Self-Updating One-Time Password Mutual Authentication Protocol for Ad Hoc Network

Feng Xu[1], Xin Lv[1, 2], Qi Zhou[1] and Xuan Liu[1]
[1] College of Computer and Information, Hohai University
Nanjing, Jiangsu 211100, CHINA
[e-mail: njxufeng@163.com]
[2] College of Water Conservancy and Hydropower Engineering
Nanjing, Jiangsu 210098, CHINA
[e-mail: lvxin.gs@163.com]
*Corresponding author: Feng Xu

Received December 26, 2013; revised March 10, 2014; accepted April 19, 2014; published May 29, 2014

Abstract

As a new type of wireless network, Ad hoc network does not depend on any pre-founded infrastructure, and it has no centralized control unit. The computation and transmission capability of each node are limited. In this paper, a self-updating one-time password mutual authentication protocol for Ad hoc network is proposed. The most significant feature is that a hash chain can update by itself smoothly and securely through capturing the secure bit of the tip. The updating process does not need any additional protocol or re-initialization process and can be continued indefinitely to give rise to an infinite length hash chain, that is, the times of authentication is unlimited without reconstructing a new hash chain. Besides, two random variable are added into the messages interacted during the mutual authentication, enabling the protocol to resist man-in-the-middle attack. Also, the user's identity information is introduced into the seed of hash chain, so the scheme achieves anonymity and traceability at the same time.

Keywords: Ad Hoc Network, Mutual Authentication, Self-Updating, Infinite Length Hash Chain, Man-in-the-middle Attack, Traceability

http://dx.doi.org/10.3837/tiis.2014.05.018

1. Introduction

As a new type of wireless network, Ad hoc network does not depend on any pre-founded infrastructure, and it has no centralized control unit. The computation and transmission capability of each node are limited. The interconnection and data transmission between the nodes can be achieved through MultiHop. Ad Hoc network has the advantages such as self-organized, dynamics, and invulnerability. However, it is open to all kinds of attacks, and obviously, the traditional security techniques cannot directly apply in the Ad hoc network, meanwhile, the existed security techniques used in Ad hoc do not address the security problems properly. Therefore, the effective methods, such as authentication, data protection, privacy-preserving, has drawn more and more concerns. Among them, authentication is the basis of communication, considering limited computing and transmitting power of each node, how to design a lightweight and secure authentication protocol has become a research hotspot.

In 1981, Lamport [1] firstly suggested the concept of one-time password based on hash function. It adds uncertain factors into the identity information sending to the server, making the message varied in each time. This method can effectively resist intercept/replay attack, but the amount of encryptions needed in the protocol is a heavy burden for the user. Later, Haller refined this method as S/KEY standard [2]. In succeeding research, one-time pass word authentication also can be implemented in two other ways: time synchronization and challenge/response. The former requires the user and the server keep time synchronized, the two parties use the identical time as calculating factor to generate one time password, and then carry out identity authentication. In the latter way, the server sends “challenge” to the user, and authenticates the identity of the user based on its “response”.

Sandirigama et al. [3] proposed a simple and secure authentication protocol (SAS) for enhancing security level apart from low processing, storage and transmission overheads. Tsuji et al. [4] improved SAS, it reduces 40% overhead of hash function adaptation, and the method has a mutual authentication phase which keeps synchronous data communications in an authentication procedure.

The traditional one-time password authentication protocol has a shortcoming that the time of authentication is limited. When the authentication times up to a certain value, the re-initialization is required. Yeh et al. [5] proposed a secure one-time password authentication scheme using smart cards. The scheme is free from any of server spoofing attacks, replay attacks, and off-line dictionary attacks. A session key here is also established to provide confidentiality. Goyal et al. [6] constructed a N/R One Time Password System. The basic idea is have the server aid the client computation by inserting “breakpoints” in the hash chains. Client computational requirements are dramatically reduced without any increase in the server computational requirements and the number of times a client may login before the system has to be reinitialized is also increased significantly. Zhang et al. [7] designed a novel scheme with

