Weaknesses of a Password-Authenticated Key Exchange Protocol between Clients with Different Passwords*

Shuhong Wang, Jie Wang, and Maozhi Xu

School of Mathematical Sciences, Peking University, China
{wshong, wangj, mzxu}@math.pku.edu.cn

Abstract. A password-authenticated key exchange scheme allows two entities, who only share a memorable password, to authenticate each other and to agree on a cryptographic session key. Instead of considering it in the classic *client and server* scenarios, Byun *et al.* recently proposed a password-authenticated key exchange protocol in a cross-realm setting where two clients in different realms obtain a secret session key as well as mutual authentication, with the help of respective servers. In this paper, we first point out that the proposed protocol is not secure, due to the choice of invalid parameters (say, subgroup generator). Furthermore, we show in detail that, even with properly chosen parameters, the protocol has still some secure flaws. We provide three attacks to illustrate the insecurity of the protocol. Finally, countermeasures are also given, which are believed able to withstand our attacks.

Keywords: Password-authenticated key exchange, Cross-realm setting, Security, Dictionary attacks.

1 Introduction

The oldest and probably the most important problem of cryptography is how to provide *private and reliable* communication among parties in a public communication channel. This significant problem is commonly reduced to the problem of generating a secure session-key. Certainly, there are many ways to establish secure session keys with the initial set-up assumption of the existence of Public Key Infrastructure (PKI). In reality, however, it is more convenient and more natural if two parties are allowed to obtain such a strong cryptographic session key without relying on the PKI, but with only a pre-shared memorable password. The solution of the problem in this scenario is known as Password-Authenticated Key Exchange (PAKE).

The concept of PAKE was first introduced by Bellovin and Merritt in 1992 [4] known as Encrypted Key Exchange (EKE) which is improved later in [5]. Since then, a number of PAKE protocols are proposed in the literature [3,11,

* Supported by the National Natural Science Foundation of China, NSFC 90104004.

M. Jakobsson, M. Yung, J. Zhou (Eds.): ACNS 2004, LNCS 3089, pp. 414–425, 2004.
© Springer-Verlag Berlin Heidelberg 2004

12,6,15,21,22,30,31] with different initial assumptions and communication workloads. As far as security is concerned, PAKE protocols are often vulnerable to dictionary attack (brute-force password search) since the possible space of memorable passwords is too small. Some security analyses of these protocols can be found in many literatures, for example, [2,26,19,29]. In practices, most of these proposed PAKE protocols are presented in the context that the two involved entities are client and server respectively and they share a common password [13,22,15]. Although some of them [11,10,28,16,17] are extended to a three-party EKE protocol, in which a trusted server exists to mediate between two communication parties to allow mutual authentication, they are less considered in a cross-realm setting like in kerberos system [27,14].

Recently in *ICICS'02*, based on the scheme in [8], Byun *et al.* designed several password-authenticated key exchange schemes between clients with different passwords, called *Client-to-Client Password-Authenticated Key Exchange* (C2C-PAKE). In these scheme, two clients (could in separate realms) fulfill the authenticated key exchange relying only on their distinct passwords and servers, without any other prior shared secret. Three C2C-PAKE schemes are presented in their paper [7]. One (CR-C2C, hereinafter) is for a cross-realm setting where two clients are in two different Kerberos realms and hence two servers (who are connected with a symmetric key) are involved. The other two are for a single-server setting where two clients are in the same realm: the Single-server Ticket Type (ST-C2C) and the Single-server Non-Ticket Type[1]. They also newly defined the security notions according to their framework for the special settings, and claimed their schemes' security under those definitions.

The goal of this paper is to show some security flaws in [7]. We show that, on the one hand, the security definition in [7] is incomplete for the new framework. That is, in the protocol with a cross-realm setting, that one server can obtain the password of a client in another realm is not considered. On the other hand, the proposed protocols are insecure even under their incomplete security definitions. We illustrate several dictionary attacks for this purpose.

The rest of the paper is arranged as follows. In Sect. 2 we give the security definitions for the PAKE protocols in a cross-realm setting. Section 3 is devoted to review the original CR-C2C protocol, followed by our security analysis in Sect. 4. In Sect. 5, we further discuss the security of the ST-C2C protocol in a single-server setting with kerberos ticket. Finally, we conclude the paper with some counter measures to resist our attacks in Sect. 7 and Sect. 6 respectively.

