Repairing Attacks on a Password-Based Group Key Agreement

Ratna Dutta and Rana Barua
Stat-Math Unit
Indian Statistical Institute
203, B.T. Road, Kolkata
India 700108
e-mail:{ratna_r,rana}@isical.ac.in

Abstract

From designing point of view, it is not a trivial task to convert a group key agreement protocol into password-based setting where the members of the group share only a human-memorable weak password and the system may not have any secure public key infrastructure. Security analysis against dictionary attacks is on the other side of the coin. The low entropy of human memorable password may enable an adversary to mount off-line dictionary attacks if careful approaches are not taken in designing the protocol. Recently, Kim *et al.* proposed a very efficient provably secure group key agreement protocol KLL, security of which relies on the Computational Diffie-Hellman (CDH) assumption in the presence of random oracles. Dutta-Barua embed the protocol KLL into password-based environment – yielding the protocol DB-PWD. Abdalla *et al.* detect certain *flaws* in the protocol DB-PWD. In this paper, we take suitable measures to overcome these attacks. We introduce a protocol MDB-PWD – an improved variant of the protocol DB-PWD and analyze its security in the security framework formalized by Bellare *et al.* in both the ideal cipher model and the random oracle model under CDH assumption.
Keywords: password-based protocol, dictionary attack, encrypted group key agreement, CDH problem

1 Introduction

Designing password-based schemes have been given considerable attention mainly because they authenticate the parties even under the circumstance that the users have restricted computing and memory devices. However, risks may arise if sufficient security measures are not taken into account while designing a protocol in password-based setting. The fundamental security threats for these protocols are dictionary attacks which arise because the passwords are relatively short and easily guessed.

The first work to deal with dictionary attacks is by Bellovin and Merritt [8]. Since then, such schemes have been extensively studied. Recently, Bellare *et al.* [7] and Boyko *et al* [11] introduced formal models and security goals for password-based authenticated key exchange schemes. An extension of the work in [7] to multi-party setting is proposed by Bresson *et al.* [14]. Bresson *et al.* presented in [13] a 2-party password-based key exchange protocol. The security of these schemes are both in the random oracle model and the ideal cipher model. Two party password-based key exchange protocols based on general assumptions were proposed by Goldreich *et al.* [23], Katz *et al.* [31] and Gennaro *et al.* [22]. These schemes are proven to be secure in the standard model in abscence of random oracles and ideal ciphers. There is an extensive history of password-based protocols in the literature. The above mentioned works are only a few of them and are not exhaustive at all.

In this paper, we concentrate on designing group key agreement in password-based setting. The works in this area are, to the best of our knowledge, by Bresson *et al.* [14], Lee *et al.* [36] and Dutta-Barua [21].

However, the protocol of [36] is not authenticated because there is no way to convince a user that the message that he receives is indeed coming from the intended participant. This leads to dictionary attack (see [1] for more details).

Very recently, Kim *et al.* [33] proposed an efficient constant round authenticated group key agreement protocol KLL which is proven to be secure under CDH assumption in the random oracle model. A recent work by Dutta-Barua [21] extends this protocol into password-based setting to obtain a protocol DB-PWD and provides a concrete security analysis in the security framework of Bellare *et al.* [7] under the CDH assumption. Security against dictionary attacks is achieved in both the random oracle model and the ideal cipher model. In [1], Abdalla *et al.* found a source of redundancy in the protocol DB-PWD that can be exploited by an attaker and thereby making it insecure against off-line dictionary attacks. Our contribution in this context is to take suitable measures to overcome these attacks. We incorporate modifications in the protocol KLL and introduce a protocol MDB-PWD – an improved variant of the protocol DB-PWD, that overcome the flaws detected by Abdalla *et al.* [1].

It is not a trivial task to convert a provably secure authenticated group key agreement into password-based group key agreement. The low entropy of the password may enable an adversary to mount off-line dictionary attacks if careful approaches are not taken while designing such protocols. Our proposed scheme MDB-PWD is an embedding of the protocol KLL in password-based setting. The security analysis of the protocol MDB-PWD is almost same as that of the protocol DB-PWD [21] except for slight minor modifications.

The organization of the paper is as follows. In Section 2, we introduce basic definitions. We present our protocol in Section 3 and provide security results in Section 4. Finally we conclude in Section 5.

2 Preliminaries

The security notion and the security model that a password-based group key agreement protocol should achieve is same as that described in the work [21]. This adopts the formal security model of Bellare *et al.* [7] as standardized by Bresson *et al.* [13, 14]. We refer the reader to [7, 13, 14] for more details. We racall below certain basic definitions. (We use the notation $a \in_R S$ to denote that a is chosen uniformly from the set S.)

2.1 Computational Diffie-Hellman (CDH) problem

Let $G = \langle g \rangle$ be a multiplicative group of some large prime order q. Then Computation Diffie-Hellman (CDH) problem in G is defined as follows:

Instance : (g, g^a, g^b) for some $a, b \in Z_q^*$.
Output : g^{ab}.