a new kind of hash chain. It achieves self-updating, fine-authentication and proactive updating, and the updating process is smooth, secure and efficient which does not need additional protocols or an independent re-initialization process. In 2008, Chefranov [8] changed the traditional construction of backward hash chain. Under C/S model, the proposed scheme based on infinite forward stepping hash chains, and not requiring re-initialization after a certain number of authentications. Wang et al. [9] proposed a novel multi-party key agreement scheme with password authentication and sharing password evolvement. The password is not only used to authenticate the mobile node's secret keys, but also used to encrypt alternating information between mobile nodes. The freshness and security of passwords are guaranteed by sharing password evolvement every time in mobile node's secret keys authentication and key agreement. Li et al. [12] proposed a two-layer authentication protocol for wireless ad hoc networks, without adopting public key computations to provide secure and anonymous communication, then it can be efficiently implemented on small ad hoc devices, moreover, the protocol accomplishes mutual authentication, session key agreement, and along with integration of non-PKI techniques, the source node can anonymously interact with the destination node. Later, to remedy the security weaknesses existed in the authentication scheme introduced by He et al. [13], such as attacks against the user anonymity, lack of user friendliness and so on, a novel privacy-preserving authentication scheme which is immune to various known types of attack was proposed in [14], and the scheme is practical for mobile wireless networking. Paterson et al. [15] considered the use of one-time passwords in the context of password-authenticated key exchange (PAKE), which allows for mutual authentication, session key agreement, and resistance to phishing attacks, and advanced a general technique for building a secure one-time-PAKE protocol from any secure PAKE protocol, based on describing a security model for the use of one-time passwords. Eldefrawy et al. [16] proposed a one-time password system, involving two different nested hash chains, which provide the authentication seed updating and the OTP production respectively. The system achieves some favourable properties, such as non-restricted one-time password generation.

Based on the above analysis, one-time password authentication protocol has characteristics of simple designation, lightweight computational overhead and high efficiency, which are suitable for Ad hoc network. But the existed methods have relative demerits as well as merits. In this paper, a self-updating one-time password mutual authentication protocol for Ad hoc network is proposed. Through capturing the secure bit of the tip of a new chain, a hash chain can update by itself smoothly, also can be continued indefinitely to give rise to an infinite length hash chain. We add identity information into authentication process, making the protocol to resist man-in-the-middle attack, and achieve anonymity and traceability at the same time.

2. Preliminaries

2.1 Hard-Core Predicate [10]

General speaking, a polynomial-time predicate b is called a hard-core of a function f if each efficient algorithm, given $f(x)$, can guess $b(x)$ with success probability that is only negligibly better than one-half.

Definition 1 (Hard-Core Predicate) A polynomial-time-computable predicate $b:\{0,1\}^* \to \{0,1\}$ is called a hard-core of a function f for each probabilistic polynomial-time algorithm A', each positive polynomial $p(\cdot)$, and all sufficiently large n has $\Pr\left[A'(f(U_n)) = b(U_n)\right] \le \frac{1}{2} + \frac{1}{p(n)}$. U_n is a random variable uniformly distributed in $\{0,1\}^n$.

2.2 Hash Chain

Definition 2 (Hash Chain) Select a cryptographic secure hash function h with security parameter $k: \{0,1\}^* \to \{0,1\}^k$. Pick a seed s randomly and apply h recursively N times to an initial seed s to generate a hash chain. The tip ω of the chain equals $h^N(s)$.

$$\omega = h^N(s) = h\left(h^{N-1}(s)\right) = \underbrace{h\left(h\left(h\left(\ldots h(s)\right)\right)\right)}_{N\ Times}$$

3. Analysis of the Yeh's Protocol

The protocol proposed in [5] is a mutual authentication protocol, utilizing hash chain to construct one-time password. The scheme has been divided into two phases: registration and authentication. C denotes the user and S denotes the server.

3.1 Registration

1) $C \leftarrow S: SEED$;

2) $C \leftarrow S: N, SEED \oplus D_0, h(D_0)$;

3) $C \to S: p_0 \oplus D_0$

$p_t = h^{N-t}(K \oplus SEED), t \ge 0$, K is the user's private key, N is the length of hash chain before reconstruction, $N >> 1$, D_0 is a temporary authenticating random number generated by the server.

