Journal of Information Hiding and Multimedia Signal Processing ©2018 ISSN 2073-4212
Ubiquitous International Volume **9**, Number **6**, November **2018**

A New Verifier-Based Anonymous Password-Authenticated Key Exchange Protocol

Chien-Ming Chen[1], Guangjie Wang[1], Weicheng Fang[1], and Tsu-Yang Wu[2,3,4*]

[1]School of Computer Science and Technology
Harbin Institute of Technology Shenzhen Graduate School
Shenzhen, China

[2]College of Computer Science and Engineering
Shandong University of Technology
Shandong, China

[3]Fujian Provincial Key Laboratory of Big Data Mining and Applications
Fujian University of Technology
Fuzhou, China

[4]National Demonstration Center for
Experimental Electronic Information and Electrical Technology Education
Fujian University of Technology
Fuzhou, China

[*]Corresponding author: wutsuyang@gmail.com

Received January, 2018; revised August, 2018

Abstract. *In the information age, privacy have aroused wide concern. User identity can be tracked in a public network if it is not stored or transmitted in a secure way. To enhance the security, many anonymous password authenticated key exchange protocols have been proposed to anonymize user's identity from the server. However, few of them focus on the stolen verifier attacks which assumes a powerful adversary who is accessible to the server's database or even secret keys. In this paper, we propose a new verifier-based anonymous password-authenticated key exchange protocols. It employs an existed algebraic MAC as verifier to resist such attacks. We also show that the protocol is secure and efficient through analysis.*

Keywords: Verifier, Anonymity, Password authentication, Security

1. **Introduction.** As computer networks have been penetrating into all walks of life, from professional works to casual activities, people start to concern their privacy. When interacting with a remote server, they may worry if they are being traced or not. Usually, authentication happens before interaction. For example, if a registered user want to submit photos to the remote, it is required to input personal information such as user name and password in order to pass the server's authentication. However, such steps may leak privacy data to attackers and the server. Thus, anonymous password-authenticated key exchange (APAKE) protocols are proposed to achieve anonymity against others during an authentication phase.

Currently, most APAKE protocols[1, 2, 3, 4, 5, 6, 7, 8] do not take stolen verifier attacks into consideration. A traditional authentication protocol insists that a user and the server share a common verifier so that when the user submit it, the server can check its validity.

Table 1. Notations in the proposed protocol

Notation	Description
G	Circular group
p	Prime order of G
g, h	Generator of G
s	Server's master secret
$H(\cdot)$	Secure hash function
U	User group's identity
S	Server's identity
ID_i	User i's identity
PW_i	User i's password
n	Size of the user group

But if the server is compromised and the verifier is obtained by an adversary, then even a legal user may be the impersonated one. To solve the problem, some protocols[9, 10, 11, 12, 13] employs assisted devices to eradicate any risks of leak. Without a verifier table, an adversary has nothing to steal. However, such solutions bring about inconvenience for users due to the use of assisted devices.

Recently, Yang et al.[14] constructed a secure and efficient verifier-based APAKE protocol using smooth projective hash function. It is the first APAKE protocol to deal with such attacks without assisted devices as far as we have studied. Although it is provably secure in the standard model and responsible for authentication with only two rounds, it incurs large computation cost on the server side even pre-computation is allowable compared with existing APAKE protocols. Therefore, in this paper, we propose a new verifier APAKE protocol. Our protocol is constructed by taking Zhang et al.'s algebraic MAC[13] as server's verifier. As long as the key of a MAC is not compromised, an adversary can do nothing but guess the password in a brute force way. And this is how our protocol resists stolen verifier attacks. Security analysis and Performance evaluation demonstrate that our protocol is secure against various known attacks and more efficient in terms of computation and communication cost.

The rest of this paper is organized as follows. In section 2, the proposed protocol is provided. Section 3 and 4 give security analysis and BAN Logic. Efficiency Analysis is described in Section 4. Finally, section 6 concludes.

2. The proposed protocol.

In this section, we present the proposed anonymous password-authenticated key exchange protocol. Some notations in our protocol are described in Table 1. The scheme includes three phases: the setup phase, the registration phase, and the authentication phase.

In the setup phase, the server S initializes the system parameters including hash functions, encryption algorithms, and $\{G, p, g, h\}$, where solutions to the discrete log problem of h with base g is hard to find. S also chooses a random secret $s \in Z_p^*$.