The success probability of any probabilistic, polynomial-time algorithm $\mathcal{A}$ in solving CDH problem in G is defined to be :

$$\mathsf{Succ}_{G,\mathcal{A}}^{\mathsf{CDH}} = \mathsf{Prob}[\mathcal{A}(g, g^a, g^b) = g^{ab} : a, b \in_R Z_q^*].$$

The probability is taken over the choice of a, b and $\mathcal{A}$'s coin tosses. We say that an algorithm $\mathcal{A}$ (t, ϵ)-breaks CDH problem in G if $\mathcal{A}$ runs in time at most t and $\mathsf{Succ}_{G,\mathcal{A}}^{\mathsf{CDH}}(t) > \epsilon$.

CDH assumption : There exists no probabilistic, polynomial-time algorithm that (t, ϵ)-breaks CDH problem in G. In other words, for every probabilistic, polynomial-time algorithm $\mathcal{A}$, $\mathsf{Succ}_{G,\mathcal{A}}^{\mathsf{CDH}} \leq \epsilon$ for sufficiently small $\epsilon > 0$.

2.2 Ideal Cipher Model

In the ideal cipher model, a keyed cipher is viewed as a family of random permutations that are queried via oracle to encrypt and decrypt. If the same query is asked twice, identical answers are provided and for each new query, a truly random value is produced by the oracle as an output. Although ideal cipher model does not provide the same security guarantees as those in the random oracle and the standard models, it is certainly superior to those provided by *ad hoc* protocol designs. Reducing ideal cipher model assumption is an interesting research problem.

2.3 Security Concerns : Dictionary Attacks

It is extremely important both to correctly define the security notions in password-based setting and prove the security of any proposed implementation. There are many security concerns associated with password-based protocols, mainly because most user's passwords are drawn from a relatively small and easily generated dictionary. The adversary may attempt to impersonate a user on-line by guessing the password. If the attack fails, the adversary can eliminate this value from the set of possible passwords. Obviously, we can not prevent a real world adversary from trying the password on-line, but can make the attack infeasible simply by imposing a limit on the number of unsuccessful impersonation attempts. We restrict the adversary incapable of eliminating more than one password after one active interaction with some user. This attack is called on-line password guessing, sometimes also referred to as impersonation attack.

Another fundamental security goal for designing traditional secure password-based protocols is to achieve security against off-line dictionary attacks. A specific focus of research has been done on preventing off-line dictionary attacks. This attack enables the adversary to record its view from past protocol executions and then scan the dictionary for a password consistent with this view. The adversary can derive the correct password if checking consistency in this way is possible and the dictionary is small. This attack is very powerful in the sense that it can be performed off-line, so the adversary need not to interact with the legitimate parties and can use a lot of computing power. Although this attack is not effective for high-entropy key, it can be very damaging when the session key is a low-entropy password, because the attacker has a non-negligible chance of winning. A secure password-based protocol should withstand this attack.

3 Protocol

We start by presenting the protocol requirements, follow it up the unauthenticated group key agreement protocol KLL of Kim, Lee, Lee [33] and the protocol DB-PWD – the password-based version of the protocol KLL presented in [21]. We then discuss certain redundancies, as noted by Abdalla *et al.* [1], in the transmitted messages during the execution of the protocol DB-PWD. Presence of these redundancies enable the protocol vulnerable to off-line dictionary attacks. Finally, we propose our protocol MDB-PWD – the modified version of the protocol DB-PWD, by appropriately modifying the unauthenticated protocol KLL and introducing encryption-based authentication mechanism using the password as a part of secret key.

3.1 Protocol Requirements

Suppose a set of $n \geq 3$ users $\mathcal{P} = \{U_1, U_2, \ldots, U_n\}$ share a low entropy secret password pw drawn uniformly from a small dictionary of size N and wish to establish a high entropy common session key among themselves. We consider the users $U_1, \ldots, U_n$ participating in the protocol are on a ring and U_{i-1}, U_{i+1} are respectively the left and right neighbors of U_i for $1 \leq i \leq n$ with $U_0 = U_n, U_{n+1} = U_1$ and U_{n+i} is taken to be U_i. Quite often we identify a user U_i with his instance $\Pi_{U_i}^{d_i}$ (for some unique integer d_i that is session specific) during a protocol execution. We denote by $A|B$ the concatenation of A, B.