3.2 Authentication

1) $C \leftarrow S: Mt = \left(M_{t1}, M_{t2}, M_{t3}\right) = \left(N-t, SEED \oplus D_t, h\left(D_t\right) \oplus p_{t-1}\right)$ with $N-1 \geq t \geq 1-\text{integer}$;

2) C: if $h(M_{12} \oplus SEED) \oplus p_{N-M_{11}-1} \neq M_{13}$, then the user determines that M_1 doesn't come from the server. Stop the process. Otherwise, continue;

3) $C \rightarrow S: U_t = p_t \oplus D_t$. p_t is a new password produced by the user, and p_{t-1} is stored on the server;

4) S: if $h\left(U_t \oplus D_t\right) = p_{t-1}$, then the user passes the authentication. Substitute the new password $p_t = U_t \oplus D_t$ for p_{t-1}, while let $t = t+1$;

However, the protocol has the following security limits:

- In the protocol, the hash chain has a finite length, thus the number of times a user may login is limited. Furthermore, if the length of hash chain is too long, the computational overhead is intensive for each node although the number of login time is grown; if the length is too short, the hash chain is required to reconstruct frequently, as a result, the registration of a new *SEED* including transmission need to be carried out repeatedly, which obviously increases communication costs and security risks.
- The protocol is vulnerable to man-in-the-middle attack. As long as an attacker masquerades as the server to ask for an authentication, the user's password will be leaked, and then it can utilize the password to impersonate the user, completing the authentication with real server.

Aiming at the security problems above-mentioned, some improvements are made on the Yeh's protocol in the next section.

4. Self-Updating One-Time Password Mutual Authentication Protocol for Ad Hoc Network

The protocol is consisted of two phrases: Initialization and Mutual Authentication.

4.1 Initialization

User A and B jointly select a secure hash function h with a security parameter N, and randomly choose a secure *SEED* from $\{0,1\}^k$. Select a predicate function $B: \{0,1\}^* \rightarrow \{0,1\}$ as the kernel of h. Each user has a unique *ID*. Take user A as an example to show the concrete initialization process.

1) Select a random number K_A as A's private key;

2) Generate a hash chain $p_{At} = h^{N-t}\left(K_A \oplus SEED \oplus h\left(ID_A\right)\right)$ with the tip value p_{A0}, t denotes the t th time authentication, $t \geq 0$, and send p_{A0} to user B securely;

3) Randomly select another private key K'_A from $\{0,1\}^k$, the corresponding tip value is p'_{A0}. Store K'_A and p'_{A0} on local device securely.

4) Generate a random number D_{At}, making $B\left(D_{At}\right) = p'_{A[i]}$, $i = 1$(initial value).

4.2 Mutual Authentication

Generally, the process of t th time authentication is as follow.

1) User A sends the authenticating information to B:

$$A \rightarrow B: MAt1 = \left(M_{At11}, M_{At12}, M_{At13}\right)$$
$$= \left(N - t, SEED \oplus D_{At}, h\left(D_{At}\right) \oplus p_{B(t-1)}\right)$$

2) User B firstly verifies the authenticity of the information, that is, checks $h\left(M_{At12} \oplus SEED\right) \oplus p_{N-MAt11-1} = M_{At13}$ whether holds or not. If not, stop authenticating. If so, computes $B\left(M_{At12} \oplus SEED\right)$, and then sends the message back to A:

$$A \leftarrow B: MBt1 = \left(M_{Bt11}, M_{Bt12}, M_{Bt13}\right)$$
$$= \left(N - t, SEED \oplus D_{Bt}, h\left(D_{Bt}\right) \oplus p_{A(t-1)}\right)$$

3) User A firstly verifies the authenticity of the information, that is, checks $h\left(M_{Bt12} \oplus SEED\right) \oplus p_{N-MBt11-1} = M_{Bt13}$ whether holds or not. If not, stop authenticating. If so, computes $B\left(SEED \oplus D_{Bt}\right)$. Then A sends the new password p_{At} to B for the next authentication:

$$A \rightarrow B: MAt2 = p_{At} \oplus D_{Bt}$$

4) User B firstly verifies the message whether comes from A, that is , checks if $h\left(MAt2 \oplus D_{Bt}\right) = p_{A(t-1)}$. If so, it returns the new p_{Bt}: $A \leftarrow B: MBt2 = p_{Bt} \oplus D_{At}$. Substitutes the new $p_{At} = MAt2 \oplus D_{Bt}$ for $p_{A(t-1)}$, and lets $t = t + 1$.

5) User A firstly checks $h\left(MBt2 \oplus D_{At}\right) = p_{B(t-1)}$ whether holds or not. If so, substitutes the new $p_{Bt} = MBt2 \oplus D_{At}$ for $p_{B(t-1)}$, and lets $t = t + 1$.

