JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 33, 907-921 (2017)
DOI: 10.6688/JISE.2017.33.4.3

A Privacy-Preserving Multi-Factor Authenticated Key Exchange Protocol with Provable Security for Cloud Computing

FU-SHAN WEI[1,2], QI JIANG[3], RUI-JIE ZHANG[1] AND CHUAN-GUI MA[4]
[1]*State Key Laboratory of Mathematical Engineering and Advanced Computing*
Zhengzhou, 450001 P.R. China
[2]*State Key Laboratory of Cryptology*
P. O. Box 5159, Beijing, 100878 P.R. China
[3]*School of Cyber Engineering*
Xidian University
Xi'an, 710071 P.R. China
[4]*Department of Basic, Army Aviation Institution*
Beijing, 101123 P.R. China
E-mail: {weifs831020; rjz_wonder}@163.com; jiangqixdu@xidian.edu.cn; chuanguima@sina.com

In the past decade, cloud computing has grown from being a promising business concept to one of the fast growing segments of the information technology's industry. More and more information of individuals and companies are stored in the cloud. However, security and privacy issues pose as the key roadblock to its fast adoption. In order to address these issues, we present an efficient MFAKE protocol based on fuzzy extractors in this paper. The novel protocol can be proven secure in the random oracle model and achieves truly multi-factor security. Compared with other related MFAKE protocols, our protocol not only enjoys stronger security, but also has greater efficiency both in terms of computation and communication. The proposed MFAKE protocol provides high-level security and protects user's privacy, so it is particular suitable for cloud computing environments.

Keywords: multi-factor key exchange, password, cloud computing, biometrics, fuzzy extractors

1. INTRODUCTION

1.1 Backgrounds

With the rapid development of the cloud computing technologies, users can conveniently enjoy the ubiquitous services via mobile devices [1-4]. However, the openness of the network and the uncontrollability of the computing environment bring serious threats to user's sensitive data and personal privacy [5]. Access control and user privacy become main concerns in cloud computing environments. The adversary is rather powerful in the cloud computing environment. Consequently, strongly secure authentication mechanism is needed under this circumstance [6-8]. Meanwhile, it is also desirable to protect user's privacy during the authentication process.

Generally speaking, user authentication is based on three types of authentication factors which are related to the user. The first type is something the user knows (typical-

Received May 31, 2016; revised July 11, 2016; accepted August 30, 2016.
Communicated by Zhe Liu.

ly a password); the second type is something the user has (*e.g.*, a smart card or a secure token that stores a private key); the third type is something the user is (*e.g.*, biometric characteristics). Unfortunately, each authentication factor has its own inherent weaknesses which cannot be solved using cryptographic techniques only [9]. In order to enhance the security of the authentication schemes, many multi-factor authenticated key exchange (MFAKE) protocols, which combine two or even more authentication factors, are proposed in recent years [10-20]. The basic idea of MFAKE protocols is that the adversary has to compromise all the authentication factors in order to breach the security of the MFAKE protocol. However, as is noted in [21], it is a non-trivial hard work to design even a secure single-factor based authenticated key exchange (AKE) protocol. Designing a truly secure MFAKE protocol can only be harder and is incredibly difficult. Great caution should be taken when designing MFAKE protocols.

1.2 Related Work

In 2006, Spantzel *et al.* [22] proposed an elegant privacy-preserving three-factor authentication scheme based on a password, a secure token and a biometric key. However, their protocol does not have rigorous security proof. In 2008, Pointcheval *et al.* [10] defined the first formal security model for MFAKE protocols. They also presented a protocol which is provably secure in the random oracle model. In 2009, Fan *et al.* [23] provided a three-factor authentication scheme which combines biometrics with passwords and smart cards. In 2010, Stebila *et al.* [24] introduced a security model for MFAKE protocols and proposed an efficient MFAKE protocol which combines passwords, one-time passwords and biometric keys. Meanwhile, Liu *et al.* [25] extended Pointcheval *et al.*'s MFAKE protocol to the three-party setting. In 2011, Huang *et al.* [26] proposed a generic framework which can combine a secure two-factor smart-card-based password authentication scheme with biometrics to produce a three-factor authentication scheme. Their main technical tool is the fuzzy extractor introduced in [27]. In 2012, Hao *et al.* [21] showed that Pointcheval *et al.*'s MFAKE protocol is vulnerable to an adversary who only compromises the password factor. In other words, the adversary only needs to compromise a single password factor in order to break down the entire system. In 2014, Yu *et al.* [28] introduced an efficient generic framework for three-factor authentication using fuzzy vaults. Their generic framework can be viewed as an improvement of Huang *et al.*'s work [26]. Huang *et al.* [29] presented a robust generic multi-factor authentication protocol for fragile communications. They also put forward the notion of stand-alone authentication which is of independent interest. He *et al.* [30] proposed a MFAKE protocol for multi-server environment using elliptic curve cryptography. Their protocol is the first three-factor authenticated scheme for multi-server environment. Recently, Fleischhacker *et al.* [31] propose a modular framework for MFAKE protocols by mixing multiple types and quantities of authentication factors in a secure way.