Let $G = \langle g \rangle$ be a multiplicative group of some large prime order q and $\overline{G} = G \setminus \{1\}$. Then $\overline{G} = \{g^x | x \in Z_q^*\}$. The password pw shared among the members of the group is used as a part of encryption/decryption key. We take a cryptographically secure hash function $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^l$ where l is a security parameter, $l \leq |q|$ ($|q|$ is the bit length of q). We also consider three block ciphers $(\mathcal{E}_k, \mathcal{D}_k)$, $(\mathcal{E}'_k, \mathcal{D}'_k)$ and $(\mathcal{E}''_k, \mathcal{D}''_k)$ where k is a password uniformly drawn from a small dictionary of size N. Here $\mathcal{E}_k$, $\mathcal{E}'_k$ and $\mathcal{E}''_k$ are keyed permutations over the sets $\mathcal{S}$, $\mathcal{S}'$ and $\mathcal{S}''$ respectively to be specified later and $\mathcal{D}_k$, $\mathcal{D}'_k$ and $\mathcal{D}''_k$ are the respective inverses of $\mathcal{E}_k$, $\mathcal{E}'_k$ and $\mathcal{E}''_k$.

3.2 Protocol KLL : Unauthenticated Group Key Agreement of [33]

The unauthenticated protocol KLL presented by Kim *et al.* in [33] involves two rounds and a key computation phase. The protocol is executed as follows among n user instances $\Pi_{U_1}^{d_1}, \ldots, \Pi_{U_n}^{d_n}$.

1. (Round 1) Each user U_i randomly chooses $k_i \in \{0,1\}^l$ and $x_i \in Z_q^*$, computes $y_i = g^{x_i}$ and keeps k_i secret. The last user U_n additionally computes $\mathcal{H}(k_n|0)$. Each user U_i broadcasts $M_i^{(1)}$ where $M_i^{(1)} = y_i$ for $1 \leq i \leq n-1$ and $M_n^{(1)} = \mathcal{H}(k_n|0)|y_n$.

2. (Round 2) User U_i on receiving $M_{i-1}^{(1)}$ and $M_{i+1}^{(1)}$ computes $K_i^L = \mathcal{H}(y_{i-1}^{x_i})$, $K_i^R = \mathcal{H}(y_{i+1}^{x_i})$ and generates $T_i = K_i^L \oplus K_i^R$. The last user U_n additionally computes $\widehat{T} = k_n \oplus K_n^R$. Each user U_i broadcasts $M_i^{(2)}$ where $M_i^{(2)} = k_i|T_i$ for $1 \leq i \leq n-1$ and $M_n^{(2)} = \widehat{T}|T_n$.

3. (Key Computation) Finally each user U_i computes $\widetilde{K}_{i+1}^R, \widetilde{K}_{i+2}^R, \ldots, \widetilde{K}_{i+(n-1)} (= \widetilde{K}_i^L)$ by using K_i^R as follows.

$$\widetilde{K}_{i+1}^R = T_{i+1} \oplus K_i^R, \widetilde{K}_{i+2}^R = T_{i+2} \oplus \widetilde{K}_{i+1}^R, \ldots, \widetilde{K}_{i+(n-1)}^R = T_{i+(n-1)} \oplus \widetilde{K}_{i+(n-2)}^R.$$

   Then U_i checks if $K_i^L = \widetilde{K}_i^L$ holds. If invalid, then aborts the protocol. Otherwise, U_i has recovered the correct K_n^R and obtains $\widetilde{k}_n$ from $\widehat{T}$. Each user U_i also checks if $\mathcal{H}(\widetilde{k}_n|0) = \mathcal{H}(k_n|0)$. If invalid, then halts the protocol, else computes the session key

$$\mathsf{sk}_{U_i}^{d_i} = \mathcal{H}(k_1|k_2|\ldots|k_{n-1}|k_n|0).$$

This protocol is very efficient and its security is proven in the random oracle model under CDH assumption.

3.3 Protocol DB-PWD : Password-Based Group Key Agreement of [21]

Converting a group key agreement protocol into password-based setting and analyzing its security is a nontrivial task. A careless protocol design may cause dictionary attacks because of low entropy of human memorable passwords. Dutta-Barua proposed a password-based group key agreement protocol in [21] which is obtained by modifying the protocol KLL introducing encryption-based authentication mechanism with the password pw as the secret encryption/descryption key. The protocol proceeds as follows among n instances $\Pi_{U_1}^{d_1}, \ldots, \Pi_{U_n}^{d_n}$. Here $\mathcal{E}_k$, $\mathcal{E}'_k$ and $\mathcal{E}''_k$ are keyed permutations over the sets $\mathcal{S}$, $\mathcal{S}'$ and $\mathcal{S}''$ respectively where $\mathcal{S} = \overline{G}$, $\mathcal{S}' = \{0,1\}^{2l}$, $\mathcal{S}'' = \{0,1\}^l$ and $\mathcal{D}_k$, $\mathcal{D}'_k$ and $\mathcal{D}''_k$ are the respective inverses of $\mathcal{E}_k$, $\mathcal{E}'_k$ and $\mathcal{E}''_k$.

1. (Round 1) Each user U_i chooses a private key $x_i \in_R Z_q^*$ and a nonce $k_i \in_R \{0,1\}^l$, computes $X_i = g^{x_i}$, encrypts it using the password pw to obtain $Y_i = \mathcal{E}_{\mathsf{pw}}(X_i)$ and sends Y_i to his neighbors U_{i-1}, U_{i+1}.