After N times authentication, user A and B both obtain N th bit tip value of the new hash chain contained in D_{At} and D_{Bt}. Once the old hash chain is exhausted, a new one is activated. This process can be continued indefinitely, achieving a hash chain self-updating

smoothly.

Fig. 1 illustrates the interaction flow between user A and B to complete the mutual authentication.

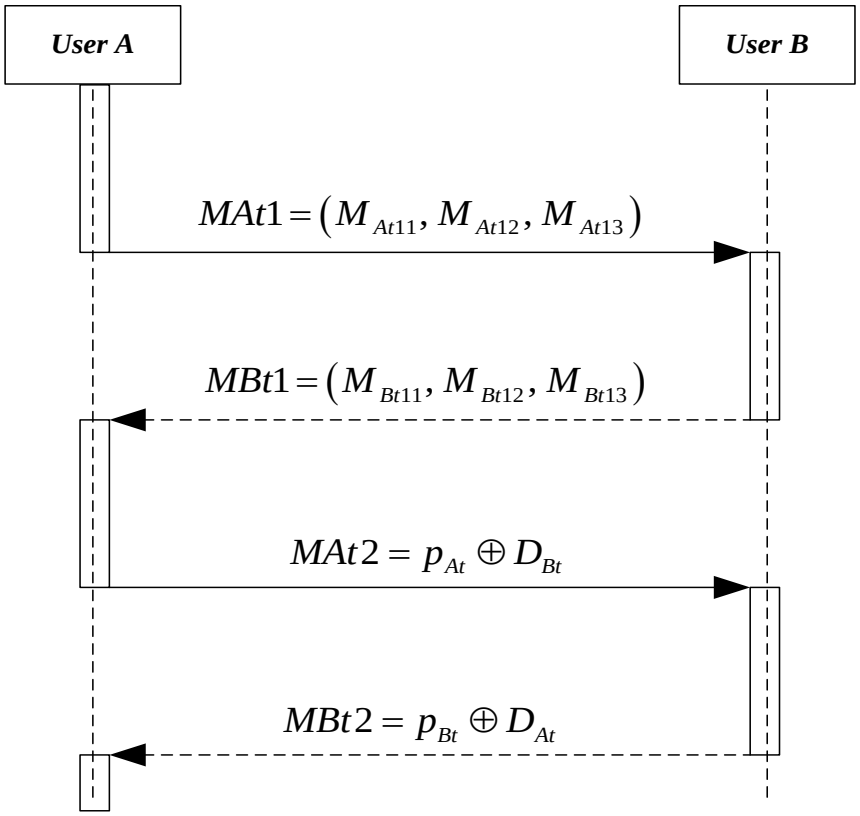

Fig. 1. The process of the t th time authentication

5. Security Analysis

The security of our protocol is totally based on the hash function and hard-core predicate function. If the hash function is one-way and collision resistant, the protocol is provably secure.

1) Anonymity

In our protocol, the node value of one-way hash chain is utilized to process authentication, without exposing any identity information of the user. So, the privacy of the user is preserved.

2) Non-Repudiation

Only the two parties who engaged in the authentication process can produce the value of the hash chain for authenticating, and the others who attempt to generate a valid one is equivalent

to break the properties of hash function, which is computationally infeasible. So, the two parties can not repudiate themselves once the session is established.

3) Traceability

The node value of the hash chain used in each time authentication is generated by the secret seed $K_A \oplus SEED \oplus h(ID_A)$, which contains the unique ID number of the corresponding user. If something misbehaved happens, the person in charge can be traced through the secret seed.

4) Non-Masquerading

The key message in the authentication process is the node value of hash chain. Any adversary attempts to impersonate user A to communicate with the others, it has to present $h^{N-t-1}\left(K_A \oplus SEED \oplus h(ID_A)\right)$ correctly, but it does not know $K_A \oplus SEED \oplus h(ID_A)$. The only way is try to obtain the hash value from $h^N\left(K_A \oplus SEED \oplus h(ID_A)\right)$. Unfortunately, it is computationally infeasible.

5) Resist Replay Attack

If the message $MAt1$ and $MBt1$ have been intercepted by the adversary, it still need to get $MAt2$ and $MBt2$ to launch a replay attack. So it has to find out p_{At} and p_{Bt}, but deriving p_{At} and p_{Bt} from the known $p_{A(t-1)}$ and $p_{B(t-1)}$ is equivalent to break one-way hash function, which is computationally infeasible.

