Key exchange protocol based on the private key

Tong Yi, Minyong Shi, Wenqian Shang

School of Computer Science, Communication University of China, Beijing 100024, China

Keywords: key exchange protocol; private key; server key disclosure attack; dictionary attack.

Abstract. At present, most existing key exchange protocols for three-party (3pake) are easily attacked by server key disclosure, dictionary attack and so on. This paper analyses the weakness of the previous work , and presents a key exchange for three-party based on the private key .This method can resist common attack through safety analysis, and the efficiency analysis result shows its efficiency. On the basis of the new 3pake, a cross-realm key exchange protocol (c2c-pake) will be also presented, the new c2c-pake only needs 3 rounds to make users generate a common session key without asymmetric or symmetric encryption, and it can be proved safe through safety analysis.

Introduction

Key exchange protocol is able to make the two sides of communication generate a common session key so that users can communicate with each other in insecure channel safely. 2010 , paper[1] presented a password-based key exchange protocol, which can resist server key disclosure attack , dictionary attack, however the protocol uses the public key of the server, so excessive consumption of calculation is generated. Paper [2] presented a verifier-based key exchange protocol for three-party, but the protocol is vulnerable to server key disclosure attack.2011 paper[3] presented a efficient 3pake, but we point out the protocol cannot resist impersonation attack subsequently, and presented a improved protocol[4]. However, this improved protocol sill cannot resist off-line dictionary attack and server key disclosure attack. So on the basis of the previous works, this paper will present a new three-party key exchange protocol which can be proved to be safe and efficient. And the new three-party key exchange protocol can be expanded into a cross-realm key exchange protocol, different from the existing c2c-pake[5][6][7], the new c2c-pake uses hash algorithm instead of symmetric and asymmetric encryption, and can make two users from different realms generate a common session key in only 3 rounds.

Analysis of protocol in paper [4]

Review of protocol in paper [4]

PWA,PWB are the passwords of user A,B respectively, assume A wants to communicate with B, the process of protocol is as follow:

Step 1:A gets random numbers $a,t \in Z_p^*$, calculates that X=g^a, X_A=PW_A*X, Z_A=H_1(A,B,X,t,PW_A), sends {Z_A,A,B,X_A,t},{A,t} to server S and user B;

Step 2: After receiving message from A ,S gets X with X=X_A/PW_A, and verifies the identity of A with $Z_A \overset{?}{=} H_1(A,B,X,t,PW_A)$, if the validation fails, the protocol stops; Else, S computes X_S=X^s*PW_B, Y_S=g^s,Z_S=H_1(A,B,X^s,t,PW_B), sends {A,B,X_S,Y_S,Z_S} to user B;

Step 3:After receiving message from S, B computes K_B=X_S/PW_B, and verifies the identity of server S with $Z_S \overset{?}{=} H_1(A,B,K_B,t,PW_B)$, if the validation fails, the protocol stops; Else, gets random number $b \in Z_p^*$,computes K_{BS}=Y_s^b ,Z_{BS}=H_1(A,B,K_{BS},t), generates session key K= K_B^b , and then sends {A,B,K_{BS},Z_B} to user A;

A verifies $Z_B \overset{?}{=} H_1(A,B,K_{BS},t)$, if the validation successes, generates session key K= K_{BS}^a .

Review of protocol in paper [4]

(1) Offline dictionary attack. If adversary E gets message {Z_A,A,B,X_A,t} which A sends to S , E can use a random number PW_E instead of password PW_A of user A, through $Z_A \overset{?}{=} H_1(A, B, X_A / PW_E, t, PW_E)$, E can get the real password of A with exhaustive method. Similarly, If E gets message Z_S which S sends to B , E can also get password of B through offline dictionary attack.

(2)Sever key disclosure attack. Almost all the password-based key exchange protocol is vulnerable to sever key disclosure attack, in this protocol , if adversary E gets the password of user C, E can pose as C to communicate with others, or pose as other user and server to communicate with C easily.

3pake based on private key

Description of protocol

First, all the parameters of the protocol will be introduced briefly:
S: a trusted third server;
A,B: Two users;
PW_A: password of user A, stored on A and server S;
PW_B: password of user B, stored on B and server S;
s_A: long-term private key of user A;
s_B: long-term private key of user B;
s_{server}: long-term private key of server S;
s_AP , s_BP , $s_{server}P$: three open parameters, P is the generating element of finite cyclic additive group G, which's order is q;

Assume user A wants to initiate a conversation with user B, the process of the protocol is as follows:

Round 1: A gets random numbers $a, t \in Z_p^*$, calculates g^a, $s_As_{server}P$, and Z_{AS}:

$$Z_{AS}=H(PW_A, s_As_{server}P, g^a, t, A, B)$$

Then A sends {g^a, Z_{AS},t,A,B}to server S and sends {A} to B;

