in our design, the leakage of the secret key of any entity does not affect the security of the system. Besides, we do not have to employ a secure sketch or error correcting procedure to obtain the exact template that was stored in the biometric database of the service provider, thus, we can avoid the performance degradation caused by the error correcting codes [15], for instance correction of a 2048 bits iris template. Since no template is stored in the server end in our system, there is no need for a detached database and to employ the computationally expensive Private Information Retrieval (PIR) system to retrieve any template from the database privately as in [6], [7]. Instead, we propose a simple client server architecture for biometric verification that could be implemented also for large scale systems such as border control applications.

Another difference to the previous systems is that our system is a hybrid system that combines server-side matching and client side-matching, where the matching score of the both sides cannot be obtained by an attacker due to the use of a range proof that does not reveal the matching score but proves that the score lies in a range based on a threshold. This way, attacks depending on the matching score (for instance hill climbing attacks, Trojan horse attacks) are avoided. Finally, revocation of the biometric templates can be easily performed by changing the transformation parameters and/or picking a different public key for the user to encrypt the stable parts.

II. Preliminaries and Definitions

Homomorphic Encryption: For a given cryptosystem with (**Keygen**,**Enc**,**Dec**) and the message and ciphertext spaces M, C that are groups $\mathbf{Dec}(\mathbf{Enc}(a) \star \mathbf{Enc}(b)) = a * b$, where $a, b \in M$, and $*, \star$ represent the group operations of M, C respectively. The homomorphic encryption schemes that we employ for our setting is ElGamal Encryption [19] and RSA encryption scheme [20], which are both multiplicatively homomorphic, namely $*, \star$ are multiplication operations in the corresponding groups.

Bilinear Pairing: A bilinear pairing is denoted by $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{F}$, where $\mathbb{G}, \mathbb{F}$ are system parameters of the elliptic curve ElGamal encryption with g the generator of the group $\mathbb{G}$. The two properties of the bilinear pairing $\hat{e}$ are: (1) $\forall\ (g_1, g_2) \in \mathbb{G} \times \mathbb{G}$ and $\forall\ (a, b) \in \mathbb{Z}$, $\hat{e}(g_1^a, g_2^b) = \hat{e}(g_1, g_2)^{ab}$ (2) pairing $\hat{e}$ is non-degenerate.

Zero Knowledge Proof (ZPK): A proof of knowledge is an interactive proof in which the prover succeeds 'convincing' a verifier that it knows something. Specifically, ZPK allow a user to have a private data, and prove its possession without releasing it. For our setting, the prover is the user trying to authenticate and the verifier is the server.

III. A New Design for Biometric Remote Authentication

In this section, we describe how to combine the ingredients defined in the previous section to obtain a provably secure and efficient MFBA protocol. We choose elliptic curve ElGamal encryption combined with a zero knowledge proof (ZPK) of plaintext (i.e. biometrics) as the cryptographic method, where ZPK proves to the server that the user knows the biometrics. In [19], Schnorr proofs of knowledge (which is based on the Schnorr signature) is combined with ElGamal encryption to obtain a non-malleable encryption scheme. For unordered biometric features, we propose the use of RSA and the associated ZPK proofs.

A. Biometric Template Generation

Basically, the biometrics of a user is represented as a set of features, where each feature can be mapped to a finite field element asi nthe fuzzy vault [2] or error corrected using a secure sketch [21]. However, the secure sketches that takes the place of classical matching algorithm can degrade the performance compared to traditional matching algorithms due to the decoding operation or even for some modalities such as fingerprints, practical secure sketches do not exist [15]. In our design, we represent the biometrics of each user as a set of features, where k denotes the size of this set and depending on the biometric modality chosen, the features could be ordered. Besides, we do not have any assumption on the secrecy of the biometrics, whereas the biometric template that is stored should be private but easily revocable.

