Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/080558

An Enhanced Password-based Group Key Agreement Protocol with Constant Rounds

Wei Yuan[1,2] *and Liang Hu*[2,*]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, 100093, Beijing, China
[2] School of Computer Science and Technology, Jilin University, Changchun 130012, China

Received: 30 Sep. 2013, Revised: 28 Dec. 2013, Accepted: 29 Dec. 2013
Published online: 1 Sep. 2014

Abstract: In PKC 2006, Abdalla et al. proposed a password-based group key exchange protocol with constant rounds and proved that protocol could resist the offline dictionary attacks in the random-oracle and ideal-cipher models. Then they proposed an open problem whether an adversary can test more than one password in the same session with online dictionary attack. To answer this question, they presented an online dictionary attack against their own protocol and declared that this new method is invalid to their protocol. In this paper, based on Abdalla et al.'s attack, we propose a modified attack and apply it to their protocol. The result shows, under the same assumption, our attack can test more than one password. We analyze the reason of this problem and develop a countermeasure to recover it. Finally, a security analysis in the random-oracle and ideal-cipher models is presented to the enhanced protocol.

Keywords: Password-based, Group key agreement, Dictionary attack, Random oracle.

1. Introduction

Authenticated group key agreement protocols [1,2,3,4,5] enable a group of players communicating over an insecure, open network to establish a shared session key and to guarantee that each user indeed shares this session key with the others. Password is one of the ideal authentication approaches to agree on a session key [6,7] in the absence of PKI or pre-distributed symmetric keys. Low-entropy passwords are easy for humans to remember but can not guarantee the same level of security as high-entropy secrets such as symmetric or asymmetric keys [8,9] so a password-based group key agreement protocol may easily suffer from the so-called dictionary attacks [10,11]. Dictionary attacks can be classified into two classes [12]: online dictionary attacks and offline ones. In online dictionary attacks, an adversary usually attempts one guessed password by participating in a key agreement protocol. If the attempts failed, the adversary shall send another message to initiate a new session until he finds out the correct password[13]. In offline dictionary attacks, an adversary selects a password from a dictionary and sends the corresponding message he generates with the password to other users. Then he repeats guessing all the possible passwords in his dictionary with the responded information.

In PKC 2006, Abdalla et al. proposed a password-based group key agreement protocol with a constant number of rounds [14] based on the protocol of Burmester and Desmedt [15]. Then they proved that their protocol could resist the offline dictionary attacks in the random-oracle and ideal-cipher models under the decisional Diffie-Hellman problem. Furthermore, they left an open problem whether an adversary could test several passwords (This is different with the usual online dictionary attacks) within one session. They presented an online dictionary attack against their own protocol and declared that it wound not threaten the security of their protocol.

Our work mainly concerns about this new online dictionary attacks, which test several passwords within one session. We try to modify Abdalla et al.'s attack so that at least more than one password can be tested in one session. Then we give our analysis on the possible reason of the problem. Finally, we propose a countermeasure and prove its security in the random-oracle and ideal-cipher models.

* Corresponding author e-mail: hul@jlu.edu.cn

2. Review of Abdalla et al.'s protocol

In this protocol, $\mathcal{E}_k, \mathcal{D}_k : G \to G$ are indexed by a $l_{\mathcal{H}}$ bit key k which is accessible (as well as their inverses) through oracle queries $\mathcal{E}$ and $\mathcal{D}$. Key generations make use of hash functions $\mathcal{H} : \{0,1\}^* \to \{0,1\}^{l_{\mathcal{H}}}$, $\mathcal{G} : \{0,1\}^* \to \{0,1\}^{l_{\mathcal{G}}}$. Key confirmations apply the function $Auth : \{0,1\}^* \to \{0,1\}^{l_{Auth}}$. The protocol runs as follows:

1. Each player U_i chooses a random number N_i and broadcasts (U_i, N_i).

2. The session $S = U_1||N_1||\ldots||U_n||N_n$ is then defined, in which each player has a specific index i, and a specific symmetric key $k_i = \mathcal{H}(S, i, pw)$. Each player U_i chooses a random exponent x_i and broadcasts $z_i^* = \mathcal{E}_{k_i}(z_i)$, where $z_i = g^{x_i}$.