In the registration phase, a user U_i fills in with personal information such as identity and password, computes

$$m_i = H(ID_i \| PW_i) \tag{1}$$

and then submits $\{ID_i, m_i\}$ to the server S through a secure channel. When S receive a registration request, it first validates the identity in case of duplicates. Then, S retrieves its master secret s to compute an algebraic MAC

$$V_i = g^{\frac{1}{m_i+s}} \tag{2}$$

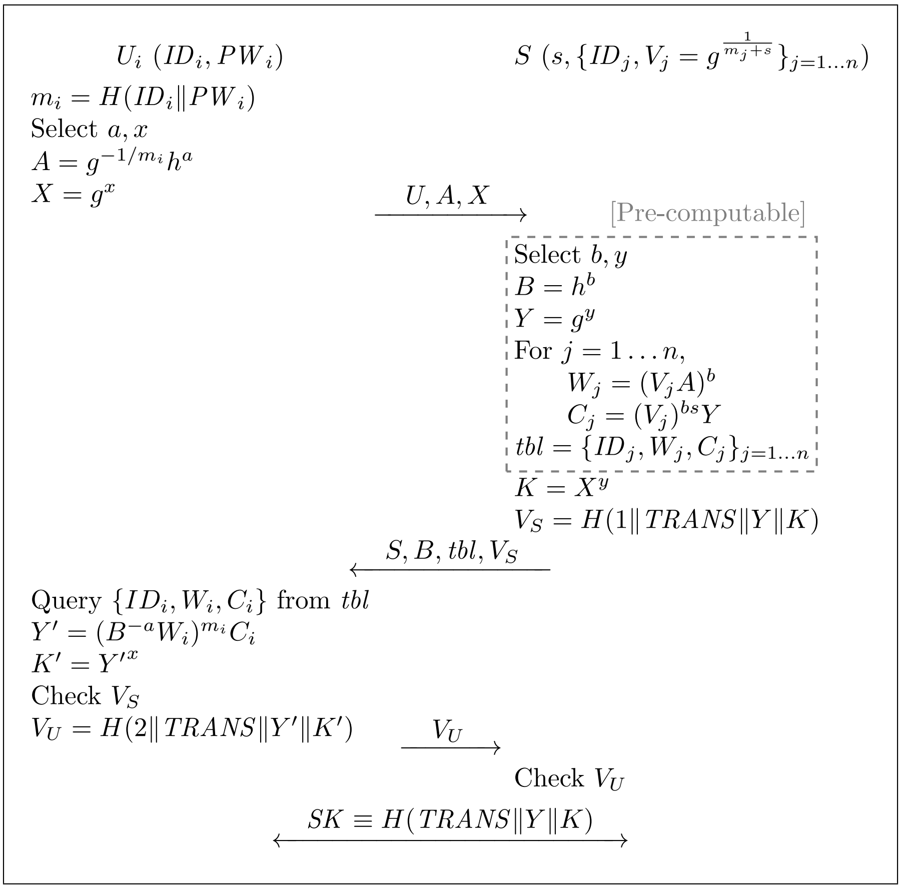

FIGURE 1. The authentication phase of the proposed APAKE protocol, where $TRANS = U \| A \| X \| S \| B \| tbl$

as a verifier for the user, and stores $\{ID_i, V_i\}$ in the database. After a quick response from S, U_i becomes a registered user.

In the authentication phase, a user U_i starts an authentication session with the server S. Figure 1 illustrates this phase, including the following steps.

(1) When the system is ready for login, U_i selects two random numbers $\{a, x\} \in Z_p^{*2}$, inputs ID_i and PW_i, derives m_i as Eq. ,the one in the registration phase, and computes

$$A = g^{-1/m_i} h^a, \tag{3}$$

$$X = g^x, \tag{4}$$

where A is a randomized authentication request and X is one component of a Diffie-Hellman key. Then, U_i sends $M_1 = \{U, A, X\}$ to S.