6) Resist Man-In-The-Middle Attack

Literature [11] pointed out Yeh's Protocol [5] is vulnerable to man-in-the-middle attack, and gave the concrete attack method.

(1) The attacker eavesdrops on the t th and $(t-1)$ th authentication, then it can obtain:

$$C \leftarrow S: Mt = \left(M_{t1}, M_{t2}, M_{t3}\right) = \left(N-t, SEED \oplus D_t, h(D_t) \oplus p_{t-1}\right);$$

$$C \rightarrow S: U_t = p_t \oplus D_t;$$

$$C \leftarrow S: M(t+1) = \left(M_{(t+1)1}, M_{(t+1)2}, M_{(t+1)3}\right) = \left(N-(t+1), SEED \oplus D_{t+1}, h(D_{t+1}) \oplus p_t\right);$$

$$C \rightarrow S: U_{t+1} = p_{t+1} \oplus D_{t+1}.$$

(2) Utilizing the stolen information, the attacker forges the password for the $t+2$ th authentication:

$$M'_{(t+2)1} = N-(t+2) = N-t-2;$$

$$M'_{(t+2)2} = M_{(t+1)2} = SEED \oplus D_{t+1};$$

$$
\begin{aligned}
M'_{(t+2)3} &= M_{(t+1)3} \oplus \left\{ \left(M_{t2} \oplus U_t \right) \oplus \left(M_{(t+1)2} \oplus U_{t+1} \right) \right\} \\
&= M_{(t+1)3} \oplus \left\{ \left(\left(SEED \oplus D_t \right) \oplus \left(p_t \oplus D_t \right) \right) \oplus \left(\left(SEED \oplus D_{t+1} \right) \oplus \left(p_{t+1} \oplus D_{t+1} \right) \right) \right\} \\
&= M_{(t+1)3} \oplus \left\{ \left(SEED \oplus p_t \right) \oplus \left(SEED \oplus p_{t+1} \right) \right\} \\
&= \left(h\left(D_{t+1} \right) \oplus p_t \right) \oplus \left(p_t \oplus p_{t+1} \right) \\
&= h\left(D_{t+1} \right) \oplus p_{t+1}
\end{aligned}
$$

Except the random number D_{t+1} used in $t+1$th authentication, other factors in the forged $t+2$th authentication password $M'(t+2) = \left(M'_{(t+2)1}, M'_{(t+2)2}, M'_{(t+2)3} \right)$ are all valid. Due to the random variable is not stored on the user side, it will not perceive that the attacker pretend to be the server and communicate with it by $M'(t+2)$, then the attacker will obtain the password for the next time authentication. Subsequently, it can masquerade as the user to complete the authentication with the server, using the true password.

In our protocol, user A and B respectively generate a random number D_{At} and D_{Bt} for securely transmitting the bit of the tip of a new chain, and store them on local; and the digit i of the bits which have been transmitted is recorded in each authentication, so the times of authentication is also recorded through this tag. Then our protocol can resist the man-in-the-middle attack described in [11].

Meanwhile, if D_{At} and D_{Bt} are tampered, then the tip value of a new hash chain will be consequentially altered, apparently, it is no longer valid for passing the verification in authenticating process.

5. Conclusion

Based on the Yeh's Protocol, a self-updating one-time password mutual authentication protocol was proposed, which allows unlimited times of authentication. Through capturing the secure bit of the tip of a new chain, the hash chain can update by itself smoothly without any re-initialization process, effectively mitigating the risk caused by the frequent interactions in the process of reconstruction. The improved protocol has a low computation overhead and traffic, which is suitable for Ad hoc environment. It can resist man-in-the-middle attack, and achieves anonymity and traceability at the same time.

Acknowledgments

This paper is supported by National Natural Science Foundation of China: "Research on Trusted Technologies for The Terminals in The Distributed Network Environment" (Grant No.

60903018), "Research on the Security Technologies for Cloud Computing Platform" (Grant No. 61272543), and "The National Twelfth Five-Year Key Technology Research and Development Program of the Ministry of Science and Technology of China" (Grant No. 2013BAB06B04).