2 Modes and Security Properties

The definition of formal security [3,6] for PAKE is somewhat technical. It means essentially that the best an active attacker can do is to guess passwords and to

[1] The later on which we are not going to discuss much more is similar to usual three-party EKE protocols where both parties (clients) share their passwords with the third trusted server only. (For more details on three-party EKE protocols, readers please refer to references [28,19]).

verify them one-by-one online through communication with an honest party. In particular, this implies that the attacker will not get any information that would allow an off-line dictionary attack. Note that when we say a PAKE protocol is subject to dictionary attack, it does not necessarily mean that the password can be found by brute force. It means that an attacker can get more information than random guess [2].

In [7], two distinct models of password-authenticated key exchange schemes (PAKE) were defined. One is called *Shared Password Authentication Model* (SPA), and the other *Different Password Authentication Model* (DPA). In SPA model, entities involved are a client and a server who share a common password. It is the case for most proposed PAKE protocols. In DPA model, we focus on the cross-realm scenario (CR-DPA, for short) where clients *Alice* and *Bob*, who are in different realms and possess distinct passwords, agree on a session key and authenticate each other with help of key distribution centers KDC_A and KDC_B. Here KDC_A and KDC_B who share a symmetric secrete cryptographic key are servers of (hence in the same realms as) *Alice* and *Bob* respectively. One can easily derives the single server DPA model (SS-DPA, for short) from CR-DPA by replacing KDC_A and KDC_B with one common server KDC. Indeed, SS-DPA is exactly the model of general three-party PAKE.

It is desirable for PAKE protocols (in both SPA and DPA models) to possess the following security attributes:

Known-key security: Each run of the protocol should result in a unique secret session key. The compromise of one session key should not compromise other session keys.

Forward secrecy: If passwords of one or more of the entities are compromised, the secrecy of previously established session keys should not be affected.

Key-compromise impersonation: Compromising passwords of any entities (clients or/and servers) should not enable the adversary to impersonate any other entities.

Unknown key share resilience: Client *Alice* should not be able to coerced into sharing a key with any client *Carol* when in fact she thinks that she is sharing the key with client *Bob*.

Key control: Any entities should not be able to force the session key to a preselected value.

Dictionary attack resilience: All passwords in the protocol must be strongly protected against a dictionary attack, and even if an attacker is given one password, other passwords must be prevented from such a attack.

In addition to above basic properties, more properties should be considered under the environments of DPA model. More precisely, the descriptions of some properties should be modified according to the new framework in DPA, especially in CR-DPA model. At least, we should consider the long-term private keys of entities instead of passwords only:

Forward secrecy - DPA: If long-term private keys (including clients' passwords and servers' cryptographic keys) of one or more of the entities are compromised, the secrecy of previously established session keys should not be affected.

Key-compromise impersonation - DPA: Compromising long-term private keys of any entities (clients or/and servers) should not enable the adversary to impersonate any other entities.

Dictionary attack resilience - DPA: All passwords in the protocol must be strongly protected against a dictionary attack, and even if an attacker is given one password, other passwords must be prevented from such an attack. Further more, the compromise of servers' shared symmetric key should not allow a dictionary attack either. And in the CR-DPA model, it is expected that any entity in one realm should not be able to mount a dictionary attack to other entities belongs to another realm.

3 The Review of the Protocol in a Cross-Realm Setting (CR-C2C)

In this section, we review the CR-C2C protocol in Sect. 4 of [7]. For convenience, we use the same notations and list them in Table 1.

Note that in the original paper of Byun et al., G is chosen as in Table 1. Subsequently g is a generator of a **subgroup** in $\mathbb{Z}_p^*$. However, it is commonly recognized that such a choice is very dangerous. We shall discuss this issue at length in section 4.1. Later, we think this flaw as a type error, and then properly take g as a generator of $G = \mathbb{Z}_p^*$.

3.1 The CR-C2C Protocol

By using notations listed in Table 1, the proposed C2C-PAKE protocol in a cross-realm setting (CR-C2C) can be described as follows (Fig. 1). This is an example of PAKE protocols under the CR-DPA model.