1.3 Motivation and Contribution

Most of the existing MFAKE protocols are based on passwords, smart cards and biometric keys, Pointcheval *et al.*'s protocol [10] is the only provably secure MFAKE protocol which combines a password, a high-entropy cryptographic key, and a biometric key. Their MFAKE protocol is very famous and lays the basis for many MFAKE proto-

cols. Unfortunately, Pointcheval *et al.*'s MFAKE protocol has the following disadvantages. First of all, Hao *et al.* [21] showed Pointcheval *et al.*'s MFAKE protocol is insecure against an adversary that only corrupts a user's password factor. The adversary can impersonate the server with the user's password and compromise all other authentication factors. Hao *et al.* found that the attack is a fundamental problem of the protocol, which seems not easily fixable within the current structural design of the protocol. Secondly, Pointcheval *et al.*'s protocol only considers unilateral client-authentication and does not achieve mutual authentication. Lack of mutual authentication opens the door for Hao *et al.*'s attack. Thirdly, Pointcheval *et al.* use the ElGamal encryption scheme to encrypt each bit of user's biometric template. Usually, user's biometric template is encoded on a 1024-bit string. As a result, their protocol is very inefficient in terms of both computation and communication. Last but not least, Pointcheval *et al.*'s protocol needs 4 rounds though it does not provide mutual authentication. However, mutual authentication actually can be achieved by 3 rounds communications.

In order to overcome the disadvantages of Pointcheval *et al.*'s MFAKE protocol, we propose a novel three-factor authenticated key exchange protocol using fuzzy extractors. In our proposal, the user and the server run a two-party EKE protocol [32] to exchange the Diffie-Hellman key materials. The session key is computed by using a special version of the HMQV protocol [33] in which a party wants to establish a secure channel with itself. The session key is established by combining the password factor and the biometric factor. The public-private key pairs are used to sign a key confirmation value of the session key. In this way, the signature can only be verified by someone who knows the session key. This trick strengthens the security of the protocol. Compared with Pointcheval *et al.*'s MFAKE protocol, our protocol provides truly three-factor security and can resist Hao *et al.*'s attack. Most notably, our protocol is very efficient in terms of computation and communication. Consequently, our protocol not only enjoys stronger security, but also has higher efficiency.

The rest of the paper is organized as follows. We recall the security model for MFAKE protocol in Section 2. In Section 3, we present the proposed MFAKE protocol. We compare the efficiency and security features of our protocol with Pointcheval *et al.*'s MFAKE protocol in Section 4. We conclude our paper in Section 5. The security proof of the proposed protocol is conducted in the random oracle model in Appendix.

2. SECURITY MODEL

In this subsection, we extend the security model for MFAKE protocol proposed in [10].

Protocol participants Participants in an MFAKE protocol are either clients C or a unique authentication server S. An instance of participant U in session i is denoted as $\prod_U^i$. We define the session identity sid_U^i as the transcript of all messages sent and received by the instance $\prod_U^i$, except for the last protocol message. By pid_U^i we denote the partner identity with which the instance $\prod_U^i$ is interacting in the protocol session. We also define a Boolean variable acc_U^i which is determined at the end of the session and denotes whether the instance $\prod_U^i$ accepts the session or not.

Partnering The two instances $\prod_U^i$ and $\prod_U^j$ are said to be partners if the following conditions are fulfilled: (1) $acc_U^i = acc_U^j = 1$; (2) $pid_U^i = U'$ and $pid_{U'}^j = U$; (3) $sid_U^i = sid_{U'}^j \neq null$.

Long-lived keys Each client C has a tuple $t_C = (\mathcal{W}_C, sk_C, pw_C)$, where $\mathcal{W}_C$ is a probability distribution for his biometric, while sk_C and pw_C are a high-entropy cryptographic key and a low-entropy password respectively. The authentication server holds a list of tuples $t_S = \langle t_S[C] \rangle$ with an entry for each client, where $t_S[C]$ is a transformed-tuple of t_C. We also assume the authentication server has its own high-entropy cryptographic key sk_S with the corresponding system-wide known public key pk_S.

Adversarial queries The adversary $\mathcal{A}$ is modeled as a probabilistic polynomial time (PPT) adversary. The abilities of $\mathcal{A}$ are modeled through oracle queries:

Execute($\prod_C^i$, $\prod_S^j$): This query models passive eavesdropping of a protocol execution between a client instance $\prod_C^i$ and a server instance $\prod_S^j$. At the end of the execution, a transcript is given to the adversary $\mathcal{A}$, which logs everything an adversary could see during the execution.

Send($\prod_U^i$, m): This query models an active attack against instance $\prod_U^i$. The adversary $\mathcal{A}$ sends message m to the protocol instance $\prod_U^i$, $\prod_U^i$ executes as specified by the protocol and sends back its response to the adversary $\mathcal{A}$.

Reveal($\prod_U^i$): This query causes the output of session key held by the instance $\prod_U^i$. This query allows the adversary $\mathcal{A}$ learn past session keys from previous executions, modeling the known key attack.

Corrupt(S): This query allows the adversary $\mathcal{A}$ learn the server's secret key sk_S and the list of tuples $t_S = \langle t_S[C] \rangle$. The corrupted server S is under the control of the adversary $\mathcal{A}$.

Corrupt(C): there are three different corruption queries to the client:
- *Corrupt*(C, sk_C, pw_C): This query enables the adversary to get the client C's password pw_C and secret key sk_C;
- *Corrupt*(C, pw_C, W_C): This query enables the adversary to get the client C's password pw_C and the biometric template W_C;
- *Corrupt*(C, sk_C, W_C): This query enables the adversary to get the client C's secret key sk_C and the biometric template W_C;

Test($\prod_U^i$): This query does not captures $\mathcal{A}$'s real attack ability. It measures the session key security of instance $\prod_U^i$. The simulator flips a coin, if the random bit is $b = 1$, then he sends the real session key to the $\mathcal{A}$. Otherwise, the simulator sends a random key of the same size to $\mathcal{A}$. $\mathcal{A}$ will guess the value of b to win the game.

