A New Security Model for Cross-Realm C2C-PAKE Protocol*

Fengjiao Wang[1] Yuqing Zhang

National Computer Network Intrusion Protection Center， GSCAS，
Beijing， 100043

Abstract. Cross realm client-to-client password authenticated key exchange (C2C-PAKE) schemes are designed to enable two clients in different realms to agree on a common session key using different passwords. In 2006, Yin-Bao presented the first provably secure cross-realm C2C-PAKE, which security is proven rigorously within a formally defined security model and based on the hardness of some computationally intractable assumptions. However, soon after, Phan *et al*. pointed out that the Yin-Bao scheme was flawed. In this paper, we first analyze the necessary security attributes in the cross-realm C2C-PAKE scenario, and then a new security model for cross-realm C2C-PAKE is given. Analogous to the general construction of 3PAKE protocol for single server C2C-PAKE setting, we give a general construction of cross-realm C2C-PAKE protocol, which security is proved in the new security model.

Key word: Password-authenticated key exchange, cross realm, client-to-client, provably secure, security model.

1 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. Over the years, there have been great deals of research on efficient and provably secure password-based authenticated key exchange schemes in the two-party or N-party settings. Most password-authenticated key exchange schemes in the two-party literature provide an authenticated key exchange between a client and a server based on pre-shared password. However, with diversity and rapid

* This work is supported by National Natural Science Foundation of China (60573048) and National “863” program(2007AA01Z427，2007AA01Z450).

[1]Corresponding author.Tel:+86-010-68860988,Fax:+86-010-68860988.
E-mail address: wangfj@nipc.org.cn.

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. Jin Wook Byun *et al*. [1] firstly considered the cross-realm scenario and divided client-to-client password authenticated key exchange protocols into two types: single-server C2C-PAKE protocol and cross-realm C2C-PAKE protocol.

1.1 Related work and Motivation

In 2002, Byun *et al*. [1] first proposed a C2C-PAKE protocol in the cross-realm setting, which is designed to enable two clients in different realms to agree on a common session key using different passwords and hence there existed two servers involved. However, this first cross-realm scheme has been found to be not secure against dictionary attacks from a malicious server in a different realm [2]. In 2004, Wang *et al*. [3] showed three dictionary attacks on the same protocol, and Kim *et al*.[4] pointed out that the protocol was susceptible to Dening-Sacco attack and an improved C2C-PAKE protocol was given. However, this improved scheme was also shown to fall to unknown key share attacks [5] later. Although several countermeasures to resist the attacks on C2C-PAKE protocol have been presented in [2~5], all these proposals and variants were designed with heuristic security analysis, not formally treated. In [6], Abdalla *et al*. first addressed the single-server C2C-PAKE protocol setting and proposed a generic method to construct provably secure single-server C2C-PAKE protocol. To reduce the generic construction's complexity the first efficient provably secure single-server C2C-PAKE protocol was given in [7], and later a similar result about cross-realm C2C-PAKE protocol based on that in [7] was given [8]. In [8], a formal model as well as the corresponding security definitions for cross-realm C2C-PAKE protocol was defined for the first time. Concurrently, a provably secure cross-realm C2C-PAKE protocol was proposed by Byun *et al*. [9]. What's interesting is, [6, 7] and [8, 9] were pointed out to be insecure at the same time by Wang *et al*. [10] and Phan *et al*. [11] respectively, and both [10] and [11] were published in INDOCRYPT 2006. As analyzed in [10], [6, 7] fall to undetectably online dictionary attacks, and an enhanced proposal were given by adding the authentication security notion for the treatment of undetectable attacks in 3-party PAKE scenario. In [11], Phan *et al*. pointed out that [8, 9] also suffered undetectably online dictionary attacks, and an unknown key-share attack to [8]. As Phan *et al*. stated in [11], designing provably secure protocols is indeed the right approach, but defining an appropriate model is not a trivial task, because not including some types of queries e.g. the Corrupt

query, or improperly defining the adversarial game may result in a security proof that fails to capture valid attacks.