3. Each player extracts $z_{i-1} = \mathcal{D}_{k_{i-1}}(z_{i-1}^*)$ and $z_{i+1} = \mathcal{D}_{k_{i+1}}(z_{i+1}^*)$, and computes the $Z_i = z_{i-1}^{x_i}$ and $Z_{i+1} = z_i^{x_{i+1}} = z_{i+1}^{x_i}$. He then broadcasts $X_i = Z_{i+1}/Z_i$.

4. Each player computes his secret as $K_i = Z_i^n X_i^{n-1} X_{i+1}^{n-2} \ldots X_{i+n-2}$, and broadcasts his key confirmation message $Auth_i = Auth(S, \{z_j^*, X_j\}_j, K_i, i)$.

5. After having received and checked all the key confirmations, each player defines his session key as $sk_i = \mathcal{G}(S, \{z_j^*, X_j, Auth_j\}_j, K_i)$.

3. Cryptanalysis of the protocol

3.1. The basic online dictionary attack

This attack is a basic online dictionary attack, in which the adversary has the entire control of the network (the formal definition of this condition can be found in reference [1]). The idea of this attack is to create a session, in which the number of dishonest players, whose roles are played by the adversary, is twice the number of honest players, and to surround each honest player with two dishonest players.

Let k be the number of honest players. The attack works as follows. First, the adversary starts a session in which all the honest players have indices of the form $3(i-1)+2$ for $i = 1, \ldots, k$. Then, let $\{pw_1, \ldots, pw_m\}$ be a list of candidate passwords that an adversary wants to try. To test whether pw_i for $i = 1, \ldots, m$ is the correct password, the adversary plays the role of players $U_{3(i-1)+1}$ and $U_{3(i-1)+3}$, and follows the protocol using pw_i as the password. Let $X_{3(i-1)+2}$ be the value that the honest player $U_{3(i-1)+2}$ outputs in the third round of this protocol. To verify whether his guess pw_i is the correct one, the adversary computes $z_{3(i-1)+2}$ from $z_{3(i-1)+2}^*$ with pw_i and checks whether equation $z_{3(i-1)+2}^{x_{3(i-1)+3}-x_{3(i-1)+1}} = X_{3(i-1)+2}$ holds. This is the case whenever pw_i is equal to the actual password. In this attack, adversary can erase one possible password after a failed test.

3.2. The improved online dictionary attack

We can modify Abdalla et al.'s basic attack to test more than one password with the following method. The preparation is the same as Abdalla et al.'s attack. Let k be the number of honest players. The aim of the adversary is to erase k possible passwords in once. First, the adversary starts a session in which all the honest players are in the position of $3(i-1)+2$ for $i = 1, \ldots, k$. The adversary plays the role of players $U_{3(i-1)+1}$ and $U_{3(i-1)+3}$, for $i = 1, \ldots, k$. Thus, there are $3k$ players in all. Then, let $\{pw_1, \ldots, pw_m\}$ be a list of candidate passwords that an adversary wants to try. The adversary gets out k candidate passwords to test. He chooses $2k$ random exponents $x_1, x_3, \ldots, x_{3(k-1)+1}, x_{3(k-1)+3}$, computes the corresponding $z_i = g^{x_i}$, and computes $z_i^* = \mathcal{E}_{k_i}(z_i)$. The main differences between Abdalla et al.'s attack and ours are that $k_{3(i-1)+1} = (S, 3(i-1)+1), pw_i$ and $k_{3(i-1)+3} = (S, 3(i-1)+3), pw_i$ in our method, where $i = 1, \ldots, k$. However, all z_is are computed from z_i^* using the same candidate password in Abdalla et al.'s method.

Let $X_{3(i-1)+2}$ be the value that the player $U_{3(i-1)+2}$ outputs in the third round. The adversary computes $z_{3(i-1)+2}$ from $z_{3(i-1)+2}^*$ using the candidate pw_i, and checks whether $z_{3(i-1)+2}^{x_{3(i-1)+3}-x_{3(i-1)+1}} = X_{3(i-1)+2}$ holds. Thus, adversary can erase k candidate passwords from the list by this method.