2. (Round 2) User U_i on receiving $Y_{i-1} = \mathcal{E}_{\mathsf{pw}}(X_{i-1})$ and $Y_{i+1} = \mathcal{E}_{\mathsf{pw}}(X_{i+1})$, recovers X_{i-1}, X_{i+1} by decryption operation with pw as key, computes his left key $K_i^L = \mathcal{H}(X_{i-1}^{x_i})$ and right key $K_i^R = \mathcal{H}(X_{i+1}^{x_i})$. User U_i, for $1 \leq i \leq n-1$ computes $\overline{X}_i = K_i^R \oplus K_i^L$, encrypts it to get $\overline{Y}_i = \mathcal{E}'_{\mathsf{pw}}(k_i | \overline{X}_i)$ and sends $\overline{Y}_i$ to the rest of the users in the second round. In contrast, user U_n computes $\overline{X}_n = k_n \oplus K_n^R$, encrypts it to obtain $\overline{Y}_n = \mathcal{E}''_{\mathsf{pw}}(\overline{X}_n)$ and sends $\overline{Y}_n$ to the rest of the users in this round. We note that right key of U_i is same as the left key of U_{i+1}.

3. (Key Computation) Finally, each user U_i on receiving the encrypted messages $\overline{Y}_j$ from all the users, decrypts those and extracts $\overline{X}_j$, for $1 \leq j \leq n$ and $n-1$ nonces k_j, $1 \leq j \leq n-1$. User U_i then recovers the nonce k_n by computing K_n^R as follows making use of his own left key K_i^L and right key K_i^R: U_i computes $K_{i-j}^L = K_{i-j+1}^L \oplus \overline{X}_{i-j}$ for $1 \leq j \leq i-1$. Note that $K_{i-j}^L = K_{i-j-1}^R$ and $K_1^L = K_n^R$. Thus U_i recovers the right key K_n^R of U_n. Then U_i computes the nonce $k_n = \overline{X}_n \oplus K_n^R$ and computes the session key $\mathsf{sk}_{U_i}^{d_i} = \mathcal{H}(k_1 | k_2 | \ldots | k_n)$.

However, Abdalla *et al.* [1] pointed out certain flaws in this protocol.

3.4 Dictionary Attacks on the Protocol DB-PWD

Following redundancies are discovered by Abdalla *et al.* [1] in the transmitted encrypted messages present in the protocol DB-PWD [21]. These make the protocol vulnerable to off-line dictionary attacks.

After first round communication, user U_i receives X_{i-1}, X_{i+1} and U_i has the value X_i. Consider the situation when any two of X_{i-1}, X_i and X_{i+1} are same.

Case (a): $X_{i-1} = X_{i+1}$: This implies $K_i^L = K_i^R$ which in turn yields a redundancy $\overline{X}_i = 0$. This helps as adversary to mount dictionary attacks in off-line.

Case (b): $X_{i-1} = X_i$: This implicitely defines $x_{i-1} = x_i$. Then $\overline{X}_{i-1} = K_{i-1}^R \oplus K_{i-1}^L = \mathcal{H}(g^{x_{i-1}x_i}) \oplus \mathcal{H}(g^{x_{i-2}x_{i-1}})$ and $\overline{X}_i = K_i^R \oplus K_i^L = \mathcal{H}(g^{x_i x_{i+1}}) \oplus \mathcal{H}(g^{x_i x_{i-1}})$. Now if in addition, it happens to be the case that $X_{i-2} = X_{i+1}$ (*i.e.* $x_{i-2} = x_{i+1}$, probability of which is very small for two honest users U_{i-2} and U_{i+1}), then $\overline{X}_{i-1} \oplus \overline{X}_i = 0$, another possible redundancy in the transmitted messages of the protocol DB-PWD. An active adversary may interleave the messages during the protocol execution and manipulate the transmitted messages to create such a redundancy as follows: adversary simply replaces the encrypted value of X_{i+1} by a copy of the encrypted value of X_{i-2} at hand in the first round communication. Later, adversary may use the redundancy $\overline{X}_{i-1} \oplus \overline{X}_i = 0$ to get advantage in off-line dictionary attack. Note that in this scenario, the adversary does not send a message that he encrypts himself by guessing the password.

Case (c): $X_i = X_{i+1}$: Arguing in a similar way as in case (b), if $X_{i-1} = X_{i+2}$ (*i.e.* $x_{i-1} = x_{i+2}$, probability of which is negligible for two honest users U_{i-1} and U_{i+2}), then one obtains the redundancy relation $\overline{X}_i \oplus \overline{X}_{i+1} = 0$. This situation can be created by an active adversary simply by replacing the encrypted value of X_{i+2} by a copy of the encrypted value of X_{i-1} (that he obtains by interleaving the protocol transmission) in the first round communication. This redundancy enables the adversary to mount off-line dictionary attack at a later time even without making a Send query on a message that is encrypted by the adversary himself by guessing the password.