(1) $Alice \to KDC_A$: $ID(A), ID(B), E_{pwa}(g^x)$

(2) $KDC_A \to Alice$: $E_R(g^x \oplus g^r, ID(A), ID(B)), E_{pwa}(g^y), Ticket_B$

(3) $Alice \to Bob$: $Ticket_B, ID(A), L$

(4) $Bob \to KDC_B$: $Ticket_B, E_{pwb}(g^{x'}), ID(A), ID(B), L$

(5) $KDC_B \to Bob$: $E_{R'}(g^{pwa \cdot r \cdot r'} \oplus g^{x'}, ID(A), ID(B)), E_{pwb}(g^{y'})$ $E_{H_4(g^{pwa \cdot r})}(g^{pwb \cdot r \cdot r'})$

(6) $Bob \to Alice$: $E_{cs}(g^a), E_{H_4(g^{pwa \cdot r})}(g^{pwb \cdot r \cdot r'})$

(7) $Alice \to Bob$: $E_{sk}(g^a), E_{cs}(g^b)$

(8) $Bob \to Alice$: $E_{sk}(g^b)$

Fig. 1. The CR-C2C protocol (Cross-realm setting)

Table 1. Parameters and Notations used in C2C-PAKE Protocols.

Notation	Meaning
p, q	two large primes satisfy $q \mid p-1$
G	a subgroup of $\mathbb{Z}_p^*$ and $\|G\| = q$
g	a generator of G
$Alice, Bob$	two clients in two different realms
$ID(A), ID(B)$	identities of *Alice* and *Bob*
pwa, pwb	passwords memorized by *Alice* and *Bob*
KDC_A, KDC_B	two key distribution centers which store password files of *Alice* and *Bob* respectively
K	a symmetric key shared between KDC_A and KDC_B
$E_X(\cdot), D_X(\cdot)$	symmetric encryption and decryption under the symmetric key X
H_1, H_2, H_3, H_4, H_5	collision-resistant one-way hash functions (e.g, SHA-1)
x, y, r, b	ephemeral secrets in Z_p^* randomly chosen by *Alice* and KDC_A
x', y', r', a	ephemeral secrets in Z_p^* randomly chosen by *Bob*'s and KDC_B
$R = H_1(g^{xy})$	session key agreed between *Alice* and KDC_A
$R' = H_2(g^{x'y'})$	session key agreed between *Bob* and KDC_B
$sk = H_3(g^{ab})$	session key agreed between *Alice* and *Bob*
cs	$cs = H_5(g^{pwa \cdot pwb \cdot r \cdot r'})$ computed by both *Alice* and *Bob*
$Ticket_B$	the Kerberos ticket issued to *Alice* for service from *Bob*, $Ticket_B = E_K(g^{pwa \cdot r}, g^r, ID(A), ID(B), L)$
L	the lifetime of $Ticket_B$, $Ticket_B$ can be reused in L

3.2 Description of the CR-C2C Protocol

1. *Alice* choose $x \in Z_p^*$ randomly, computes and sends $E_{pwa}(g^x)$ to KDC_A together with $ID(A)$ and $ID(B)$ in (1).

2. KDC_A obtains g^x by decrypting $E_{pwa}(g^x)$, chooses $y, r \in Z_p^*$ randomly and computes $E_{pwa}(g^y)$ and $g^{pwa \cdot r}$. KDC_A makes $Ticket_B$ and also specifies L, a lifetime of $Ticket_B$. Then KDC_A sends $E_R(g^x \oplus g^r, ID(A), ID(B)), E_{pwa}(g^y)$ and $Ticket_B$ to *Alice*.
   Upon receiving the message from KDC_A, *Alice* computes a session key $R = H_1(g^{xy})$ and decrypts $E_R(g^x \oplus g^r, ID(A), ID(B))$ to find g^r.

3. *Alice* just forwards $Ticket_B, ID(A)$ and L to *Bob*.

4. *Bob* chooses $x' \in Z_p^*$ randomly and computes $E_{pwb}(g^{x'})$. Then he sends $E_{pwb}(g^{x'}), ID(A)$ and $ID(B)$ to KDC_B together with $Ticket_B$ and L.
   Upon the receipt of $Ticket_B$, KDC_B obtains $g^{pwa \cdot r}$ by decrypting $Ticket_B$. Note that KDC_B also can obtain g^r from this decryption.

