Cryptanalysis of a client-to-client password-authenticated key agreement protocol

Fengjiao Wang and Yuqing Zhang

(National Computer Network Intrusion Protection Center, GSCAS, Beijing, China)

***Abstract*—Recently, Byun *et al*. proposed an efficient client-to-client password-authenticated key agreement protocol (EC2C-PAKA), which was provably secure in a formally defined security model. This letter shows that EC2C-PAKA protocol is vulnerable to password compromise impersonate attack and man-in-the-middle attack if the key between servers is compromised.**

***Index Terms* — Cryptanalysis, EC2C-PAKA, impersonate attack, man-in-the-middle attack.**

I. INTRODUCTION

Using a human memorable password to achieve authentication and agree on a common secret value (a session key) over an insecure open network, is a popular method because of its easy-to-memorize property. With the rapid development of modern communication environments in the fields such as mobile networks, home networking and etc., there is a need to construct a secure end-to-end channel between clients, which is quite different from the existing client-server model that based on a pre-shared password. Byun *et al*. firstly considered the cross-realm scenario in [1]. Later, their scheme was found to be flawed in [2], and attacks and improvements were successively given, such as [3] etc. In 2007, Byun *et al*. [4] proposed an efficient client-to-client password-authenticated key agreement protocol (EC2C-PAKA) which was provably secure in a formally defined security model. As has been indicated in [4], passwords may be revealed inadvertently during a conversation or by malicious insider adversaries. The previously used session keys may also be lost for various reasons such as hijacking or careless clients. Therefore, in proposing the protocol, Byun *et al*. proved the security of their protocol under the assumption that realistic active adversaries could get session keys and passwords.

However, we found that the EC2C-PAKA protocol is insecure under this assumption. Password compromise enables an adversary a chance to impersonate a valid client. Furthermore, the leakage of the symmetric encryption key between servers enables the adversary to launch a man-in-the-middle attack to the communication between clients.

This letter reviews the EC2C-PAKA protocol proposed by Byun *et al.* [5] and shows that it suffers from password compromise impersonate attack and key compromise man-in-the-middle attack. We note that the password compromise impersonate attack cannot be prohibited only by sharing a password between client and server, nor does the key compromise man-in-the-middle attack by adopting symmetric encryption between servers.

II. REVIEW OF *Byun et al.*'s EC2C-PAKA PROTOCOL

A concise view of EC2C-PAKA protocol proposed by Byun *et al.* [4] is given in figure 1. Readers are referred to [4] for details. Throughout the letter, notations are used as in Table 1.

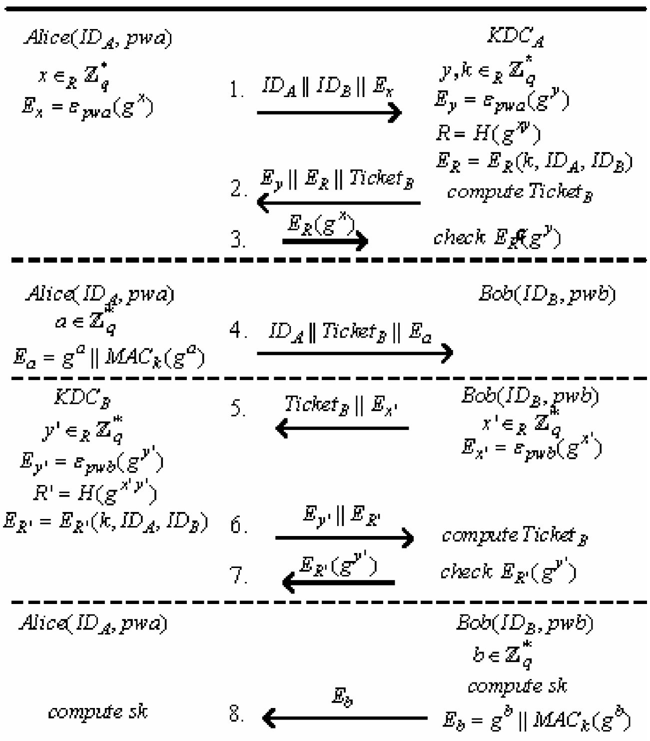

Figure1. EC2C-PAKA protocol

Let the public parameters of the system be a prime order q and a generator g over a cyclic group $\mathbb{Z}_q^*$.

