An Efficient Generic Framework for Three-Factor Authentication With Provably Secure Instantiation

Jiangshan Yu, Guilin Wang, Yi Mu, *Senior Member, IEEE*, and Wei Gao

***Abstract*— Remote authentication has been widely studied and adapted in distributed systems. The security of remote authentication mechanisms mostly relies on one of or the combination of three factors: 1) something users know—password; 2) something users have—smart card; and 3) something users are—biometric characteristics. This paper introduces an efficient generic framework for three-factor authentication. The proposed generic framework enhances the security of existing two-factor authentication schemes by upgrading them to three-factor authentication schemes, without exposing user privacy. In addition, we present a case study by upgrading a secure two-factor authentication scheme to a secure three-factor authentication scheme. Furthermore, implementation analysis, formal proof, and privacy discussion are provided to show that the derived scheme is practical, secure, and privacy preserving.**

***Index Terms*— Authentication, security, privacy, password, smart card, biometrics.**

I. Introduction

THE need of user authentication is a fundamental security requirement in computer society. With wide-spread of distributed computer networks, remote user authentication has been introduced to identify a user remotely, and has been widely studied (see [1]–[3]). In general, authentication services may require three factors, i.e., password, smart card and biometric characteristics. The authentication based on a password is called password-based authentication (e.g. Facebook login system). A system which authenticates users by using password and smart card is called two-factor authentication (e.g. HSBC Internet banking login system). In which, a client can pass authentication only if the client has correct password and the corresponding authentic smart card. The biometric-based authentication mainly employs the biometric characteristics, e.g. fingerprint, palm print, and iris.

Manuscript received February 13, 2014; revised April 28, 2014 and August 8, 2014; accepted October 8, 2014. Date of publication October 14, 2014; date of current version November 12, 2014. This work was supported in part by The University of Birmingham under EPSRC Grant EP/H005501/1. The work of W. Gao was supported in part by the National Natural Science Fundation of China under Grant 61202475. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jianying Zhou. *(Corresponding author: Jiangshan Yu.)*

J. Yu is with the Department of Computer Science, University of Birmingham, Birmingham B15 2TT, U.K. (e-mail: jxy223@cs.bham.ac.uk).

G. Wang is with Huawei International Pte Ltd., Singapore 486035 (e-mail: wang.guilin@huawei.com).

Y. Mu is with the Centre for Computer and Information Security Research, School of Computer Science and Software Engineering, University of Wollongong, Wollongong, NSW 2522, Australia (e-mail: ymu@uow.edu.au).

W. Gao is with the Department of Mathematics and Informatics, Ludong University, Yantai 264025, China (e-mail: sdgaowei@gmail.com).

Digital Object Identifier 10.1109/TIFS.2014.2362979

The earliest user authentication mechanism through the Internet is based on password. The concept of password based authentication was first proposed by Lamport in 1981 [4]. Such authentication systems remain the most common mechanism for internet applications (e.g. email services, conference management systems, and social networks). However, the security of such systems is not always reliable. For example, poor password selections, the password capture Trojans, and the reuse of passwords could break the security. One popular attack is called dictionary attack, which targets to find the correct password by trying a large amount of likely possibilities, such as words in a dictionary or the likely combination of words. This attack is usually efficient since most users prefer to choose human memorisable passwords, e.g. the user's name, address, or mobile number. A good remedy is additionally using hardware authentication tokens (usually smart cards) to authenticate clients. Such a remedy is called two factor authentication, which has become popular and has been used by applications with higher security guarantees, e.g. internet banking services. In 1991, Chang and Wu [5] introduced this idea of using password and smart card to authenticate clients. Afterwards, many two-factor authentication schemes have been proposed. However, the security of two factor authentication could be compromised since the smart card may be stolen and the data stored in the smart card can be duplicated, and the range of possible passwords could be small and users may forget or lose their passwords. Due to such concerns, biometric identification was introduced to authenticate users by using their biometric features.

In 1999, Juels and Watenberg [6] proposed a biometric authentication scheme, called "fuzzy commitment," that improves some aspects of two-factor authentication because biometric characteristics have higher entropy, and they cannot be forgotten and are rarely to be lost [7]. However, one problem is that biometric characteristics are not completely private since one can "steal" biometric characteristics from others; e.g., the fingerprint can be obtained from a mug that the victim has used, and the facial features may be obtained from a user's photograph. A way to alleviate these problems is to combine all these three factors together. This approach is also known as three-factor authentication, which has been greatly adapted by cloud-based applications (see [8]).

A. Related Work

The introduction of password-based authentication by Lamport in 1981 [4] has inspired numerous password based authentication protocols [9]–[13]. In 1999, Yang and Shieh [9] proposed two two-factor authentication schemes: one is based

on timestamp and the other is based on random nonce. Both of them support contact-less password changing, i.e., users do not need to contact/inform a server to change their password. A system satisfying such requirement can save the computation cost on the server side, and save the communication cost on both server and user side. Later, Chan and Cheng [14], and Fan *et al.* [10] identified impersonation attacks on the Yang-Schieh scheme. To overcome this flaw, Shen, Lin and Hwang [11], and Yang, Wang and Chang [12] suggested improvements on the Yang-Schieh scheme. However, Yoon *et al.* [13] showed possible attacks on the YWC-scheme [12], and introduced an improvement. In 2006, however, Wang and Bao [15] pointed out that both the SLH-scheme [11] and Yoon *et al.*'s scheme [13] are vulnerable to impersonation attack.

On the other direction, in 2003, Kim *et al.* [16] proposed two constructions of three-factor authentication schemes by using password, smart card, and fingerprints, without requiring public key directory tables. However, Scott [17] pointed out that a passive eavesdropper, without accessing to any smart card, password, or fingerprint, could impersonate any identity to pass authentication after successfully eavesdropping only once legitimate log-in.

In 2004, Uludag *et al.* [18] surveyed various types of biometric authentication systems, and recommended to use digital rights management (DRM) systems [19] to address the problem of biometric authentication systems. In their method, the cryptographic key is bound with a biometric template and stored in a database. Thus, the key cannot be revealed without passing biometric authentication. However, the requirement of the biometric database has increased the cost and put users' privacy at risk. To protect users' privacy, in 2006 Bhargav-Spantze *et al.* [20], [21] proposed a novel privacy preserving two-phase three-factor authentication scheme, based on zero knowledge proof (ZKP), in which user privacy is preserved by using the Pedersens commitments [22]. However, the scheme is expensive because of modular exponentiation operations, and the requirement that all users' commitments are stored on the server side. In 2009, Fan and Lin [23] constructed an efficiency enhancing and privacy preserving three-factor authentication scheme, but it does not support contact-less password changing. There are also many other research [24], [25] have been done on preserving user privacy in distributed systems.

Recently, Li and Hwang [26] proposed an efficient three factor user authentication scheme, without requiring synchronized clocks. Later, Li *et al.* [27] pointed out that the Li-Huang scheme does not meet proper authentication since it is vulnerable to the man-in-the-middle attack. To address this shortcoming, they provided a further improvement. In 2011, however, Das *et al.* [28] found that Li *et al.*'s improved scheme neither provided strong authentication nor supported contact-less password changing. They then proposed an improvement on Li *et al.*'s scheme. However, the improved scheme is still insecure as an adversary who obtained a victim's smart card can launch off-line password guessing attack.

To tackle the problem caused by insecure proposals and improvements, Huang *et al.* [29] proposed a generic framework to upgrade two factor authentication schemes to three-factor authentication schemes, while preserving security and privacy. The basic idea is to use a fuzzy extractor to generate the biometric key from the biometric characteristics, and run twice the underlying two-factor authentication scheme. The first run is the normal underlying two-factor scheme using passwords and smart cards. In the second run of the underlying scheme, the password is replaced with the generated biometric key. This framework does not require any change on the underlying two-factor authentication protocol, and in the derived scheme users do not need to hand their biometric characteristics over to the server, so that servers do not need to store any data related to user's biometric characteristics. Thus, user privacy is preserved and the cost on the server side is reduced.

B. Motivation

Huang *et al.* [29] offer a good framework to produce three-factor authentication schemes from existing two factor authentication schemes. This framework eases the design of three factor authentication systems, provides higher security guarantee, and preserves user privacy. To generate biometric keys from the biometric characteristics, Huang *et al.*'s framework employs the "fuzzy extractor" [30]. Fuzzy extractor generates a pair of strings (P, R) from user biometric characteristics, where P is the auxiliary string and R should be kept secret as private key. The private R can be recovered if a user can provide the corresponding auxiliary string P and a close enough biometric characteristics. The error tolerance in the scheme depends on three error correcting techniques, namely Hamming distance, set difference, and edit distance. The fuzzy extractor provides a good insight into biometric identification by extracting a unique and random 'private' key directly from the user's biometric features. However, the fuzzy extractor has not been widely implemented since the distance measures in it are less accepted than the Euclidean distance measurement in biometric applications [31].

