View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Institutional Knowledge at Singapore Management University

Singapore Management University
Institutional Knowledge at Singapore Management University

Research Collection School Of Information Systems

School of Information Systems

8-2011

A generic framework for three-factor authentication: preserving security and privacy in distributed systems

Xinyi HUANG

Yang Xiang

Ashley Chonka

Jianying Zhou

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

DOI: https://doi.org/10.1109/TPDS.2010.206

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research

Part of the Information Security Commons

Citation

HUANG, Xinyi; Xiang, Yang; Chonka, Ashley; Zhou, Jianying; and DENG, Robert H.. A generic framework for three-factor authentication: preserving security and privacy in distributed systems. (2011). *IEEE Transactions on Parallel and Distributed Systems*. 22, (8), 1390-1397. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/1421

This Journal Article is brought to you for free and open access by the School of Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email libIR@smu.edu.sg.

Published in IEEE Transactions on Parallel and Distributed Systems, 2011 August, Volume 22, Issue 8, Pages 1390-1397
https://doi.org/10.1109/TPDS.2010.206

A Generic Framework for Three-Factor Authentication: Preserving Security and Privacy in Distributed Systems

Xinyi Huang, Yang Xiang, *Member*, *IEEE*, Ashley Chonka,
Jianying Zhou, and Robert H. Deng, *Senior Member*, *IEEE*

Abstract—As part of the security within distributed systems, various services and resources need protection from unauthorized use. Remote authentication is the most commonly used method to determine the identity of a remote client. This paper investigates a systematic approach for authenticating clients by three factors, namely password, smart card, and biometrics. A generic and secure framework is proposed to upgrade two-factor authentication to three-factor authentication. The conversion not only significantly improves the information assurance at low cost but also protects client privacy in distributed systems. In addition, our framework retains several practice-friendly properties of the underlying two-factor authentication, which we believe is of independent interest.

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

1 Introduction

In a distributed system, various resources are distributed in the form of network services provided and managed by servers. Remote authentication is the most commonly used method to determine the identity of a remote client. In general, there are three authentication factors:

1. Something the client knows: password.
2. Something the client has: smart card.
3. Something the client is: biometric characteristics (e.g., fingerprint, voiceprint, and iris scan).

Most early authentication mechanisms are solely based on password. While such protocols are relatively easy to implement, passwords (and human generated passwords in particular) have many vulnerabilities. As an example, human generated and memorable passwords are usually short strings of characters and (sometimes) poorly selected. By exploiting these vulnerabilities, simple dictionary attacks can crack passwords in a short time [1]. Due to these concerns, hardware authentication tokens are introduced to strengthen the security in user authentication, and smart-card-based password authentication has become one of the most common authentication mechanisms.

Smart-card-based password authentication provides two-factor authentication, namely a successful login requires the client to have a valid smart card and a correct password. While it provides stronger security guarantees than password authentication, it could also fail if both authentication factors are compromised (e.g., an attacker has successfully obtained the password and the data in the smart card). In this case, a third authentication factor can alleviate the problem and further improve the system's assurance.

Another authentication mechanism is biometric authentication [2], [3], [4], where users are identified by their measurable human characteristics, such as fingerprint, voiceprint, and iris scan. Biometric characteristics are believed to be a reliable authentication factor since they provide a potential source of high-entropy information and cannot be easily lost or forgotten. Despite these merits, biometric authentication has some imperfect features. Unlike password, biometric characteristics cannot be easily changed or revoked. Some biometric characteristics (e.g., fingerprint) can be easily obtained without the awareness of the owner.[1] This motivates the three-factor authentication, which incorporates the advantages of the authentication based on password, smart card, and biometrics.

1.1 Motivation

The motivation of this paper is to investigate a systematic approach for the design of secure three-factor authentication with the protection of user privacy.

Three-factor authentication is introduced to incorporate the advantages of the authentication based on password, smart card, and biometrics. A well designed three-factor authentication protocol can greatly improve the information assurance in distributed systems. However, the previous

- *X. Huang and R.H. Deng are with the School of Information Systems, Singapore Management University, 80 Stamford Road, Singapore 178902. E-mail: {xyhuang, robertdeng}@smu.edu.sg.*
- *Y. Xiang is with the School of Information Technology, Deakin University, Melbourne Campus at Burwood, 221 Burwood Highway, Burwood, Victoria 3125, Australia. E-mail: yang@deakin.edu.au.*
- *A. Chonka is with the Faculty of Science and Technology, School of Information Technology, Pigdons Road, Waurn Ponds Campus, Deakin University, Geelong, Victoria 3216, Australia. E-mail: ashley.chonka@deakin.edu.au.*
- *J. Zhou is with the Institute for Infocomm Research (I^2R), A*STAR, 1 Fusionopolis Way, #21-01 Connexis, South Tower, Singapore 138632. E-mail: jyzhou@i2r.a-star.edu.sg.*

Manuscript received 15 Mar. 2010; revised 18 July 2010; accepted 16 Sept. 2010; published online 9 Nov. 2010.
Recommended for acceptance by D. Turgut.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2010-03-0160.
Digital Object Identifier no. 10.1109/TPDS.2010.206.