Round2: After receiving the messages from user A, S computes $s_{server}s_AP$, verifies :

$$Z_{AS} \overset{?}{=} H\left(PW_A,\ s_{server}s_AP,\ g^a, t,\ A, B\right)$$

If the validation fails, the protocol stops; else gets random number $s \in Z_p^*$,calculates $S_{server}S_BP$, g^{sa}, and Z_{SB}:

$$Z_{SB} = h(g^{sa}, PW_B, s_{server}s_BP, t, A),$$

Then S sends (Z_{SB},t,A,g^{sa}) to user B;

After received message from A, B gets random number $b \in Z_p^*$,computes g^b, $s_Bs_{server}P$, and Z_{BS}:

$$Z_{BS} = H(PW_B, s_Bs_{server}P, g^b)$$

Then B sends{g^b, $Z_{BS,}$ R_A) to S;

Round3: After receiving message from B, S verifies:

$$Z_{BS} \overset{?}{=} H(PW_B, s_{server}s_BP, g^b)$$

If validation fails , stops the protocol; else computes g^{bs}, and Z_{SA}:

$$Z_{SA}=H(PW_A, s_{server}s_AP, g^{bs})$$

Then S sends {Z_{SA},g^{bs}} to A;

After receiving message from server, B verifies:

$$Z_{SB} \overset{?}{=} h(g^{sa}, PW_B, s_Bs_{server}P, t, A)$$

If validation successes, calculates session key k=g^{asb};

After receiving message from server, A verifies

$$Z_{SA} \overset{?}{=} H(PW_A, s_A s_{server} P, g^{bs})$$

If validation successes, generates the common session key k=g^{asb};

Safety analysis

(1) Offline dictionary attack.

Transmitted message (ZAS,ZSB,ZBS,ZAS) all have two or more unknown parameters for attacker, so attacker cannot get users' password or private key through brute force.

(2) Server key disclosure attack.

Assume adversary E gets PWA of session initiator A from server, E attempts to pose as A to initiate a conversation, because E dose not know private key sA of A , cannot generate sAsserverP correctly ,so the verification $Z_{AS} \overset{?}{=} H(PW_A, s_{server} s_A P, g^a, t, A, B)$ included by S will fail ;

If E attempts to pose as server S and user B to communicate with A, he cannot pass the validation in round 3 because private key sserver of server S is unknown. Another situation, if E gets the password PWB of session responder B, he can also not pose as B or other users to communicate, because both the private key sB and sserver are unknown for E.

(3) Forward security.

Because each session key consists of random number, and Diffie-Hellman problem is hard, even if password and private key of user is got by adversary E, previous session key and communication content is sill safe. So the protocol has forward security.

(4)Replay attack.

Because there is fresh identifier t, it is hard for adversary to impact the operation of protocol through sending the old messages.

(5)Man-in-the-middle attack.

Adversary can not pose as user to communicate with server, because the private key and password of user is unknown, he cannot pass the validation on server:

$$\mathrm{Z_{AS}} \overset{?}{=} \mathrm{H}\left(\mathrm{PW_A},\ \mathrm{s_{server} s_A P},\ \mathrm{g^a}, t,\ A, B\right)$$

$$Z_{BS} \overset{?}{=} H(PW_B, s_{server} s_B P, g^b)$$

On the other hand, the adversary also cannot pose as server to communicate with user, because he doesn't know the password of user and private key of server. So the adversary can not use man-in-the-middle-attack to influence the operation of protocol.

(6)The insider attack.

Assume C is a legitimate user with password PW_C and private key s_C, he attempts to pose as the session initiator A to communicate with user B, C computes g^c, $s_C s_{server} P$, and $Z_{CS:}$

$$Z_{CS}=H(PW_C, s_C s_{server} P, g^c, t, A, B)$$

C sends {g^c, Z_{CS},t,A,B},{A}to server S and B, respectively. However, C cannot pass the validation on S, because

$$\mathrm{H}\left(\mathrm{PW_C},\ \mathrm{s_C s_{server} P},\ \mathrm{g^c}, \mathrm{t}, \mathrm{A}, \mathrm{B}\right) \neq \mathrm{H}\left(\mathrm{PW_A},\ \mathrm{s_A s_{server} P},\ \mathrm{g^c}, \mathrm{t}, \mathrm{A}, \mathrm{B}\right)$$

Obviously, user C cannot pose as session initiator A to communicate. Similarly, C also can not pose as the session responder B to communicate. So the protocol can resist the insider attack.