Moreover, we observe that the efficiency of Huang *et al.*'s framework can be improved from running underlying scheme twice to running it once – which saves almost half of the cost in total. Moreover, the study on the concrete three-factor authentication scheme with formally security analysis, which is recognised as an open problem and a challenging issue [32], are missing in their work.

C. Contributions

The main contributions of this paper are the improved generic framework for three-factor authentication and a provably secure instantiation. The merits of this paper are as follows.

First, the proposed generic framework enhances efficiency by combining the user's password and the user's biometric key together and using the hash value of this combination as the user's secret key. Consequently, the resulted three-factor scheme only needs to run the underlying two-factor scheme one time. This saves almost half of the communication cost

and computation cost for each login among potential billions of users.

Second, the proposed generic framework is more practical. We employ the improved finger print-based "fuzzy vault" [33] to identify the user's biometric features. Literature shows that the fuzzy extractor has not been implemented yet, while researchers implemented and improved the fuzzy vault scheme in recent years [31], [33]–[36]. Moreover, the fuzzy vault has been widely accepted because the Euclidean distance measurement which is used in the fuzzy vault are widely accepted by majority of biometric applications [31]. Therefore, the improved framework selects the fuzzy vault to employ the third factor, biometric features.

Last, a provably secure instantiation is presented. In particular, this paper discusses the practicability analysis of the concrete scheme, compares our concrete scheme with other existing three-factor schemes, provides privacy discussion, and shows formal security proof on the concrete scheme.

D. Organization

The rest of this paper is organised as follows. Section II reviews and discusses two well-known biometric identification mechanisms. Section III reviews Huang *et al.*'s framework and provides an improved generic framework for three-factor authentication. The instantiation with analysis and comparison are given in Section IV. In section V, formal security proof and privacy discussion for this instantiation are provided.

II. Biometric Identification Mechanisms

In 1999, Juels and Wattenberg [6] proposed "fuzzy commitment", the first biometric identification scheme, which deploys Hamming distance to tolerate errors. Later, Juels and Sudan [37] introduced a provably secure biometric identification scheme, called fuzzy vault, in which a user can generate a long-bit secret key, and encrypt it by using his/her extracted biometric template. The long-bit secret key can be recovered by providing the encrypted data and the corresponding authentic biometric characteristics. In 2003, Clancy *et al.* [34] proposed a secure smart card based fingerprint authentication scheme by using Juels and Sudan's fuzzy vault. Later, in 2007, Nandakumar *et al.* [35] proposed a fully automatic implementation by employing the fuzzy vault and using helper data to align unidentified fingerprints accurately. Their scheme used both location (x, y) and orientation attribute θ of a minutia point to record the biometric data, where (x, y) is the row and column indicators in the image as the location, and θ is the orientation on the X-axis. The helper data is high curvature points extracted from the fingerprint orientation field, thus it neither affects the security nor leaks any information about the biometric template. One year later, Nagar, Nandakumar and Jain [33] improved the security and matching accuracy of Nandakumar *et al*'s fingerprint-based fuzzy vault scheme by employing additional minutiae descriptors [38], which capture local ridge orientation and ridge frequency information in the neighbourhood of a minutia. The results in [33] show that the improved scheme reduces the false acceptance rate (FAR) and significantly increases the vault security.

On the other direction, in 2004, Dodis *et al.* [30] proposed fuzzy extractor, which has two procedures: a generation procedure and a reproduction procedure. After a user scanned his biometric features and obtained the biometric template w, the generation procedure extracts a random R and a corresponding auxiliary P from w. In the authentication phase, the inputs of reproduction procedure are P and an unidentified biometric template w'; the output of this reproduction procedure is exactly the same R if and only if the difference between w and w' is within an acceptable error tolerance. In 2008, Teoh and Ong [39] proposed a randomised dynamic quantisation transformation (RDQT), which is based on fuzzy commitment, to binarize biometric data, and satisfy both randomness and uniqueness. Meanwhile, Sheng *et al.* [40] presented a template-free biometric-key generation, which can also generate a key directly from a biometric template.

A. Fuzzy Vault

Fuzzy vault is a cryptographic construction for data protection and user authentication, whose security relies on unexposed biometric characteristics and smart card. The error tolerance in fuzzy vault is achieved by using the Euclidean distance measurement which has been widely accepted by the majority of biometric applications. The operations of the fuzzy vault are described as follows.

First, a user extracts biometric template X by scanning her biometric characteristics (e.g. fingerprint). Then, she encodes a pre-self-generated secret string K into a self-selected polynomial Pol, and evaluates the polynomial on all elements in X. She also needs to choose a large number of random points which do not lie on Pol as the noise. The final vault V is the collection of the points which lie on Pol and the noise points which do not lie on Pol.

She can recover the secret string K from vault V by providing a biometric template X' such that the difference between X and X' satisfies $|X - X'| < \epsilon$, where $X - X' = \{x | x \in X, x \notin X'\}$, and ϵ is an integer which is the fuzziness parameter. This is because that the polynomial Pol can be reconstructed if a sufficient number of points on Pol can be identified. Thus, K can be successfully recovered from Pol. The detail operation is defined as follows:

1) Lock:

1) $\xrightarrow[K,Pol]{X}$ $\boxed{Gen(\cdot)}$ $\rightarrow L$: Taking input a user's biometric template X, secret K and polynomial Pol, $Gen(\cdot)$ outputs a set L of points which lie on the Pol.
2) $\xrightarrow[L]{CP}$ $\boxed{Enc(\cdot)}$ $\rightarrow V$: Taking input L and a set CP of "chaff points" (i.e. random noise points) which do not lie on Pol, $Enc(\cdot)$ outputs a vault V such that $V = CP \cup L$. CP is generated on the user side, and if we denote r the number of points in L, and s the number of points in CP, then we require $s \gg r$.

2) Unlock:

1) $\xrightarrow[V]{X'}$ $\boxed{Dec(\cdot)}$ $\rightarrow$ Pol: Taking input V and biometric template X', $Dec(\cdot)$ outputs Pol if and only

if $|X - X'| < \epsilon$, where $X - X' = \{x | x \in X, x \notin X'\}$ and ϵ is the fuzziness parameter.

2) $\xrightarrow{Pol}$ $\boxed{Rec(\cdot)}$ $\rightarrow K$: Taking input Pol, $Rec(\cdot)$ outputs the secret key K.

Remark 1: The security of the fuzzy vault is based on the difficulty of distinguishing genuine points from chaff points in vault V, and the difficulty to reconstruct the polynomial Pol in vault V. So, the security guarantee is in proportion to the number of added chaff points.

III. A Generic Three-Factor Authentication Framework

A. Review of Huang et al.'s Framework

Huang *et al.*'s framework employs the fuzzy extractor to generate a uniquely long-bit random string as the biometric key for users. By running the underlying two-factor scheme twice, a three-factor scheme is constructed. In particular, the first run uses password and smart card as normal two factor authentication system. In the second run, framework replaces the password by a biometric key and runs the underlying protocol again, thus a three-factor authentication is obtained. Huang *et al.*'s framework consists of three phases:

1) Registration: The processes of registration includes the following steps:

1) User U_i chooses initial password PW_1 and extracts biometric template X by scanning her biometric features;
2) U_i generates a pair (R, P) by providing X to the fuzzy extractor;
3) Let the second password PW_2 be $h(R)$, where $h(\cdot)$ is a cryptographic hash function;
4) $U_i\,[PW_1] \xLeftrightarrow{2-Factor-Reg} S\,[SK_1] \rightarrow Data_1$;
U_i runs the underlying two-factor registration protocol (2-Factor-Reg) with initial password PW_1, and server S uses secret key SK_1 to generate $Data_1$;
5) $U_i\,[PW_2] \xLeftrightarrow{2-Factor-Reg} S\,[SK_2] \rightarrow Data_2$.
U_i runs the registration protocol again with PW_2, and S issues $Data_2$ by using another secret SK_2;
6) U_i obtains a smart card SC which stores $Data_1$, $Data_2$, and $Data_3 = (P, h(\cdot), Rep(\cdot))$, where $h(\cdot)$ and $Rep(\cdot)$ are the corresponding hash function and the reproduction procedure, respectively.

The scheme supposes that PW_1, PW_2 will be deleted immediately from the server side upon completion of the corresponding registration steps. This means that in the registration phase, the server is fully trusted.