[1] Section 3.2 presents three other subtle issues in biometric authentication (especially in distributed systems).

research on three-factor authentication is confusing and far from satisfactory.

1.1.1 *Security Issues*

As we will show shortly (in Section 1.3), most existing three-factor authentication protocols are flawed and cannot meet security requirements in their applications. Even worse, some improvements of those flawed protocols are not secure either. The research history of three-factor authentication can be summarized in the following diagram.

NEW PROTOCOLS → BROKEN → IMPROVED PROTOCOLS → BROKEN AGAIN → $\cdots$.

1.1.2 *Privacy Issues*

Along with the improved security features, three-factor authentication also raises another subtle issue, namely how to protect the biometric data. Not only is this the privacy information of the owner, it is also closely related to the security in the authentication. As biometrics cannot be easily changed, the breached biometric information (either on the server side or the client side) will make the biometric authentication totally meaningless. However, this issue has received less attention than it deserves from protocol designers.

We believe it is worthwhile, both in theory and in practice, to investigate a generic framework for three-factor authentication, which can preserve the security and the privacy in distributed systems.

1.2 Contributions

The main contribution of this paper is a generic framework for three-factor authentication in distributed systems. The proposed framework has several merits as follows:

First, we demonstrate how to incorporate biometrics in the existing authentication based on smart card and password. Our framework is generic rather than instantiated in the sense that it does not have any additional requirements on the underlying smart-card-based password authentication. Not only will this simplify the design and analysis of three-factor authentication protocols, but also it will contribute a secure and generic upgrade from two-factor authentication to three-factor authentication possessing the practice-friendly properties of the underlying two-factor authentication system.

Second, authentication protocols in our framework can provide true three-factor authentication, namely a successful authentication requires password, smart card, and biometric characteristics. In addition, our framework can be easily adapted to allow the server to decide the authentication factors in user authentication (instead of all three authentication factors).

Last, in the proposed framework clients' biometric characteristics are kept secret from servers. This not only protects user privacy but also prevents a single-point failure (e.g., a breached server) from undermining the authentication level of other services. Furthermore, the verification of all authentication factors is performed by the server. In particular, our framework does not rely on any trusted devices to verify the authentication factors, which also meets the imperfect feature of distributed systems where devices cannot be fully trusted.

1.3 Related Work

Several authentication protocols have been proposed to integrate biometric authentication with password authentication and/or smart-card authentication. Lee et al. [5] designed an authentication system which does not need a password table to authenticate registered users. Instead, smart card and fingerprint are required in the authentication. However, due to the analysis given in [6], Lee et al.'s scheme is insecure under conspiring attack.

Lin and Lai [7] showed that Lee et al.'s scheme is vulnerable to masquerade attack. Namely, a legitimate user (i.e., a user who has registered on the system) is able to make a successful login on behalf of other users. An improved authentication protocol was given by Lin and Lai to fix that flaw. The new protocol, however, has several other security vulnerabilities. First, Lin-Lai's scheme only provides client authentication rather than mutual authentication, which makes it susceptible to the server spoofing attack [8]. Second, the password changing phase in Lin-Lai's scheme is not secure as the smart card cannot check the correctness of old passwords [9]. Third, Lin-Lai's scheme is insecure under impersonation attacks due to the analysis given by Yoon and Yoo [10], who also proposed a new scheme. However, the new scheme is broken and improved by Lee and Kwon [11].

In [12], Kim et al. proposed two ID-based password authentication schemes where users are authenticated by smart cards, passwords, and fingerprints. However, Scott [13] showed that a passive eavesdropper (without access to any smart card, password or fingerprint) can successfully login to the server on behalf of any claiming identity after passively eavesdropping only one legitimate login.

Bhargav-Spantzel et al. proposed a privacy preserving multifactor authentication protocol with biometrics [14]. The authentication server in their protocol does not have the biometric information of registered clients. However, the biometric authentication is implemented using zero knowledge proofs [15], which requires the server to maintain a database to store all users' commitments and uses costly modular exponentiations in the finite group.

In [16], Uludag et al. presented various methods of binding a cryptographic key with the biometric template of a user stored in the database. The cryptographic key cannot be revealed without a successful biometric authentication. However, the biometric database could put client privacy at risk. In order to protect client privacy, Fan and Lin [17] proposed a three-factor authentication scheme with privacy protection on biometrics. The essential approach of their scheme is as follows: 1) During the registration, the client chooses a random string and encrypts it using his/her biometric template; 2) The result (called sketch) is stored in the smart card; and 3) During the authentication, the client must convince the server that he/she can decrypt the sketch, which needs correct biometrics (close to the biometric template in the registration). As we shall show shortly, our framework employs a different approach. The client in our framework uses his/her biometrics to generate a random string. This leads to a generic three-factor authentication protocol from smart-card-based password authentication. Very recently, Li and Hwang [18] proposed another biometric-based remote client authentication scheme using smart card and password. Our analysis, which will be given shortly, points out two limitations of Li-Hwang's scheme in practical application. In addition, there are no satisfactory solutions for three-factor authentication with additional properties (e.g., key agreement with forward security), which have been studied intensively in smart-card-based password authentication.