In addition to the classical representation of biometrics, one can also implement the biometric template extraction method of bipartite biotokens [3], where each feature is transformed using a scaling and a translation in order to separate the stable v and non stable r part of each feature. This way, the stable part v can be encrypted using standard public key encryption schemes as one bit of change in the plaintext (i.e. nonstable biometrics) will result in a completely different ciphertext when encrypted with standard cryptosystems. Next, the encrypted stable parts are stored at the service provider and the non stable parts are stored as a plaintext in the smart card of the user which is tamper proof. The main difference of this approach is that there is no need to employ a secure sketch or error correcting procedure to obtain the exact template that was stored in the biometric database of the service provider. Moreover, this method of template generation applies to any type of biometrics that can be processed as stable and non stable parts. The authors of [3], [4] implements this approach for fingerprints.

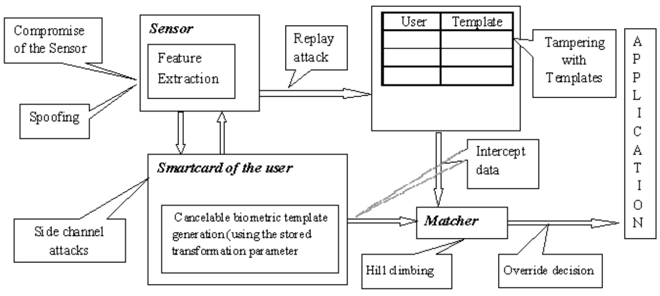

Fig. 1. Security and Privacy Issues of Multi-factor Biometric Systems

As different from the systems of [3], [4], we store the non-stable (residual) part, the transformation (i.e. scaling and translation) parameters and the window parameter E in the user's tamper proof smart card.

B. The Security Model

We propose a MFBA scheme that consists of three components, which communicate via an encrypted channel.

Sensor Client SC: This is the entity that obtains the fresh biometrics of the user during verification. As required for any biometric system, the liveliness assumption should be satisfied as it guarantees with high probability that the biometrics is coming from a live human user. The sensor client is always honest as in any biometric system and it is trusted by everyone.

Service Provider SP: This entity stores the identity information (name, personalized usernames...) for each user and the encrypted stable parts of each user's biometric template. Since no biometric template of a user either as a plaintext or in encrypted form is stored at the *SP*, there is no need for a detached biometric database.

User U with a smart card: Each user possesses a tamper proof smart card that stores the non stable parts of his biometrics and the parameters of the biometric template extraction method. We emphasize that no complete template is stored as in other match on card systems.

1) Adversarial Capabilities and Goals: In order to define the adversaries capabilities and goals, one has to determine the security and privacy issues for MFBA systems, which is summarized in figure 1. In our security model, the goal of the adversary is to impersonate a user. We model the adversary's power by allowing him to interact with protocol instances through several oracles as below.

Reveal: This query models the leakage of information about the authentication requests, where an eavesdropper listening to the communication channel can obtain the encrypted stable parts and the associated ZPK's if the session key is compromised. Namely, it models the leakage of information about the session key agreed on by the client and the server as in the case of a misuse of it afterward. Moreover, the authentication data of a user can also be leaked from the server due to an insider attack.

Corrupt: This query models corruption capabilities of the adversary. She can indeed steal/break either one of the authentication factors of the clients. In particular, the oracle can output the biometrics of the user. It models the attack against the sensor client or the user by compromising the sensor/biometrics of the user. Or the oracle can output the parameters (or some part of the stored data) that are stored in the tamperproof smart card of the user. It models the side channel attack against the smart card of the user. Clearly, the adversary is restricted to query the corrupt oracle at most for one authentication factor. Besides, no corruption can be performed during an authentication session, but before a new session starts. For even higher security, a user password can be added as a third factor.