5. KDC_B chooses $r^{'} \in Z_p^*$ randomly and computes $(g^{pwa\cdot\ r\cdot r^{'}})$. KDC_B also selects another random number $y^{'} \in Z_p^*$, and computes $R^{'} = H_2(g^{x^{'}y^{'}})$. Next KDC_B computes $E_{R'}(g^{pwa\cdot r\cdot r^{'}} \oplus g^{x^{'}}, ID(A), ID(B))$ using $R^{'}$, and sends $E_{R'}(g^{pwa\cdot r\cdot r^{'}} \oplus g^{x^{'}}, ID(A), ID(B))$, $E_{pwb}(g^{y^{'}})$ and $E_{H_4(g^{pwa\cdot r})}(g^{pwb\cdot r\cdot r^{'}})$ to *Bob*.

6. *Bob* decrypts $E_{pwb}(g^{y^{'}})$ to find $g^{y^{'}}$ and computes $R^{'} = H_2(g^{x^{'}y^{'}})$, and then decrypts $E_{R'}(g^{pwa\cdot r\cdot r^{'}} \oplus g^{x^{'}}, ID(A), ID(B))$ using $R^{'}$ to obtain $g^{pwa\cdot r\cdot r^{'}}$ from $g^{pwa\cdot r\cdot r^{'}} \oplus g^{x^{'}}$. He makes $cs = H_5(g^{pwa\cdot pwb\cdot r\cdot r^{'}})$. Then *Bob* chooses a random number $a \in Z_p^*$ and computes $E_{cs}(g^a)$. He finally sends $E_{cs}(g^a)$ and $E_{H_4(g^{pwa\cdot r})}(g^{pwb\cdot r\cdot r^{'}})$ to *Alice*.

7. *Alice* computes $H_4(g^{pwa\cdot r})$ with her pwa and g^r and uses it to decrypts $g^{pwb\cdot r\cdot r^{'}}$. *Alice* also can computes $cs = H_5(g^{pwa\cdot pwb\cdot r\cdot r^{'}})$ using $g^{pwb\cdot r\cdot r^{'}}$ and her password. Next, *Alice* selects $b \in Z_p^*$ randomly, and computes $sk = H_3(g^{ab})$ as well $E_{cs}(g^b)$. Finally she sends $E_{sk}(g^a)$ and $E_{cs}(g^b)$ for session key confirmation.

8. Upon the receipt of $E_{sk}(g^a), E_{cs}(g^b)$, *Bob* retrieves g^b and computes sk with g^b and a. Then he verifies g^a by decrypting $E_{cs}(g^a)$ with sk. And *Bob* also sends $E_{sk}(g^b)$ to *Alice* for session key confirmation. Till now the execution of protocol 1 completes.

4 Attacks on the CR-C2C Protocol

In this section, we analyze the security of the CR-C2C protocol by presenting three dictionary attacks.

First of all, we demonstrate the danger (it is a damage!) to chose generator g in a subgroup of Z_p^* (Attack 1). Then we consider g to be a generator of the whole group Z_p^*, and present other two attacks. Note that Attack 2 is also effective to the case where g is a subgroup generator.

In Attack 2, a malicious key distribution center in one (say,*Bob*'s) realm (KDC_B) can extract the passwords of the users belong to another (*Alice*'s) realm. Note that this attack can be looked as symmetric on the whole system's point of the view, that is to say, if it is *Bob* who requests the access to *Alice*'s service, then *Alice*'s key distribution center (KDC_A) can extract *Bob*'s password. It is this attack that makes us to extend the concept of security against dictionary attacks for password-authenticated key exchange protocols in cross-realm settings. Obviously, the protocol above does not satisfy the *Dictionary attack resilience - DPA* and *Key-compromise impersonation - DPA* requirements as desired.

The last attack is somehow technical and self-symmetric (i.e, in the same implementation, both *Alice* and *Bob* can reduce the passwords space of the opposing entity). Precisely, *Alice* can reduce *Bob*'s password space to half and *Bob* can excludes *Alice*'s passwords too, both succeed with a probability higher than $1-(\frac{3}{4})^t$ after implementing the CR-C2C protocol t times. This attack shows that the C2C-PAKE protocols are insecure under the dictionary attacks.