2) Authentication: User U_i' first inserts SC into a card reader, enters her password, and scans her biometric features. We use X' to denote the extracted biometric template. The authentication phase is as follows.

1) The smart card recovers R' through $Rep(\cdot)$, and calculates $PW_2' = h(R')$. $R' = R$ if and only if $|X - X'| < \epsilon$ for some fuzziness parameter ϵ;
2) $U_i'\left[PW_1', Data_1\right] \xLeftrightarrow{2-Factor-Auth} S\,[SK_1]$; U_i' with $(PW_1', Data_1)$ runs the authentication phase (2-Factor-Auth) of the underlying two-factor authentication protocol with server S;
3) $U_i'\left[PW_2', Data_2\right] \xLeftrightarrow{2-Factor-Auth} S\,[SK_2]$; U_i' with PW_2', $Data_2$ runs the 2-Factor-Auth with S.

The user successfully passes user authentication if and only if both step 2 and step 3 succeeded.

3) Password Changing: The password can be changed by running password changing protocol (2-Factor-Password-Changing) in the underlying two-factor scheme after successfully logging and updating the SC accordingly. The biometrics can be changed by running step 2 and step 3 in the registration phase, then the user and server execute 2-Factor-Password-Changing and update the corresponding data in SC.

B. Improved Framework

We assume that the server in the registration phase is trusted. The details are specified as follows:

1) Three-Factor-Registration: The processes of registration include the following steps:

1) User U_i chooses an initial password PW_1, a long-bit secret key PW_2.
2) The fuzzy vault device extracts biometric template X by scanning her biometric features.
3) Taking X, PW_2, and polynomial Pol as inputs, $Gen(\cdot)$ outputs a set L, and by taking the set CP of noise chaff points and L, the $Ence(\cdot)$ outputs the encrypted data V.
4) $U_i\,[PW] \xLeftrightarrow{2-factor-Reg} S\,[SK] \rightarrow Data_1$, where $PW = h(PW_1||PW_2)$ and $||$ is concatenation operation.
The user with PW and the server with SK run the registration phase of the underlying protocol.
5) Server stores $Data_1$ and $Data_2 = (V, Rec(\cdot), Dec(\cdot), h(\cdot))$ in smart card SC, and gives it to U_i.

2) Three-Factor-Authentication: To access services, user U_i' inserts SC to a card reader, which can extracts the data from the SC. Then, U_i' inputs PW_1' and scans her biometric features, the extracted biometric template is X'. The details are as follows:

1) The card reader extracts X' from U_i''s biometric features, and reproduces PW_2' such that $PW_2' = PW_2$ if and only if $|X - X'| < \epsilon$;
2) The smart card calculates $PW' = h(PW_1'||PW_2')$;
3) $U_i'\left[PW', Data_1\right] \xLeftrightarrow{2-factor-Auth} S\,[SK]$;
The user can successfully pass authentication if and only if this step is success.

3) Three-Factor-Password-Changing: The PW_1 can be changed by following steps.

1) After passing authentication, U_i' sends the password changing request, inputs new password PW_1'', and scans the biometric template.
2) The 'fuzzy vault' device will recover the PW_2 by using the 'fuzzy vault' decoding scheme.
3) The smart card calculates $PW'' = h(PW_1''||PW_2)$.
4) PW'' is taken as the password and runs the password changing phase of the underlying protocol.

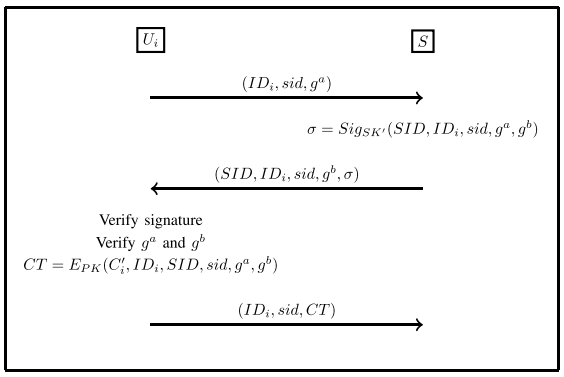


Fig. 1. Login phase of Yang *et al.*'s scheme.

Biometric key PW_2 can be changed in a similar way. For this purpose, U_i' chooses a new biometric key as PW_2'', then encrypts it via the fuzzy vault device, outputs V' which replaces current V in SC. The SC calculates $PW'' = h(PW_1||PW_2'')$, then takes PW'' as the password and runs the password changing phase of the underlying protocol.

IV. Instantiation

Our instantiation will use the Yang *et al.*'s two-factor authentication scheme [41], which is provably secure, as the underlying scheme.

Let G be a group of prime order q and g a generator, $H : \{0,1\}^* \rightarrow \{0,1\}^k$ denote a collision resistant hash function, $H' : \{0,1\}^* \rightarrow \{0,1\}^k$ a hash function which preserves the entropy of its input (e.g. add paddings after the input); $PRF_K : \{0,1\}^k \rightarrow \{0,1\}^k$ a pseudo-random function keyed by K. In addition, we assume that a server S has a long term secret x such that $x \in \{0,1\}^k$, and encryption and signature key pairs (PK, SK) and (PK', SK'), respectively.

Let $E_{PK}(M)$ denotes the asymmetric key encryption on message M under public key PK; $Sig_{SK'}(M)$ a signature on M issued by using signing key SK'.

A. Review of Yang et al.'s Scheme

In the registration phase, a user U_i chooses a unique identity ID_i and sends it to the server S. After receiving the request, S issues a credential $C_i = PRF_x(H(ID_i))$, and hides it by calculating $B = C_i \oplus H'(PW_0)$, where PW_0 is the initial password chosen by S; then S sends the initial password PW_0 and a smart card which contains $(PK, PK', ID_i, B, p, g, q)$ to U_i.

The login phase is presented in the Fig. 1. To log in, U_i attaches her smart card to a card reader device, and enters her password PW'. The smart card calculates $C_i' = B \oplus H'(PW')$ and sends (ID_i, sid, g^a) to S, where sid is the session identifier, and a is a new selected random number. S should send $(SID, ID_i, sid, g^b, Sig_{SK'}(SID, ID_i, sid, g^a, g^b))$ to U_i, where SID is the identity of S and the signature is used for server side authentication. If the signature is valid, then U_i believes that he is talking to the real server, and sends (ID_i, sid, CT) to S, where $CT = E_{PK}(C_i', ID_i, SID, sid, g^a, g^b))$. S accepts U_i as a genuine user if $C_i' = PRF_x(H'(ID_i))$. Now both parties believe that they have shared the same session key g^{ab}.

In addition, U_i can change her password at anytime after she receiving the smart card and initial password PW_0 from S. To change the password, she picks a new password PW_{new}, and performs $B_{new} = B \oplus H'(PW_0) \oplus H'(PW_{new})$, then replaces B with B_{new}.

B. Protocol

The basic idea of our concrete protocol is that using $PW = H'(PW_1||PW_2)$ as the password in Yang *et al*'s scheme, where PW_1 is the real password, and PW_2 is the biometric key encrypted through fuzzy vault scheme. A user can pass authentication only if s/he provides the correct password, smart card, and the biometric features which is close enough with the one used in the registration phase.

1) Registration: In the registration phase, a user U_i performs exactly the same as in Yang *et al.*'s scheme. However, after U_i receiving the smart card and the initial password PW_0, she needs to additionally selects a new password PW_1, a polynomial Pol and a biometric key PW_2. In addition, she extracts her biometric template X, encrypts PW_2 through fuzzy vault device which outputs a vault V. Then U_i writes V into the smart card, and calculates $PW = H'(PW_1||PW_2)$, and updates B by computing $B = C_i \oplus H'(PW_0) \oplus PW$. The 'fuzzy vault' procedures are reviewed in the Section II-A, thus we omit the detail here.

2) Login-and-Authentication Phase: User U_i' attaches her smart card to a card reader device, inputs password PW_1' and scans her biometric features. The fuzzy vault device extracts the biometric template X', then the fuzzy vault device calculates $Pol' = Dec(X', V)$, and $PW_2' = Rec(Pol')$. The smart card SC calculates $C_i' = B \oplus PW'$, where $PW' = H'(PW_1'||PW_2')$. Then, the protocol runs the login phase as the same as Yang *et al*'s scheme by using PW'.

3) Password-Changing: To change an old password PW_1, U_i performs the following steps.

1) Chooses a new password PW_1'.
2) Calculates $PW_{new} = H'(PW_1'||PW_2)$ and computes $B_{new} = B \oplus PW \oplus PW_{new}$, where $PW = H'(PW_1||PW_2)$.
3) Replace B with B_{new} in the smart card.