3.3. Problem Discussion

It is easy to understand the improved attack. Suppose that there are six players. Player 2 and player 5 are honest players; the others are simulated by the adversary. Then player 1 and player 3 can participate in the protocol with a common guessed password pw_1. Player 4 and player 6 can participate in the protocol with another common guessed password pw_2. Finally, two wrong passwords can be erased from the dictionary if this session fails. It should be pointed out that the session may be established in the basic online dictionary attack if the guessed password is equal to the correct one. The session is sure to fail in our modified online dictionary attack even though the correct password is guessed by the adversary. However, the partial test may succeed between the two dishonest players and the honest player if corresponding test password is identical to the right one. Then the adversary gets the correct password. That is to say, to erase more passwords in one session if the guessed password is wrong, our method sacrifices the possibility to establish a session key when the guessed password is right.

In our method, that the number can be tested in one session is identical to the number of honest players. If the number of honest players is very few, this attack can not lead to security problem. However, if many honest

players participate in this protocol, the security problem can not be ignored. To present their protocol, Abdalla et al. summarizes a principle from their attack to Dutta and Barua's protocol [16]: *A player should make sure that the encryption key used by each player is unique to that player.* Thus, they develop a hash function $\mathcal{H}$, which each player's symmetric key k_i can be computed by the secret password pw, so that each player can verify it by pw. However, in their protocol, only the final session key can be confirmed in the step4 and step5. Namely, they do not verify whether the broadcasted message z_i^* is encrypted by the expected k_i before X_i is broadcasted and the adversary can gather enough information to test his guess with z_i^* and X_i. Therefore, the order of the steps of their protocol is not reasonable, which leads that their protocol did not achieve their principle.

4. The enhanced protocol

To cover the gaps, we propose an enhanced protocol. The protocol runs as follows:

1. Each player U_i chooses a random number N_i and broadcasts (U_i, N_i);

2. The session $S = U_1||N_1||\ldots||U_n||N_n$ is then defined, in which each player has a specific index i, and a specific symmetric key $k_i = \mathcal{H}(S, i, pw)$. Each player U_i chooses a random number x_i and broadcasts $z_i^* = \mathcal{E}_{k_i}(z_i||k_i)$, where $z_i = g^{x_i}$;

3. Each player extracts $z_{i-1}||k_{i-1} = \mathcal{D}_{k_{i-1}}(z_{i-1}^*)$ and $z_{i+1}||k_{i+1} = \mathcal{D}_{k_{i+1}}(z_{i+1}^*)$, and checks whether $k_{i-1} = \mathcal{H}(S, i-1, pw)$ and $k_{i+1} = \mathcal{H}(S, i+1, pw)$. If both the two equations hold, he computes $Z_i = z_{i-1}^{x_i}$ and $Z_{i+1} = z_i^{x_{i+1}} = z_{i+1}^{x_i}$, then broadcasts $X_i = Z_{i+1}/Z_i$. Otherwise, he broadcasts an error and terminates the protocol.

4. Each player computes his secret as $K_i = Z_i^n X_i^{n-1} X_{i+1}^{n-2} \ldots X_{i+n-2}$, and gets the session key $sk_i = \mathcal{G}(S, K_i)$.

5. Security analysis of the enhanced protocol

5.1. Security definitions

In 2005, Abdalla et al. proposed the real-or-random (ROR) model instead of the find-and-guess model of Bellare and Rogaway to prove their three-party password-based authenticated key exchange protocol. This model seems more suitable for the password-based setting and we shall prove our scheme under this model.

A player may have numerous instances, called oracles, of distinct concurrent executions of the protocol. We denote the j-th instance of U_i by U_i^j. The interaction between the adversary $\mathcal{A}$ and players occurs only via oracle queries, which describe the capabilities of $\mathcal{A}$. In the ROR model, Reveal queries are replaced by Test queries; Execute queries are introduced to model passive attack and can easily be simulated with the Send queries. The Send query and Test query are described as follows:

Send (U_i^j, m): This query models an active attack. can intercept a message and then either modify it or create a new one to the intended player. The output of this query is the response generated by the instance U_i^j upon receipt of the message m according to the execution of protocol P. The adversary can initiate the execution of P by sending a query (U_i^j, start).