4.1 Attack 1

Suppose an attacker eavesdrops the implementation of the protocol. He can obtain the exchanged messages $E_{pwa}(g^x)$, $E_{pwa}(g^y)$, $E_{pwb}(g^{x'})$ and $E_{pwb}(g^{y'})$. Then he can mount an off-line dictionary attack to recover pwa and pwb. We only show the process of extracting password pwa as follows. It is the same for password pwb.

1. Decrypts the $E_{pwa}(g^x)$ using a candidate password pwa': $\widetilde{g^x} = D_{pwa'}(E_{pwa}(g^x))$.
2. Raises $\widetilde{g^x}$ to power q and checks whether 1 is obtained.
3. If 1 is obtained, excludes pwa' from *Alice*'s password space; Otherwise,
4. Chooses another password and repeats above steps until all the passwords are checked.

If the correct password is not found, one should continue this excluding process by decrypting ciphertext $E_{pwa}(g^y)$ with another candidate password. Note that a candidate password pwa' can not be excluded only if $D_{pwa'}(E_{pwa}(g^x))^q = 1$. We assume that the decryption results randomly in Z_p^* if the pwa' is incorrect. Then it is obvious that the probability of $D_{pwa'}(E_{pwa}(g^x))^q = 1$ is $\frac{q}{p-1}$. Consequently, the valid passwords space of both *Alice* and *Bob* will be reduced by a factor of up to $(\frac{q}{p-1})^2$, on average, through once eavesdropping of session execution. Over a number of sessions the space of valid passwords will be narrowed down to a single password at a logarithm rate.

4.2 Attack 2

As noted above, upon the receipt of $Ticket_B$, KDC_B can obtain $g^{pwa \cdot r}$ as well as g^r by decrypting $Ticket_B$. It is easy to see that a password guessing attack on pwa is available to KDC_B. The start point of the attack is similar to the first one, i.e. an attacker can get enough information to verify the correctness of a guessed password. While there are still some difference: the first attack is probabilistic and this attack is decisional. Since the equality $g^{pwa' \cdot r} = g^{pwa \cdot r}$ if and only if $pwa' = g^{pwa}$.

On the opposite, KDC_A can disclose *Bob*'s correct password when *Bob* requests the service from *Alice*. Therefore, using above CR-C2C protocol, all passwords of the users in one realm may be exposed to a malicious KDC in another realm. This is very dangerous in practice, especially for example, between two realms (corporations) which keeping cooperation as well as competition.

The reason why this attack succeeds is based on the constitution of kerberos ticket $Ticket_B$, in which, both $g^{pwa \cdot r}$ and g^r are included simultaneously, and have nothing to do with the generator g, hence this attack is also effective when g is chosen from the subgroup as in the original paper.

4.3 Attack 3

To reduce the space of *Bob*'s valid passwords, after receiving message (6), *Alice* does the following:

1. Checks whether g is a quadratic residue modulo p, if yes, gives up, otherwise goes on to next step.
2. Computes $H_4(g^{pwa \cdot r})$ with her pwa and g^r and decrypts $g^{pwb \cdot r \cdot r'}$ as she does in the protocol.
3. Checks whether $g^{pwb \cdot r \cdot r'}$ is a quadratic residue modulo p. if not, she claims that pwb is odd and stops. Otherwise, Otherwise,
4. Implements the protocol and repeats above steps again up to t times.
5. If in t times, $g^{pwb \cdot r \cdot r'}$ is always quadratic residue modulo p, then she claim pwb is even and stop.

Now we proceed to the correctness and success probability of this attack. We assume both r and r' are uniformly chosen from Z_p^*.

- For the case pwb is actually an odd number:
  Alice can correctly claim that pwb is odd once $g^{pwb \cdot r \cdot r'}$ is a quadratic non-residue modulo p, this happens if . Under the assumption that r and r' are uniformly chosen, the probability that both r and r' are odd numbers should be $1 - (\frac{3}{4})^t$, i.e, the probability of her success.
- For the case pwb is actually an even number:
  Obviously, $g^{pwb \cdot r \cdot r'}$ is always a quadratic residue modulo p in this case. According to the attack, *Alice* firmly claims pwb even. This claim will be incorrect only if pwb is odd as well as $r \cdot r'$ is even for all the t times. This condition happens with probability $(\frac{3}{4})^t$. So, the probability of her success should be $1 - (\frac{3}{4})^t$.