The biometric key PW_2 and the biometric features can be changed in a similar way, in which case, the vault V in the smart card should also be updated.

C. Analysis of Implementation

To analyze the derived three-factor authentication scheme, we take the fingerprint based fuzzy vault scheme [35] proposed by Nandakumar, Jain, and Pankanti in 2007, though any secure biometrics authentication protocol can be used. In their fuzzy vault scheme, each element $v_i \in V$ $(i \in \{1, 2, \ldots, r+s\})$ is represented as three-tuple such that $v_i = (x, y, \theta)$, where r is the number of points in L (w.r.t. the points in V which lie on P) and s denotes the number of noise points in V which do not lie on P, (x, y) is the row and column coordinates

TABLE I
COMPARISON OF SCHEMES (A)

Efficiency \ Scheme		Li-Hwang scheme [26]	Li *et al.*'s scheme [27]	Das's scheme [28]	Kim-Lee-Yoo scheme [16]	Bhargav-Spantze *et al.*'s scheme [21], [20]	Fan-Lin scheme [23]	Proposed scheme
No server storage[1]		√	√	√	√	×	×	√
Reg-Cost	Server	Very Low	Very Low	Very Low	2 EXP	1 EXP	Low	Low
	Client	Very Low	Very Low	Very Low	Very Low	2 EXP	Low	Very low
Auth-Cost	Server	Very Low	Very Low	Very Low	2 EXP	3 EXP	1 PKD	1 SigSign, 2 EXP, 1 PKD
	Client	Very Low	Very Low	Very Low	2 EXP	2 EXP	PKE	1 SigVer, 2 EXP, 1 PKE

Very Low The most expensive operation is hash function.
Low The most expensive operation is symmetric key encryption/decryption.
EXP: Large exponentiation computation is required.
PKE (or PKD): Asymmetric key encryption (or decryption) is required.
SigSign (or SigVery): Digital signature signing (or verification) is required.

TABLE II
COMPARISON OF SCHEMES (B)

Scheme \ Property	Contactless password changing	Biometrics privacy	Session key establishment supportance	Security
Li-Hwang scheme [26]	√	×	×	Vulnerable to man-in-the-middle attack
Li *et al.*'s scheme [27]	√	×	√	Fail to provide strong authentication
Das's scheme [28]	√	×	√	Vulnerable to off-line password guessing attack
Kim-Lee-Yoo scheme [16]	√	√	×	Vulnerable to impersonation attack
Bhargav-Spantze *et al.*'s scheme [21], [20]	×	√	×	Secure under three-factor requirements
Fan-Lin scheme [23]	×	√	√	Secure under three-factor requirements
Proposed scheme	√	√	√	Secure under three-factor requirements

in the image showing the location, θ is the orientation which respect to the X-axis.

In addition, we take $s \approx 10r$ to satisfy the requirement that $s \gg r$. Moreover, 8-degree polynomial is used to encrypt 128-bit secrets, and the lengths of x, y, θ (quantized and represented in bit strings) are $6, 5, 5$, respectively. As the parameter showed in [35], there are around 30 points which lie on the selected polynomial in a 640×480 at 500 dpi resolution fingerprint image, so we could conclude $r = 30$ and $s = 300$. Thus, V contains 330 points which requires 660 Bytes space. Furthermore, the length of help data used in this fuzzy vault scheme is depended on the points of maximum curvature in the flow curves, and it can be ignored. Thus, only less than 1 KB additional data are required if compared with the underlying two factor authentication scheme.

The genuine acceptance rate (GAR) and false acceptance rate (FAR) are influenced by the degree of polynomial. In the above setting, the FAR falls in $0.01\% - 0.04\%$ and GAR is grater than 90%. In fact, GAR is acceptable even if $GAR = 50\%$, as this means that genuine users can pass authentication by scanning their fingerprint about twice.

We compare our instantiation with other schemes into two tables, namely Table I and Table II. The focus of the first table is on the efficiency, and the second table is mainly focusing on the security and privacy. In Table I, *Reg-Cost* and *Auth-Cost* present the computational cost in the registration phase and authentication phase, respectively; the number indicates that how many times the corresponding operation is required by the protocol, e.g. *2 EXP* means that exponentiation computation is required twice.

These two tables show that Li-Hwang scheme [26], Li *et al.*'s scheme [27], Das's scheme [28], and Kim-Lee-Yoo scheme [16] support contactless password changing, and the first three schemes only have very small computation cost. However, all of them have security flaws. In contrast, both Bhargav-Spantze *et al.*'s scheme [21] and Fan-Lin scheme [23] are secure under the three-factor adversary model, but they do not support contactless password changing and Bhargav-Spantze *et al.*'s scheme does not support session key establishment, so perfect forward secret cannot been guaranteed. While the derived protocol protects user privacy, offers contactless password changing, and supports session key establishment, with acceptable computation cost.

V. SECURITY AND PRIVACY ANALYSIS

The hypothesis of the security of our proposed generic framework is that (A) the underlying two-factor authentication protocol is secure when any one factor is compromised, and (B) the fuzzy vault system is secure when the biometric template is kept secret. In a system derived by using our framework, the authentication process is actually the same as

[1]This guarantees that the server does not store users' password and biodata.

the underlying two-factor authentication protocol. However, the difference is that, the "password" $PW = h(PW_1||PW_2)$ is the output of a hash function, where the input data are the human memorisable password PW_1, and the secret bitstring PW_2 which is protected by using the fuzzy vault system.

Considering three different cases: the PW_1 and the biometric template are exposed to the attacker, the PW_1 and the smart card are exposed to the attacker, and the biometric template and the smart card are exposed to the attacker.

To make the analysis easier to be understood, we assume a very strong attacker, who can recover PW_2 if the biometric template is compromised, though actually the attacker also needs the information stored in the smart card. However, this assumption will not affect our security since if the system is secure against a very strong attacker, then the system is also secure against a weak attacker.

Loosely speaking, if the PW_1 and biometric template (so the PW_2) are compromised, then it is the similar case as that in the underlying two-factor authentication protocol, the password is corrupted while the smart card remains secure (since PW can be computed in this case). So the derived system will remain secure. Otherwise, we can build a probabilistic polynomial time (PPT) Turing machine to break the security of the underlying two-factor authentication protocol, which contradicts to the hypothesis (A).

If the case that PW_1 and the smart card are compromised, we have that PW_2 is secure thanks to the hypothesis (B). In addition, by hypothesis (A), we have that the system is secure if PW remains secure. So, the only way the attacker can pass the authentication is to discover the value of PW. If there is a way to discover the value of PW with overwhelming probability, then we can either construct a PPT Turing machine that is able to discover the value of the password in the underlying two-factor authentication protocol, which is a contradiction of hypothesis (A); or we can find the hash collision which contradicts to the assumption of a secure hash function. The case that the biometric template and smart card are composed is similar to this case. Now, we present the formal security analysis of the instantiation given in Section IV.

Considering two communicating parties A and B, a mutual authentication protocol is secure if and only if participant A accepting participant B implies B accepting A. The generic security model of mutual authentication have been well studied [42]–[44]; however, more strict security model is desired for the three-factor authentication systems due to the more intricate authentication conditions. Currently, the formal security analysis of multiple factor authentication scheme remains as a challenging issue [32], although there are some existing works [23], [45], [46].

This section proposes a security model for three-factor authenticated key exchange schemes by extending and adopting the existing generic model [42]. Based on the proposed model, we prove the security of the derived scheme.

A. Security Model

We place probabilistic polynomial time (PPT) adversary A between user U_i in user set U and sever S_j in server set S. Let $\prod_{U,S}^{sid}$ be the user oracle interacting with the server in session sid; and $\prod_{S,U}^{sid}$ denotes the server oracle interacting with user in the session sid. It is obvious that if protocol $\prod$ is secure when A knows two out of three factors, then $\prod$ is still secure when only one factor has been leaked to A. Therefore, we only consider the case of two corrupted factors. A can make following oracle queries.