(2) After S receives M_1, it selects two random numbers $\{b, y\} \in Z_p^{*2}$, and computes two variables using the random numbers

$$B = h^b, \tag{5}$$

$$Y = g^y, \tag{6}$$

where Y is the other component of a Diffie-Hellman key. Next, S traverses the user table and retrieves the stored $\{ID_j, V_j\}_{j=1\dots n}$ from the database. For each pair, it randomizes V_j as W_j and hides Y with a simple but fast operation in C_j

$$W_j = (V_j A)^b, \tag{7}$$

$$C_j = (V_j)^{bs} Y. \tag{8}$$

After the loop computation, S packages all these values in an indexed table $tbl = \{ID_j, W_j, C_j\}$. Note that almost all the exponentiation operations are pre-computable before or in the loop. Thus, S can pre-process values such as B, Y, V_j^b, and C_j, only waiting for incoming A to compute A^b. These values together with the following computations are sent to U_i by S as $M_2 = \{S, B, tbl, V_S\}$, where V_S is an authenticator.

$$K = X^y, \tag{9}$$

$$V_S = H(1\|U\|A\|X\|S\|B\|tbl\|Y\|K). \tag{10}$$

(3) When M_2 is received by the user, U_i retrieves $\{ID_i, W_i, C_i\}$ from tbl, and recover the hidden value according to the following equations

$$Y' = (B^{-a} W_i)^{m_i} C_i, \tag{11}$$

$$K' = (Y')^x. \tag{12}$$

By substituting the recovered values, U_i can verifies the correctness of V_S. If it turns out to be false, the protocol ends with failure. Otherwise, U_i believes that S is a trusted server, computes another authenticator

$$V_U = H(2\|U\|A\|X\|S\|B\|tbl\|Y\|K) \tag{13}$$

and sends $M_3 = \{V_U\}$ to S.

(4) After S receives M_3, it checks V_U against its own computation. Incorrect result will lead to denial of the user's login request. On the contrary, S will accept the anonymous user as a legal one, and compute the session key

$$SK = H(U\|A\|X\|S\|B\|tbl\|Y\|K). \tag{14}$$

U_i can also compute the session key with corresponding Y' and K' after successful verification of V_S.

3. **Security Analysis.** In fact, various key exchange protocols have been proposed. However, many of them have been proven insecure [15, 16, 17, 18, 19]. For this reason, a security analysis is required. In this section, we show that the proposed protocol meets common security requirements and resists various known attacks.

1. Mutual Authentication:In the proposed protocol, both the registered user U_i and the server S authenticate each other to complete mutual authentication. U_i verifies V_S, while S verifies V_U. On the one side, to check the correctness of V_S, U_i must obtain the same value of Y. Since the computation $Y' = (B^{-a} W_i)^{m_i} C_i = V_i^{-bs} C_i$ is equivalent to Y, if the user inputs an honest m_i and the server inputs correct s and V_i, U_i will trust S. On the other side, S will regard anyone who manage to recover Y from tbl as legal user.
2. Fairness of key exchange: The session key in our protocol involves X and Y from the user and the server respectively. Thus both party enjoy equal contributions to the key exchange process.
3. Forward secrecy: Even if the user's password or the server's master secret are leaked by accident, previously established session keys will be protected by random exponents x and y chosen by U_i and S.

Table 2. Symbols used in BAN logic

Notation	Description
$A \mid\equiv X$	A believes X
$A \triangleleft X$	A saw X
$A \mid\sim X$	A said X
$A \Rightarrow X$	A has control over X
$\#(X)$	X is fresh
$A \overset{K}{\leftrightarrow} B$	A and B share a key K
$A \overset{X}{\Leftrightarrow} B$	A and B share a secret X
$\{X\}_K$	X is encrypted under a key K
$(Y)_X$	Y contains secret X

4. Anonymity: The proposed scheme provides anonymity for the registered users. In the protocol, U_i sends a different A containing the identity for each authentication request. Therefore, neither S nor other attackers can trace the user. They cannot even guess the identity from A because it contains a random exponent a.
5. Resistance to Replay Attacks: An adversary who replays messages will find the receiver's response immediately. Each replay incurs a new random nonce generated by the receiver. At the moment, those who replay should answer the response correctly to finish the authentication. However, the answer consists of a random number a or b involved in the replayed message. Knowing nothing about it, the adversary cannot offer an answer. Therefore, our protocol can resist replay attacks.
6. Resistance to Off-line Password Guessing Attacks: In these attacks, an adversary tries to guess a user's password from existing protocol transcripts. In the protocol, passwords are inputs of a hash function that outputs m_i. Although the value A can be the adversary's target, it contains a random nonce that cannot be guessed. Similarly, the adversary cannot guess from *tbl* since it contains server's master secret.
7. Resistance to Stolen Verifier Attacks: It is assumed that the server's database is unbelievably accessible to an adversary in these attacks. In this case, our protocol guarantees that the adversary at least relies on some brute force methods to break users' passwords. This is because the database stores an algebraic MAC for U_i. Without the server's key s, the adversary can do nothing. If the key is unlikely stolen, the only way to obtain PW_i is to guess from V_i. Thus, the protocol is secure against such attacks.