On all accounts, by this attack, *Alice* correctly judges the parity of pwb with probability $1 - (\frac{3}{4})^t$. Thus she can exclude half of the valid passwords of *Bob* with the same probability by implementing the protocol t times.

To see how *Bob* can reduce *Alice*'s password space, one only need to observe that *Bob* can obtain $g^{pwa \cdot r \cdot r'}$ from message (5): $E_{R'}(g^{pwa \cdot r \cdot r'} \oplus g^{x'}, ID(A), ID(B))$ and that he knows R', his session key shared with KDC_B.

It is valuable to point out that this attack is invalid when g is a subgroup generator, since a subgroup generator is always a quadratic residue modulo p.

5 The Protocol in Single-Server Setting with Ticket (ST-C2C)

5.1 The ST-C2C Protocol

In the original paper, the authors pointed out that the ST-C2C protocol can be easily constructed by modifying the CR-C2C protocol in following way. That is converting the shared key (K) between two *Kerberos* servers into a private key (PK) of the single server KDC and identifying the rest part to those of CR-C2C protocol.

We note that the ticket in the CR-C2C protocol should also be modified to suit the settings of ST-C2C. In detail, the $Ticket_B$ is encrypted by PK and may not be necessarily includes g^r, since g^r is generated by the KDC himself, he can just store it for later use. Hence $Ticket_B^{'} = E_{PK}(g^{pwa \cdot r}, ID(A), ID(B), L)$. As for the other parameters such as $R, R^{'}, sk$ and cs are computed in the same way as those in the CR-C2C protocol.

The ST-C2C protocol behaves as follows.

(1) $Alice \rightarrow KDC$: $ID(A), ID(B), E_{pwa}(g^x)$
(2) $KDC \rightarrow Alice$: $E_R(g^x \oplus g^r, ID(A), ID(B)), E_{pwa}(g^y), Ticket_B^{'}$

(3) $Alice \rightarrow Bob$: $Ticket_B^{'}, ID(A), L$
(4) $Bob \rightarrow KDC$: $Ticket_B^{'}, E_{pwb}(g^{x^{'}}), ID(A), ID(B), L$
(5) $KDC \rightarrow Bob$: $E_{R'}(g^{pwa \cdot r \cdot r^{'}} \oplus g^{x^{'}}, ID(A), ID(B)), E_{pwb(g^{y'})}$
$E_{H_4(g^{pwa \cdot r})}(g^{pwb \cdot r \cdot r^{'} \cdot x})$
(6) $Bob \rightarrow Alice$: $E_{cs}(g^a), E_{H_4(g^{pwa \cdot r})}(g^{pwb \cdot r \cdot r^{'}})$
(7) $Alice \rightarrow Bob$: $E_{sk}(g^a), E_{cs}(g^b)$
(8) $Bob \rightarrow Alice$: $E_{sk}(g^b)$

Fig. 2. The ST-C2C protocol (Single-server with Ticket).

5.2 Analysis of ST-C2C Protocol

It is obviously that above attacks to the CR-C2C protocol can be directly applied to the ST-C2C protocol except for Attack 2, since we change the $ticket_B$ to $Ticket_B^{'} = E_{PK}(g^{pwa \cdot r}, ID(A), ID(B), L)$. This is also because the clients *Alice* and *Bob* are both in the same realm with the unique KDC. While for Attack 1 and 3, the former is due to an eavesdropper and the latter is carried by one client to another. Therefore, they still effective.

6 The Counter Measures

The counter measures to above attacks are very simple. For the subgroup generator attack (Attack 1), we can just select the generator $g \in Z_p^*$.