Freshness The freshness notion captures the intuitive fact that a session key is not

trivially known to the adversary $\mathcal{A}$. A client C is fully corrupted if and only if all the authentication factors of C have been corrupted. For example, a client C is fully corrupted if an adversary asks more than one corruption query to the client. A server S is fully corrupted if and only if a *Corrupt*(S) query has been asked. We say that the session key of an instance $\prod_U^i$ is fresh if the following conditions hold: (1) Upon acceptance of the instance $\prod_U^i$, neither the participant U nor its intended partner is fully corrupted; (2) No Reveal query has been sent to either $\prod_U^i$ or its partner (if such instance exists).

Session key security Suppose $\mathcal{P}$ is an MFAKE protocol and $\mathcal{A}$ is a probabilistic polynomial time (PPT) adversary against session key security of $\mathcal{P}$. If the adversary can correctly guess the random bit b used by the *Test* oracle, then we say the adversary succeeds in this attack game. We denote this event by *Succ*.

Definition 5: The advantage of an adversary $\mathcal{A}$ in violating the MFAKE semantic security of the protocol $\mathcal{P}$, is defined as $Adv_{\mathcal{P},\mathcal{D}}^{mfake}(\mathcal{A}) = 2 \cdot \Pr[Succ] - 1$. The advantage function of the protocol $\mathcal{P}$ is defined as $Adv_{\mathcal{P},\mathcal{D}}^{mfake}(t, R) = \max\{Adv_{\mathcal{P},\mathcal{D}}^{mfake}(\mathcal{A})\}$, where maximum is over all $\mathcal{A}$ with time-complexity at most t and using resources at most R (such as the number of oracle queries). A MFAKE protocol $\mathcal{P}$ is said to be semantically secure if the advantage $Adv_{\mathcal{P},\mathcal{D}}^{mfake}(t, R)$ is only negligibly larger than $kq_{send}/|\mathcal{D}|$, where q_{send} is the number of *Send* queries, $\mathcal{D}$is the dictionary space and k is a constant.

Authentication security Authentication security ensures that an adversary should not be able to impersonate a participant unless the participant is fully corrupted. We say an adversary $\mathcal{A}$ breaks client authentication if there exists a server instance $\prod_S^i$ that has accepted with partner identity C, but there exists no client instance $\prod_C^j$ that is partnered with $\prod_S^i$ and the client C is not fully corrupted. Let $Succ_{CAuth}^{mfake}(\mathcal{A})$ denote the adversary $\mathcal{A}$'s success probability in breaking client authentication.

Definition 6: A MFAKE protocol achieves client authentication security if for all PPT adversary $\mathcal{A}$, the success probability $Succ_{CAuth}^{mfake}(\mathcal{A})$ is only negligibly larger than $kq_{send}/|\mathcal{D}|$, where q_{send} is the number of *Send* queries, $\mathcal{D}$ is the dictionary space and k is a constant.

Likewise, we can define server authentication security. We omit it for simplicity. We say a MFAKE protocol provide mutual authentication if it achieves both client authentication security and server authentication security.

3. DESCRIPTION OF THE PROTOCOL

In this section, we propose an efficient multi-factor authenticated key exchange protocol using fuzzy extractors. Our protocol is based on a cyclic group with parameters (p, q, g). p and q are two big primes such that $q|p - 1$. Let G_q be a subgroup in Z_p^* with prime order q, and g be a generator of G_q. Let u and v be two random elements in G_q. In the registration phase, a client C chooses his password pw_C from the dictionary space $\mathcal{D}$. C also imprints his biometric template W_C and computes $Gen(W_C) = (R_C, P_C)$ using the generation algorithm of the fuzzy extractor, where R_C is a private string and P_C is a public string. R_C can be treated as a biometric key. Without loss of generality, we assume

pw_C, $R_C \in Z_p^*$. C sends (pw_C, R_C, P_C) to the server S through a secure channel. C has his private-public key pair (sk_C, pk_C) for signature and S also has its private-public key pair (sk_S, pk_S) for signature. For simplicity, we assume S stores all clients' public keys in its database and the public key of the server is known system-wide. Thus, we can omit public key credential exchange and verification in the key exchange phase.