Efficiency analysis

Table 1 Efficiency comparison

Protocol in	[3]			[4]			this paper		
	A	B	S	A	B	S	A	B	S
index operation	4	2	4	2	2	2	2	2	2
hash	3	2	1	2	2	2	2	2	4
inverse operation	9	0	1	0	1	1	0	0	0
multiplication	2	0	0	1	0	1	1	1	2
round		4			3			3	

From Table 1 we can see that the new protocol is more efficient than protocol of [3], compared with the protocol of [4], the new 3pake is almost the same in efficiency, but the safety performance has been enhanced.

The c2c-pake presented by this paper

Description of protocol

A,B: two users from different realms;
SA: a trusted third server in the realm of user A;
SB: a trusted third server in the realm of user B;
PWA: password of user A, stored on A and server SA;
PWB: password of user B, stored on B and server SB;
sA1, sA2: two long-term private keys of user A;
sB1, sB2: two long-term private keys of user B;
sserverA1, sserverA2,: two long-term private keys of server SA;
sserverB1, sserverB2,: two long-term private keys of server SB;
sA1P , sA2P ,sB1P , sB2P ,sserverA1P, sserverA2P, sserverB1P, sserverB2P: several open parameters;

Assume user A wants to communicate with B, the process of protocol is as follows:

Round 1: A gets random numbers $a,t \in Z_p^*$ computes ga,sA1sserverA1P, and Z_{AS_A} :

$$Z_{AS_A} = H(g^a, PW_A, s_{A1}s_{serverA1}P, t, A, B)$$

Then A sends { Z_{AS_A} ,ga,t,A,B}to SA, meanwhile, sends {A,sA1P,sA2P} to user B;

Round 2: After receiving message from A, server SA computes sserverA1sA1P,verifies that

$$Z_{AS_A} \overset{?}{=} H(g^a, PW_A, s_{serverA1}s_{A1}P, t, A, B)$$

If the validation fails, stops the protocol; else, server S_A computes that $s_{serverA1}s_{serverB1}P$, $s_{serverA2}s_{serverB2}P$, and $Z_{S_AS_B}$:

$$Z_{S_AS_B} = H(g^a, s_{serverA1}s_{serverB1}P, s_{serverA2}s_{serverB2}P, t, A, B) \ ,$$

Then S_A sends { $Z_{S_AS_B}$,t,g^a,A,B} to S_B;

After receiving message from A, B gets random number $b \in Z_p^*$, computes $s_{B1}s_{serverB1}P$, g^b, and Z_{BS_B} :

$$Z_{BS_B} = H(g^b, PW_B, s_{B1}s_{serverB1}P, B)$$

Then B sends {g^b, Z_{BS_B} ,B} to server S_B;

Round 3: After receiving message from server S_A and user B, server S_B computes $s_{serverB1}s_{serverA1}P$,$s_{serverB2}s_{serverA2}P$,$s_{serverB1}s_{B1}P$,verifies that

$$Z_{S_AS_B} \overset{?}{=} H(g^a, s_{serverB1}s_{serverA1}P, s_{serverB2}s_{serverA2}P, t, A, B), Z_{BS_B} \overset{?}{=} H(g^b, PW_B, s_{serverB1}s_{B1}P, B)$$

If both the validation success, S_B gets $s \in Z_p^*$, computes that g^{sb},g^{sa},$s_{serverB1}s_{A1}P$, $s_{serverB2}s_{A2}P$, Z_{S_BA}, Z_{S_BB}. Z_{S_BA} and Z_{S_BB} are as follows:

$$Z_{S_BA} = H(g^{sb}, s_{serverB1}s_{A1}P, s_{serverB2}s_{A2}P), Z_{S_BB} = H(g^{sa}, t, PW_B, s_{serverB1}s_{B1}P),$$

Then S_B sends {Z_{S_BA},g^{sb}},{Z_{S_BB},g^{sa},t} to user A and B ,respectively; else, the protocol stops

After receiving message from S_B, A computes $s_{A1}s_{serverB1}P$, $s_{A2}s_{serverB2}P$,verifies

$$Z_{S_BA} \overset{?}{=} H(g^{sb}, s_{A1}s_{serverB1}P, s_{A2}s_{serverB2}P),$$

If the validation successes, generates the session key g^{asb};

For user B, after receiving message from S_B, verifies:

$$Z_{S_BB} \overset{?}{=} H(g^{sa}, t, PW_B, s_{B1}s_{serverB1}P)$$

If the validation successes, computes the common key g^{bsa}.

Safety analysis

(1) Off-line dictionary attack.

The offline dictionary is infeasible because all of the message transmitted have more than two unknown parameters for adversary.

(2) The insider attack.