To resist against Attack 2, we try to unable KDC_B to obtain $g^{pwa \cdot r}$ and g^r simultaneously from the Kerberos ticket $Ticket_B$. Actually, we can set $Ticket_B = E_K(g^{pwa \cdot r}, g^{xr}, ID(A), ID(B), L)$ by involving $Alice$'s contribution x as a mask of g^r. And the followed communication messages should be changed in an obvious way. For example, in step (5), KDC_B will use $H_4(g^{pwa \cdot r})$ to compute and send $E_{H_4(g^{pwa \cdot r})}(g^{pwb \cdot xr \cdot r'})$ instead of $E_{H_4(g^{pwa \cdot r})}(g^{pwb \cdot r \cdot r'})$. After receiving message (6) from Bob, $Alice$ will decrypt $g^{pwb \cdot xr \cdot r'}$, and then compute cs as $H_5(g^{pwa \cdot pwb \cdot r \cdot r'}) = H_5((g^{pwb \cdot xr \cdot r'})^{\frac{1}{x} \cdot pwa)}$ using $g^{pwb \cdot xr \cdot r'}, pwa$ and x.

In the last attack, the weakness which an attacker can make use of is when g is a quadratic non-residue modulo p. Therefore, if we simply select g to be a quadratic residue modulo p, Attack 2 would be invalid. In group Z_p^*, there may exist many number of such generators.

The countermeasures for the ST-C2C protocol is the same, one can figure them straightforward from that of CR-C2C protocol.

7 Conclusion

In this paper, we show the insecurity of the C2C-PAKE protocols [7] in both cross-realm setting and single-server setting with ticket, by presenting three effective dictionary attacks. In the original parameters environment, the proposed protocols collapse under the subgroup generator attack. Even configured with the powerful parameters, they are still susceptible to various dictionary attacks in both SPA and DPA/CR-DPA senses. We also provide the corresponding countermeasure against our attacks. At least one lesson can be taken from our attacks, that PAKE protocols in a cross-realm setting are more vulnerable than classic or three-party PAKE protocols because of their intrinsic relationship between different realms. Therefore, more precautions should be taken to prevent various attacks such as compromise of the symmetric key shared between two servers.

References

1. R. Anderson and S. Vaudenay, Minding Your p's and q's, in *Advaces in Cryptology - ASIACRYPT'96*, LNCS 963, pages 236-247, Springer-Verlag, 1995.
2. F. Bao, Security Analysis of a Password Authenticated Key Exchange Protocol, in *Proceedings of ISC'03*, LNCS 2851, pages 208-217, Springer-Verlag, 2003.
3. M. Bellare and P. Rogaway, Entity Authentication and Key Distribution, in *Advances in cryptology - Crypto'93*, pages 232-249, Springer-Verlag, 1993.
4. S. Bellovin and M. Merritt, Encrypted Key Exchange: Password-based Protocols Secure Against Dictionary Attacks, in *Proceedings of IEEE Security and Privacy*, pages 72-84, 1992.