U_1 | U_2 | U_3 | U_4 | U_5
pw | pw | pw | pw | pw
x_1, k_1 | x_2, k_2 | x_3, k_3 | x_4, k_4 | x_5, k_5

U_i computes $\mathsf{str}_i^{(1)} = U_i|d_i|1$ and $T_i = \mathsf{str}_i^{(1)}|g^{x_i}$ for $1 \leq i \leq 5$

$\mathsf{str}_1^{(1)}|\mathcal{E}_{\mathsf{pw}|U_1}(T_1)$ $\quad$ $\mathsf{str}_2^{(1)}|\mathcal{E}_{\mathsf{pw}|U_2}(T_2)$ $\quad$ $\mathsf{str}_3^{(1)}|\mathcal{E}_{\mathsf{pw}|U_3}(T_3)$ $\quad$ $\mathsf{str}_4^{(1)}|\mathcal{E}_{\mathsf{pw}|U_4}(T_4)$ $\quad$ $\mathsf{str}_5^{(1)}|\mathcal{E}_{\mathsf{pw}|U_5}(T_5)$ $\quad$: Round-1

Communications : U_i sends $\mathsf{str}_i^{(1)}|\mathcal{E}_{\mathsf{pw}|U_i}(T_i)$ to U_{i-1}, U_{i+1}, $1 \leq i \leq 5$, $U_0 = U_5, U_6 = U_1$

U_i on decryption recovers $g^{x_{i-1}}, g^{x_{i+1}}$, $1 \leq i \leq 5$, $x_0 = x_5, x_6 = x_1$
U_i aborts the protocol if $g^{x_{i-1}} = g^{x_{i+1}}$ or if $g^{x_i} = g^{x_{i-1}}$ or if $g^{x_i} = g^{x_{i+1}}$
Else U_i computes $K_i^L = \mathcal{H}(g^{x_{i-1}x_i})$, $K_i^R = \mathcal{H}(g^{x_i x_{i+1}})$, $\mathsf{str}_i^{(2)} = U_i|d_i|2$ for $1 \leq i \leq 5$, $x_0 = x_5, x_6 = x_1$
U_i, $1 \leq i \leq 4$ computes $\overline{X}_i = K_i^R \oplus K_i^L$, $\overline{T}_i = \mathsf{str}_i^{(2)}|(k_i|\overline{X}_i)$ and U_5 computes $\overline{X}_5 = k_5 \oplus K_5^R$, $\overline{T}_5 = \mathsf{str}_5^{(2)}|\overline{X}_5$.

$\mathsf{str}_1^{(2)}|\mathcal{E}'_{\mathsf{pw}|U_1}\left(\overline{T}_1\right)$ $\quad$ $\mathsf{str}_2^{(2)}|\mathcal{E}'_{\mathsf{pw}|U_2}\left(\overline{T}_2\right)$ $\quad$ $\mathsf{str}_3^{(2)}|\mathcal{E}'_{\mathsf{pw}|U_3}\left(\overline{T}_3\right)$ $\quad$ $\mathsf{str}_4^{(2)}|\mathcal{E}'_{\mathsf{pw}|U_4}\left(\overline{T}_4\right)$ $\quad$ $\mathsf{str}_5^{(2)}|\mathcal{E}''_{\mathsf{pw}|U_5}\left(\overline{T}_5\right)$ $\quad$: Round-2

Communications : U_i, $1 \leq i \leq 4$ sends $\mathsf{str}_i^{(2)}|\mathcal{E}'_{\mathsf{pw}|U_i}\left(\overline{T}_i\right)$ to U_j, $1 \leq j \leq 5, j \neq i$
and U_5 sends $\mathsf{str}_5^{(2)}|\mathcal{E}''_{\mathsf{pw}|U_5}\left(\overline{T}_5\right)$ to U_j, $1 \leq j \leq 4$

U_i, $1 \leq i \leq 5$ on decryption recovers $k_j|\overline{X}_j$, $1 \leq j \leq 4, j \neq i$ and $\overline{X}_5$ and extracts $k_j, 1 \leq j \leq 4$
U_i, $1 \leq i \leq 5$ recovers K_5^R and and computes $k_5 = \overline{X}_5 \oplus K_5^R$
U_i computes his session key $\mathsf{sk}_{U_i}^{d_i} = \mathcal{H}(k_1|k_2|k_3|k_4|k_5)$ and sets his session identity
$\mathsf{sid}_{U_i}^{d_i} = \{(U_1, d_1), \ldots, (U_5, d_5)\}$

Figure 1: Key agreement in protocol MDB-PWD among $n = 5$ users

Other kind of redundancies may be created by an active adversary by manipulating the messages exchanged during execution of the protocol DB-PWD. So while transforming the unauthenticated protocol KLL into password-based setting, we should pay careful efforts to make an active adversary unable to get any advantage in guessing the password off-line. This is by no means a trivial task.