Organization of this paper. The remainder of this paper is organized as follows: Section 2 briefly reviews the preliminaries of our framework. After that, we describe the challenges of biometric authentication in distributed systems in Section 3. The generic framework for three-factor authentication is given in Section 4. Section 5 provides the analysis of the proposed framework, and its formal security proofs are given in the supplementary file, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2010.206. Section 6 concludes this paper.

2 PRELIMINARIES

This section reviews the definitions of smart-card-based password authentication, three-factor authentication, and fuzzy extractor.

2.1 Smart-Card-Based Password Authentication

Definition 1. *A smart-card-based password authentication protocol (hereinafter referred to as SCPAP) consists of four phases.*

2-Factor-Initialization: The server (denoted by $\mathcal{S}$) generates two system parameters PK and SK. PK is published in the system, and SK is kept secret by $\mathcal{S}$. An execution of this algorithm is denoted by 2-Factor-Initialization$(\kappa) \rightarrow (PK, SK)$. Here, κ is system's security parameter which determines the size of PK and SK, and the security level of cryptographic algorithms.

2-Factor-Reg: The client (denoted by $\mathcal{C}$), with an initial password PW, registers on the system by running this interactive protocol with $\mathcal{S}$. The output of this protocol is a smart card SC. An execution of this protocol is denoted by

$$\mathcal{C}[PW] \xLeftrightarrow{\mathsf{2-Factor-Reg}} \mathcal{S}[SK] \rightarrow SC.$$

The information in square brackets indicates the secret value(s) known by the corresponding party. (The same notation will be used in the remainder of this paper.)

2-Factor-Login-Auth: This is another interactive protocol between the client and the server, which enables the client to login successfully using PW and SC. An execution of this protocol is denoted by

$$\mathcal{C}[PW,\ SC] \xLeftrightarrow{\mathsf{2-Factor-Login-Auth}} \mathcal{S}[SK] \rightarrow \{1, 0\}.$$

The output of this protocol is "1" (if the authentication is successful) or "0" (otherwise).

2-Factor-Password-Changing: This protocol enables a client to change his/her password after a successful authentication (i.e., 2-Factor-Login-Auth outputs "1"). The data in the smart card will be updated accordingly.

Security requirements. The attacker on SCPAP can be classified from two aspects: the behavior of the attacker and the information compromised by the attacker.

As an interactive protocol, SCPAP may face passive attackers and active attackers.

Passive attacker. A passive attacker can obtain messages transmitted between the client and the server. However, it cannot interact with the client or the server.

Active attacker. An active attacker has the full control of the communication channel. In addition to message eavesdropping, the attacker can arbitrarily inject, modify, and delete messages in the communication between the client and the server.

On the other hand, SCPAP is a two-factor authentication protocol, namely a successful login requires a valid smart card and a correct password. According to the compromised secret, an attacker can be further classified into the following two types.

Attacker with smart card. This type of attacker has the smart card, and can read and modify the data in the smart card. Notice that there are techniques to restrict access to both reading and modifying data in the smart card. Nevertheless, from the security point of view, authentication protocols will be more robust if they are secure against attackers with the ability to do that.

Attacker with password. The attacker is assumed to have the password of the client but is not given the smart card.

Definition 2 (Secure SCPAP). *The* basic *security requirement of SCPAP is that it should be secure against a passive attacker with smart card and a passive attacker with password. It is certainly* more desirable *that SCPAP is secure against an active attacker with smart card and an active attacker with password.*

2.2 Three-Factor Authentication

Three-factor authentication is very similar to smart-card-based password authentication, with the only difference that it requires biometric characteristics as an additional authentication factor.

Definition 3 (Three-Factor Authentication). *A three-factor authentication protocol involves a client $\mathcal{C}$ and a server $\mathcal{S}$, and consists of five phases.*

3-Factor-Initialization: $\mathcal{S}$ generates two system parameters PK and SK. PK is published in the system, and SK is kept secret by $\mathcal{S}$. An execution of this algorithm is denoted by 3-Factor-Initialization$(\kappa) \rightarrow (PK, SK)$, where κ is system's security parameter.

3-Factor-Reg: A client $\mathcal{C}$, with an initial password PW and biometric characteristics $BioData$, registers on the system by running this interactive protocol with the server $\mathcal{S}$. The output of this protocol is a smart card SC, which is given to $\mathcal{C}$. An execution of this protocol is denoted by

$$\mathcal{C}[PW, BioData] \xLeftrightarrow{\mathsf{3-Factor-Reg}} \mathcal{S}[SK] \rightarrow SC.$$

3-Factor-Login-Auth: This is another interactive protocol between the client $\mathcal{C}$ and the server $\mathcal{S}$, which enables the client to login successfully using PW, SC, and $BioData$. An execution of this protocol is denoted by

$$\mathcal{C}[PW,\ SC,\ BioData] \xLeftrightarrow{\mathsf{3-Factor-Login-Auth}} \mathcal{S}[SK] \rightarrow \{1, 0\}.$$

The output of this protocol is "1" (if the authentication is successful) or "0" (otherwise).