1.2 Our Contributions

As mentioned above, a suitable formal security model for cross-realm C2C-PAKE protocol is needed after the scheme proven to be secure in Yin-Bao security model [8] was found to be flawed. This paper works out this problem and a suitable security model is presented based on the work of [8, 10]. In section 2, we briefly describe the security model in [10] and [8] first; Necessary security attributes in a cross-realm C2C-PAKE protocol setting are analyzed, and a new formal security model for this scenario is defined in section 3; In section 4, analogous to the general construction of 3PAKE protocol for single server C2C-PAKE setting [10], we give a general construction of cross-realm C2C-PAKE protocol, which security is proved in the new security model. Comparison with the first formal security model [8] for cross-realm C2C-PAKE protocol is given in section 5, and finally we conclude the paper in section 6.

2 Two Formal Security Model for C2C-PAKE Protocol

Before defining the improved new security model for cross-realm C2C-PAKE protocol, we first provide a brief description to the security model for single-server C2C-PAKE protocol setting in [10] and that for cross-realm C2C-PAKE protocol in [8].

2.1 Security Model for Single-server C2C-PAKE Protocol [10]

As is well recognized, designing provably secure protocols is indeed the right approach. Recently, with the importance of 3-party PAKE protocols been realized by protocol researchers, the research into precise security definitions and formalization are needed and paid more attention.

In [6], Abdalla *et al*. first addressed the single-server C2C-PAKE protocol setting and proposed a generic method to construct provably secure single-server C2C-PAKE protocol. To reduce the generic construction's complexity the first efficient provably secure single-server C2C-PAKE protocol was given in [7]. Unfortunately, soon after both of them were pointed out to suffer from undetectable on-line dictionary attacks [10]. As analyzed in [10], only adding mutual authentication between two communicating parties in the end can not enhance those protocols to resist to undetectable on-line dictionary

attacks. The reason is that there are inside attackers in the 3-party scenario, who themselves can play the legal role of one of the involved client users and impersonate the other client party by guessing the value of its password. After finishing the protocol with the trusted server, inside attackers can verify whether a password guess is correct by comparing the session keys obtained from legal and impersonating identifications respectively. So if only communicating parties authenticate each other, inside attackers can still guess the correct password by keeping on-line interacting with the trusted server which cannot detect such attacks. To solve out this problem, a new security notion called authentication security [10] is added as an extension for the ROR model of single-server 3-party PAKE protocols [6].

2.2 Security Model for Cross-Realm C2C-PAKE Protocol [8]

Different from single-server 3-party PAKE protocol setting, in a cross-realm C2C-PAKE protocol setting two servers are considered. Denote *A*, *B* as two clients belonging to two different realms. Client *A* shares his password pw_A with server S_1, and client *B* shares his password pw_B with another server S_2. Additionally, assume that there is an authenticated private channel between server S_1 and server S_2. All clients' passwords are chosen from the same small dictionary *D* whose distribution is D_{pw}.

During the execution of a protocol in cross-realm setting, an adversary could also interactive with protocol participants via several oracle queries, which model adversary's possible attacks in the real execution, and all possible oracle queries are: *Execute* query, *Reveal* query, *SendClient* query, *SendServer* query and *Test* query, which are defined as in a Find-Then-Guess model sense in [6]. Furthermore, security notions as *opened*, *unopened*, *partnering* and *freshness* are also defined as that in [6]. However, different from the two-party password-based protocol setting that only *off-line dictionary attack* on the password needs to be considered, in the cross-realm C2C-PAKE protocol setting, besides the *off-line dictionary attack*, *client's inside attack* (a legal client may try to learn other client's password after the execution of the protocol) and *server's inside attack* (a legal server may try to learn the password of the client not belonging to his realm) are also necessary to be considered. Thus, the security definition proposed in [8] consists of the following four security requirements: (1) the session key cannot be distinguished from random number by an outside malicious adversary; (2) the server does not know the session key between clients; (3) the client does not know other client's password; and (4) client's passwords are not revealed to other servers except for their own servers.