Test (U_i^j): This query models the indistinguishability of the real session key from a random string. Once the instance U_i^j has accepted a session key, the adversary attempts to distinguish it from a random key. A random bit b is chosen. If b=1, the real session key is returned. If b=0, a random key is returned. Adversary outputs a guess bit b'. If $b = b'$, where b is the hidden bit used by U_i^j, Adversary $\mathcal{A}$ wins the game.

5.2. Computational assumptions

• Decisional Diffie-Hellman (DDH) problem

Let G be a finite cyclic group of prime order q. g is a generaotr of G. Given (g, g^x, g^y, g^{xy}) and (g, g^x, g^y, g^z) where $x, y, z \in z_q$, it is difficult to distinguish between g^z and g^{xy}. Formally, define the advantage function $Adv_G^{DDH}(\mathcal{A}) = |Pr[\mathcal{A}(X) = 1] - Pr[\mathcal{A}(Y) = 1]|$, where $X \in (g, g^x, g^y, g^{xy})$,$Y \in (g, g^x, g^y, g^z)$. The DDH problem is hard in group G if $Adv_G^{DDH}(\mathcal{A})$ is negligible for any probabilistic polynomial time adversary $\mathcal{A}$. $Adv_G^{DDH}(t)$ is the maximum value of $\mathcal{A}$ running in time at most t.

• Multi-Decisional Diffie-Hellman (MDDH) assumption

Let G be a finite cyclic group of prime order q. g is a generaotr of G.Given $(g, g^{x_1}, g^{x_2}, \ldots, g^{x_n}, g^{x_1 \cdot x_2, \ldots, \cdot x_n})$ and $(g, g^{x_1}, g^{x_2}, \ldots, g^{x_n}, g^y)$, where $x_1, x_2, \ldots, x_n, y \in Z_q$, it is difficult to distinguish between $g^{x_1 \cdot x_2, \ldots, \cdot x_n}$ and g^y. Define the advantage function $Adv_G^{MDDH}(\mathcal{A}) = |Pr[\mathcal{A}(X) = 1] - Pr[\mathcal{A}(Y) = 1]|$, where $X \in (g, g^{x_1}, g^{x_2}, \ldots, g^{x_n}, g^{x_1 \cdot x_2, \ldots, \cdot x_n})$, $Y \in (g, g^{x_1}, g^{x_2}, \ldots, g^{x_n}, g^y)$. The MDDH problem is hard in group G if $Adv_G^{MDDH}(\mathcal{A})$ is negligible for any probabilistic polynomial time adversary $\mathcal{A}$. $Adv_G^{MDDH}(t)$ is the maximum value of $Adv_G^{MDDH}(\mathcal{A})$ running in time at most t.

Lemma 1. For any group G and integer n, the MDDH problem can be reduced to the DDH problem and the advantage $Adv_G^{MDDH}(t) \leq n \cdot Adv_G^{DDH}(t)$. The proof of this lemma can be found in reference [6].

Lemma 2. Let E, E', and F be events defined on a probability space such that $Pr[E|\neg F] = Pr[E'|\neg F]$. Then we have $|Pr[E] - Pr[E']| \leq P[F]$. The proof of this lemma can be found in reference [7].

5.3. Security analysis

Theorem 1. Let P denote the proposed protocol in which the password is chosen in a dictionary of size N. For any adversary $\mathcal{A}$ running in time t, that makes at most q_{active} attempts within at most $q_{session}$ sessions, his advantage in breaking the semantic security of the session key, in the ideal-cipher model, is upper-bounded by:

$Adv_P^{ror-aka}(\mathcal{A}) \leq 2q_{\mathcal{G}}^2/2^{l_{\mathcal{G}}} + 2nq_{session}Adv_G^{DDH}(t)$
$+ \frac{2q_{\mathcal{H}}^2+2q_{\mathcal{D}}q_{\mathcal{H}}+2nq_{session}Adv_G^{DDH}(t)+2q_{\mathcal{G}}/|G|+2q_{active}/N}{2^{l_{\mathcal{H}}}} +$
$\frac{(q_{\mathcal{E}}+q_{\mathcal{D}})^2+2q_{\mathcal{D}}+2nq_{session}q_{\mathcal{E}}+2q_{\mathcal{G}}}{|G|}$

$q_{\mathcal{H}}$, $q_{\mathcal{G}}$ denote the number of oracle queries to the random oracles $\mathcal{H}$ and $\mathcal{G}$, and $q_{\mathcal{E}}$, $q_{\mathcal{D}}$ denote the number of oracle queries to the ideal-cipher oracles $\mathcal{E}$ and $\mathcal{D}$.