2) User Privacy: We define the security notion for MFBA as User Privacy. To formally model this notion, we describe a security game between a challenger that simulates the server and an adversary that tries to impersonate a user. The adversary can ask several queries, but to the server only: We only consider adversaries whose goal is to impersonate a client to the server. Briefly, user privacy means that the adversary cannot impersonate a user to the server and thus cannot access user-specific applications. The formal definition of user privacy is as follows:

Given an adversary A running against the biometric authentication scheme and a challenger that simulates the registration phase of the scheme, we consider the following game between A and the simulator S. At the end of the game, A makes an authentication request. If successfully authenticated, she wins, otherwise, she looses

```
Experiment $Exp_A$
For $1 \leq j \leq N-1, (ID_j, c_j, ZKP_j) \leftarrow Registration$
$(i \neq j, ID_i) \leftarrow A$
$(ID_i, c^*, ZKP^*) \leftarrow Registration$
$c', ZKP' \leftarrow A^O(Verification)$
If $c' \approx c^*$ and $ZKP'$ is verified, return 1, else return 0
```

A biometric authentication scheme satisfies the notion of User Privacy if $Succ_A = Pr[b=1]$ is negligible.

Here, the simulator S simulates the registration phase by registering the encrypted authentication data and the corresponding zero knowledge proofs ZKP_j for $N-1$ users. Next the adversary A declares a user U_i with identity ID_i to attack the user privacy of the scheme. The simulator S registers the challenge $c^* =< w_1, ..., w_k >$ and the corresponding zero knowledge proofs $ZKP^* =< ZKP(w_1), ..., ZKP(w_k)) >$ that S got from its challenger C, where the challenge is the encryption of the stable parts of U_i's biometrics. Having access to reveal and corrupt oracles, the adversary A tries to impersonate the user U_i. If A succeeds, the S returns the challenger the answer of A, thus the simulator is able to break the non-malleable ElGamal encryption scheme [19] using A.

C. The concrete scheme

In this section, we present our MFBA scheme, which consists of three components: The user U with a smart card, a sensor client SC and a service provider SP. An overview of the verification phase is presented in figure 3. For the first construction that is based on the elliptic curve ElGamal encryption scheme and a non-malleable ZPK [19] based on the Schnorr signature, we assume that biometrics is represented as an ordered set of features such as face [21].

Setup Phase: The parameters of the elliptic curve ElGamal encryption scheme are initialized with a bilinear pairing $\hat{e}: \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{F}$ and a hash function $H: \{0,1\}^* \rightarrow \mathbb{Z}_q^*$. Each user U_i posesses an ElGamal public key pk_{U_i} that is used to encrypt the stable parts of the biometrics. Also, SP and SC generate their key pairs to build a secure communication channel between the entities.

Enrollment Phase: U_i registers to the system as follows:

(1)-SC extracts U_i's raw biometrics b and the raw data is transformed via a translation and scaling. Next, each transformed biometrics υ_j for $1 \leq j \leq k$, is separated to fraction (residual parts) r_j's and integer (stable) part ν_j's using a reflected modulus $rmod$ that does not increase the distance between points [3], [4]. The stable part ν_j's are hashed using H and the resulting values $\mu_j = H(\nu_j)$ are encrypted using the public key pk_{U_i} of the user U_i to obtain $w_j =$**Enc**$((g^{\mu_j})) = (w_j^1, w_j^2)$, whereas the residual r_j's are stored in the smart card in clear for each $1 \leq j \leq k$. To enforce the secrecy of the non-encrypted r_j's, we use an approach similar to the Match On Card (MOC) system [15], where fresh biometrics are acquired by the sensor client but the matching of the residual parts are made inside the card. This way, the confidentiality of the r_j's relies on inherent protections of smart cards against physical threats, where r_j's do not go out of the card.

(2)-U_i registers his ID_i and the encrypted stable parts together with the ZPK's at the SP. U_i stores the residual parts (i.e. r_j's) and the parameters (i.e. transformation parameters, reflected modulus $rmod$, windowing parameter E) in his smart card. U_i does not store the secret key of his public key pk_{U_i} as it will not be used at all.

Verification Phase: U_i authenticates to SP as follows:

(1)-The sensor client SC extracts U_i's fresh b' and communicates with the smart card of U_i to send b'.

(2)-The user's smart card seperates the stable and residual parts of each feature using the parameters stored in his card and encrypts the stable parts of each feature using the ElGamal public key pk_{U_i} of U_i. The residual parts are matched to the fresh residual parts on card and the encrypted stable parts $w'_j = (w'^1_j, w'^2_j)$ are sent to SC.