3.5 Protocol MDB-PWD : Modified Version of the Protocol DB-PWD

We now describe our password-based group key agreement protocol MDB-PWD which is an improvement over the protocol DB-PWD to overcome dictionary attack. The protocol proceeds as follows among n instances $\Pi_{U_1}^{d_1}, \ldots, \Pi_{U_n}^{d_n}$. Here $\mathcal{E}_k$, $\mathcal{E}'_k$ and $\mathcal{E}''_k$ are keyed permutations over the sets $\mathcal{S}$, $\mathcal{S}'$ and $\mathcal{S}''$ respectively specified below and $\mathcal{D}_k$, $\mathcal{D}'_k$ and $\mathcal{D}''_k$ are the respective inverses of $\mathcal{E}_k$, $\mathcal{E}'_k$ and $\mathcal{E}''_k$.

$\mathcal{S} = \{U|d|t|X :$ User U with instance number d sends t^{th} message $X \in \overline{G}\}$
$\mathcal{S}' = \{U|d|t|(k_i|\overline{X}_i) :$ User U with instance number d sends t^{th} message $k_i|\overline{X}_i \in \{0,1\}^{2l}\}$
$\mathcal{S}'' = \{U|d|t|\overline{X} :$ User U with instance number d sends t^{th} message $\overline{X} \in \{0,1\}^{l}\}$

Let $\mathsf{str}_i^{(l)} = U_i|d_i|l$, $1 \leq i \leq n$, $l = 1, 2$.

1. (Round 1) In the first round, each user U_i chooses a private key $x_i \in_R Z_q^*$ and a nonce $k_i \in_R \{0,1\}^l$, computes $X_i = g^{x_i}$, encrypts $T_i = \mathsf{str}_i^{(1)}|X_i$ using $\mathsf{pw}|U_i$ [1] as encryption key to obtain $Y_i = \mathcal{E}_{\mathsf{pw}|U_i}(T_i)$

[1]If necessary, one may use $\mathcal{H}_0(\mathsf{pw}|U_i)$, where $\mathcal{H}_0$ is a hash function. This can be computed off-line and used if $\mathsf{pw}|U_i$ is longer than the desirable limit on the encryption key length.

and sends $\mathsf{str}_i^{(1)}|Y_i$ to his neighbors U_{i-1}, U_{i+1}.

2. (Round 2) In the second round, each user U_i on receiving $\mathsf{str}_{i-1}^{(1)}|Y_{i-1}$, $\mathsf{str}_{i+1}^{(1)}|Y_{i+1}$ from his neighbors, decrypts Y_{i-1}, Y_{i+1} with the decryption function $\mathcal{D}$ and the respective decryption keys $\mathsf{pw}|U_{i-1}$, $\mathsf{pw}|U_{i+1}$ and obtains $T_{i-1} = \mathsf{str}_{i-1}^{(1)}|X_{i-1}$, $T_{i+1} = \mathsf{str}_{i+1}^{(1)}|X_{i+1}$ respectively. U_i aborts the protocol if any two of X_{i-1}, X_i and X_{i+1} are same which occurs with negligible probability. Otherwise, U_i computes his left key $K_i^L = \mathcal{H}(X_{i-1}^{x_i})$ and right key $K_i^R = \mathcal{H}(X_{i+1}^{x_i})$. Each user U_i for $1 \leq i \leq n-1$ computes $\overline{X}_i = K_i^R \oplus K_i^L$, $\overline{T}_i = \mathsf{str}_i^{(2)}|(k_i|\overline{X}_i)$, $\overline{Y}_i = \mathcal{E}'_{\mathsf{pw}|U_i}(T_i)$ and sends $\mathsf{str}_i^{(2)}|\overline{Y}_i$ to the rest of the users in the second round. U_n computes $\overline{X}_n = k_n \oplus K_n^R$, $\overline{T}_n = \mathsf{str}_n^{(2)}|\overline{X}_n$, $\overline{Y}_n = \mathcal{E}''_{\mathsf{pw}|U_n}(T_n)$ and sends $\mathsf{str}_n^{(2)}|\overline{Y}_n$ to all the users. We note that right key of U_i is same as the left key of U_{i+1}.

3. (Key Computation) Each user U_i on receiving $\mathsf{str}_j^{(2)}|\overline{Y}_j$ from U_j, $1 \leq j \leq n$, decrypts using decryption function $\mathcal{D}'$ (or $\mathcal{D}''$) and $\mathsf{pw}|U_j$ as the decryption key to recover $\overline{X}_j$ and instance number d_j. U_i also extracts k_j for $1 \leq j \leq n-1$. User U_i then computes K_n^R as follows making use of his own left key K_i^L and right key K_i^R: U_i computes $K_{i-j}^L = K_{i-j+1}^L \oplus \overline{X}_{i-j}$ for $1 \leq j \leq i-1$. Note that $K_{i-j}^L = K_{i-j-1}^R$ and $K_1^L = K_n^R$. Thus U_i recovers the right key K_n^R and computes $k_n = \overline{X}_n \oplus K_n^R$. Finally user U_i computes his session key $\mathsf{sk}_{U_i}^{d_i} = \mathcal{H}(k_1|k_2|\ldots|k_n)$ and sets his session identity $\mathsf{sid}_{U_i}^{d_i} = \{(U_1, d_1), \ldots, (U_n, d_n)\}$.