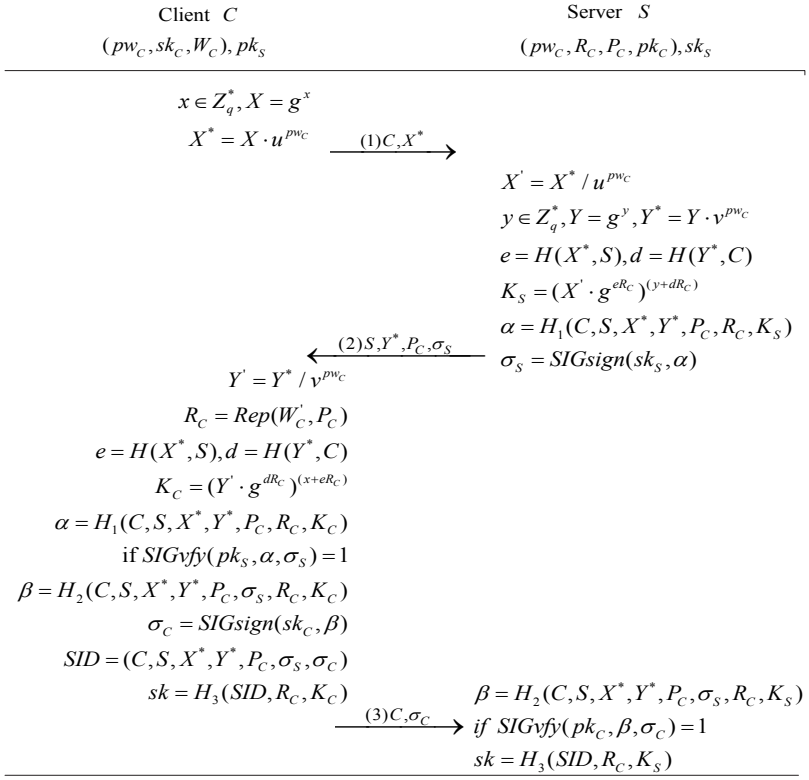

Fig. 1. Our multi-factor authenticated key exchange protocol.

Detailed steps of the key exchange phase of the proposed protocol, as shown in Fig. 1, are described as follows:

Round 1: The client C randomly chooses $x \in Z_p^*$, compute $X = g^x$ and $X^* = X \cdot u^{pw_C}$. Finally, C sends (C, X^*) to the server S.

Round 2: Upon receiving the message (C, X^*), S chooses a random number $y \in Z_p^*$, then computes $Y = g^y$, $Y^* = Y \cdot v^{pw_C}$ and $X' = X^*/u^{pw_C}$. S also computes a secret value $K_S = (X \cdot g^{eR_C})^{(y+dR_C)}$, where $e = H(X^*, S)$ and $d = H(Y^*, C)$. Next, S computes $\alpha = H_1(C, S, X^*, Y^*, P_C, R_C, K_S)$ and signs α using his private key sk_S to generate the signature σ_S. Finally, S sends the message (S, Y^*, P_C, σ_S) back to the client C.

Round 3: Upon receiving the message (S, Y^*, P_C, σ_S), C imprints another biometric tem-

plate W'_C and computes $R_C = Rep(W'_C, P_C)$ using the reproduction algorithm of the fuzzy extractor. C also computes the secret value $K_C = (Y' \cdot g^{dR_C})^{(x+eR_C)}$, where $e = H(X^*, S)$ and $d = H(Y^*, C)$. C computes $\alpha = H_1(C, S, X^*, Y^*, P_C, R_C, K_C)$ and verifies the validity of the signature σ_S. If it is valid, then C computes $\beta = H_2(C, S, X^*, Y^*, P_C, \sigma_S, R_C, K_C)$ and signs β using his private key sk_C to generate the signature σ_C. Finally, C computes the common session key $sk = H_3(C, S, X^*, Y^*, P_C, \sigma_S, \sigma_C, R_C, K_C)$, sends the message (C, σ_C) to S and accepts the session.

Upon receiving the message (C, σ_C), S computes $\beta = H_2(C, S, X^*, Y^*, P_C, \sigma_S, R_C, K_C)$ and verifies the validity of the signature σ_C. If the verification is successful, then S also computes the session key $sk = H_3(C, S, X^*, Y^*, P_C, \sigma_S, \sigma_C, R_C, K_S)$ and accepts the session.

4. PERFORMANCE ANALYSIS

To the best of knowledge, Pointcheval *et al.*'s MFAKE protocol [10] is the only MFAKE protocol in the literature which combines passwords, high-entropy keys and biometrics. So we compare the efficiency and security features of the proposed protocol with Pointcheval *et al.*'s MFAKE protocol in this section.

With respect to computation, we only consider the number of modular exponentiations since these are the most expensive type of computation. Note that the product of two modular exponentiations in the formulation of $g^x h^y$ can be computed at the cost of a single modular exponentiation. The computation cost of the fuzzy extractor [27] is no more than the cost of several hash function operations, so we omit it for simplicity. We instantiate the signature scheme used in our protocol using the well-known Schnorr signature scheme [34]. In terms of communication, the length of the identifications is assumed to be 32 bits, an element in G_q is assumed to be 160 bits, the output size of secure hash functions is 160 bits and the length of the helping data P_C in the fuzzy extractor is about 62464 bits [35]. The length of the biometric template is 1024 bits and the length of the authentication tag is assumed to be 4 bits in Pointcheval *et al.*'s MFAKE protocol. The comparison of computation cost and communication cost of the proposed protocol and Pointcheval *et al.*'s protocol is presented in Table 1. We can see from Table 1 that our protocol is very efficient both in terms of computation and communication. Particularly, the computation cost of protocol is negligible compared with that of Pointcheval *et al.*'s protocol.

Table 2 summarizes security features of the proposed protocol compared with Pointcheval *et al.*'s protocol [10]. We can see from Table 2 that our protocol provides more security features than Pointcheval *et al.*'s protocol does. Their protocol does not provide mutual authentication. Due to Hao *et al.*'s attack [21], their protocol does not achieve three-factor security either. It is claimed that biometric privacy is preserved in Pointcheval *et al.*'s protocol because the server only stores an ElGamal encryption of the reference biometric template. Unfortunately, Hao *et al.*'s attack showed that the server is able to discover the biometric template with the password factor. Moreover, our protocol is secure against key compromise impersonation (KCI) attack. The only disadvantage of our protocol is that we assume the biometric template is private. However, the biometric template is assumed to be public in Pointcheval *et al.*'s protocol, which is more general and realistic.

Table 1. Comparisons of efficiency.

Protocols	Computation cost of the client	Computation cost of the server	Bandwidth	Message rounds
PZ08[10]	1026E	2050E	172416bits	4
Our protocol	4E	4E	63520bits	3

Table 2. Comparisons of security features.