TABLE 1. NOTATIONS USED IN EC2C-PAKA

Alice, Bob	The client belongs to different realms.
KDC_A, KDC_B	The server in each realm, respectively.
ID_A, ID_B	Identity of client Alice and Bob, respectively.
pwa, pwb	Passwords shared between client an server in local realm.
K	Pre-distributed encryption key between servers.
k	A short-term random key.
x, y, a, b, x', y'	Random numbers.
m	A message.
$E_k(m)$	A symmetric encryption of *m* using a symmetric key *k*.
L	A lifetime of a $Ticket_X$.
$Ticket_B$	$E_K(k, ID_A, ID_B, L)$, computed by KDC_A.
$MAC_k(m)$	A message authentication code on *m* using a secret key *k*.
$H(m)$	A pseudo-random value of *m* using a pseudo-random function *H*.

III. PASSWORD COMPROMISE IMPERSONATE ATTACK ON EC2C-PAKA

Assuming that valid client *Alice* wants to establish a shared session key with client *Bob* in a different realm, and an adversary *Eve* has got *Alice*'s password *pwa* making use of the *Corrupt* ( ) query defined in the security model in [4]. Obviously, *Eve* can impersonate *Alice* to communicate with *Bob*, which is treated as a trivial attack. We now show that on obtaining *Alice*'s password, *Eve* is able to impersonate *Bob* to communicate with *Alice*. The attack proceeds as follows:

(1) *Eve* hijacks the message ($ID_A||ID_B||E_x$) from *Alice* to KDC_A, chooses $y', k' \in \mathbb{Z}_q^*$ randomly and computes $E_{y'} = \varepsilon_{pwa}(g^{y'})$, $R' = H(g^{xy'})$, $E_{R'} = E_{R'}(k', ID_A, ID_B)$. Then *Eve* chooses a random number as $Ticket_B$ and sends $E_{y'} \,||\, E_{R'} \,||\, Ticket_B$ to *Alice*.

(2) On receiving the message ($E_{y'} \,||\, E_{R'} \,||\, Ticket_B$) from *Eve*, *Alice* obtains $g^{y'}$ by decrypting $E_{y'} = \varepsilon_{pwa}(g^{y'})$ with her own password *pwa*, computes $R' = H((g^{y'})^x)$ and decrypts $E_{R'} = E_{R'}(k', ID_A, ID_B)$ to check the validity, which makes *Alice* believe that this message is from KDC_A. Meanwhile, *Alice* computes and sends $E_{R'}(g^x)$ to KDC_A that is also hijacked by *Eve*.

(3) *Alice* chooses $a \in \mathbb{Z}_q^*$ randomly, computes $E_a = g^a \,||\, MAC_{k'}(g^a)$ and sends it to *Bob*. *Eve* hijacks the message, chooses $b' \in \mathbb{Z}_q^*$, computes $E_{b'} = g^{b'} \,||\, MAC_{k'}(g^{b'})$ and sends it back to *Alice*.

(4) *Alice* checks the validity of $E_{b'} = g^{b'} \,||\, MAC_{k'}(g^{b'})$ with *k'*, if it is valid, *Alice* believes that she is communicating with *Bob* who is actually *Eve*. Finally, a session key *sk* is computed and shared between *Alice* and *Eve*, which *Alice* believes is shared with *Bob*.

The process is shown as in Figure 2, where the messages in the dashed pane are hijacked and utilized by adversary *Eve*. *Eve*(KDC_A) denotes that *Eve* acts as KDC_A, and *Eve*(*Bob*, KDC_B) denotes that *Eve* acts as *Bob* and KDC_B.

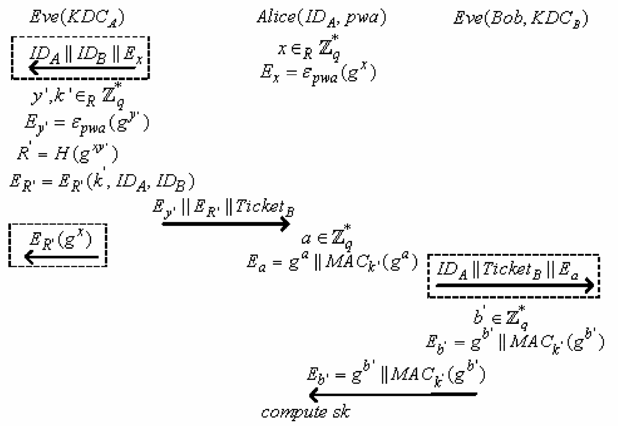

Figure 2. Passwords compromise impersonate attack

IV. KEY COMPROMISE MAN-IN-THE-MIDDLE ATTACK

In addition to the above mentioned attack, EC2C-PAKA protocol is also vulnerable to key compromise man-in-the-middle attack. Assuming that the adversary *Eve* has got the encryption key *K* between servers, then the attack proceeds as follows:

(1) Adversary *Eve* wiretaps the communication between client *Alice* and her server KDC_A, decrypts $Ticket_B$ with the encryption key *K* he has got, and gets the random *k*.