1) *Register($\prod$, S_j):* Upon receiving this query from A, the server oracle acts as S_j to run the registration phase with A, and issues identity ID_i and sends smart card SC to A.
2) *Execute(U_i, S_j, Sid):* This oracle query models all passive attackers who can eavesdrop on all messages transmitted between U and S in session sid in $\prod$. Upon receiving this query, $\prod_{U,S}^{sid}$ and $\prod_{S,U}^{sid}$ will execute protocol as U_i and S_j in $\prod$, respectively. The messages exchanged between them will be recorded and sent to A.
3) *Send(U_i, S_j, Sid, M_m, m):* This query sends message M_m with sequence of message flow m to server oracle $\prod_{S,U}^{sid}$ which simulates S_j, and then, the oracle will compute a response honestly in $\prod$, and send the response to A.
4) *Send(S_j, U_i, Sid, $M_{m'}$, m'):* This query sends message $M_{m'}$ with a sequence of message flow m' to user oracle $\prod_{U,S}^{sid}$ which simulates U_i, and then, the user oracle will compute a response honestly in $\prod$, and send the response to A. Upon receiving the query with $m' = \lambda$, where λ is an empty set, from A, the user oracle will start a new session and send a service request message to A.
5) *Reveal($\prod$, U_i, S_j, Sid):* This query models the leakage of a session key in session sid between user U_i and server S_j. This query can only be made when a session key has been shared between the server and the user in session sid. Upon receiving this query, the user oracle will send the shared session key to A.
6) There are three corruption queries:
   a) *Corrupt(U_i, pw, SC):* Upon receiving this query, user oracle will output the user U_i's password pw and the data stored in the smart card SC;
   b) *Corrupt(U_i, pw, Bio):* Upon receiving this query, user oracle will output the user U_i's password pw and the biometric template Bio;
   c) *Corrupt(U_i, SC, Bio):* Upon receiving this query, user oracle will output the user U_i's biometric template Bio and the data stored in the smart card SC;

   Note that A can only make one corruption query on the same target.
7) *Test(U_i, S_j, Sid):* This query can be made by A only after a session key has been shared between U_i and S_j in a fresh session sid. If so, then a coin b is tossed, if it lands $b = 0$, then this oracle outputs the session key. Otherwise, a fixed-length random string is returned. A needs to output $b' = 0$ or $b' = 1$ as the result of distinguishing the session key from the random string. A can only ask this query once.

The definitions of matching conversations, secure mutual authentication and secure key exchange [42] are reviewed as follows.

Definition 1 (Matching Conversations): Considering fix number of moves $R = 2\rho - 1$ and R-move protocol $\prod$. Run $\prod$ in the presence of adversary A and consider two oracles $\prod_{U,S}^{sid}$ and $\prod_{S,U}^{sid}$ that engage in conversations K and K', respectively. (τ, α, β) denotes that A obtains response β by sending α to an oracle at time τ. $\alpha_1 = \lambda$ indicates the start point of a new session in protocol $\prod$. "$*$" denotes the final decision of R-move protocol $\prod$.

1) We say that K' is a matching conversation to K if there exist $\tau_0 < \tau_1 < \cdots < \tau_R$ and $\alpha_1, \beta_1, \ldots, \alpha_\rho, \beta_\rho$ such that K is prefixed by $(\tau_0, \lambda, \alpha_1)$, $(\tau_2, \beta_1, \alpha_2), \ldots, (\tau_{2\rho-4}, \beta_{\rho-2}, \alpha_{\rho-1})$, $(\tau_{2\rho-2}, \beta_{\rho-1}, \alpha_\rho)$ and K' is $(\tau_1, \alpha_1, \beta_1)$, $(\tau_3, \alpha_2, \beta_2), \ldots, (\tau_{2\rho-3}, \alpha_{\rho-1}, \beta_{\rho-1})$.
2) We say that K is a matching conversation to K' if there exist $\tau_0 < \tau_1 < \cdots < \tau_R$ and $\alpha_1, \beta_1, \ldots, \alpha_\rho, \beta_\rho$ such that K' is prefixed by $(\tau_1, \alpha_1, \beta_1)$, $(\tau_3, \alpha_2, \beta_2), \ldots, (\tau_{2\rho-3}, \alpha_{\rho-1}, \beta_{\rho-1})$, $(\tau_{2\rho-1}, \alpha_\rho, *)$ and K is $(\tau_0, \lambda, \alpha_1)$, $(\tau_2, \beta_1, \alpha_2), \ldots, (\tau_{2\rho-4}, \beta_{\rho-2}, \alpha_{\rho-1})$, $(\tau_{2\rho-2}, \beta_{\rho-1}, \alpha_\rho)$.

Let $\prod_{U_i,S_j}^{sid}$ (resp. $\prod_{S_j,U_i}^{sid}$) be the oracle which acts as user U_i (resp. server S_j) communicating with server S_j (resp. user U_i). Let $No - Matching^{A,U_i}(k)$ (resp. $No - Matching^{A,S_j}(k)$) be the event that there exist U_i, S_j and sid such that $\prod_{U_i,S_j}^{sid}$ (resp. $\prod_{S_j,U_i}^{sid}$) has accepted A as $\prod_{S_j,U_i}^{sid}$ (resp. $\prod_{U_i,S_j}^{sid}$), while $\prod_{S_j,U_i}^{sid}$ (resp. $\prod_{U_i,S_j}^{sid}$) has not engaged in a matching conversation. In other words, it is the event that user U_i (resp. server S_j) believes that server S_j (resp. user U_i) is communicating with him, but in fact, it is the adversary A who has impersonated server S_j (resp. user U_i).

Remark 2: The above definition is defined for the case of $R = 2\rho - 1$. The case of $R = 2\rho$ is similar and we omit it here.

Definition 2 (Secure Three-Factor Mutual Authentication (STMA)): We say that $\prod$ is a secure mutual authentication protocol if the following properties are satisfied in presence of PPT adversary A defined in the adversary model.

1) If oracle $\prod_{U_i,S_j}^{sid}$ and $\prod_{S_j,U_i}^{i}$ have matched conversations, then they accept each other.
2) $\prod_{U_i,S_j}^{sid}$ accepted implies a matching conversation: the probability of $No - Matching^{A,U_i}(k)$ is negligible, where S_j should not be registered by A. (Secure server authentication).
3) $\prod_{S_j,U_i}^{sid}$ accepted implies a matching conversation: the probability of $No - Matching^{A,S_j}(k)$ is negligible, where U_i should not be registered by A. (Secure user authentication).

Definition 3 (Secure Three-Factor Authenticated Key Exchange (STAKE)): A Protocol $\prod$ is called $STAKE$ if the following properties hold in presence of PPT adversary A defined in the adversary model:

- $\prod$ is an $STMA$ protocol;
- if the session is fresh in protocol $\prod$, and both $\prod_{U_i,S_j}^{sid}$ and $\prod_{S_j,U_i}^{i}$ complete matching conversations, then they have shared the same session key;
- the advantage $Adv^A(k)$ is negligible.

Note that:

- A session is called fresh if both $\prod_{U_i,S_j}^{sid}$ and $\prod_{S_j,U_i}^{sid}$ accepted each other and no session key reveal query has been made to $\prod_{U_i,S_j}^{sid}$ or $\prod_{S_j,U_i}^{sid}$.
- $Adv^A(k) = |\Pr[Gguess^A(k)] - \frac{1}{2}|$, where the $\Pr[Gguess^A(k)]$ is the probability such that A has won in the $Test(U_i, S_j, sid)$.

B. Formal Security Analysis

To prove the security of our concrete scheme, we shall show that if A can successfully pass user or server authentication with a non-negligible probability, then we can construct a PPT Turing machine T to solve the underlying hard problem under the help of A with a non-negligible probability. The concrete protocol is reviewed as follows:

1) $U_i \rightarrow S$: $M_1 = (ID_i, sid, g^a)$
2) $S \rightarrow U_i$: $M_2 = (SID, sid, g^b, Sig_{SK'}(SID, ID_i, sid, g^a, g^b))$
3) $U_i \rightarrow S$: $M_3 = (ID_i, sid, CT)$, where $CT = E_{PK}(C_i', ID_i, SID, sid, g^a, g^b)$
4) S checks credential C_i'. U_i will pass user authentication if and only if $C_i' = PRF_x(H(ID_i))$.
   Now, the shared session key is g^{ab}.

Lemma 1 (Secure User Authentication): In the proposed protocol $\prod$, if the pseudo-random function (PRF) is replaced by an ideal random function, the public key encryption scheme is secure against CCA2 attack, and $\prod_{S_j,U_i}^{sid}$ has accepted, then the probability of $No - Matching^{A,S_j}(k)$ is negligible even in presence of PPT adversary A in the adversary model.

Proof: This can be proved by contradiction. If there exists an adversary A who can pass user authentication with non-negligible probability ϵ, then we can construct a PPT Turing machine T to solve the underlying hard problem without knowing secret key x, i.e. winning the game of PRF (Game-PRF), with a non-negligible probability by using A.