3-Factor-Password-Changing: This protocol enables a client to change his/her password after a successful authentication. The data in the smart card will be updated accordingly.

3-Factor-Biometrics-Changing[2]: An analogue of password-changing is biometrics-changing, namely the client can change his/her biometrics used in the authentication, e.g., using a different finger or using iris instead of finger.

While biometrics-changing is not supported by previous three-factor authentication protocols, we believe it provides the client with more flexibility in the authentication.

Cost effectiveness. In general, three-factor authentication is less computationally efficient than smart-card-based password authentication, since the former requires additional computational resources for biometric authentication. To make three-factor authentication practical, biometric-related operations must be performed fast and accurately. As indicated in [16], the performance of extracting and authenticating certain types of biometrics (e.g., face and keystroke) is not satisfactory, but others (e.g., fingerprint and iris) can satisfy practical requirements. (Examples include fingerprint recognition in laptops and biometric visa.)

Security requirements. A three-factor authentication protocol can also face **passive attackers** and **active attackers** as defined in SCPAP (Section. 2.1). A passive (an active) attacker can be further classified into the following three types.

Type I attacker has the smart card and the biometric characteristics of the client. It is not given the password of that client.

Type II attacker has the password and the biometric characteristics. It is not allowed to obtain the data in the smart card.

Type III attacker has the smart card and the password of the client. It is not given the biometric characteristics of that client. Notice that such an attacker is free to mount any attacks on the (unknown) biometrics, including biometrics faking and attacks on the metadata (related to the biometrics) stored in the smart card.

Definition 4 (Secure Three-Factor Authentication). *For a three-factor authentication protocol, the* basic *security requirement is that it should be secure against passive type I, type II, and type III attackers. It is certainly* more desirable *that a three-factor authentication protocol is secure against active type I, type II, and type III attackers.*

2.3 Fuzzy Extractor

This section briefly reviews the fuzzy extractor introduced in [21].

2.3.1 Metric Space

A metric space is a set $\mathcal{M}$ with a distance function $\mathrm{dis}: \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}^+ = [0, \infty)$ which obeys various natural properties. One example of metric space is *Hamming metric*: $\mathcal{M} = \mathcal{F}^n$ is over some alphabet $\mathcal{F}$ (e.g., $\mathcal{F} = \{0, 1\}$) and $\mathsf{dis}(w, w')$ is the number of positions in which they differ.

2.3.2 Statistic Distance

The statistical distance between two probability distributions A and B is denoted by $\mathbf{SD}(A, B) = \frac{1}{2}\sum_v |\Pr(A = v) - \Pr(B = v)|$.

2.3.3 Entropy

The min-entropy $\mathbf{H}_\infty(A)$ of a random variable A is $-\log(\max_{\mathrm{a}} \Pr[\mathrm{A} = \mathrm{a}])$.

2.3.4 Fuzzy Extractor

A fuzzy extractor extracts a nearly random string R from its biometric input w in an error-tolerant way. If the input changes but remains close, the extracted R remains the same. To assist in recovering R from a biometric input w', a fuzzy extractor outputs an auxiliary string P. However, R remains uniformly random even given P. The fuzzy extractor is formally defined as below.

Definition 5 (Fuzzy Extractor). *An* $(\mathcal{M}, m, \ell, t, \epsilon)$ *fuzzy extractor is given by two procedures* (Gen, Rep).

1. $$\xrightarrow{BioData:w} \boxed{\textsf{Gen}} \rightarrow \begin{cases} R: & \text{Random String;} \\ P: & \text{Auxiliary String.} \end{cases}$$

Gen is a probabilistic generation procedure, which on (biometric) input $w \in \mathcal{M}$ outputs an "extracted" string $R \in \{0, 1\}^\ell$ and an auxiliary string P. For any distribution W on $\mathcal{M}$ of min-entropy m, if $<R, P> \leftarrow \mathrm{Gen}(W)$, then we have $\mathbf{SD}(<R, P>, <U_\ell, P>) \le \epsilon$. Here, U_ℓ denotes the uniform distribution on ℓ-bit binary strings.

2. $$\xrightarrow[P]{BioData:w'} \boxed{\textsf{Rep}} \rightarrow R \text{ if } \mathsf{dis}(w, w') \le t.$$

Rep is a deterministic reproduction procedure allowing to recover R from the corresponding auxiliary string P and any vector w' close to w: for all $w, w' \in \mathcal{M}$ satisfying $\mathrm{dis}(w, w') \le t$, if $<R, P> \leftarrow \mathrm{Gen}(w)$, then we have $\mathrm{Rep}(w', P) = R$.

3 Challenges in Biometric Authentication

This section is devoted to a brief description of three subtle issues in biometric authentication, namely privacy issues, error tolerance, and nontrusted devices.