(2) *Eve* hijacks the message $E_a = g^a \,||\, MAC_k(g^a)$ from *Alice* to *Bob*, replaces it by $E_{a'} = g^{a'} \,||\, MAC_k(g^{a'})$, and sends it to *Bob*.

(3) On receiving the message from *Eve* (*Bob* thinks that it is from *Alice*), *Bob* communicates with KDC_B as usual, and finally *Bob* computes and sends *Alice* $E_b = g^b \| MAC_k(g^b)$ which is hijacked by *Eve* and then replaced by $E_{b'} = g^{b'} \| MAC_k(g^{b'})$.

After finishing the execution of the protocol, *Alice* and *Bob* think that they have shared a session key between them. However, both of them actually shared a different key with adversary *Eve.*

The process is shown as in Figure 3, where the messages in the dashed pane are hijacked and utilized by adversary *Eve*.

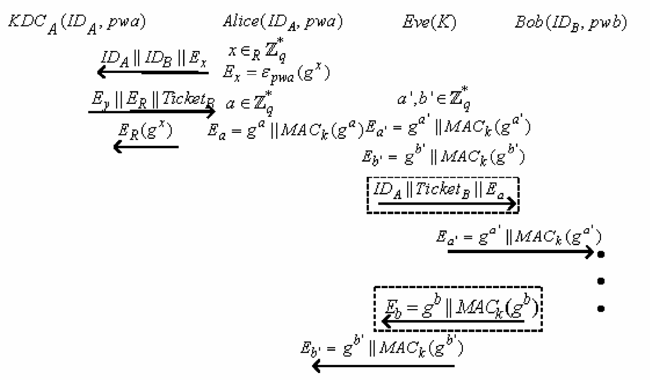

Figure 3. Key compromise man-in-the-middle attack

V. DISCUSSION

It is worthwhile to discuss why EC2C-PAKA [4] falls to the two attacks given in this letter. Firstly, although they assumed that the adversary could get client's password by *Corrupt* ( ) query, no security definition was given to resist the impersonate attacks, and therefore the goal that our first attack achieves was not considered in [4]. That is to say, the attacking approach considered in [4] making use of a password is too specific to catch our attack.

Secondly, a client whose password is compromised is not able to distinguish between interactions with other honest parties or adversary; therefore we suggest that some authentication information of server (public key) should be kept by a client, besides the shared password.

Another point is that the long-term symmetric key K is the only security association between two servers, and therefore the compromise of key K can lead to several attacks. To avoid this kind of attacks, public key encryption between servers is suggested in this letter.

VI. CONCLUSION

The main goal of client-to-client key agreement protocol is to establish a sole shared session key between two clients. This letter shows that the EC2C-PAKA protocol proposed by Byun *et al*. is still vulnerable to password compromise impersonate attack and key compromise man-in-the-middle attack.

ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China (60573048,60773135,90718007) and the High Technology Research and Development Program of China (863 program) (2007AA01Z427，2007AA01Z450). Meanwhile, Professor Jing Xu is thanked for enlightening me with this idea.

REFERENCES

[1] J. W. Byun, I. R. Jeong, D. H. Lee, and C.S. Park. Password-authenticated key exchange between clients with different passwords. In *ICICS* 2002, LNCS, Springer-Verlag, Vol.2513, pp.134-146.

[2] S. Wang, J. Wang, and M. Xu. Weaknesses of a password-authenticated key exchange protocol between clients with different passwords. In *ACNS* 2004, LNCS, Springer-Verlag, Vol.3089, pp. 414-425.

[3] J. Kim, S. Kim, J. Kwak, and D. Won. Cryptanalysis and improvement of password authenticated key exchange scheme between clients with different passwords. In ECCSA 2004, Part *I*, LNCS ,Vol3043, pp.895-902.

[4] J. W. Byun, D. H. Lee, and J.I. Lim. EC2C-PAKA: An efficient client-to-client password-authenticated key agreement. *INFORMATION SCIENTCES*, Vol.177, pp. 3995-4013, 2007.