Figure 1 illustrates the protocol for $n = 5$. The formal description of the protocol is given in the appendix.

Note 1 : From a received message of the form $\mathsf{str}_k^{(1)}|Y_k$ in the first round, user U_i gets the information about its possible sender U_k and use $\mathsf{pw}|U_k$ as decryption key to decrypt Y_k. U_i on decryption recovers a plaintext of the form $\mathsf{str}_{\tilde{k}}^{(1)}|X_{\tilde{k}}$ and aborts the protocol if $\mathsf{str}_{\tilde{k}}^{(1)} \neq \mathsf{str}_k^{(1)}$. Similar check is done by each user after receiving the messages in the second round communication.

Note 2 : Generally, while executing this protocol, we identify a user U_i with its instance $\Pi_{U_i}^{d_i}$, where d_i is the instance number of the user in the session being executed. At the start of the protocol, the session identity $\mathsf{sid}_{U_i}^{d_i}$ is not known and is built up (by each participant) as the protocol proceeds. Moreover, when an instance Π_U^i aborts the protocol, it sets $\mathsf{acc}_U^i = 0$ and $\mathsf{sk}_U^i = \mathsf{NULL}$. Note that the algorithms are correct provided the users are honest, *i.e.* they do not deviate from the protocol (we additionally assume that the adversary never participates as a user). Then after the execution of the protocol, the group of users agree upon a common session key.

We next point out certain observations regarding the modifications incorporated in the protocol DB-PWD to yield the protocol MDB-PWD in the light of the dictionary attacks mentioned in Section 3.4 [2].

Observation 1: The protocol MDB-PWD is obtained by modifying the unauthenticated protocol KLL of Kim *et al.* [33] and introducing encryption-based authentication mechanism. Note that each user U_i uses $\mathsf{pw}|U_i$ as encryption key. Only those users who have the knowledge of pw, would be able to decrypt the encrypted messages. The direct replacement of the signature scheme used for authenticated version of KLL in [33] by a symmetric encryption scheme using the password pw as secret key does not yield a secure password-based protocol and one can mount an off-line dictionary attack as follows: Observe that in the unauthenticated protocol KLL with n users, each user U_i for $1 \leq i \leq n-1$ sends $k_i|T_i$ whilst user U_n sends $k_n \oplus K_n^R|T_n$ in the second round, where $T_i = K_i^L \oplus K_i^R$ for $1 \leq i \leq n$. We thus obtain the relation

[2] We are grateful to anonymous referees for these observations

$T_1 \oplus T_2 \oplus \ldots \oplus T_n = 0$. Now when we introduce encryption-based authentication mechanism using the password as the secret key, the ciphertexts in the second round communication are simply the encryption of $k_i|T_i$ for $1 \leq i \leq n-1$ and the encryption of $k_n \oplus K_n^R|T_n$. Thus the plaintexts are co-related instead of being random. This redundancy enables an adversary to make the protocol vulnerable to dictionary attacks by guessing the password off-line and verifying whether the decrypted values ($k_1|T_i$ for $1 \leq i \leq n-1$, $k_n \oplus K_n^R|T_n$) in the second round communication leads to $T_1 \oplus T_2 \oplus \ldots \oplus T_n = 0$. If so, the adversary's guess for password is correct. To prevent such attacks, we remove the redundancy by restricting U_n to send the encryption of only $k_n \oplus K_n^R$ instead of $k_n \oplus K_n^R|T_n$ in this round. As a result, the key computation is appropriately modified.

Observation 2: The protocol MDB-PWD is aborted if $X_{i-1} = X_{i+1}$ which in turn implies $K_i^L = K_i^R$. This step is essential to disable an adversary from mounting off-line password guessing attack, because if the protocol proceeds with $K_i^L = K_i^R$ for some $i \neq n$, then $\overline{X}_i = 0$ and the corresponding publicly transmitted value is simply the encryption of a constant string. By maintaining a list, the attacker can exhaustively search for the password.