4. **Analysis of BAN Logic.**

BAN (Burrows-Abadi-Needham) logic is used to prove the security of an authentication protocol. Generally, with a list of logic symbols and inference rules, it can help analyze the authenticity and freshness of the messages transmitted and its source, which shows whether the protocol is vulnerable to eavesdropping and replaying. Table 2 lists the symbols used in BAN logic.

The following inference rules are used in our proof.

R1: $\frac{A \mid\equiv A \overset{X}{\Leftrightarrow} B, A \triangleleft (Y)_X}{A \mid\equiv B \mid\sim Y}$

R2: $\frac{A \mid\equiv \#(X), A \mid\equiv B \mid\sim X}{A \mid\equiv B \mid\equiv X}$

R3: $\frac{A \mid\equiv B \Rightarrow X, A \mid\equiv B \mid\equiv X}{A \mid\equiv X}$

R4: $\frac{A \mid\equiv \#(X)}{A \mid\equiv \#(X,Y)}$

R5: $\frac{A \mid\equiv B \mid\equiv (X,Y)}{A \mid\equiv B \mid\equiv X)}$

The goal of BAN logic analysis is to prove both parties believe in the established key. To achieve them, we follow the general proof routine including four steps.

(1) We set the following goals according to the design of our authentication phase.
G1: $U_i \mid\equiv U_i \overset{SK}{\leftrightarrow} S$
G2: $S \mid\equiv U_i \overset{SK}{\leftrightarrow} S$
G3: $U_i \mid\equiv S \mid\equiv U_i \overset{SK}{\leftrightarrow} S$
G4: $S \mid\equiv U_i \mid\equiv U_i \overset{SK}{\leftrightarrow} S$

(2) In order to adapt the protocol to the symbols in BAN logic, we idealize the messages.
M1: $S \triangleleft (A, X)$
M2: $U_i \triangleleft (B, W_i, C_i, V_S, U_i \overset{K}{\Leftrightarrow} S)_{V_i^{bs}}$
M3: $S \triangleleft (A, X, V_U, U_i \overset{K}{\Leftrightarrow} S)_{V_i^{bs}})$

(3) Besides the common reference rules, we assume the following as known conditions. Among these conditions, A3 and A4 state that S and U_i share a secret V_i^{bs}. It is also a randomized MAC computable by both S and U_i. On the one side, S owns the random number b and the MAC V_i. On the other side, U_i can recover the secret by the equation $(B^{-a}W_i)^{-m_i} \equiv V_i^{bs}$. Therefore, it is reasonable to include them in our assumptions.
A1: $U_i \mid\equiv \#(a, x)$
A2: $S \mid\equiv \#(b, y)$
A3: $U_i \mid\equiv U_i \overset{V_i^{bs}}{\Leftrightarrow} S$
A4: $S \mid\equiv U_i \overset{V_i^{bs}}{\Leftrightarrow} S$
A5: $U_i \mid\equiv S \Rightarrow U_i \overset{SK}{\leftrightarrow} S$
A6: $S \mid\equiv U_i \Rightarrow U_i \overset{SK}{\leftrightarrow} S$

(4) The detailed steps to the goals are listed as followed. From S5, S6, S11 and S12, we show that the goals can be inferred.
S1: $\frac{A3,M2}{U_i \mid\equiv S \mid\sim (B,W_i,C_i,V_S,U_i \overset{K}{\Leftrightarrow} S)}$
S2: $\frac{A1}{U_i \mid\equiv \#(B,W_i,C_i,V_S,U_i \overset{K}{\Leftrightarrow} S)}$
S3: $\frac{S1,S2}{U_i \mid\equiv S \mid\equiv (B,W_i,C_i,V_S,U_i \overset{K}{\Leftrightarrow} S)}$
S4: $\frac{S3}{U_i \mid\equiv S \mid\equiv U_i \overset{K}{\Leftrightarrow} S}$
S5: $\frac{S4}{U_i \mid\equiv S \mid\equiv U_i \overset{SK}{\leftrightarrow} S}$
S6: $\frac{S5,A5}{U_i \mid\equiv U_i \overset{SK}{\leftrightarrow} S}$
S7: $\frac{A4,M3}{S \mid\equiv U_i \mid\sim (A,X,V_U,U_i \overset{K}{\Leftrightarrow} S)}$
S8: $\frac{A2}{S \mid\equiv \#(A,X,V_U,U_i \overset{K}{\Leftrightarrow} S)}$
S9: $\frac{S7,S8}{S \mid\equiv U_i \mid\equiv (A,X,V_U,U_i \overset{K}{\Leftrightarrow} S)}$
S10: $\frac{S9}{S \mid\equiv U_i \mid\equiv U_i \overset{K}{\Leftrightarrow} S}$
S11: $\frac{S10}{S \mid\equiv U_i \mid\equiv U_i \overset{SK}{\leftrightarrow} S}$
S12: $\frac{S11,A6}{S \mid\equiv U_i \overset{SK}{\leftrightarrow} S}$