3.1 Privacy Issues

A trivial way to include biometric authentication in smart-card-based password authentication is to scan the biometric characteristics and store the extracted biometric data as a template in the server. During the authentication, a comparison is made between the stored data and the input biometric data. If there is a sufficient commonality, a biometric authentication is said to be successful. This method, however, will raise several security risks, especially in a multi-server environment where user privacy is a concern (e.g., in a distributed system). First, servers are not 100 percent secure. Servers with weak security protections can be broken in by attackers, who will obtain the biometric data on those servers. *Second*, servers are not 100 percent trusted. Server-A (equivalently, its curious administrator) could try to login to Server-B on behalf of their common clients, or distribute users' biometric information in the system. In either case,

[2] This is motivated by the reviewer's comment.

user privacy will be compromised, and a single-point failure on a server will downgrade the whole system's security level from three-factor authentication to two-factor authentication (since clients are likely to register the same biometric characteristics on all servers in the system).

Notice that there is a potential solution to preserve user privacy even the server has a copy of clients' biometric data. The method is called "cancellable biometrics" [22]: Biometric data can be intentionally distorted in a repeatable manner. This allows the client to generate different biometrics for different purposes and register different biometric data on different servers. Furthermore, the client can cancel his/her biometric data on the server and enroll a new one whenever necessary (e.g., if the biometric data stored on the server is compromised). However, cancellable biometrics has certain limitations [23]. To date, there are generally two methods to implement cancellable biometrics: 1) Biometric Salting and 2) Noninvertible Transforms. The former method needs an auxiliary data which must be kept secret, and it remains as a challenging work to design a noninvertible transform function satisfying both performance and noninvertibility requirements. Due to these concerns, our framework does not use cancellable biometrics.

3.2 Error Tolerance and Nontrusted Devices

One challenge in biometric authentication is that biometric characteristics are prone to various noise during data collecting, and this natural feature makes it impossible to reproduce precisely each time biometric characteristics are measured. A practical biometric authentication protocol cannot simply compare the hash or the encryption of biometric templates (which requires an exact match). Instead, biometric authentication must tolerate failures within a reasonable bound. Another issue in biometric authentication is that the verification of biometrics should be performed by the server instead of other devices, since such devices are usually remotely located from the server and cannot be fully trusted. The above two subtle issues seem to be neglected in a recent three-factor authentication protocol proposed by Li and Hwang [18]. The detailed analysis of their protocol is given in the supplementary file (Section 1), which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2010.206.

4 A Generic Framework for Three-Factor Authentication

This section describes a generic approach for three-factor authentication from a smart-card-based password authentication protocol (SCPAP, Definition 1) and a fuzzy extractor (Definition 5). The design philosophy of our approach can be found in the supplementary file (Section 2), which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2010.206, where a graphical representation (Fig. 1) is given to illustrate the three-factor authentication process.

4.1 3-Factor-Initialization

We first describe the initialization phase in the proposed framework. This phase generates a public parameter and a secret parameter for three-factor authentication. Let 2-Factor-Initialization be the initialization algorithm in the underlying SCPAP. Given a security parameter κ, the authentication server $\mathcal{S}$ in our framework runs 2-Factor-Initialization twice:

1. 2-Factor-Initialization$(\kappa) \rightarrow (PK_1, SK_1)$.
2. 2-Factor-Initialization$(\kappa) \rightarrow (PK_2, SK_2)$.

Notice that the two pairs (PK_1, SK_1) and (PK_2, SK_2) are generated in an independent manner.

The public parameter in three-factor authentication is the pair (PK_1, PK_2), and the corresponding secret parameter is the pair (SK_1, SK_2).

4.2 3-Factor-Reg

The registration in our framework is made up of the following steps. In the following, let h be a cryptographic hash function chosen by the client $\mathcal{C}$.

1. An initial password PW_1 is chosen by the client $\mathcal{C}$.
2. $\mathrm{Gen}(\mathit{BioData}) \rightarrow (R, P)$. A pair (R, P) is generated using $\mathcal{C}$'s biometric template $\mathit{BioData}$ and the algorithm Gen in the fuzzy extractor. We assume there is a device extracting the biometric template and carrying out all calculations in the fuzzy extractor. Notice that this step does not involve any interaction with the authentication server.
3. Let $PW_2 = h(R)$. The second "password" PW_2 is calculated from the random string R. R will be deleted immediately once the calculation of PW_2 is complete.
4. $$\mathcal{C}[PW_1] \xLeftrightarrow{\text{2-Factor-Reg}} \mathcal{S}[SK_1] \rightarrow \mathsf{Data}_1.$$
   $\mathcal{C}$ (using PW_1) and $\mathcal{S}$ (using SK_1) first execute the 2-Factor-Reg protocol of SCPAP. Let Data_1 be the data generated by $\mathcal{S}$ at this step.
5. $$\mathcal{C}[PW_2] \xLeftrightarrow{\text{2-Factor-Reg}} \mathcal{S}[SK_2] \rightarrow \mathsf{Data}_2.$$
   $\mathcal{C}$ and $\mathcal{S}$ have another run of 2-Factor-Reg protocol, where $\mathcal{C}$ registers PW_2 and $\mathcal{S}$ uses SK_2 to generate the corresponding data Data_2. PW_2 will be deleted immediately once the registration is complete.
6. $\mathcal{S}$ generates a smart card SC which contains Data_1 and Data_2. The client $\mathcal{C}$ is given SC.
7. $\mathcal{C}$ updates the data in the smart card SC by adding $\mathrm{Data}_3 = \{$the auxiliary string P, the description of the hash function h, the reproduction algorithm Rep$\}$.