Let's assume that PRF is an ideal random function. The Game-PRF is defined as follows: there are two participants, a challenger and a PRF oracle $\prod_{PRF}$ which has the secret x. The challenger has the power to ask $\prod_{PRF}$ for the $PRF_x(M)$ of any message M as many times as she wants. The game is that this challenger sends two different plaintexts P_0 and P_1 to the PRF oracle, which will output $PRF_x(P_b)$ to the challenger, where P_0 and P_1 have not been asked by the challenger, and b is either 0 or 1 according to the result of coin tossing. After that, the challenger needs to output $b' = 0$ or $b' = 1$ as her guess of value b. If $b' = b$, then the challenger won the game. Let $\Pr_{adv}[PRF] = \Pr_{win} - \frac{1}{2}$ be the advantage of correct guessing of b, where $\Pr_{win}$ denotes the probability of the event that this challenger won the game.

The basic idea is that to win Game-PRF, T simulates an environment of our concrete protocol to convince adversary A that this simulation is the real environment of concrete protocol execution. On the other side, A should only has a negligible probability to know the truth, i.e. this is not a real protocol environment but a simulation. In such a simulation,

T communicates with A who has the ability to break our concrete protocol in some way in a session with session ID sid with a non-negligible probability. Then, in order to win Game-PRF, T will make use of A's ability to make the decision of which input message has been used to generate the output $PRF_x(P_b)$ with a non-negligible probability.

The simulation is constructed as follows. In the simulation, T answers all oracle queries made by A. To achieve this goal, T needs to setup (SK, PK) for the public key scheme and (SK', PK') for the signature scheme, while T does not know the value of long term secret key x which is for $\prod_{PRF}$. $\prod_{U_i,S_j}^{sid}$ denotes the user oracle which has password PW_1, smart-card SC, and corresponding biometric template X which can recover biometric key PW_2 with the SC. $\prod_{S_j,U_i}^{sid}$ denotes the server oracle which has PRF oracle $\prod_{PRF}$. In our concrete protocol, A can make the following queries:

- *Register($\prod, S_j$):* Upon receiving this query from A, T runs the registration phase with A with the help of $\prod_{PRF}$.
- *Execute(U_i, S_j, Sid):* In $\prod$, $\prod_{U_i,S_j}^{sid}$ and $\prod_{S_j,U_i}^{sid}$ generate and record all messages transmitted between U_i and S_j in session sid, then send these messages to A.
- *Send(U_i, S_j, Sid, M_m, m):* A can send M_1 to T, then T responds to M_2 by using SK' to sign a signature as the protocol specified. Upon receiving M_3 from A, T sends the result of user authentication according to M_1 and M_3 by using SK to decrypt the ciphertext and asking $\prod_{PRF}$ in order to verify the credential.
- *Send($S_j, U_i, Sid, M_{m'}, m'$):* Upon receiving a new session query $Send(S_j,U_i,sid, M_\lambda,\lambda)$, T asks $\prod_{U_i,S_j}^{sid}$ to send first message M_1 to A. After receiving corresponding message M_2, T checks the signature by using PK'. If the signature is valid, T asks $\prod_{PRF}$ and encrypts its output to form message M_3.
- *Corrupt($U_i, Factor_a, Factor_b$):* Upon receiving this query, $\prod_{U_i,S_j}^{sid}$ will send the corresponding two factors according to a and b, where $a, b \in \{pw, SC, Bio\}$ and $a \neq b$.

If A can pass user authentication successfully with a non-negligible probability without asking $\prod_{U_i,S_j}^{sid}$, there must exist a matching conversation between A and T who simulates server S_j if the following happens. First, A asks $Corrupt(U_i, factor_a, factor_b)$ to obtain two factors, then sends the first message to T who then responds with the second message. Finally, A forms the third message to T.

Now, we show how T makes use of A to win Game-PRF with non-negligible advantage as follows. We assume that A attacks at least once among q_s sessions, while T does not know which session A is going to attack. Now, T chooses a session out of q_s sessions randomly. Then, the probability of A passing user authentication in this session is $\frac{1}{q_s} \cdot \epsilon$.

To avoid the case that A found that this environment is only a simulation, in the rest $q_s - 1$ sessions, T redirects the identity ID_r, which is included in the first message, to oracle $\prod_{PRF}$ which will respond $PRF_x(ID_r)$ back to T. Then, T records this identity into the compromised table and checks whether A has passed the user authentication by matching $PRF_x(ID_r)$ with the credential which is encrypted in the third message. If they are matched, then T responds to A that T accepts A's login request. Otherwise, T rejects A's request. For these sessions, T just randomly guesses the value of b, so the probability that T wins the game is $\frac{1}{2}$.

To use A, after receiving first message $M_1 = (ID_{new}, sid, g^a)$, T forms $M_2 = (SID, sid, g^b, Sig_{SK'}(SID, ID_{new}, sid, g^a, g^b))$ by using SK' and sends it to A. If A can successfully pass user authentication, s/he must be able to forge third message $M_3 = (ID_{new}, sid, CT)$, where $CT = E_{PK}(C'_{new}, ID_{new}, SID, sid, g^a, g^b)$. Now, T requires to start the Game-PRF by choosing two distinct messages $y_0 = H(ID_{new})$ and $y_1 = R_1$, and sends (y_0, y_1) to the PRF test query. The query responds $PRF_x(y_b)$ to T, then T decrypts CT to recover C'_{new} and checks whether the response is the same as C'_{new}. If it is, then it outputs $b' = 0$ as the guessed result of b. Otherwise, it outputs $b' = 1$.

We now analyze the probability of game winning. We assume that A forges user U_{new}, and passes user authentication successfully in polynomial time τ, with non-negligible probability ϵ, after asking q_R times $Register(\prod, S_j)$, q_E times $Execute(U_i, S_j, sid)$, q_S times *send* query in q_s sessions. The formula of calculating probability $\Pr_{adv}[PRF]$ of three different corrupting cases should be the same but with different ϵ because we do not care how A can pass the user authentication. If A does not select this special session, the probability of game wining without the help of A is $\frac{1}{2}$. Otherwise, if A indeed attacks this special session chose by T, then the probability is concerned as follows. The probability of A pass authentication is ϵ, so the probability that we win the Game-PRF is $(\epsilon \cdot 1 + (1 - \epsilon) \cdot \frac{1}{2})$. Because if A has passed authentication, then we have 100% probability to win the game. However, A may fail with the probability of $(1-\epsilon)$, in this case, we have $\frac{1}{2}$ probability to win the game. Thus,

$$\begin{aligned}\Pr_{adv}[PRF] &= \frac{1}{q_s} \cdot (\epsilon \cdot 1 + (1 - \epsilon) \cdot \frac{1}{2}) + \frac{q_s q_s - 1}{q_s} \cdot \frac{1}{2} - \frac{1}{2} \\ &= \frac{\epsilon + q_s}{2q_s} - \frac{1}{2} \\ &= \frac{\epsilon}{2q_s}\end{aligned}$$

It is clear that $\Pr_{adv}[PRF]$ is non-negligible since ϵ is non-negligible, and T spends $\tau' = \tau + \tau_2$ time to win games, where τ_2 is the executing time of T interaction with the test query. It is obvious that both τ and τ'_2 are polynomial times, thus, τ' is also a polynomial time. Therefore, T can win Game-PRF with non-negligible advantage $\Pr_{adv}[PRF]$, and this contradicts assumption. ■

Lemma 2 (Secure Server Authentication): In proposed protocol $\prod$, if the signature scheme is unforgeable against adaptive chosen message attack, and $\prod_{U_i,S_j}^{sid}$ has accepted, then for any PPT adversary A in the adversary model, the probability of $No - Matching^{A,U_i}(k)$ is negligible.

Proof: This can be proved by contradiction as well. If A has been accepted by $\prod_{U_i,S_j}^{sid}$ with non-negligible probability of $No - Matching^{A,U_i}(k)$, then we can construct a PPT machine T which can win the Game of unforgeable against chosen message attack (Game-UFCMA) [47] by employing A.

In Game-UFCMA, there is a signature signing oracle $\prod_{Sign}$. A challenger who has PK' can make signing queries on messages, and can also verify the signature by using PK'. To win the game, the challenger needs to output a fresh message M_{new} with valid signature on it. Let $\Pr_{win}[SIG]$ be the probability of the advantage of game winning.

The basic idea is that to win Game-UFCMA, T simulates an environment of our concrete protocol to convince adversary A that this simulation is the real concrete protocol. In addition, A should only have a negligible probability to know the truce, i.e. this is not a real protocol environment but a simulation. In such simulation, T communicates with A who has the ability to successfully forge server's signature in a session with session ID sid with a non-negligible probability. Then, T will make use of A's ability to win Game-UFCMA with a non-negligible probability.