Protocols	Security Proof	Hardness Assumption	Three-Factor Security	Mutual Authentication	KCI Security	Biometrics Privacy	Biometrics Assumption
PZ08 [10]	Y	CDH	N	N	N	N	Public
Our protocol	Y	CDH	Y	Y	Y	Y	Private

5. CONCLUSIONS

In this paper, we design a simple and privacy-preserving MFAKE protocol using fuzzy extractors. The proposed protocol is proven secure in the random oracle model based on the hardness of the CDH problem. Security and performance comparisons show that our protocol achieves both higher efficiency and stronger security. To the best of our knowledge, this is the most efficient MFAKE protocol with provable security which combines passwords, high-entropy cryptographic keys and biometric templates.

ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China (Nos. 61309016, 61379150, 61501515, 61672413), Postdoctoral Science Foundation of China (No. 2014M562493), Postdoctoral Science Foundation of Shanxi Province, Natural Science Basic Research Plan in Shaanxi Province of China (Program No. 2016JM6005), Fundamental Research Funds for the Central Universities (Program No. JB161501).

REFERENCES

1. Y. Ren, J. Shen, J. Wang, *et al.*, "Mutual verifiable provable data auditing in public cloud storage," *Journal of Internet Technology*, Vol. 2, 2015, pp. 317-323.
2. F. Zhangjie, S. Xingming, L. Qi, *et al.*, "Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing," *IEICE Transactions on Communications*, Vol. 1, 2015, pp. 190-200.
3. Z. Xia, X. Wang, X. Sun, *et al.*, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 2, 2016, pp. 340-352.
4. F. Zhangjie, R. Kui, S. Jiangang, and S. Xingming, "Enabling personalized search over encrypted outsourced data with efficiency improvement," *IEEE Transactions on Parallel and Distributed Systems*, 2015, DOI: 10.1109/TPDS.2015.2506573.

5. Z. Liu, X. Huang, Z. Hu, *et al.*, "On emerging family of elliptic curves to secure internet of things: ECC comes of age," *IEEE Transactions on Dependable and Secure Computing*, Vol. 14, 2016, pp. 237-248.
6. P. Guo, J. Wang, B. Li, and S. Lee, "A variable threshold-value authentication architecture for wireless mesh networks," *Journal of Internet Technology*, Vol. 15, 2014, pp. 929-936.
7. J. Shen, H. Tan, J. Wang, *et al.*, "A novel routing protocol providing good transmission reliability in underwater sensor networks," *Journal of Internet Technology*, Vol. 16, 2015, pp. 171-178.
8. Q. Jiang, M. Jianfeng, L. Guangsong, *et al.*, "Improvement of robust smartcard-based password authentication scheme," *International Journal of Communication Systems*, Vol. 2, 2015, pp. 383-393.
9. Z. Liu, H. Seo, *et al.*, "Efficient implementation of NIST-compliant elliptic curve cryptography for 8-bit AVR-based sensor nodes," *IEEE Transactions on Information Forensics and Security*, Vol. 7, 2016, pp. 1385-1397.
10. D. Pointcheval and S. Zimmer, "Multi-factor authenticated key exchange," in *Proceedings of Applied Cryptography and Network Security*, 2008, pp. 277-295.
11. S. Kumari, X. Li, F. Wu, *et al.*, "A user friendly mutual authentication and key agreement scheme for wireless sensor networks using chaotic maps," *Future Generation Computer Systems*, Vol. 63, 2016, pp. 56-75.
12. D. Wang, D. He, P. Wang, *et al.*, "Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment," *IEEE Transactions on Dependable and Secure Computing*, Vol. 12, 2015, pp. 428-442.
13. S. Kumari, M. Khan, and M. Atiquzzaman, "User authentication schemes for wireless sensor networks: a review," *Ad Hoc Networks*, Vol. 27, 2015, pp. 159-194.
14. Q. Jiang, M. Jianfeng, and W. Fushan, "On the security of a privacy-aware authentication scheme for distributed mobile cloud computing services," *IEEE Systems Journal*, 2016, DOI: 10.1109/JSYST.2016.2574719.
15. D. He, N. Kymar, H. Shen, *et al.*, "One-to-many authentication for access control in mobile pay-TV systems," *Science China-Information Sciences*, Vol. 59, 2016, pp. 1-14.
16. Q. Jiang, W. Fushan, M. Jianfeng, *et al.*, "Robust extended chaotic maps-based three-factor authentication scheme preserving biometric template privacy," *Nonlinear Dynamics*, Vol. 83, 2016, pp. 2085-2101.
17. J. Shen, H. Tan, S. Moh, *et al.*, "Enhanced secure sensor association and key management in wireless body area networks," *Journal of Communications and Networks*, Vol. 17, 2015, pp. 453-462.
18. D. He, N. Kymar, and N. Chilamkurti, "A secure temporal-credential based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks," *Information Sciences*, Vol. 321, 2015, pp. 263-277.
19. S. Kumari, M. Khan, and X. Li, "Design of a user anonymous password authentication scheme without smart card," *International Journal of Communication Systems*, Vol. 29, 2016, pp. 441-458.
20. D. He, S. Zeadally, B. Xu, *et al.*, "An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad-hoc networks," *IEEE Transactions on Information Forensics and Security*, Vol. 10, 2015, pp. 1681-2691.