This completes the description of the 3-Factor-Reg protocol in our framework. As in the existing authentication protocols, we assume the registration phase is performed in a secure and reliable environment, and particularly the device at Step 2 is trusted for its purpose. After a successful registration, the client $\mathcal{C}$ will have a smart card SC (contains $\{\mathrm{Data}_1, \mathrm{Data}_2, \mathrm{Data}_3\}$). The initial password is PW_1. Notice that neither the server nor the smart card has a copy of client's biometric characteristics.

4.3 3-Factor-Login-Auth

The client $\mathcal{C}$ first inserts the smart card SC into a card reader, which will extract the data $\{\mathrm{Data}_1, \mathrm{Data}_2, \mathrm{Data}_3\}$. (Recall that $\mathrm{Data}_3 = (P, h, \mathrm{Rep})$.) After that, $\mathcal{C}$ inputs the password PW_1 and his/her biometric data.[3] Let $BioData'$ be the biometric template extracted at this phase. The login is made up of the following three steps.

1. Calculate $R = \mathrm{Rep}(BioData', P)$ and $PW_2 = h(R)$. A random string R is calculated from the biometric template $BioData'$ and the auxiliary string P (which is stored in the smart card) by running the algorithm Rep. The random string R will be the same as the one generated at the registration phase if $BioData'$ is close to $BioData$. More precisely, one can obtain an identical R if $\mathrm{dis}(BioData, BioData') < t$ in an $(\mathcal{M}, m, \ell, t, \epsilon)$ fuzzy extractor (Definition 5).
2. $$\mathcal{C}[PW_1, SC(\mathsf{Data}_1)] \overset{\mathsf{2-Factor-Login-Auth}}{\Longleftrightarrow} \mathcal{S}[SK_1].$$
   $\mathcal{C}$ (using PW_1 and Data_1) and $\mathcal{S}$ (using SK_1) first execute the 2-Factor-Login-Auth protocol of SCPAP.
3. $$\mathcal{C}[PW_2, SC(\mathsf{Data}_2)] \overset{\mathsf{2-Factor-Login-Auth}}{\Longleftrightarrow} \mathcal{S}[SK_2].$$
   $\mathcal{C}$ and $\mathcal{S}$ have another run of 2-Factor-Login-Auth, where $\mathcal{C}$ uses PW_2 and Data_2, and $\mathcal{S}$ uses SK_2.

This completes the description of the 3-Factor-Login-Auth protocol in our framework. The protocol outputs "1" if and only if both executions of 2-Factor-Login-Auth protocol output "1." Otherwise, the protocol outputs "0."

Remark. In order to make the protocol clear, we separate the authentication by two steps (Step 2 and Step 3). This also shows the flexibility of our protocol: According to the criticality of the requested service, the service provider (i.e., the server $\mathcal{S}$ in our protocol) can determine the factors used in the authentication. Namely, the service provider can authenticate the client based on "smart card and password" (Step 2), "smart card and biometrics" (Step 1 and Step 3), or "smart card, password, and biometrics" (Steps 1-3). In the case of all three factors are required, client and server can carry out the authentication more efficiently. Let $\{\mathcal{M}_1, \mathcal{R}_1, \mathcal{M}_2, \mathcal{R}_2, \ldots, \mathcal{M}_n, \mathcal{R}_n\}$ be a transcript of the authentication at Step 2, and let $\{\mathcal{M}'_1, \mathcal{R}'_1, \mathcal{M}'_2, \mathcal{R}'_2, \ldots, \mathcal{M}'_n, \mathcal{R}'_n\}$ be that at Step 3.

4.4 3-Factor-Password-Changing

After a successful login (i.e., 3-Factor-Login-Auth outputs "1"), the client and the server can execute 2-Factor-Password-Changing of SCPAP to change the password PW_1 to PW'_1 and update the data in the smart card accordingly.

4.5 3-Factor-Biometrics-Changing

Similarly, one can change the biometrics used in the authentication. To do that, the client can generate a new "password" PW'_2 (determined by the new biometrics) by running Step 2-3 in the registration phase. After that, the client and the server can execute 2-Factor-Password-Changing of SCPAP to change PW_2 to PW'_2 and update the data in the smart card accordingly. As in the registration, PW'_2 will be deleted immediately once this phase is complete.

5 Scheme Analysis

This section is devoted to the analysis of the generic framework.

5.1 Security Analysis

In the supplementary file (Sections 3.1, 3.2, and 3.3), which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2010.206, we show that the generic construction satisfies all security requirements of three-factor authentication, if the underlying SCPAP satisfies Definition 2 and the fuzzy extractor satisfies Definition 5. Other security properties of the proposed framework are also investigated in the supplementary file (Section 3.4), which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2010.206.