(3)-SC signs the encrypted stable parts w'_j and transmits them together with his signature σ to SP.

(4)-SP verifies σ and compares the fresh encrypted stable parts w'_j's of U_i to the previously stored data w_j's by using the homomorphic property of ElGamal encryption scheme. For $1 \leq j \leq k$, SP selects $s_j \overset{R}{\leftarrow} \mathbb{Z}_p^*$ to compute

$$R_j = \left(R_j^1, R_j^2\right) = \left(\left(\frac{w_j^1}{w_j'^1}\right)^{s_j}, \left(\frac{w_j^2}{w_j'^2}\right)^{s_j}\right)$$

(5)-SP checks for $1 \leq j \leq k$ whether $\hat{e}(g^U, R_j^1) = \hat{e}(g, R_j^2)$ by computing $2k$ bilinear pairings.

(6)-Finally, SP counts the number of the equations satisfying the above condition and computes the matching score ms, which is compared to the matching score mns of the non-stable parts stored on card. The comparison is made by using an efficient range proof [22], which does not reveal the mns even to the server but proves that $mns \approx ms$. If the user can prove to the server that the mns lies within the range determined by the predefined threshold of the system, SP decides to authenticate U_i.

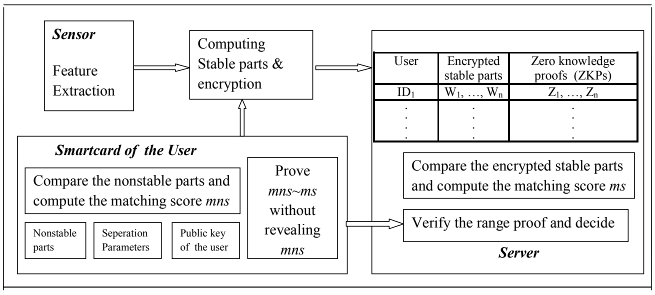
Sensor
Feature Extraction
Computing Stable parts & encryption
User
Encrypted stable parts
Zero knowledge proofs (ZKPs)
ID1
W1, …, Wn
Z1, …, Zn
Smartcard of the User
Compare the nonstable parts and compute the matching score mns
Nonstable parts
Seperation Parameters
Public key of the user
Prove mns~ms without revealing mns
Compare the encrypted stable parts and compute the matching score ms
Verify the range proof and decide
Server

Fig. 2. Verification Phase of the new Protocol

Theorem 3.1: Assume that an attacker running against our protocol breaks the user privacy by making at most two queries to the **Corrupt**, q_H queries to the oracle H, q_R queries to **Reveal**, then, we can break the one-wayness of the ElGamal scheme.

Due to the page limitations the proof will be presented in the full version of the paper.

D. Biometrics as an Unordered Set

Although some biometric modalities can be represented as an ordered set of features such as face biometric, for fingerprints this is not a trivial task [3]. Fuzzy vault based systems try to find a solution for biometrics that consists of an unordered set of features, however, however, there exists many attacks against these systems that reveal both the secret used for authentication and the biometric template that hides this secret. Since ordering or grouping of features are not possible for some biometric modalities, we cannot use a probabilistic encryption scheme such as ElGamal encryption system since the matching of the non stable part on card will not be consistent with the matching of the encrypted stable parts at the remote server. (There could be accidental matches on card that results in different matching scores). However, if we use a deterministic scheme like RSA, the remote server can send the indices of the fresh encrypted parts that exactly match the stored encrypted stable parts and thus, the match on card system performs the matching according to the instruction of the remote server, which will result in similar matching scores at the both entities. We note that the stored stable features at the SP and the non-stable parts stored at the smart card share the same order at the enrollment phase, i.e. if a specific feature is stored as the second feature in the server, than the unstable part of this feature is also stored at the 2. place on card. If the number of matched stable features is above the threshold, SP sends the signed order information of the matched stable parts to the SC. For instance, if the first stable part in the fresh query matched the third stable part stored in the gallery, than SP sends [1 → 3] to the SC. In order to leak no information about the actual matching score, SP sends to the client random order information for the non-matching parts. We note that computing the indices of the matching stable parts is also possible when ElGamal encryption system is used, however, the remote server has to compute in worst case $O(k^2)$ bilinear pairings and modular divisions, where the computation of one bilinear pairing is approximately 9 modular exponentiations. (k is the size of the feature set.) Thus, the use of ElGamal is impractical compared to a deterministic encryption scheme for unordered biometric features. Finally, replay attacks should be considered when a deterministic scheme is used as encryption of the same message results in the same ciphertext, whereas the encryption of the same message results in a different ciphertext due to the random coins used in the probabilistic encryption scheme. Thus, the communication channel should be encrypted using a session key and ZPK's designed for RSA [20] should be attached to the ciphertext with a time stamp as before.