3 New Security Model for Cross-Realm C2C-PAKE Protocol

As analyzed in [11], the *undetectable on-line dictionary attacks* are caused by parties in general not being able to distinguish between interactions with other honest parties or with the adversary, which is illustrated in [8].Therefore incorporating security against undetectable on-line dictionary attacks directly into the security model of a PAKE protocol is suggested; Moreover, as in the 3-party PAKE setting, we also have to incorporate *authentication security* directly into our security model to resist the undetectable on-line dictionary attacks in cross-realm C2C-PAKE setting.

Furthermore, due to the existence of malicious client insider in a cross-realm C2C-PAKE setting, incorporate Corrupt queries into the security model is necessary. As analyzed in [12], exclusion of the **Corrupt** queries in the C2C-PAKE-YB model [8] gives rise to *unknown key-share attacks* by a malicious client insider, which cannot be captured by a proof in the security model. Note that corruption of a client is essentially equivalent to having a malicious client as an adversary. Although the C2C-PAKE-YB model claims security against the latter case, the former is not considered. Therefore, we add the Corrupt queries and the security definition of mutual authentication between clients into the security model.

As analyzed above, on the basis of guaranteeing semantic security this paper extends the C2C-PAKE-YB model [8] by adding the **Corrupt** queries, mutual authentication and the security against undetectable online dictionary attacks directly into the security model.

3.1 Communication Model

We still adopt the notations to denote protocol participants and client's instances that used in [8] here. Denote A, B as two clients belonging to two different realms. Client A shares his password pw_A with server S_1, and client B shares his password pw_B with another server S_2. Additionally, assume that there is an authenticated private channel between server S_1 and server S_2. In practice, this can be implemented by a pre-distributed common key shared between S_1 and S_2 or their public keys. All clients' passwords are chosen from the same small dictionary D whose distribution is D_{pw}. Each participant U's (maybe a client or a server) i-th instances as U^i. Denote the set of all clients as $\boldsymbol{\mu}$, and denote the set of all servers as $\boldsymbol{S}$.

The cross-realm PAKE protocol is an interactive protocol among four participants' instances: A^i, B^j, S_1^s, S_2^t. After the protocol, A^i and B^j establish a session key sk. It is assumed that an adversary $\mathcal{A}$ has full control over the communication channels except that the channels between servers and can

create several concurrent instances of the protocol. During the execution of protocol, the interaction between an adversary and the protocol participants occurs only via oracle queries, which model the adversary capabilities in a real attack. The oracle queries an adversary $\mathcal{A}$ could launch are as follows:

(1) *Execute*($U_1^{i_1}$, S^j , $U_2^{i_2}$): This query models passive attacks, where the attacker gets access to honest executions among the client instances $U_1^{i_1}$ and $U_2^{i_2}$ and trusted server instance S^j by eavesdropping. The output of this query consists of the message that was exchanged during the honest execution of the protocol.

(2) *SendClient*(U^i , *m*): This query models an active attack against clients, in which the adversary sends a message to the client instance U^i . The output of the query is the message that client instance U^i would generate upon receipt of message *m*.

(3) *SendServer*(S^j, *m*): This query models an active attack against the server, in which the adversary sends a message to server instance S^j . It outputs the message which server instance S^j would generate upon receipt of message *m*.

(4) *Reveal*(U^i): This query models the misuse of session keys by clients. Only if the session key of the client instance U^i is defined, the query is available and returns to the adversary the session key.

(5) *Corrupt*(U^i): This query models the attacks resulting in the password pw to be revealed. Adversary $\mathcal{A}$ gets back from this query pw, but doesn't get any internal data on a corrupted client. This is weak corruption model, corresponding to another notion, called strong corruption model where the internal data of a corrupted client also gets revealed on *Corrupt*(U^i) query.

(6) *Test*(U^i): This query is used to measure the semantic security of the session key of the client instance U^i. If the session key is not defined, it returns⊥. Otherwise, it returns either the session key held by the client instance U^i if *b*=0 or a random number of the same size if *b*=1, where *b* is the hidden bit selected at random prior to the first call.