5.2 Comparison with Previous Protocols

The purpose of this paper is to investigate a systematic approach for the design of secure three-factor authentication. Thus, like almost all generic constructions, our framework does not have advantages from the computational point of view. Nevertheless, it is still affordable for smart-card applications, due to the efficient designs of SCPAP and fuzzy extractor: There are a number of efficient SCPAPs in the literature, and fuzzy extractors can be constructed from error-correcting code and standard pairwise-independent hashing [21], both of which require only lightweight operations. In addition, the proposed framework enjoys several desirable properties of SCPAP. This saves the time and effort on the design of three-factor authentication with those properties, and more importantly avoids the confusing "broken and improved" process in the existing research on three-factor authentication.

6 Conclusion

Preserving security and privacy is a challenging issue in distributed systems. This paper makes a step forward in solving this issue by proposing a generic framework for three-factor authentication to protect services and resources from unauthorized use. The authentication is based on password, smart card, and biometrics. Our framework not only demonstrates how to obtain secure three-factor authentication from two-factor authentication, but also addresses several prominent issues of biometric authentication in distributed systems (e.g., client privacy and error tolerance). The analysis shows that the framework satisfies all security requirements on three-factor authentication and has several other practice-friendly properties (e.g., key agreement, forward security, and mutual authentication). The future work is to fully identify the practical threats on three-factor authentication and develop concrete three-factor authentication protocols with better performances.

[3] The biometric extractor is trusted to extract biometrics properly and never divulges the biometric information. This is a weaker assumption than using a fully trusted device to verify biometrics.

References

[1] D.V. Klein, "Foiling the Cracker: A Survey of, and Improvements to, Password Security," *Proc. Second USENIX Workshop Security,* 1990.

[2] *Biometrics: Personal Identification in Networked Society,* A.K. Jain, R. Bolle, and S. Pankanti, eds. Kluwer, 1999.

[3] D. Maltoni, D. Maio, A.K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition.* Springer-Verlag, 2003.

[4] Ed. Dawson, J. Lopez, J.A. Montenegro, and E. Okamoto, "BAAI: Biometric Authentication and Authorization Infrastructure," *Proc. IEEE Int'l Conf. Information Technology: Research and Education (ITRE '03),* pp. 274-278, 2004.

[5] J.K. Lee, S.R. Ryu, and K.Y. Yoo, "Fingerprint-Based Remote User Authentication Scheme Using Smart Cards," *Electronics Letters,* vol. 38, no. 12, pp. 554-555, June 2002.

[6] C.C. Chang and I.C. Lin, "Remarks on Fingerprint-Based Remote User Authentication Scheme Using Smart Cards," *ACM SIGOPS Operating Systems Rev.,* vol. 38, no. 4, pp. 91-96, Oct. 2004.

[7] C.H. Lin and Y.Y. Lai, "A Flexible Biometrics Remote User Authentication Scheme," *Computer Standards Interfaces,* vol. 27, no. 1, pp. 19-23, Nov. 2004.

[8] M.K. Khan and J. Zhang, "Improving the Security of 'A Flexible Biometrics Remote User Authentication Scheme'," *Computer Standards Interfaces,* vol. 29, no. 1, pp. 82-85, Jan. 2007.

[9] C.J. Mitchell and Q. Tang, "Security of the Lin-Lai Smart Card Based User Authentication Scheme," Technical Report RHUL-MA20051, http://www.ma.rhul.ac.uk/static/techrep/2005/RHUL-MA-2005-1.pdf, Jan. 2005.

[10] E.J. Yoon and K.Y. Yoo, "A New Efficient Fingerprint-Based Remote User Authentication Scheme for Multimedia Systems," *Proc. Ninth Int'l Conf. Knowledge-Based Intelligent Information and Eng. Systems (KES),* 2005.

[11] Y. Lee and T. Kwon, "An improved Fingerprint-Based Remote User Authentication Scheme Using Smart Cards," *Proc. Int'l Conf. Computational Science and Its Applications (ICCSA),* 2006.

[12] H.S. Kim, J.K. Lee, and K.Y. Yoo, "ID-Based Password Authentication Scheme Using Smart Cards and Fingerprints," *ACM SIGOPS Operating Systems Rev.,* vol. 37, no. 4, pp. 32-41, Oct. 2003.

[13] M. Scott, "Cryptanalysis of an ID-Based Password Authentication Scheme Using Smart Cards and Fingerprints," *ACM SIGOPS Operating Systems Rev.,* vol. 38, no. 2, pp. 73-75, Apr. 2004.

[14] A. Bhargav-Spantzel, A.C. Squicciarini, E. Bertino, S. Modi, M. Young, and S.J. Elliott, "Privacy Preserving Multi-Factor Authentication with Biometrics," *J. Computer Security,* vol. 15, no. 5, pp. 529-560, 2007.

[15] S. Goldwasser, S. Micali, and C. Rackoff, "The Knowledge Complexity of Interactive Proof-Systems," *SIAM J. Computing,* vol. 18, no. 1, pp. 186-208, Feb. 1989.

[16] U. Uludag, S. Pankanti, S. Prabhakar, and A.K. Jain, "Biometric Cryptosystems: Issues and Challenges," *Proc. IEEE,* Special Issue on Multimedia Security for Digital Rights Management, vol. 92, no. 6, pp. 948-960, June 2004.