21. F. Hao and D. Clarke, "Security analysis of a multi-factor authenticated key exchange protocol," in *Proceedings of the 10th International Conference on Applied Cryptography and Network Security*, 2012, pp. 1-11.
22. B. Spantzel, A. Squicciarini, *et al.*, "Privacy preserving multifactor authentication with biometrics," *Journal of Computer Security*, Vol. 15, 2007, pp. 529-560.
23. C. Fan and Y. Lin, "Provably secure remote truly three-factor authentication scheme with privacy protection on biometrics," *IEEE Transactions on Information Forensics and Security*, Vol. 4, 2009, pp. 933-945.
24. D. Stebila, P. Udupi, and S. Chang, "Multi-factor password-authenticated key exchange," in *Proceedings of the 8th Australasian Conference on Information Security*, 2010, pp. 56-66.
25. Y. Liu, F. Wei, and C. Ma, "Multi-factor authenticated key exchange protocol in the three-party setting," in *Proceedings of International Conference on Information Security and Cryptology*, 2011, pp. 255-267.
26. X. Huang, Y. Xiang, A. Chonka, *et al.*, "A generic framework for three-factor authentication: preserving security and privacy in distributed systems," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 22, 2011, pp. 1390-1397.
27. Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Proceedings of International Conference on Cryptology-Eurocrypt*, 2004, pp. 523-540.
28. J. Yu, G. Wang, Y. Mu, *et al.*, "An efficient generic framework for three-factor authentication with provably secure instantiation," *IEEE Transactions on Information Forensics and Security*, Vol. 9, 2014, pp. 2302-2313.
29. X. Huang, Y. Xiang, E. Bertino, *et al.*, "Robust multi-factor authentication for fragile communications," *IEEE Transactions on Dependable and Secure Computing*, Vol. 11, 2014, pp. 568-581.
30. D. He and D. Wang, "Robust biometrics-based authentication scheme for multi-server environment," *IEEE Systems Journal*, Vol. 9, 2015, pp. 816-823.
31. N. Fleischhacker, M. Manulis, and A. Azodi, "A modular framework for multi-factor authentication and key exchange," in *Proceedings of International Conference on Security Standardization Research*, 2014, pp. 190-214.
32. M. Abdalla and D. Pointcheval, "Simple password-based encrypted key exchange protocols," in *Proceedings of International Conference on CT-RSA*, 2005, pp. 191-208.
33. H. Krawczyk, "A high-performance secure Diffie-Hellman protocol," in *Proceedings of International Conference on CRYPTO*, 2005, pp. 546-566.
34. C. Schnorr, "Efficient signature generation by smart cards," *Journal of Cryptology*, Vol. 4, 1991, pp. 161-174.
35. R. Marino and F. Alvarez, "A crypto-biometric scheme based on iris-templates with fuzzy extractors," *Information Sciences*, Vol. 195, 2012, pp. 91-102.

APPENDIX: SECURITY PROOF

In this section, we present the security proof of our protocol within the security model given in section 2.

Theorem 1: Let $\mathcal{P}$ be the MFAKE protocol in Fig. 1. Assume the signature scheme used in our protocol is existentially unforgeable against adaptive chosen message attacks and the fuzzy extractor in our protocol is an (m, l, t, ε) fuzzy extractor. Let $\mathcal{A}$ be an adversary makes q_{send}, $q_{send} \leq |\mathcal{D}|$, queries of type *Send* to different instances. Then under the CDH assumption, the adversary's advantage in attacking the semantic security of the proposed protocol is bounded by

$$Adv_{\mathcal{P},\mathcal{D}}^{ake}(\mathcal{A}) \leq \frac{q_{send}}{|\mathcal{D}|} + negl(l).$$

***Proof*:** For each experiment Exp_i, we define the following events:

S_i^{ake} (for session key security): this event occurs if the adversary correctly guesses the hidden bit b used by the *Test* oracle;

A_i^{CAuth} (for client authentication): this event occurs if the adversary impersonates a client who is not fully corrupted and makes a server instance accepts a session with the impersonated client, but there is no partner instance of the client.

A_i^{SAuth} (for server authentication): this event occurs if the adversary impersonates the server which is uncorrupted and makes a client instance accepts a session with the server, but there is no partner server instance.

We also denote by Δ_i the distance between experiments Exp_i and Exp_{i+1}.

Experiment Exp_0 This is the real attack experiment against the MFAKE protocol. According to the definitions, we have:

$$Adv_{\mathcal{P}}^{mfake}(\mathcal{A}) = 2 \cdot Pr[S_0^{ake}] - 1,$$
$$Succ_{CAuth}^{MFAKE}(\mathcal{A}) = Pr[A_0^{CAuth}],$$
$$Succ_{SAuth}^{MFAKE}(\mathcal{A}) = Pr[A_0^{SAuth}].$$

Experiment Exp_1 We simulate the random oracles $H_i (i = 1, 2, 3)$ in this experiment as usual by maintaining hash lists $\wedge_{H_i} (i = 1, 2, 3)$. We additionally simulate three private random oracles $H_i' (i = 1, 2, 3)$ which will be used in later experiments by maintaining hash lists $\wedge_{H_i'} (i = 1, 2, 3)$. We also simulate all the instances as the real players would do. From this simulation, we see that this experiment is indistinguishable from the previous experiment. Thus we have:

$$\Delta_0 = 0$$

Experiment Exp_2 For an easier analysis, we cancel the sessions in which some unlikely collision appear in this experiment. We first cancel the sessions in which some collision on the session transcripts $((C, X^*), (S, Y^*, P_C, \sigma_S))$ occur, and we also cancel the sessions in which some collision appear on the output of the hash functions. Since transcripts involve at least one honest party and hash functions are modeled as random oracles, thus we have:

$$\Delta_1 \le \frac{(q_{exe} + q_{send})^2}{2q} + \frac{3q_h^2}{2^{l-1}}.$$

Experiment *Exp*$_3$ In this experiment, we begin to modify the simulation rules for passive sessions. It should be noted that if the adversary simply forwards the messages it receives from the oracle instances, such sessions are viewed as passive sessions although the adversary use *Send* queries in these sessions. For passive sessions in which one of the participants is fully corrupted by the adversary, we compute the authenticators α, β and the session key *sk* using the corresponding private random oracles $H_i'(i = 1, 2, 3)$ simulated in Exp_1. More precisely, we compute $\alpha = H_1(C, S, X^*, Y^*, P_C, R_C, K_C)$, $\beta = H_2(C, S, X^*, Y^*, P_C, \sigma_S, R_C)$ and $sk = H_3'(C, S, X^*, Y^*, P_C, \sigma_S, \sigma_C, R_C)$ without using the Diffie-Hellman key K_C/K_S. This experiment actually deals with forward security.

The experiments Exp_3 and Exp_2 are indistinguishable unless $\mathcal{A}$ queries the hash function $H_i(i = 1, 2, 3)$ using the messages $(C, S, X^*, Y^*, P_C, R_C, K_S)$, $(C, S, X^*, Y^*, P_C, \sigma_S, R_C, K_C)$ and $(C, S, X^*, Y^*, P_C, \sigma_S, \sigma_C, R_C, K_C)$, respectively. We denote such an event by *PassiveAskH*. To upper bound the probability of this event, we consider an auxiliary experiment Exp_3'. The simulation of the participants changes in the experiment Exp_3', but the distributions remain the same. Let us be given an instance of the CDH problem (U, V). We simulate the above-mentioned passive sessions using the self-reducibility of the CDH problem. To simulate these passive sessions, we first choose a_0, a_1, b_0, b_1 from Z_p^*. We compute $X^* = U^{a_0}g^{a_1} \cdot u^{pw_C}$ and $Y^* = V^{b_0}g^{b_1} \cdot v^{pw_C}$. All other simulation is the same as the one in Exp_3, which means we use private hash oracles to compute α, β and *sk* in these sessions. If the event *PassiveAskH* occurs, we can extract

$$\begin{aligned} &CDH(U^{a_0}g^{a_1}, V^{b_0}g^{b_1}) \\ &= CDH(U^{a_0}, V^{b_0}) \cdot CDH(U^{a_0}, g^{b_1}) \cdot CDH(g^{a_1}, V^{b_0}) \cdot CDH(g^{a_1}, g^{b_1}) \\ &= CDH(U, V)^{a_0b_0} \cdot U^{a_0b_1} \cdot V^{a_1b_0} \cdot g^{a_1b_1} \end{aligned}$$

from $\wedge_{H_i}(i = 1, 2, 3)$. If an adversary can distinguish between the experiments Exp_2 and Exp_3, then we can solve the CDH problem with exactly the same advantage. Hence we have, thus we have:

$$\Delta_2 \le q_h \cdot Adv_{\mathbb{G}}^{cdh}(O(t)).$$

Experiment *Exp*$_4$ In this experiment, we again modify the simulation rules for passive sessions. For passive sessions in which the password factor of the client is uncorrupted by the adversary, we simply choose X^* and Y^* randomly from G without using the password. Moreover, we compute the authenticators α, β and the session key *sk* of these sessions using the private random oracles $H_i'(i = 1, 2, 3)$ as we did in the previous experiment. With a same analysis with the previous experiment, we have

$$\Delta_3 \le q_h \cdot Adv_{\mathbb{G}}^{cdh}(O(t)).$$

Experiment *Exp*$_5$ In this experiment, we begin to modify the simulation rules for *Send* queries. If the adversary asks $Corruption(C_i, pw_{C_i}, W_{C_i})$ query to a client C_i, we change

the simulation rules of the server with respect to client C_i. When the server receives a *Send*(S, (C_i, X^*)) query, it sends back the message (S, X^*, P_{C_i}, σ_S) according to the description of the protocol. However, when the adversary sends back a *Send*(S, (C_i, σ_{C_i})) to the server, the server simply rejects and terminates the session without verifying the validity of the signature σ_{C_i}. The experiments Exp_4 and Exp_5 are undistinguishable unless the adversary successfully forges a valid signature σ_{C_i} without the knowledge of sk_{C_i}. If the adversary forges a valid signature σ_{C_i} on the value β, which means it breaks the existentially unforgeable security of the signature scheme. Thus, we have

$$\Delta_4 \le Adv_{sig}(O(t)).$$

Experiment Exp_6 In this experiment, we again modify the simulation rules for *Send* queries. If the adversary asks *Corruption*(C_i, pw_{C_i}, sk_{C_i}) query to a client C_i, we change the simulation rules of the server with respect to client C_i. More precisely, when the server receives a *Send*(S, (C_i, X^*)) query from the adversary, it randomly chooses Y^* from G_q without using the password. Moreover, the server computes $\alpha = H'_1(C, S, X^*, Y^*, P_{C_i})$ using the private hash function H'_1 without using the biometric key R_{C_i} and the Diffie-Hellman value K_S. The server generates a signature σ_S of the value α and sends the message (S, Y^*, P_{C_i}, σ_S) to the adversary. When the adversary sends back a *Send*(S, (C_i, σ_{C_i})) query to the server, the latter simply rejects and terminates the session without verifying the validity of the signature σ_{C_i}. The experiments Exp_5 and Exp_6 are undistinguishable unless the adversary queries the hash function H_1 on message (C_i, S, X^*, Y^*, P_{C_i}, R_{C_i}, K_{C_i}) or queries the hash function H_2 on message (C_i, S, X^*, Y^*, P_{C_i}, σ_S, R_{C_i}, K_{C_i}), where $K_{C_i} = CDH(X^* / u_{pw_{C_i}} g^{eR_{C_i}}, Y^* / v_{pw_{C_i}} g^{dR_{C_i}})$. If the adversary makes such queries to hash functions, we can recover the secret biometric key R_{C_i} from the hash lists with the help of the adversary, which violates the security of the fuzzy extractor. Thus, we have