IV. Discussion

In table 1, we analyse the success of the attacker against our system in case of 4 classes of attacks.

TABLE I

ATTACKS AGAINST MULTI-FACTOR BIOMETRIC SYSTEMS

	Compromise of User biometrics	Impersonation
(1) Server Compromise	⊖	⊖
(2) Side channel attack	⊖	⊖
(3) Session key compromise	⊖	⊖
(4) Sensor Compromise	Unavoidable	⊖if no (2)

In our security model, we only assume that the attacker cannot compromise both the sensor and the smart card of the user, otherwise, the attacker with the true biometrics of the user and the transformation parameters stored at the card can impersonate a user trivially. We note that this assumption may be relaxed if the user has its own biometric smart card reader, then we can obtain higher security against sensor compromise. Also, the server stores the encrypted stable parts together with the ZPK's, thus, tampering with the stable parts is not possible since the ZPK's are non-malleable i.e. cannot be modified to work with the new stable parts. Another advantage of the new system is that revoking of the templates is possible since the user can choose a different public key in the encryption of the stable parts and use different transformation parameters in the separation of the stable/nonstable parts. This also prevents linkability of the stored templates of the same user at different servers. We emphasize that the smart card of the user does not output a matching score, but a range proof on this score, which does not leak any information about the score and proves the server that the score lies within a range that the server accepts. Thus, no information useful for a hill climbing attack can be obtained due to the tamper-proofness of the smart card. Alternatively, SP can also store the non-stable parts and can perform the matching of the two parts himself. This way, there is no need for a range proof and a matching score, instead the server outputs an accept/reject decision. Thus, hill climbing attacks are not applicable and the server cannot compute the true biometrics of the user due to the encrypted stable parts and the secrecy of the the transformation parameters that are stored in the tamperproof smart card of the user. To prevent a malicious SP from learning the biometrics of the user through a pre-computed dictionary attack, the stable parts of the small features can be padded with a random string, which will be stored in the tamperproof smartcard to be used later in the verification. This way, a malicious insider cannot guess the real biometrics of the user due to the secrecy of the randomness stored in the smartcard. Besides, the systems summarized in [7] require the use of secure sketches and store biometrics in the detached database as encrypted using the public key of the service provider SP. Thus, the collusion of the server end components results in the violation of the user privacy. If the secret key of SP is leaked, than every user has to re-register to the system before the compromise of the database. Therefore, our new design stores each biometrics as encrypted with the user's public key.

V. Conclusion

In this paper, we present the security model for MFBA and describe two schemes for ordered/unordered set of biometric features that combine a different extraction method, zero knowledge proofs and homomorphic encryption schemes. The security notion for MFBA is defined as user privacy, which is achieved for our protocols even in the case of simultaneous attacks against the system.

References

[1] A. Juels and M. Wattenberg, "A fuzzy commitment scheme", in *ACM CCS'99*, 1999, pp. 28–36.

[2] A. Juels and M. Sudan, "A fuzzy vault scheme", *Des. Codes Cryptography*, vol. 38, no. 2, pp. 237–257, 2006.

[3] T. E. Boult, W. J. Scheirer, and R. Woodworth, "Revocable fingerprint biotokens: Accuracy and security analysis", in *CVPR'07*. 2007, IEEE.