This theorem shows the advantage of the adversary essentially grows linearly with the number of active attempts that the adversary makes and the passive attacks are essentially negligible because an honest transcript does not help a computationally bounded adversary in guessing the password.

Proof. We incrementally define a sequence of experiments from the experiment Exp_0 to Exp_7. In each experiment, various adversary behaviors are simulated and the advantages of an adversary $\mathcal{A}$ are upper-bounded. At the end of the experiments, we measure the probability $|Pr[Suc_i] - Pr[Suc_{i-1}]|$ between Exp_i and Exp_{i-1}. Finally, we get the result of the Theorem 1 by the difference of the probability.

Experiment Exp_0. This experiment simulates the real attack. The advantage of $\mathcal{A}$ in this protocol is defined as $Adv_P^{ror-dka}(\mathcal{A}) = 2Pr[Suc_0] - 1$

Experiment Exp_1. In this experiment, we simulate the random oracles $\mathcal{H}$ and $\mathcal{G}$ by maintaining the list $L_{\mathcal{H}}$ and $L_{\mathcal{G}}$, respectively. If $\mathcal{A}$ asks a query of the form (S, i, pw) such that a record (S, i, pw, r) exists in the list $L_{\mathcal{H}}$, then r is returned. Otherwise, r is chosen randomly from $\{0,1\}^{L_{\mathcal{H}}}$, and (S, i, pw, r) is recorded to $L_{\mathcal{H}}$. Define the collision event in the output of $\mathcal{H}$ by $Col_{\mathcal{H}}$. Then the probability of that bad event is upper-bounder by $q_{\mathcal{H}}^2/2^{L_{\mathcal{H}}}$. Similarly, the probability of the collision event $Col_{\mathcal{G}}$ in the output of $\mathcal{G}$ is upper-bounder by $q_{\mathcal{G}}^2/2^{L_{\mathcal{G}}}$. Exp_0 and Exp_1 are perfectly indistinguishable unless that the bad event $Bad_1(= Col_{\mathcal{H}} \vee Col_{\mathcal{H}})$ occurs. Thus, we got $|Pr[Suc_1 \neg Bad_1]| = Pr[Suc_0 \neg Bad_1]$. By lemma 2, we have $|Pr[Suc_1] - Pr[S$
$uc_0]| \leq Pr[Bad_1] = q_{\mathcal{G}}^2/2^{L_{\mathcal{G}}} + q_{\mathcal{H}}^2/2^{L_{\mathcal{H}}}$

Experiment Exp_2. This experiment simulates the ideal-cipher oracles $\mathcal{E}$ and $\mathcal{D}$ by maintaining a list $L_{\mathcal{E},\mathcal{D}}$, which keeps track of the previous queries-answers and links each query to a specific player. $L_{\mathcal{E},\mathcal{D}}$ has the form $(S, i, e, type,$
$k, z, z^*)$, where $type \in enc, dec$. Such a record means that $Z^* = \mathcal{E}_k(z||k)$, and type indicates which kind of queries generated the record. The index i indicates which player is associated with the key k, while S indicates the session with which should be handled. These values are all set to null if k does not come from a $\mathcal{H}$ query of the form $(S, i, *)$ with $i \in 1, \ldots, n$. The e will be explained in the next experiment. $\mathcal{E}$ and $\mathcal{D}$ can be simulated as follows:

Encryption query: For an encryption query $\mathcal{E}_k(z||k)$, if a record $(\cdot, \cdot, \cdot, \cdot, k, z, *)$ exists in the list $L_{\mathcal{E},\mathcal{D}}$, the element * is returned. Otherwise, a random value $z^* \in G$ is returned and $(\cdot, \cdot, \cdot, enc, k, z, *)$ is added into $L_{\mathcal{E},\mathcal{D}}$.