Table 3. Efficiency comparison of APAKE protocols

Protocols	The number of modular exponentiations				Communication costs
	User U_i		Server S		
	Total	Total–Precomp.	Total	Total–Precomp.	
APAKE [1]	6	4	$4n+2$	$3n+1$	$(n+2)\|p\| + (n+1)\|h\|$
TAP [2]	3	2	$n+1$	n	$2\ \|p\| + (n+1)\|h\|$
NAPAKE [3]	4	3	$n+3$	2	$(n+3)\|p\| + \|h\|$
VEAP [5]	2	1	$n+2$	1	$3\ \|p\| + 2\ \|h\| + n\|e\|$
VAPAKE [13]	13	6	$8n+9$	$6n+1$	$(2n+8)\|p\|$
APSTD [7]	8	6	$9n+2$	$7n$	$(6n+2)\|p\| + 3\ \|h\|$
Ours	6	3	$2n+3$	2	$(2n+3)\|p\| + 2\ \|h\|$

5. **Efficiency Analysis.** To analyze the efficiency, we compare our scheme with other APAKE protocols in terms of computation and communication cost. The comparison follows Shin et al.'s criterion[5]. The computation cost is evaluated by the number of modular exponentiations performed by a clients or the server since it takes rather little time to process symmetric encryption/decryption, hash, and message authentication code. We also analyze their computation cost under the situation where pre-computation is allowed. The communication cost consists of the size of all transmitted messages in single authentication phase. The messages usually involve group elements ($|p|$), hash result ($|h|$), and encrypted text ($|e|$). Table 3 presents the comparison data.

Our proposed protocol performs better than most other ones. In the table, only VAPAKE and ours can withstand stolen verifier attacks. By comparison, our protocol is more efficient. Compared to the insecure, currently, all existed protocols require linear computation on the server side. But pre-computation can release its burden to accelerate the time to respond during an authentication. Our protocol requires only 2 exponentiations in such situation, which is next to the best VEAP. As for the communication cost, it also come with a linear coefficient. To better understand the cost, we assume a common security setting which states $|e| = 128, |h| = 160, |p| = 1024$. Thus the cost of our scheme can be ranked as a middle one among all the others. Although it does not excel in communication cost, besides the VAPAKE, it is the only two protocols that withstand stolen verifier attacks. Therefore, our protocol has advantages in computation cost and security.

6. **Conclusions.** In this paper, we present a new verifier-based anonymous password-authenticated key exchange protocol. By setting an algebraic MAC as server's verifier, we design a protocol secure against stolen verifier attacks. The analysis shows that our protocol can withstand various known attacks and achieve high efficiency.

Acknowledgement. The work of Chien-Ming Chen was supported in part by Shenzhen Technical Project under Grant number JCYJ20170307151750788 and in part by Shenzhen Technical Project under Grant number KQJSCX20170327161755. The work of Tsu-Yang Wu was supported in part by the Science and Technology Development Center, Ministry of Education, China under Grant no. 2017A13025 and the Natural Science Foundation of Fujian Province under Grant no. 2018J01636.

REFERENCES

[1] C.-T. Li, C.-L. Chen, C.-C. Lee, C.-Y. Weng, and C.-M. Chen, "A novel three-party password-based authenticated key exchange protocol with user anonymity based on chaotic maps," *Soft Computing*, vol. 22, no. 8, pp. 2495–2506, 2018.

[2] T.-Y. Wu, W. Fang, C.-M. Chen, and G. Wang, "Cryptanalysis of an anonymous mutual authentication scheme for secure inter-device communication in mobile networks," in *International Conference on Intelligent Information Hiding and Multimedia Signal Processing.* Springer, 2017, pp. 206–213.