To use A, T needs to simulate A's view as follows. In the simulation, T answers all oracle queries made by A. To achieve this goal, T needs to setup all parameters except signing key SK'. In our concrete scheme, A can ask following quires:

- *Execute*(U_i, S_j, Sid)*:* In $\prod$, $\prod^{sid}_{U_i,S_j}$ and $\prod^{sid}_{S_j,U_i}$ generate and record all messages transmitted between U_i and S_j, then send them to A.
- *Send*(U_i, S_j, Sid, M, m)*:* A can send M_1 to T, then T responds M_2 by asking the $\prod_{Sign}$ of $\prod^{sid}_{S_j,U_i}$. Upon receiving M_3 from A, T sends the result of user authentication according to M_1 and M_3.
- *Send*$(S_j, U_i, Sid, M_{m'}, m')$*:* Upon receiving new session query $Send(S_j, M_\lambda, \lambda)$, T asks $\prod^{sid}_{U_i,S_j}$ to send first message M_1 to A. After receiving corresponding M_2, T checks the signature, and forms M_3 if the signature is valid.

If A can successfully pass server authentication with a non-negligible probability, there must exist a matching conversation between A and T who simulates user U_i if the following happens. In the simulation, first, T chooses message $M_1 = (T, sid, g^a)$, and sends it to A. If A can successfully pass server authentication, then A will form message $M_2 = (SID, sid, g^b, Sig_{SK'}(SID, T, sid, g^a, g^b))$ and send it to T.

To win the Game-UFCMA with A's help, T sends $M = (SID, T, sid, g^a, g^b)$ together with the signature in M_2 to the test query. We assume that A forges server S and passes server authentication successfully in polynomial time τ, with non-negligible probability ϵ, asking q_E times to $Execute(U_i, S_j, sid)$ and q_S times to send a query, which contains q_s times $Send(S_j, U_i, sid, M_{m'}, m')$. Let η be the probability of T winning Game-UFCMA when A has failed to pass server authentication. The probability is analysed as follows. In q_s times *send* query made by A, we choose one query to help us to answer the Game-UFCMA. The probability of A pass sever authentication is ϵ, so the probability of we win the Game-UFCMA is $(\epsilon \cdot 1 + (1-\epsilon) \cdot \eta)$. Because that if A has passed authentication, then we have 100% probability to win the game. On the other side, A may also failed with the probability of $(1-\epsilon)$, in this case, we have the probability of η to win the game. For the rest queries, the probability of game wining without the help of A is η. Thus,

$$\begin{aligned}\Pr_{win}[SIG] &= \frac{1}{q_s} \cdot (\epsilon \cdot 1 + (1-\epsilon) \cdot \eta) + \frac{q_s - 1}{q_s} \cdot \eta \\ &= \frac{\epsilon + \eta \cdot (q_s - \epsilon)}{q_s}\end{aligned}$$

It is clear that $\Pr_{win}[SIG]$ is non-negligible since ϵ is non-negligible and η is negligible. The time T spent to win the games is $\tau' = \tau + \tau_3$, where t_3 is the executing time of T spends in GAME-UFCMA. τ' is a polynomial time because both τ and τ_3' are polynomial times. Therefore, we can construct PPT machine T to win Game-UFCMA of the signature scheme, with non-negligible probability, and this is a contradiction. ■

Theorem 1 (Secure Three-Factor Mutual Authentication (STMA)): In proposed protocol $\prod$, if: (A) the PRF is replaced by an ideal random function and PKE scheme is secure against $CCA2$ attack; (B) the signature scheme is unforgeable against chosen message attack; (C) at least one of $\prod^{sid}_{U_i,S_j}$ and $\prod^{sid}_{S_j,U_i}$ has accepted; then for any PPT adversary A in the adversary model, the probabilities of both $No - Matching^{A_{U_i}}(k)$ and $No - Matching^{A_{S_j}}(k)$ are negligible.

Proof: Obviously, the first condition of Definition 2 holds because it is easy to verify that our concrete protocol is correct. In addition, by Lemma 1 and Lemma 2, the second and third conditions of Definition 2 also hold. Therefore, Theorem 1 holds. ■

Theorem 2 (Secure Three-Factor Authenticated Key Exchange (STAKE)): In proposed protocol $\prod$, if (A) the PRF is replaced by an ideal random function and the PKE scheme is secure against $CCA2$ attack; (B) the signature scheme is unforgeable against adaptive chosen message attack; then for any PPT adversary A in the adversary model, the advantage $Adv^A(k)$ of A winning the game of $AKEP$ in a fresh session is negligible.

Proof: According to the Definition 3, $STAKE$ needs to meet three conditions. The first condition is that protocol $\prod$ is required to satisfies $STMA$. This condition is achieved because Theorem 1. The second condition is that for a fresh session in protocol $\prod$, if complete conversations are matched, then the same session key must be shared between these two communicating parties. This condition is achieved because that in our concrete scheme, the key exchange is the plain two-move Diffie-Hellman protocol [43], and this condition is a well-known property and it was proved. For the third condition, the advantage $Adv^A(k) = |\Pr[Gguess^A(k)] - \frac{1}{2}|$ is non-negligible due to [43]. Thus, $\prod$ is a secure three-factor authenticated key exchange protocol. ■

C. Privacy Discussion

The proposed framework preserves user privacy due to the following reasons. First, the server does not know any information about the user's biometric template since the user does not need to provide biometric templates to the server. Second, the data stored in SC will not leak biometric information since V contains a large amount of noise. Thus, the probability

of successful recovering the biometric template is negligible due to [35]. Moreover, the helper data H which is required in the fingerprint based fuzzy vault scheme are global features, and two very different fingerprint can have very similar helper data. So, H also will not leak biometric characteristics [35].

VI. Conclusion

The proposed framework can systematically and efficiently upgrade two-factor authentication schemes to three-factor authentication schemes. The derived scheme protects user's privacy, and enhances security. In addition, we made a case study by applying the framework on an existing two factor authentication scheme [41]. Our analysis, discussion, and formal proof show that the resulted three-factor protocol achieves higher security guarantee and preserves user privacy.

References

[1] A. K. Jain, A. Ross, and S. Pankanti, "Biometrics: A tool for information security," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 2, pp. 125–143, Jun. 2006.

[2] J. Yu, G. Wang, and Y. Mu, "Provably secure single sign-on scheme in distributed systems and networks," in *Proc. IEEE 11th TrustCom*, Jun. 2012, pp. 271–278.

[3] G. Wang, J. Yu, and Q. Xie, "Security analysis of a single sign-on mechanism for distributed computer networks," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 294–302, Feb. 2013.

[4] L. Lamport, "Password authentication with insecure communication," *Commun. ACM*, vol. 24, no. 11, pp. 770–772, Nov. 1981.

[5] C.-C. Chang and T.-C. Wu, "Remote password authentication with smart cards," *IEE Proc.-E Comput. Digital Techn.*, vol. 138, no. 3, pp. 165–168, May 1991.

[6] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *Proc. 6th ACM CCS*, 1999, pp. 28–36.

[7] V. Matyas, Jr., and Z. Ríha, "Toward reliable user authentication through biometrics," *IEEE Security Privacy*, vol. 1, no. 3, pp. 45–49, May/Jun. 2003.

[8] Z. Siddiqui, A.-H. Abdullah, M. K. Khan, and A. S. Alghamdi, "Smart environment as a service: Three factor cloud based user authentication for telecare medical information system," *J. Med. Syst.*, vol. 38, no. 1, pp. 1–14, 2014.

[9] W.-H. Yang and S.-P. Shieh, "Password authentication schemes with smart cards," *Comput. Security*, vol. 18, no. 8, pp. 727–733, 1999.

[10] L. Fan, J.-H. Li, and H.-W. Zhu, "An enhancement of timestamp-based password authentication scheme," *Comput. Security*, vol. 21, no. 7, pp. 665–667, 2002.

[11] J.-J. Shena, C.-W. Linb, and M.-S. Hwang, "Security enhancement for the timestamp-based password authentication scheme using smart cards," *Comput. Security*, vol. 22, no. 7, pp. 591–595, 2003.

[12] C.-C. Yanga, R.-C. Wang, and T.-Y. Chang, "An improvement of the Yang–Shieh password authentication schemes," *Appl. Math. Comput.*, vol. 162, no. 3, pp. 1391–1396, 2005.

[13] E.-J. Yoon, W.-H. Kim, and K.-Y. Yoo, "Security enhancement for password authentication schemes with smart cards," in *Proc. TrustBus*, 2005, pp. 311–320.

[14] C.-K. Chan and L.-M. Cheng, "Cryptanalysis of timestamp-based password authentication scheme," *Comput. Security*, vol. 21, no. 1, pp. 74–76, 2002.