Partnering. The definitional approach of partnering uses the notion of session identifications (*sid*). Specifically, two instances U_1^i and U_2^j are partners if the following conditions are met: (1) Both U_1^i and U_2^j accept; (2) Both U_1^i and U_2^j share the same *sid*; (3) The partner identification for U_1^i is U_2^j and vice-versa; and (4) No instance other than U_1^i and U_2^j accepts with a partner identification equal to U_1^i or U_2^j .

Freshness. An oracle U_1^i is *fresh* (*or holds a fresh session key*) at the end of execution, if and only if (1) U_1^i has accepted with or without a partner oracle U_2^j ,(2) both U_1^i and U_2^j oracles have not been sent a *Reveal* query, (3) both U_1 and U_2 have not been sent a *Corrupt* query.

3.2 Security Definition

Corresponding to the security definition given in [8], we present the security definition in our new formal security model here.

A secure cross-realm C2C-PAKE protocol is defined by the following four requirements: (1) the session key cannot be distinguished from a random number by an adversary (outside malicious adversary or passive servers); (2) the server does not know the session key between clients; (3) the server can distinguish interactions with honest user or an adversary; (4) the client can be sure that he has been talking to his intended partner client.

Semantic Security in the ROR Model [6]: For any adversary, the advantage for him of guessing the random bit *b* used in the *Test* query is lager than 1/2 with non-negligible probability. Let $Succ^{ake}$ denote the event that the adversary correctly guesses the value of *b*, and let *D* be user's password dictionary. For any adversary, we define his advantage Adv_D^{ake} as

$$Adv_D^{ake}(\mathcal{A}) = 2 \cdot \Pr[Succ^{ake}] - 1$$
$$Adv_D^{ake}(t, R) = \max_{\mathcal{A}} \{Adv_D^{ake}(\mathcal{A})\}$$

where the maximum is over all adversaries with time-complexity at most *t* and using at most *R* times oracle queries.

A protocol *P* is said to be semantically secure if the advantage Adv_D^{ake} is only negligibly larger than $O(q_s) \cdot D_{pw}$, where q_s is the number of all send queries, D_{pw} is the distribution of the password dictionary.

Remark 1. Although corrupt queries are added into this security model, note that corruption of a client is essentially equivalent to having a malicious insider client as the adversary and this doesn't increase the advantage of guessing a random bit *b*. Therefore, we still consider the semantic security in the ROR sense, and this security notion is identical to that defined in [10].

Key Privacy Against Passive Server: We require that no information about the session key is revealed to the server. Note that the server knows all passwords of his members. So a malicious server is always able to impersonate one of its members and exchange a session key with another

client by active attack. As a result, we cannot require a malicious server cannot learn the session key.

The passive server *S* could query two oracles: Execute and Test. Let $Succ^{kp}$ denote the event that *S* correctly guess the value of the random bit *b* used in the Test query. Let *D* is the user's password dictionary. For any passive server *S*, we define his advantage $Adv_D^{kp}(S)$ as

$$Adv_D^{kp}(S) = 2 \cdot \Pr[Succ^{kp}] - 1$$
$$Adv_D^{kp}(t, R) = \max_S \{Adv_D^{kp}(S)\}$$

where the specification of the maximum is as that in security notion 1.

A protocol *P* is key private against passive server if the advantage Adv_D^{kp} is negligible.

Authentication Security [10]: To resist the undetectable on-line dictionary attacks to 3-party PAKE protocol, unilateral authentication from the client to the trusted server is indispensable, wherein the adversary may be an inside attacker and impersonates another client user. Let $Succ^{auth(C \to S)}$ denote the event that an adversary $\mathcal{A}$ successfully impersonates a client instance during executing the protocol *P* while the trusted server does not detect it. For any adversary $\mathcal{A}$, we define his advantage $Adv_D^{auth(C \to S)}(\mathcal{A})$ as

$$Adv_D^{auth(C \to S)}(\mathcal{A}) = \Pr[Succ^{auth(C \to S)}]$$
$$Adv_D^{auth(C \to S)}(t, R) = \max_{\mathcal{A}} \{Adv_D^{auth(C \to S)}(\mathcal{A})\}$$

where the specification of the maximum is as above. *D* is the user's password dictionary.