Decryption query: For a decryption query $\mathcal{D}_k(z^*)$, if a record $(\cdot, \cdot, \cdot, \cdot, k, *, z^*)$ exists in the list $L_{\mathcal{E},\mathcal{D}}$, the element * is returned. Otherwise, if k has been returned to a hash query of the form $(S, i, *)$, we choose number $z \in G \ \{0\}$ randomly and update the list $L_{\mathcal{E},\mathcal{D}}$ with $(S, i, \cdot, dec, k, z, z^*)$. Otherwise, we choose $z \in G \ \{0\}$ randomly and update the list $L_{\mathcal{E},\mathcal{D}}$ with $(\cdot, \cdot, \cdot, dec, k, z, z^*)$. Finally, z is returned.

The simulation above is perfect, except the following three bad events. First, that the collisions may appear contradicts the permutation property of the ideal-cipher. The probability can be upper-bounded by $(q_{\mathcal{E}} + q_{\mathcal{D}})^2/2|G|$. Second, z may be equal to 0 and we avoid it in the decryption query. At last, in the case of the decryption query simulation, one will abort executions if the value k involved in a decryption query is outputted by $\mathcal{H}$. The probability is at most $q_{\mathcal{H}}/2^{l_{\mathcal{H}}}$ for each decryption query. For any k involved in a decryption query, if it comes from a $\mathcal{H}$ query, we know the corresponding pair (S, i). Define $Bad_2(= Col_{\mathcal{E},\mathcal{D}} \vee Col_k \vee Col_0)$, we get $|Pr[Suc_2 \neg Bad_2] = Pr[Su$
$c_1 \neg Bad_2]|$, thus, we have $|Pr[Suc_2] - Pr[Suc_1]| \leq (q_{\mathcal{E}} + q_{\mathcal{D}})^2/2|G| + q_{\mathcal{D}}/|G| + q_{\mathcal{D}}q_{\mathcal{H}}/2^{l_{\mathcal{H}}}$

Experiment Exp_3. In this experiment, we change the simulation of the decryption queries, and make use of our challenger to embed an instance of the MDDH problem in the protocol simulation. Let the challenger output tuples $(\gamma_1, \gamma_2, \cdots, \gamma_n, \lambda_1, \lambda_2, \cdots, \lambda_n)$. We simulate the decryption queries properly with these tuples. More preciously, we make a new tuple each time when a new session S appears in a decryption query. However, if several queries are asked with the same S, the challenger outputs the same tuple.

The latter tells us that, given a tuple outputted by the challenger, and for any randomly chosen $(e_1, e_2, \cdots, e_n)$, the tuple $(\gamma_1^{e_1}, \gamma_2^{e_2}, \cdots, \gamma_n^{e_n}, \lambda_1^{e_1 e_2}, \lambda_2^{e_2 e_3}, \cdots, \lambda_n^{e_n e_1})$ has the same distribution as the original tuple. We make this property as follows, by modifying the sub-case previously considered for new decryption queries in the experiment Exp_2.

Decryption query: For a decryption query $\mathcal{D}_k(z^*)$ such that $k = \mathcal{H}(S, i, *)$ was previously obtained from $\mathcal{H}$ for some valid index i, we query challenger for getting a tuple $(\gamma_1, \gamma_2, \cdots, \gamma_n, \lambda_1, \lambda_2, \cdots, \lambda_n)$. Then we choose $e \in Z_q^*$ randomly, add the record $(S, i, e, dec, k, z = \gamma^e, z^*)$ into the list $L_{\mathcal{E},\mathcal{D}}$, and return z.

The records in the list $L_{\mathcal{E},\mathcal{D}}$ has been defined. The changes above of the simulation on the decryption queries does not modify it in the view of the adversary. Hence, $Pr[Suc_3] = Pr[Suc_2]$

Experiment Exp_4. In this experiment, we simulate the Send query in the second and the third round. When the session S is defined, U_i computes the symmetric keys as $k_j = \mathcal{H}(S, j, pw)$, for all player j.

In the second round, U_i chooses a random number $z_i^* \in G$ to be broadcasted, and asks $\mathcal{D}_{k_i}(z_i^*)$ with the simulation in Exp_3. As the result, e_i is added to the list $L_{\mathcal{E},\mathcal{D}}$, if z_i^* hasn't been in the list. But the latter event can not happen with probability greater than $q_{\mathcal{E}/|G|}$.