$$\Delta_5 \le \frac{\varepsilon}{q_H}.$$

Experiment Exp_7 In this experiment, we continue to modify the simulation rules for *Send* queries. If the adversary asks *Corruption*(C_i, W_{C_i}, sk_{C_i}) query to a client C_i, we change the simulation rules of the server with respect to client C_i. More precisely, when the server receives a *Send*(S, (C_i, X^*)) query from the adversary, it randomly chooses Y^* from G_q without using the password. Moreover, the server computes $\alpha = H'_1(C, S, X^*, Y^*, P_{C_i}, R_{C_i})$ using the private hash function H'_1 without computing the Diffie-Hellman value K_S. The server generates a signature σ_S of the value α and sends the message (S, Y^*, P_{C_i}, σ_S) to the adversary. When the adversary sends back a *Send*(S, (C_i, σ_{C_i})) query to the server, the server simply rejects and terminates the session without verifying the validity of the signature σ_{C_i}. The experiments Exp_6 and Exp_7 are undistinguishable unless the adversary queries the hash function H_1 on message (C_i, S, X^*, Y^*, P_{C_i}, R_{C_i}, K_{C_i}) or queries the hash function H_2 on message (C_i, S, X^*, Y^*, P_{C_i}, σ_S, R_{C_i}, K_{C_i}), in which $K_{C_i} = CDH(X^* / u_{pw_{C_i}} g^{eR_{C_i}}, Y^* / v_{pw_{C_i}} g^{dR_{C_i}})$. We denote this bad event by *ActiveAskH*. Thus, we have

$$\Delta_6 \le \Pr[ActiveAskH_7].$$

Experiment Exp_8 In this experiment, we again modify the simulation rules for *Send* queries. If the adversary fully corrupts a client C_i, we change the simulation rules of the client C_i. More precisely, when C_i receives a *Send*(C_i, *Start*) query, he responds according to the description of the protocol and sends (C_i, X^*) to the adversary. Moreover, when C_i receives a *Send*(C_i, (S, Y^*, P_{C_i}, σ_{C_i})) query, it simply rejects and terminates the session without verifying the validity of σ_S. It should be noted that the server is uncorrupted. As a result, sk_S is unknown to the adversary. The experiments Exp_7 and Exp_8 are undistinguishable unless the adversary forges a valid signature σ_S. With a similar analysis with Exp_5, we have

$$\Delta_7 \le Adv_{sig}(O(t)).$$

Experiment Exp_9 In this experiment, we modify the simulation rules for *Send* queries for the last time. If the adversary corrupts the server, we change the simulation rules of the server. More precisely, when S receives a *Send*(S, (C_i, X^*)) query, it responds according to the description of the protocol and sends (S, Y^*, P_{C_i}, σ_S) to the adversary. Moreover, when S receives a *Send*(S, (C_i, σ_{C_i})) query, it simply rejects and terminates the session without verifying the validity of σ_{C_i}. It should be noted that sk_{C_i} is unknown to the adversary. The experiments Exp_8 and Exp_9 are undistinguishable unless the adversary forges a valid signature σ_{C_i}. With a similar analysis with Exp_5, we have

$$\Delta_8 \le Adv_{sig}(O(t)).$$

In this final experiment, only the passive sessions accept and generate session keys. These session keys are all randomly chosen. All the active sessions (except for the session in which the adversary simply relays the messages) are terminated without accepting. Consequently, the adversary has no advantage in distinguishing the session key and break the mutual authentication. Note that if the password is uncorrupted, it is never used in the simulation. The event $ActiveAskHC_9$ corresponds to an attack in which the adversary tries to impersonate the client C_i to the server S without the password. Since the authenticator β sent by the adversary has been computed with at most one pw_{C_i} value. Thus, we have

$$Pr[ActiveAskH_9] \le \frac{q_{send}}{|\mathcal{D}|}.$$

Combining all the above equations together, we get the announced result.

Fu-Shan Wei (魏福山) received the Ph.D. degrees in the Zhengzhou Information Science and Technology Institute, China. He is currently a Lecturer in Zhengzhou Information Science and Technology Institute. His current research interest includes gateway protocol and code analysis.

Qi Jiang (姜奇) received Ph.D. degree in Computer Science from Xidian University in 2011. He is now an Associate Professor at School of Cyber Engineering, Xidian University. His research interests include security protocols and wireless network security, cloud security.

Rui-Jie Zhang (张瑞杰) received Ph.D. from the Zhengzhou Information Science and Technology Institute. Now, she is currently a Lecturer in Zhengzhou Information Science and Technology Institute. Her current research interest includes image analysis and processing.

Chuan-Gui Ma (马传贵) received Ph.D. degree in Mathematic from Zhejiang University of China. He is a Professor in Zhengzhou Information Science and Technology Institute, Zhengzhou, China. His research field is information security.