[17] C.-I. Fan and Y.-H. Lin, "Provably Secure Remote Truly Three-Factor Authentication Scheme with Privacy Protection on Biometrics," *IEEE Trans. Information Forensics and Security,* vol. 4, no. 4, pp. 933-945, Dec. 2009.

[18] C.T. Li and M.-S. Hwang, "An Efficient Biometrics-Based Remote User Authentication Scheme Using Smart Cards," *J. Network and Computer Applications,* vol. 33, no. 1, pp. 1-5, 2010.

[19] P.C. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," *Proc. Int'l Cryptology Conf. (CRYPTO),* pp. 388-397, 1999.

[20] T.S. Messerges, E.A. Dabbish, and R.H. Sloan, "Examining Smart-Card Security under the Threat of Power Analysis Attacks," *IEEE Trans. Computers,* vol. 51, no. 5, pp. 541-552, May 2002.

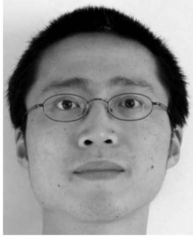
[21] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data," *Proc. Int'l Conf. Theory and Applications of Cryptographic Techniques (Eurocrypt),* pp. 523-540, 2004.

[22] N.K. Ratha, J.H. Connell, and R.M. Bolle, "Enhancing Security and Privacy in Biometrics-Based Authentication Systems," *IBM Systems J.,* vol. 40, no. 3, pp. 614-634, 2001.

[23] M.-H. Lim and A.B.J. Teoh, "Cancelable Biometrics," *Scholarpedia,* vol. 5, no. 1, p. 9201, 2010.

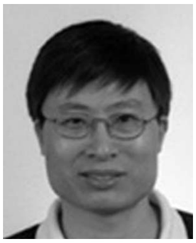
[24] H. Tian, X. Chen, and Y. Ding, "Analysis of Two Types Deniable Authentication Protocols," *Int'l J. Network Security,* vol. 9, no. 3, pp. 242-246, July 2009.

Xinyi Huang received the PhD degree in computer science (information security) in 2009, from the School of Computer Science and Software Engineering, the University of Wollongong, Australia. He is currently a post-doctoral fellow in the School of Information Systems, Singapore Management University. His research interests focus on the cryptography and its applications in information systems. He has published more than 40 referred research papers at international conferences and journals. His research results have more than 350 citations.



Yang Xiang received the PhD degree in computer science from Deakin University, Melbourne, Australia, in April 2007. He is currently with School of Information Technology, Deakin University. His research interests include network and system security, distributed systems, and wireless systems. In particular, he is currently leading in a research group developing active defense systems against large-scale network attacks and new Internet security countermeasures. He has published more than 100 research papers in international journals and conferences. He has served as program/general chair for many international conferences such as ICA3PP 11, IEEE HPCC 10/09, IEEE ICPADS 08, and NSS 10/09/08/07. He is on the editorial board of Journal of Network and Computer Applications. He is a member of the IEEE.

Ashley Chonka received the bachelor of computer science degree in 2001 and the master's of information techology (professional) degree in 2005. He also received the PhD degree from Deakin University on 5 May 2010. He has successfully published more than 20 peer-reviewed papers and is currently a lecturer at Deakin University. His research interests are in the area of Network security, MultiCore, CyberWarfare, Chaos Theory, and Honeypot systems.



Jianying Zhou received the PhD degree in information security from the University of London, in 1997. He is a senior scientist at Institute for Infocomm Research (I^2R), and heads the Network Security Group. His research interests are in computer and network security, cryptographic protocol, mobile and wireless communications security. He has published about 150 referred papers at international conferences and journals. He is actively involved in the academic community, having served in many international conference committees as general chair, program chair and PC member, having been in the editorial board and as a regular reviewer for many international journals. He is a cofounder and steering committee member of International Conference on Applied Cryptography and Network Security (ACNS).

Robert H. Deng received the bachelor's degree from National University of Defense Technology, China, the MSc and PhD degrees from the Illinois Institute of Technology. He has been with the Singapore Management University since 2004, and is currently professor, associate dean for Faculty & Research, School of Information Systems. Prior to this, he was principal scientist and manager of Infocomm Security Department, Institute for Infocomm Research, Singapore. He has 26 patents and more than 200 technical publications in international conferences and journals in the areas of computer networks, network security and information security. He has served as general chair, program committee chair, and program committee member of numerous international conferences. He is an associate editor of the *IEEE Transactions on Information Forensics and Security*, associate editor of *Security and Communication Networks Journal (John Wiley)*, and member of Editorial Board of *Journal of Computer Science and Technology (the Chinese Academy of Sciences)*. He received the University Outstanding Researcher Award from the National University of Singapore in 1999 and the Lee Kuan Yew Fellow for Research Excellence from the Singapore Management University in 2006. He was named Community Service Star and Showcased Senior Information Security Professional by $(ISC)^2$ under its Asia-Pacific Information Security Leadership Achievements program in 2010. He is a senior member of the IEEE.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.