5. S. Bellovin and M. Merritt, Augumented Encrypted Key Exchange: A Password-based Protocol Secure Against Dictionary Attacks and Password File Compromise. *ACM Secrity '93*, pages 244-250.
6. V. Boyko, P. MacKenzie, and S. Patel, Provably-secure Password Anthentiation and Key Exchange Using Diffie-Hellman. in *Advances in cryptology - EUROCRYPT 2000*, LNCS 1807, pages 156-171, Springer-Verlag, 2000.
7. J. W. Byun, I. R. Jeong, D. H. Lee, and C-S. Park, Password-Authenticated Key Exchange Between Clients with Different Passwords, in *Proceedings of Information and Communications Security - ICICS 2002*, LNCS 2513, pages 134-146, Springer-Verlag, 2002.
8. G. D. Crescenzo, and O. Kornievskaia, Efficient Kerberized multicast in Practical distributed setting, in *Proceedings of ISC'01*, LNCS 2000, pages 27-45, Springer-Verlag, 2001.
9. W. Diffie, P. Van Oorschot and M. Wiener, Authentication and Authenticated Key Exchange, *Designs, Codes and Cryptography*, 2, 1992, pages 107-125.
10. L. Gong, Optimal Authentication Protocols Resistant to Password Guessing Attacks. in *8th IEEE Computer Security Foundations Workshop*, pages 24-29, 1995.
11. L. Gong, M. Lomas, R. Needham, and J. Saltzer, Protecting Poorly Chosen Secrets from Guessing Attacks, *IEEE Journal on Selected Areas in Communications*, 11(5), pages 648-656, 1993.
12. Y. H. Hwang, D. H. Yum, and P. J. Lee, EPA: An Efficient Password-Based Protocol for Authenticated Key Exchange, in *Proceedings of ACISP 2003*, LNCS 2727, Springer-Verlag Berlin Heidelberg 2003, pages 452-463, 2003
13. D. Jablon, Strong Password-Only Authenticated Key Exchange, *ACM Computer Communications Review*, vol.26, no.5, pp. 5-20, October 1996.
14. B. Jaspan, Dual-workfactor Encrypted Key Exchange: Efficiently Preventing Password Chaining and Dictionary Attacks, in *Proceedings of the 6th Annual USENIX Security Conference*, pages 43-50, July, 1996.
15. T. Kwon, Authentication and Key Agreement via Memorable Password, in *Proceedings of the ISOC Symposium - NDSS'01*, 2001.
16. T. Kwon, M. Kang, S. Jung, and J. Song, An Improvment of the Password-Based Authentication Protocol (K1P) on Security against Replay Attacks, *IEICE Trans. Commun.,* E82-B(7), pages 991-997, 1999.
17. T. Kwon, M. Kang, and J. Song, An Adaptable and Reliable Authentication Protocol for Communication Networks, in *Proceedings of IEEE INFOCOM'97*, pages 737-744, 1997.
18. C. H. Lim and P. J. Lee, Several Practical Protocols for Authentication to Threshold Cryptosystems, *Information Processing Letters*, 53, 1995, pages 91-96.
19. C.-L. Lin, Hung-Min Sun and T. Hwang. Three-party Encrypted Key Exchange: Attacks and A Solution. *ACM Operating Systems Review*, 34(4):12-20, 2000.
20. C.-L. Lin, H.-M. Sun and T. Hwang. Three-party Encrypted Key Exchange Without Server Public-Keys. *IEEE Communications Letters*, 5(12):497-499, December 2001.
21. S. Lucks, Open Key Exchange: How to Defeat Dictionary Attacks Without Encrypting Public Keys, in *Proceedings of Security Protocols Workshop*, LNCS 1361, pages 79-90, Springer-Verlag, 1997.
22. P. MacKenzie, S. Patel, and R. Swaminathan, Password-Authenticated Key Exchange Based on RSA, in *Proceedings of AsiaCrypt 2000*, LNCS 1976, pages 599-613, Springer-Verlag, 2000.
23. P. MacKenzie, The PAK suite: Protocols for Password-Authenticated Key Exchange, *Submission to IEEE P1363.2*, April 2002.

24. P. MacKenzie, More Efficient Password-Authenticated Key Exchange, *Progress in Cryptology – CT-RSA 2001*, pages 361-377, 2001.
25. W. Mao and C.H.Lim, Cryptanalysis in Prime Order Subgroups of $\mathbb{Z}_n^*$, in *Advances in cryptology-ASIACRYPT'98*, LNCS 1514, Spinger-Verlag, 1998, pp.214-226.
26. S. Patel, Number Theoretic Attacks on Secure Password Schemes, in in *Proceedings of IEEE Symposium on Research in Security and Privacy*, pages 236-247, 1997.
27. J. G. Steiner, B. C. Newman, and J. I. Schiller, Kerberos: An Authentication Service for Open Network Systems. in *USENIX Conference Proceedings*, pages 191-202, February, 1988.
28. M. Steiner, G. Tsudik and M. Waidner. Refinement and Extension of Encrypted Key Exchange. *ACM SIGOPS Operating Systems Review*, 29(3), pages 22-30, 1995.
29. Z. Wan, and S. Wang, Cryptanalysis of Two Password-Authenticated Key Exchange Protocols, in *Proceedings of ACISP 2004*, LNCS, Springer-Verlage (to appear), 2004.
30. T. Wu, Secure Remote Password Protocol, in *ISOC Network and Distributed System Security Symposium*, 1998.
31. F. Zhu, D. S. Wong, A. H. Chan, and R. Ye, Password authenticated key exchange based on RSA for imbalanced wireless networks, in *Proceedings of ISC 2002*, LNCS 2433, pp. 150-161, Springer-Verlag, 2002.