[15] G. Wang and F. Bao, "Cryptanalysis of timestamp-based password authentication schemes using smart cards," in *Proc. ICICS*, 2006, pp. 399–409.

[16] H.-S. Kim, S.-W. Lee, and K.-Y. Yoo, "ID-based password authentication scheme using smart cards and fingerprints," *SIGOPS Oper. Syst. Rev.*, vol. 37, no. 4, pp. 32–41, Oct. 2003.

[17] M. Scott, "Cryptanalysis of an ID-based password authentication scheme using smart cards and fingerprints," *SIGOPS Oper. Syst. Rev.*, vol. 38, no. 2, pp. 73–75, Apr. 2004.

[18] U. Uludag, S. Pankanti, A. K. Jain, and S. Prabhakar, "Biometric cryptosystems: Issues and challenges," *Proc. IEEE*, vol. 92, no. 6, pp. 948–960, Jun. 2004.

[19] A. K. Jain and D. Maltoni, *Handbook of Fingerprint Recognition*. Secaucus, NJ, USA: Springer-Verlag, 2003.

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

[21] A. Bhargav-Spantzel, A. C. Squicciarini, S. K. Modi, M. Young, E. Bertino, and S. J. Elliott, "Privacy preserving multi-factor authentication with biometrics," *J. Comput. Security*, vol. 15, no. 5, pp. 529–560, 2007.

[22] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Advances in Cryptology*. Berlin, Germany: Springer-Verlag, 1991, pp. 129–140.

[23] C.-I. Fan and Y.-H. Lin, "Provably secure remote truly three-factor authentication scheme with privacy protection on biometrics," *IEEE Trans. Inf. Forensics Security*, vol. 4, no. 4, pp. 933–945, Dec. 2009.

[24] X. Sun, H. Wang, J. Li, and Y. Zhang, "Satisfying privacy requirements before data anonymization," *Comput. J.*, vol. 55, no. 4, pp. 422–437, 2012.

[25] L. A. Dunning and R. Kresman, "Privacy preserving data sharing with anonymous ID assignment," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 2, pp. 402–413, Feb. 2013.

[26] C.-T. Li and M.-S. Hwang, "An efficient biometrics-based remote user authentication scheme using smart cards," *J. Netw. Comput. Appl.*, vol. 33, no. 1, pp. 1–5, Jan. 2010.

[27] X. Li, J.-W. Niu, J. Ma, W.-D. Wang, and C.-L. Liu, "Cryptanalysis and improvement of a biometrics-based remote user authentication scheme using smart cards," *J. Netw. Comput. Appl.*, vol. 34, no. 1, pp. 73–79, 2011.

[28] A. K. Das, "Cryptanalysis and further improvement of a biometric-based remote user authentication scheme using smart cards," *Int. J. Netw. Security Appl.*, vol. 3, no. 2, pp. 13–28, 2011.

[29] X. Huang, Y. Xiang, A. Chonka, J. Zhou, and R. H. Deng, "A generic framework for three-factor authentication: Preserving security and privacy in distributed systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 8, pp. 1390–1397, Aug. 2011.

[30] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Advances in Cryptology*. Berlin, Germany: Springer-Verlag, 2004, pp. 523–540.

[31] Y. Wu and B. Qiu, "Transforming a pattern identifier into biometric key generators," in *Proc. IEEE ICME*, Jul. 2010, pp. 78–82.

[32] D. He, M. Ma, Y. Zhang, C. Chen, and J. Bu, "A strong user authentication scheme with smart cards for wireless communications," *Comput. Commun.*, vol. 34, no. 3, pp. 367–374, 2011.

[33] A. Nagar, K. Nandakumar, and A. K. Jain, "Securing fingerprint template: Fuzzy vault with minutiae descriptors," in *Proc. 19th Int. Conf. Pattern Recognit.*, Dec. 2008, pp. 1–4.

[34] T. C. Clancy, N. Kiyavash, and D. J. Lin, "Secure smartcardbased fingerprint authentication," in *Proc. ACM Workshop Biometrics, Methods Appl.*, 2003, pp. 45–52.

[35] K. Nandakumar, A. K. Jain, and S. Pankanti, "Fingerprint-based fuzzy vault: Implementation and performance," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 4, pp. 744–757, Dec. 2007.

[36] U. Uludag and A. Jain, "Securing fingerprint template: Fuzzy vault with helper data," in *Proc. CVPR Workshop*, Jun. 2006, p. 163.

[37] A. Juels and M. Sudan, "A fuzzy vault scheme," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2002, p. 408.

[38] J. Feng, "Combining minutiae descriptors for fingerprint matching," *Pattern Recognit.*, vol. 41, no. 1, pp. 342–352, 2008.

[39] A. Teoh and S. O. Thian, "Secure biometric template protection via randomized dynamic quantization transformation," in *Proc. IEEE Int. Symp. Biometrics Security Technol.*, Apr. 2008, pp. 1–6.

[40] W. Sheng, G. Howells, M. Fairhurst, and F. Deravi, "Template-free biometric-key generation by means of fuzzy genetic clustering," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 2, pp. 183–191, Jun. 2008.

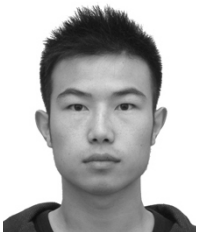
[41] G. Yang, D. S. Wong, H. Wang, and X. Deng, "Two-factor mutual authentication based on smart cards and passwords," *J. Comput. Syst. Sci.*, vol. 74, no. 7, pp. 1160–1172, Nov. 2008.

[42] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in *Advances in Cryptology*. Berlin, Germany: Springer-Verlag, 1993, pp. 232–249.

[43] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Advances in Cryptology*. Berlin, Germany: Springer-Verlag, 2001, pp. 453–474.

[44] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Advances in Cryptology*. Berlin, Germany: Springer-Verlag, 2000, pp. 139–155.

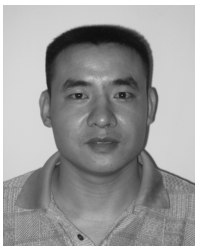
[45] J. Xu, W.-T. Zhu, and D.-G. Feng, "An improved smart card based password authentication scheme with provable security," *Comput. Standards Inter.*, vol. 31, no. 4, pp. 723–728, 2009.
[46] J. Xu, W.-T. Zhu, and W. Jin, "A generic framework for constructing cross-realm C2C-PAKA protocols based on the smart card," *Concurrency Comput., Pract. Exper.*, vol. 23, no. 12, pp. 1386–1398, 2011.
[47] S. Goldwasser, S. Micali, and R. L. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," *SIAM J. Comput.*, vol. 17, no. 2, pp. 281–308, 1988.



Jiangshan Yu received the M.Sc. and M.Phil. degrees in computer science (information security) from the School of Computer Science and Software Engineering, University of Wollongong, Wollongong, NSW, Australia, in 2011 and 2012, respectively. He is currently pursuing the Ph.D. degree at the University of Birmingham, Birmingham, U.K. The focus of his research has been on information security and privacy.

Guilin Wang received the Ph.D. degree in computer science from the Institute of Software, Chinese Academy of Sciences, Beijing, China, in 2001. He is currently a Senior Researcher with Huawei International Pte Ltd., Singapore. He was a Senior Lecturer with the University of Wollongong, Wollongong, NSW, Australia, a Lecturer with the University of Birmingham, Birmingham, U.K., a Research Scientist with the Institute for Infocomm Research, Singapore, and an Assistant Professor with the Chinese Academy of Sciences. He has authored or coauthored over 80 research publications in applied cryptography and telecommunication security. His main research interests include the analysis, design, and applications of digital signatures and security protocols. He has served as the Program Cochair for six international security conferences, a Committee Member for over 60 international conferences or workshops, and a reviewer for over 20 international journals.

Yi Mu (SM'03) received the Ph.D. degree from Australian National University, Canberra, ACT, Australia, in 1994. He is currently a Professor, the Head of the School of Computer Science and Software Engineering, and the Codirector of the Centre for Computer and Information Security Research with the University of Wollongong, Wollongong, NSW, Australia. His current research interests include information security and cryptography. He is the Editor-in-Chief of the *International Journal of Applied Cryptography* and serves as an Associate Editor for 10 other international journals. He has authored over 300 research papers. He is a member of the International Association for Cryptologic Research.



Wei Gao was born in 1978. He received the B.Sc. degree from Ludong University, Yantai, China, in 1996, the M.S. degree from Guangzhou University, Guangzhou, China, in 2003, and the Ph.D. degree from Hunan University, Changsha, China, in 2006. Since 2007, he has been an Associate Professor with Ludong University. His main research interests are applied mathematics, cryptography, and information security.