In the third round, U_i recovers $z_{i-1}||k_{i-1} = \mathcal{D}_{k_{i-1}}(z_{i-1}^*)$, $z_{i+1}||k_{i+1} = \mathcal{D}_{k_{i+1}}(z_{i+1}^*)$, and checks whether $k_{i-1} = \mathcal{H}(S, i\text{-}1, pw)$ and $k_{i+1} = \mathcal{H}(S, i\text{+}1, pw)$. If z_{i-1}^* and z_{i+1}^* have been simulated in the second round, we gets e_{i-1} and e_{i+1} in the list $L_{\mathcal{E},\mathcal{D}}$ such that $z_{i-1} = \gamma_{i-1}^{e_{i-1}}$ and $z_{i+1} = \gamma_{i+1}^{e_{i+1}}$. Otherwise, this z_j^* has been previously answered by the encryption oracle $\mathcal{E}_k(z||k)$, where $k = \mathcal{H}(S, i, pw)$ is the correct key for U_j in session S. We mark such an event by $Encrypt$. In such a case, the simulation is terminated and the adversary wins. Thus, we gets $z_i = \gamma_i^{e_i}$, $z_{i-1} = \gamma_{i-1}^{e_{i-1}}$, $z_{i+1} = \gamma_{i+1}^{e_{i+1}}$ correctly and then computes $Z_i = CDH(z_{i-1}, z_i) = \lambda_{i-1}^{e_{i-1}e_i}$,$Z_{i+1} = CDH(z_i, z_{i+1}) = \lambda_i^{e_i e_{i+1}}$. $X_i = Z_{i+1}/Z_i$ is broadcast. After this round, each player can compute the session key as before. The simulation is still perfect, unless the above bad events happen. Therefore, we get $|Pr[Suc_4] - Pr[Suc_3]| \leq q_{passive} \cdot q_{\mathcal{E}}/|G| + Pr[Encrypt_1]/2^{l_{\mathcal{H}}} \leq nq_{session}q_{\mathcal{E}}/|G| + Pr[Encrypt_1]/2^{l_{\mathcal{H}}}$

Experiment Exp_5. Since it is clear that the security of the above protocol still relies on the DDH assumption, let the challenger output tuples $(\gamma_1, \gamma_2, \cdots, \gamma_n, \lambda_1, \lambda_2, \cdots, \lambda_n)$. We have $|Pr[Suc_5] - Pr[Suc_4]| \leq q_{session} \cdot Adv_G^{MDDH}(t) \leq n \cdot q_{session} \cdot Adv_G^{DDH}(t)$
$|Pr[Encrypt_2] - Pr[Encrypt_1]| \leq q_{session} \cdot Adv_G^{MDDH}(t) \leq n \cdot q_{session} \cdot Adv_G^{DDH}(t)$

Experiment Exp_6. In this experiment, we derive the session keys from a private random oracle $\mathcal{G}' : sk_i = \mathcal{G}'(S, K_i)$. After the modification of the derivation of the session key, the probability for the adversary to tell the difference between the previous experiments and the current one is to query $sk_i = \mathcal{G}(s, K_i)$. Since the previous game, we know no information has been leaked about and these queries are identical inside each session: the probability of such an event can also be upper-bounded by $q_{\mathcal{G}}/|G|$. Thus, we have $|Pr[Suc_6] - Pr[Suc_5]| \leq q_{\mathcal{G}}/|G|$ and $|Pr[Encrypt_2] - Pr[Encrypt_1]| \leq q_{\mathcal{G}}/|G|$. Since the private oracle is private to the simulator, $Pr[Suc_6] = 1/2$. . Experiment Exp_7. The password pw is only used in the simulation of the second and third rounds to compute k_{i-1}, k_i, and k_{i+1} with the element γ_{i-1}, γ_i, and γ_{i+1}. But only X_i, which is computed from λ_{i-1} and λ_i, is outputted. In this experiment, we can simplify the simulation of the second and third rounds as follows: In the second round, U_i randomly chooses $z_i^* \in G$, and sends it with no decryption. In the third round, U_i simply computes and sends $X_i = \lambda_i/\lambda_{i-1}$. This simulation is perfect since we do not need anymore to compute K_i. Thus, the probability of the Encrypt event is less than the probability of first flows manufactured by the adversary. We have $Pr[Encrypt_3] \leq q_{active}/N$

In the above, the collisions in the output of $\mathcal{H}$ have been eliminated in previous experiments and we can get the Theorem 1.