[4] W. J. Scheirer and T. E. Boult, "Bipartite biotokens: Definition, implementation, and analysis", in *ICB'09*. 2009, vol. 5558 of *LNCS*, pp. 775–785, Springer.

[5] N. K. Ratha, J. H. Connell, and R. M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems", *IBM Systems Journal*, vol. 40, no. 3, pp. 614–634, 2001.

[6] J. Bringer and H. Chabanne, "An authentication protocol with encrypted biometric data", in *AFRICACRYPT'08*. 2008, vol. 5023 of *LNCS*, pp. 109–124, Springer.

[7] N. D. Sarier, "A survey of distributed biometric authentication systems", in *BIOSIG'09*. 2009, vol. 155 of *LNI*, pp. 43–55, GI.

[8] Ahmad-Reza Sadeghi, Thomas Schneider, and Immo Wehrenberg, "Efficient privacy-preserving face recognition", in *ICISC*. 2010, vol. 5984 of *LNCS*, pp. 229–244, Springer.

[9] B. Schoenmakers and P. Tuyls, "Computationally secure authentication with noisy data", in *Security with Noisy Data: Private Biometrics, Secure Key Storage and Anti-Counterfeiting*. 2007, pp. 141–149, Springer.

[10] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft, "Privacy-preserving face recognition", in *PETS'09*. 2009, vol. 5672 of *LNCS*, pp. 235–253, Springer.

[11] M. Upmanyu, A. M. Namboodiri, K. Srinathan, and C.V. Jawahar, "Efficient biometric verification in encrypted domain", in *ICB'09*. 2009, vol. 5558 of *LNCS*, pp. 899–908, Springer.

[12] A. Bhargav-Spantzel, A. C. Squicciarini, and E. Bertino, "Privacy preserving multi-factor authentication with biometrics", in *Digital Identity Management*. 2006, pp. 63–72, ACM.

[13] K. M. Apampa, T. Zhang, G. B. Wills, and D. Argles, "Ensuring privacy of biometric factors in multi-factor authentication systems", in *SECRYPT'08*. 2008, pp. 44–49, INSTICC Press.

[14] N. D. Sarier, "A new approach for biometric template storage and remote authentication", in *ICB'09*. 2009, vol. 5558 of *LNCS*, pp. 909–918, Springer.

[15] J. Bringer, H. Chabanne, D. Pointcheval, and S. Zimmer, "An Application of the Boneh and Shacham Group Signature Scheme to Biometric Authentication", in *IWSEC'08*. 2008, vol. 5312 of *LNCS*, pp. 219–230, Springer.

[16] Y. Itakura and S. Tsujii, "Proposal on a multifactor biometric authentication method based on cryptosystem keys containing biometric signatures", *Int. J. Inf. Sec.*, vol. 4, no. 4, pp. 288–296, 2005.

[17] S. Hirata and K. Takahashi, "Cancelable biometrics with perfect secrecy for correlation-based matching", in *ICB'09*. 2009, vol. 5558 of *LNCS*, pp. 868–878, Springer.

[18] T. Sakashita, Y. Shibata, T. Yamamoto, K. Takahashi, W. Ogata, H. Kikuchi, and M. Nishigaki, "A proposal of efficient remote biometric authentication protocol", in *IWSEC'08*. 2009, vol. 5824 of *LNCS*, pp. 212–227, Springer.

[19] Y. Tsiounis and M. Yung, "On the security of elgamal based encryption", in *PKC'98*. 1998, vol. 1431 of *LNCS*, pp. 117–134, Springer.

[20] R. Rivest, "Lecture notes 9: Homomorphic encryption", `web.mit.edu/6.857/OldStuff/Fall01/handouts/L09-homomorphic.ps`.

[21] Y. Sutcu, Q. Li, and N. Memon, "Secure sketch for biometric templates", in *ASIACRYPT'06*. 2006, vol. 4284 of *LNCS*, pp. 99–113, Springer.

[22] K. Peng and F. Bao, "Batch range proof for practical small ranges", in *AFRICACRYPT'10*. 2010, vol. 6055 of *LNCS*, pp. 114–130, Springer.