We say *P* satisfies client-to-server authentication security if $Adv_D^{auth(C \to S)}(t, R)$ is negligible in the security parameter.

Mutual Authentication Between Two Clients: It's necessary for two corresponding parties to authenticate each other over an insecure network in the help of the trusted servers in a cross-realm C2C-PAKE protocol setting. Let $No - Matching^{\mathcal{A}}(k)$ be the event that an oracle U_1^i (or U_2^j) has accepted but there's no oracle U_2^j (or U_1^i) who has a matching conversation with S_2 (or S_1).

We say *P* is a secure mutual authentication protocol, if for any polynomial adversary $\mathcal{A}$, it satisfies:

(1) If U_1^i and S_1, U_2^j and S_2 have a matching conversation respectively, both U_1^i and U_2^j accept;

(2) The probability that $No-Matching^{A}(k)$ occurs is negligible.

4 Generic Construction of Cross-Realm C2C-PAKE Protocol

To our knowledge, hitherto, all the proposals for cross-realm C2C-PAKE protocols have been found to be flawed, so a secure scheme for cross-realm C2C-PAKE setting is needed. As analyzed above, authentication security from the client to server is indispensable in a 3PAKE protocol. The new generic construction of 3PAKE protocol for single-server setting [10] enlightened us to extend it to a generic construction of 3PAKE protocol for cross-realm setting, namely ENGPAKE in this paper. We assume that there is an authenticated private communication channel between the trusted servers. Moreover, we attach the message for constructing session keys between clients to the last round of the 2PAKE protocol between client and sever, which reduces the number of communication rounds. See Fig.1 for more details.

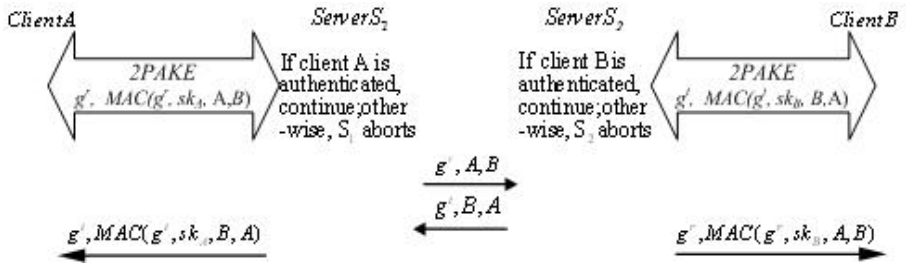

Fig. 1. ENGPAKE:extended new generic construction of 3PAKE for cross-realm setting.

Assume that user A in the realm of $S_1$and user B in the realm of S_2 want to establish a secure session key with the cooperation of servers S_1 and S_2, whose passwords pw_A and pw_B are stored in S_1 and S_2 respectively. User A and user B first run a 2PAKE protocol with server S_1 and S_2 respectively, and two secure high-entropy session keys sk_A and sk_B are established. The 2PAKE protocol used here can be any semantic secure 2-PAKE protocol. Furthermore, if the authentication message from client to server in 2PAKE protocol is not omitted, we attach the message for constructing session key between clients to it. Take the communication between client A and server S_1 for example, on receiving the authentication message from client A, server S_1 first verifies the authentication messages from A using the session key sk_A, if it is invalid, S_1aborts; Otherwise, S_1 sends $<g^r, A, B>$ to server S_2. Analogously, if S_2 has authenticated client B, it send $<g^t, B, A>$ to S_1, compute g^r, $MAC(g^r, sk_B, A, B)$ and send it to B. After receiving $<g^t, B, A>$, S_1 first checks

the identities of client *A* and *B*, then compute g^t, $MAC(g^t, sk_A, B, A)$ and send it to *A*. In this manner, user *A* and user *B* establish a session key in an authenticated way finally with the help of the trusted servers in each realm.