Assume adversary gets password and private key of user C, he attempts to pose as A to initiate a session, E calculates ge, $Z_{AS_A} = (g^e, PW_C, s_{C1}s_{severA1}P, t,\ A, B)$, sends ($Z_{AS_A}$,ge,t,A,B) to SA. However, the validation included by SA will fail because of $(g^e, PW_C, s_{C1}s_{severA1}P, t, A, B) \neq (g^e, PW_A, s_{A1}s_{severA1}P, t, A, B)$. If E wants to pose as B to communicate with other user as a responder, also cannot pass the validation included by SB.

(3) Sever key disclosure attack.

If adversary E gets password PWA of session initiator A, E attempts to pose as A to communicate with other user, E cannot pass the validation included by SA in round 2, because the private key sA1 is unknown for E; If E wants to pose as session responder B and SB to respond A, the private key sserverB1, sserverB2 is unknown for E, which makes the validation included by A fail. In another way , when E gets password PWB of session responder B , if E attempts to pose as B to respond other user, validation $Z_{BS_B} \overset{?}{=} H(g^b, PW_B, s_{serverB1}s_{B1}P, B)$ included by server SB will fail because private key sB1 is unknown for E; If E attempts to pose as session initiator A and server SB to communicate with B , the private key sserverB1 is unknown ,which makes validation $Z_{S_BB} \overset{?}{=} H(g^{sa}, t, PW_B, s_{B1}s_{serverB1}P)$ included by B fail. So the protocol can resist sever key disclosure attack.

(4)Replay attack.

Because there is fresh identifier t, the replay attack is infeasible.

(5) Forward security.

Several temporary random numbers are add into the calculation of the session key, in addition ,there is intractability of the Diffie-Hellman problem, so even if adversary gets the password and private key of user , previous communication content is still unknown for the adversary.

(6) Impersonation attack.

Adversary E cannot pose as the session initiator A to communicate because both the password and private key of A are unknown for E; Similarly, E also cannot pose as the session responder B to respond the session.

Efficiency analysis

Table 2 Efficiency comparison (c2c-pake)

Protocol in	[5]	[6]	this paper
Asymmetric encryption	√	×	×
Symmetric encryption	√	√	×
Rounds	6	8	3

From the Table 2 we can see that the c2c-pake presented by this paper has no use for asymmetric or symmetric encryption, and can generates a common session key in only three rounds, so the new protocol is more efficient.

Summary

This paper introduces private keys of server and user into 3pake, and presents a improved protocol on the basis of a existing one. The new protocol is almost as same as the old one in efficiency, but the safety has been improved greatly. And on the basis of the new 3pake, an efficient c2c-pake without asymmetric or symmetric encryption is presented, and it is proved that the c2c-pake can resist all kinds of common attacks. In future work, we will further improve the efficiency of protocols, and plan to reduce the round number of cross-realm key exchange protocol.

Acknowledgements

In this paper, the research was sponsored by the project of computer science and technology major comprehensive reform（ZL140103）and the Communication University of China Engineering Course planning project（XNG1436）

References

[1] Xiao-Fei Ding, Chuan-Gui MA. The Three-party Password-Authenticated Key Exchange Protocol with Stronger Security[J].Chinese Journal of Computers, 2010, 33(1):111-118,

[2] Shunbo XIANG, Wende KE. Efficient password-authenticated tripartite key exchange protocol[J]. Computer Engineering and Applications,2012, 48(16):107-110.

[3] Jun CHEN, Feng LIU, Wei GAO. Password-based Simple Three-party Key Exchange Protocol [J]. Computer Engineering [J],2011,37(8):112-114.

[4] Tong Yi , Hongchao Chen , Dailin Wu. Improved Password-based key exchange Protocol for Three-parties [J]. Computer Applications and Software, 2013,30 (4):313-315.

[5]Guang-wei Liu, En-guang Zhou, Hong Yan, Fu-cai Zhou. An Improved Cross-Realm Client-to-Client Password-Authenticated Key Exchange Protocol[J].Journal of Northeastern University(Nature Science),2009,30(1):42-45.

[6] Jing-feng Li, Wei-feng Guo, Lai-shun Zhang, Yun-peng Li. Cryptoanalysis and Improvement of Verifier-based Key Agreement Protocol in Cross-Realm Setting[C], 2011 Seventh International Conference on Computational Intelligence and Security(CIS'7), IEEE Press, December,2010, 3-4 Dec. 2011

[7] Po-Jen Chuang and Yi-Ping Liao. Efficient and Secure Cross-Realm Client-to-Client Password-Authenticated Key Exchange[C].2012 26th IEEE International Conference on Advanced Information Networking and Applications (AINA'26), IEEE Press, 26-29 March 2012.