References

[1] L. Lamport, "Password Authentication with Insecure Communication," *Communications of the ACM*, vol. 24, issue 11, pp. 770-772, November, 1981. Article (CrossRef Link)

[2] N. Haller, The S/Key One-Time Password System, *http://mail.tools.ietf.org/html/rfc1760*, 1995.

[3] M. Sandirigama, A. Shimizu, and M. T. Noda, "Simple and Secure Password Authentication Protocol (SAS)," *IEICE TRANSACTIONS on Communications*, vol. E83-B, no. 6, pp. 1363-1365, June, 2000.

[4] T. Tsuji, T. Kamioka, and A. Shimizu, "Simple and Secure Password Authentication Protocol Ver.2 (SAS-2)," *IEICE TRANSACTIONS on Communications*, vol. 102, no. 314, pp. 7-11, September, 2002.

[5] T. C. Yeh, H. Y. Shen, and J. J. Hwang, "A Secure One-Time Password Authentication Scheme Using Smart Cards," *IEICE TRANSACTIONS on Communications*, vol. E85-B, no. 11, pp. 2515-2518, November, 2002.

[6] V. Goyal, A. Abraham, S. Sanyal, and S. Y. Han, "The N/R One Time Password System," in *Proc. of International Conference on Information Technology: Coding and Computing*, pp. 733-738, April 4-6, 2005. Article (CrossRef Link)

[7] H. Zhang, and Y. Zhu, "Self-Updating Hash Chains and their Implementations," in *Proc. of 7th International Conference on Web Information Systems Engineering*, pp. 387-397, October 23-26, 2006. Article (CrossRef Link)

[8] A. G. Chefranov, "One-Time Password Authentication with Infinite Hash Chains," *Novel Algorithms and Techniques in Tele-Communications, Automation and Industrial Electronics*, pp. 283-286, 2008. Article (CrossRef Link)

[9] X. Wang, J. Zhang, S. Wang, and et al, "A Key Agreement Scheme for Mobile Ad Hoc Networks Based on Password Authentication," *Journal of Software*, vol. 17, no. 8, pp. 1811-1817, August, 2006. Article (CrossRef Link)

[10] O. Goldreich, *Foundation of Cryptography: Basic Tools*, Cambridge University Press, pp. 64-74, 2001. Article (CrossRef Link)

[11] D. H. Yum, and P. J. Lee, "Cryptanalysis of Yeh-Shen-Hwang's One-Time Password Authentication Scheme," *IEICE TRANSACTIONS on Communications*, vol. E88-B, no. 4, pp. 1647-1648, April, 2005. Article (CrossRef Link)

[12] C. Li, and M. Hwang, "A Lightweight Anonymous Routing Protocol without Public Key En/Decryptions for Wireless Ad Hoc Networks," *Information Sciences*, vol. 181, issue 23, pp. 5333-5347, December, 2011. Article (CrossRef Link)

[13] D. He, M. Ma, Y. Zhang, and et al, "A Strong User Authentication Scheme with Smart Cards for Wireless Communications," *Computer Communications*, vol. 34, issue 3, pp. 367-374, March, 2011. Article (CrossRef Link)

[14] C. Li, and C. Lee, "A Novel User Authentication and Privacy Preserving Scheme with Smart Cards for Wireless Communications," *Mathematical and Computer Modelling*, vol. 55, issue 1-2, pp. 35-44, January, 2012. Article (CrossRef Link)

[15] K. G. Paterson, and D. Stebila, "One-Time-Password-Authenticated Key Exchange," in *Proc. of 15th Australasian Conference on Information Security and Privacy*, pp. 264-281, July 5-7, 2010. Article (CrossRef Link)

[16] M. H. Eldefrawy, M. K. Khan, and K. Alghathbar, "One-Time Password System with Infinite Nested Hash Chains," in *Proc. of International Conferences on Security Technology, Disaster Recovery and Business Continuity*, pp. 161-170, December 13-15, 2010. Article (CrossRef Link)

Feng XU is a Professor in the College of Computer and Information at Hohai University. His main research interests are trusted computing and network information security.

Xin LV is a Post doctorate in the College of Water Conservancy and Hydropower Engineering at Hohai University. His research is mainly focused on digital signature applications and security in e-commerce

Qi Zhou is a graduate student in the College of Computer and Information at Hohai University. She majors in cloud computing and information security.

Xuan Liu is a Ph.D candidate in the College of Computer & Information at Hohai University. She majors in cloud computing and information security.