4.1 Assumptions

Obviously, the two cryptographic primitives, DDH assumption and Message authentication code, used as building blocks in our scheme are identical to that in NGPAKE [10], since only communications between trusted servers are added via authenticated private channel. For simplicity, we don't repeat the notions here, refer to [6] for more details.

4.2 Security of ENGPAKE

We prove the security of our scheme by showing that all security requirements defined in subsection 3.2 are met.

Semantic Security in the ROR Model and Authentication Security: The communication channel between servers is authenticated and private, so the communication between servers is secure against any outside adversary. From this viewpoint, server S_1 and S_2 can be treated as a single server when considering outside adversary, and therefore our scheme ENGPAKE is identical to NGPAKE for single-server setting [10], which semantic security was proven rigorously and authentication security from client to server was guaranteed. Thus, ENGPAKE is also semantically secure against outside adversary in the ROR model and providing authentication security from client to server.

Key Privacy Against Passive Server: Since the sub-protocol executed by both clients in the last stage of the scheme for constructing session key is substantially an authenticated Diffie-Hellmam key exchange as in NGPAKE, it is apparent that key privacy against passive server is met. For simplicity, we don't provide its proof here.

Mutual Authentication Between clients: we prove the security of this security requirement through the following theorem.

Theorem 1. ENGPAKE is a secure mutual authentication protocol between clients, assuming that the MAC algorithm and the 2PAKE protocol between client and server are secure.

Proof: The first condition in the definition of mutual authentication between clients is easily verified; it merely equals to say that when the messages are faithfully relayed to each other, each party accepts. In addition, the session

key sk_A and sk_B uniquely bind the values of g^r and g^t to these particular matching sessions and differentiate them from the messages that the parties may exchange in other sessions. As far as the second condition is concerned, it is proved by the following lemma.

Lemma1. $\Pr[No-Matching^{A}(k)] < 2(Adv^{ror-ake}_{2PAKE,D}(t, q_{exe}, q_{exe}+q^{A}_{send}, q^{A}_{send}) + q^{A}_{ake} \cdot Adv^{euf-cma}_{MAC}(t,2,0))$.

The proof of this lemma can be easily derived from the proof of the authentication security in the Theorem 1 in [10]; therefore, we don't bother to repeat it here due to the high complexity of the proof process. #

5 Comparisons and Discussion

C2C-PAKE-YB model is the first formal security model for cross-realm C2C-PAKE protocol setting. However, because of no consideration of authentication security from client to server and mutual authentication between the clients, the protocol proved to be secure in this model was later founded to be insecure against undetectable on-line dictionary attacks and unknown key-share attacks.

Our newly defined security model for cross-realm C2C-PAKE protocol overcomes these problems by adding the corrupt queries capabilities for adversary and defining authentication security from client to server and mutual authentication between clients. Relations between the security definition in our model and that in C2C-PAKE-YB model are analyzed as follows:

(1) Semantic Security. In our model, semantic security is defined in the ROR model, which is stronger than that defined in C2C-PAKE-YB model in FTG sense. The relation between ROR model and FTG model was discussed in [6].

(2) Key privacy. This is identical to that defined in C2C-PAKE-YB model, assuming that the servers are passive.

(3) Authentication Security. As analyzed above, to resist undetectable on-line dictionary attacks, authentication security from client to server is indispensable in 3-party PAKE protocol. Moreover, a cross-realm C2C-PAKE protocol which has this security attribute can not only protect the password from outside adversary but also from the malicious client and server.

(4) Mutual Authentication between clients. Mutual authentication between clients guarantees that two corresponding parties can be sure of whom they have been talking to at the end of a protocol execution. Obviously, this property is necessary in cross-realm C2C-PAKE protocol setting because of

the existence of malicious insider client.

Obviously, these are the advantages that our security model over the Yin-Bao security model.

Furthermore, the generic construction of 3-PAKE for cross-realm setting inherits the excellent properties from NGPAKE in [10], and it is more efficient than NGPAKE in the authentication stage when the last message for authentication from client to server in 2PAKE protocol is not omitted. All existing protocols for cross-realm PAKE are summarized in the following table.