[3] J. Yang and Z. Zhang, "A new anonymous password-based authenticated key exchange protocol," in *International Conference on Cryptology in India.* Springer, 2008, pp. 200–212.

[4] H.-Y. Lin, W.-G. Tzeng, and others, "Anonymous Password Based Authenticated Key Exchange with Sub-Linear Communication," *Journal of Information Science and Engineering*, vol. 25, no. 3, pp. 907–920, 2009.

[5] S. Shin, K. Kobara, and H. Imai, "Very-efficient anonymous password-authenticated key exchange and its extensions," in *International Symposium on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes.* Springer, 2009, pp. 149–158.

[6] X. Hu, J. Zhang, Z. Zhang, and J. Xu, "Universally composable anonymous password authenticated key exchange," *Science China Information Sciences*, vol. 60, no. 5, p. 52107, 2017.

[7] X. Hu, J. Zhang, Z. Zhang, and F. Liu, "Anonymous Password Authenticated Key Exchange Protocol in the Standard Model," *Wireless Personal Communications*, pp. 1–24, 2017. [Online]. Available: http://dx.doi.org/10.1007/s11277-017-4250-z

[8] C.-M. Chen, W. Fang, S. Liu, T.-Y. Wu, J.-S. Pan, and K.-H. Wang, "Improvement on a chaotic map-based mutual anonymous authentication protocol." *Journal of Information Science & Engineering*, vol. 34, no. 2, 2018.

[9] Z. Chai, Z. Cao, and R. Lu, "Efficient password-based authentication and key exchange scheme preserving user privacy," in *International Conference on Wireless Algorithms, Systems, and Applications.* Springer, 2006, pp. 467–477.

[10] Y. Yang, J. Zhou, J. Weng, and F. Bao, "A new approach for anonymous password authentication," in *Computer Security Applications Conference, 2009. ACSAC'09. Annual.* IEEE, 2009, pp. 199–208.

[11] Y. Yang, J. Zhou, J. W. Wong, and F. Bao, "Towards practical anonymous password authentication," in *Proceedings of the 26th Annual Computer Security Applications Conference.* ACM, 2010, pp. 59–68.

[12] S. Shin and K. Kobara, "A secure anonymous password-based authentication protocol with control of authentication numbers," in *Information Theory and Its Applications (ISITA), 2016 International Symposium on.* IEEE, 2016, pp. 325–329.

[13] Z. Zhang, K. Yang, X. Hu, and Y. Wang, "Practical Anonymous Password Authentication and TLS with Anonymous Client Authentication," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security.* ACM, 2016, pp. 1179–1191.

[14] X. Yang, H. Jiang, Q. Xu, M. Hou, X. Wei, M. Zhao, and K.-K. R. Choo, "A Provably-Secure and Efficient Verifier-Based Anonymous Password-Authenticated Key Exchange Protocol," in *Trustcom/BigDataSE/I?? SPA, 2016 IEEE.* IEEE, 2016, pp. 670–677.

[15] K.-H. Wang, C.-M. Chen, W. Fang, and T.-Y. Wu, "On the security of a new ultra-lightweight authentication protocol in iot environment for rfid tags," *The Journal of Supercomputing*, vol. 74, no. 1, pp. 65–70, 2018.

[16] C.-M. Chen, C.-T. Li, S. Liu, T.-Y. Wu, and J.-S. Pan, "A provable secure private data delegation scheme for mountaineering events in emergency system," *IEEE Access*, vol. 5, pp. 3410–3422, 2017.

[17] C.-M. Chen, W. Fang, K.-H. Wang, and T.-Y. Wu, "Comments on an improved secure and efficient password and chaos-based two-party key agreement protocol," *Nonlinear Dynamics*, vol. 87, no. 3, pp. 2073–2075, 2017.

[18] C.-M. Chen, L. Xu, T.-Y. Wu, and C.-R. Li, "On the security of a chaotic maps-based three-party authenticated key agreement protocol," *Journal of Network Intelligence*, vol. 1, no. 2, pp. 61–65, 2016.

[19] K.-H. Wang, C.-M. Chen, W. Fang, and T.-Y. Wu, "A secure authentication scheme for internet of things," *Pervasive and Mobile Computing*, vol. 42, pp. 15–26, 2017.