6. Conclusions

In this paper, based on Abdalla et al.'s attack, we propose a modified one and apply it to their protocol. The result shows, under the same assumption, our attack can test more than one password. We analyze the reason of this problem and develop a countermeasure to recover it. Finally, a security analysis in the random-oracle and ideal-cipher models is presented to the enhanced protocol.

Acknowledgement

This paper is supported by the Strategic Priority Research Program of CAS under Grant No. XDA06010701, the National 973 Program of China under Grant No.2011CB302400, and the National Natural Science Foundation of China under Grant No.61379139.

References

[1] W. Diffie, M.E. Hellman, New directions in cryptography, IEEE Transaction on Information Theory, **22**, 644-654 (1976).

[2] M. K. Boyarsky, Public-key cryptography and password protocols: The multi-user case. ACM CCS 99: 6th Conference on Computer and Communications Security, 63-72.

[3] S. M. Bellovin and M. Merritt, Encrypted key exchange: Password-based protocols secure against dictionary attacks, 1992 IEEE Symposium on Security and Privacy, 72-84 (1992).

[4] M. Bellare, D. Pointcheval, and P. Rogaway, Authenticated key exchange secure against dictionary attacks, Advances in Cryptology-EUROCRYPT 2000, Lecture Notes in Computer Science, **1807**, 139-155 (2000).

[5] M. S. Hwang, Dynamic participation in a secure conference scheme for mobile communications, IEEE Trans. Veh. Technol., **48**, 1469 (1999).

[6] K. F. Hwang and C. C. Chang, A self-encryption mechanism for authentication of roaming and teleconference services, IEEE Trans. on Wireless Communications, **2**, 400-407 (2003).

[7] Feng Bao, Analysis of a secure conference scheme for mobile communication, IEEE Trans. on Wireless Communications, **5**, 1984-1986 (2006).

[8] M.O. Rabin, Technical Report LCS/TR-212, MIT Laboratory for Computer Science, (1979).
[9] R.L. Rivest, A. Shamir, and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Communications of ACM, **21**, 120-126 (1978).
[10] Liu, C.L. Introduction to Combinatorial Mathematics. McGrawHill, New York, (1968)
[11] A. Shamir, How to share a secret, communications of the ACM, **22**, 612-614 (1979)
[12] L Lamport, Password Authentication with Insecure Communication, Communications of the ACM, **24**, 770-772 (1981).
[13] M. H. Zheng, H. H. Zhou, J. Li and G. H. Cui, Efficient and provably secure password-based group key agreement protocol, Computer Standards & Interfaces, **31**, 948-953 (2009).
[14] M. Abdalla, E. Bresson, O. Chevassut, D. Pointcheval, Password-based group key exchange in a constant number of rounds, PKC 2006, Lecture Notes in Computer Science, **3958**, 427-442 (2006).
[15] M. Burmester, Y. Desmedt, A secure and scalable group key exchange system, Information Processing Letters, **94**, 137-143 (2005).
[16] R. Dutta, R. Barua, Password-based encrypted group key agreement, International Journal of Network Security, **3**, 30-41 (2006).

Wei Yuan was born in 1984. He began the study of computer science at Jilin University in 2003 and got his bachelor degree in 2007. Then he continued his research on information security and got his master degree in 2010 and doctor degree in 2013 at the college of computer science and technology of Jilin University. Now he is working for the State Key Laboratory of Information Security of the Institute of Information Engineering of Chinese Academy of Sciences. His main research interests include cryptography and information security. he have participated in several projects include two National Natural Science Foundations of China and one National Grand Fundamental Research 973 Program of China and published more than 30 research papers from 2010.

Liang Hu was born in 1968. He has his BS degree on Computer Systems Harbin Institute of Technology in 1993 and his PhD on Computer Software and Theory in 1999. Currently, he is the professor and PhD supervisor of College of Computer Science and Technology, Jilin University, China. His main research interests include distributed systems, computer networks, communications technology and information security system, etc. As a person in charge or a principal participant, Dr Liang Hu has finished more than 20 national, provincial and ministerial level research projects of China.