Table 1. Comparison with Existing protocols for cross-realm PAKE.

Attacks / Existing protocols	Off-line dictionary attack	Undetectable on-line dictionary attack	Client's inside attack	Server's inside attack
C2C-PAK-BJLP[1]	R	R	S	S
C2C-PAKE-WWX[3]	R	R	S	R
C2C-PAKE-KKW[4]	S	R	S	S
EC2C-PAKE[9]	R	S	S	S
C2C-PAKE-YB[8]	R	S	S	R
Our generic protocol	R	R	R	R

Where "R" denotes resist, "S" denotes suffer here.

6 Conclusions

This paper first introduces the security model of 3-PAKE for single-server setting and cross-realm setting with two servers involved. Through analysis, we summarize the security attributes needed by 3-PAKE protocol for cross-realm setting and present with a suitable security model for this scenario. Protocols proven secure in this model have the security attributes of authentication security from client to server and mutual authentication between clients, which can effectively avoids protocols suffering from several attacks, such as on-line dictionary attacks (both detectable and undetectable on-line dictionary attacks) and unknown key-share attacks. Furthermore, we extend the generic construction of 3-PAKE for single-server setting [10] to a generic construction of 3-PAKE protocol for cross-realm setting, the security of which is proved in the new security model. Finally, the advantages of this newly defined security model over the Yin-Bao model are discussed and comparison with Existing protocols for cross-realm PAKE is given.

Reference

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

[2] L. chen. A weakness of the password-authenticated key agreement between clients with different passwords scheme. ISO/IEC JTC 1/SC27 N3716.

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

[4] J. Kim, S. Kim, J. Kwak, and D.Won. Cryptanalysis and improvements of password authenticated key exchange scheme between clients with different passwords. In Proceedings of ICCSA 2004, LNCS Vol. 3044, pp.895-902, Springer-Verlag, 2004.

[5] R.C.-W. Phan and B. Goi, Cryptanalysis of an Improved Client-to-Client Password-Authenticated Key Exchange (C2C-PAKE) Scheme. In Proceedings of ACNS 2005, LNCS Vol.3531, pp.33-39, Springer-Verlag, 2005.

[6] M. Abdalla, P.-A. Fouque, D. Pointcheval. Password-based authenticated key exchange in the three-party setting. In S. Vaudenay, editor, Public Key Cryptography-PKC 2005, Vol 3386, LNCS, pp.65-84, 2005.

[7] M. Abdalla, D. Pointcheval. Interactive diffie-hellman assumptions with applications to password-based authentication. In Financial Cryptography and Data Security-FC 2002, Vol 3570, LNCS, pp.341-356, 2005.

[8] Y. Yin and L. Bao. Secure Cross-Realm C2C-PAKE Protocol. Proc. ACISP'06, LNCS 4058, pp.395-406, 2006.

[9] J. W. Byun, D.H. Lee, and J. Lim. Efficient and Provably Secure Client-to-Client Password-Based Key Exchange Protocol. Proc. APWeb'06, LNCS 3841, pp.830-836, 2006.

[10] W. WeiJia, H. Lei. Efficient and provably secure generic construction of three-party password-based authenticated key exchange protocols. Progress in Cryptology - INDOCRYPT'06, LNCS 4329, pp.118-132, 2006.

[11] R. C.-W. Phan, B. Goi. Cryptanalysis of two provably secure C2C-PAKE protocols. Progress in Cryptology - INDOCRYPT'06, LNCS 4329, pp.104-117, 2006.

[12] K.-K.R. Choo, C. Boyd, and Y. Hitchcock. Examining Indistinguishability-Based Proof Models for Key Establishment Protocols. Advances in Cryptology-Asiacrypt'05, LNCS 3788, pp.585-604, 2005.

[13] M. Bellare and P. Rogaway. Entity authentication and key distribution. Advances in Cryptology-CRYPTO'93, LNCS 773, pp.232-249, 1993.