Observation 3: Also the protocol MDB-PWD is aborted if $X_i = X_{i-1}$ or $X_i = X_{i+1}$. As discussed earlier, an active adversary may manipulate the transmitted messages during protocol execution by Send query so that $\overline{X}_{i-1} \oplus \overline{X}_i = 0$ (adversary simply replaces in the first round the ciphertext of U_{i+1} by the ciphertext of U_{i-2} in the same round) or $\overline{X}_i \oplus \overline{X}_{i+1} = 0$ (adversary replaces the ciphertext of U_{i+2} in the first round by the ciphertext of U_{i-1} in the same round). Now if the encryption-based mechanism uses pw as the encryption key instead of $\mathsf{pw}|U_i$ for user U_i, then an active adversary in the on-line phase may manipulate the messages as described above, search exhaustively for the password by checking $\overline{X}_{i-1} \oplus \overline{X}_i = 0$ or $\overline{X}_i \oplus \overline{X}_{i+1} = 0$ and thus can mount an off-line dictionary attack. To resist these type of manipulation-based redundancies, the encryption by user U_i is done using $\mathsf{pw}|U_i$ as the secret key instead of pw.

4 Security Analysis

We will now state the security result of our password based authenticated group key agreement protocol MDB-PWD. We omit the proof because the proof is similar to that provided in [21] for the protocol DB-PWD except for certain minor modifications. Similar to the protocol DB-PWD, our proposed scheme MDB-PWD is also based on the CDH assumption and security is achieved in both the random oracle model and the ideal cipher model in the security framework formalized by Bellare *et al.* [7]. However, this proof does not deal with forward secrecy.

Theorem 4.1 *The password based encrypted key agreement protocol P described in Section 3.5 satisfies the following:*

$$\mathsf{Adv}_P^{\mathsf{AKA}}(t, q_{\mathcal{E}}, q_H, q_E, q_S) \leq \frac{q_{\mathcal{E}}^2}{L} + \frac{2q_S}{N} + 4q_H q_S^2 \mathsf{Succ}_G^{\mathsf{CDH}}(t)$$

where t is the time bound of the protocol execution P, N is the size of the dictionary of all possible passwords and $q_{\mathcal{E}}, q_H, q_E, q_S$ are respectively the maximum number of encryption/decryption, hash, Execute, Send *queries an adversary may make and $L = \min\{|\mathcal{S}|, |\mathcal{S}'|, |\mathcal{S}''|\}$, $\mathcal{S}$, $\mathcal{S}'$ $\mathcal{S}''$ being as specified in Section 3.5.*

5 Conclusion

We appropriately modify the protocol DB-PWD, proposed by Dutta-Barua [21] to overcome the flaws discovered by Abdalla *et al.* [1] and present its improved variant MDB-PWD. The proposed scheme MDB-PWD is secure under CDH assumption in both the random oracle model and the ideal cipher model in the security framework of Bellare *et al.* [7]. To obtain secure password-based efficient group key agreement protocol under standard assumption without using random oracle is an interesting research topic and this area requires to be studied for further improvement.

6 Acknowledgement

We are thankful to Prof. Emmanuel Bresson for a personal communication to bring in attention the redundancies mentioned in Section 3.4 present in an initial draft of this manuscript and also discussing the idea to overcome those.

References

[1] M. Abdalla, E. Bresson, O. Chevassut and D. Pointcheval. *Password-based Group Key Exchange in a Constant Number of Rounds.*. In proceedings of PKC 2006, LNCS, Springer-Verlag, 2006.

[2] M. Abdalla, M. Bellare and P. Rogaway. *DHIES: An Encryption Scheme Based on the Diffie-Hellman Problem.* In proceedings of CT-RSA 2001, LNCS 2020, pp. 143-158, Springer-Verlag, 2001.

[3] M. Abdalla, P. A. Fouque and D. Pointcheval. *Password-Based Authenticated Key Exchange in the Three-Party Setting.* In proceedings of PKC 2005, LNCS 3386, pp. 65-84, Springer-Verlag, 2004.

[4] N. Asokan and P. Ginzboorg. *Key Agreement in Ad-hoc Networks.* In Computer Communications, 23(18), pp. 1627-1637, 2000.

[5] M. Bellare and P. Rogaway. *Entity Authentication and Key Distribution.* In proceedings of Crypto 1993, LNCS 773, pp. 231-249, Springer-Verlag, 1994.

[6] M. Bellare and P. Rogaway. *Provably Secure Session Key Distribution: The Three-party Case.* In proceedings of STOC 1995, pp. 57-66, ACM Press, 1995.

[7] M. Bellare, D. Pointcheval, and P. Rogaway. *Authenticated Key Exchange Secure Against Dictionary Attacks.* In proceedings of Eurocrypt 2000, LNCS 1807, pp. 139-155, Springer-Verlag, 2000.

[8] S. M. Bellovin and M. Merritt. *Augmented Encrypted Key Exchange : A Password-based Protocol Secure against Dictionary Attacks and Password File Compromise.* In proceedings of ACM CCS 1993, pp. 244-250, ACM Press, 1993.

[9] Bluetooth. *Specification of Bluetooth System.* Available at http://www.bluetooth.com/developer/specification/specification.asp.

[10] M. Boyarsky. *Public-key cryptography and password protocols : The multi-user case.* In proceedings of ACM CCS 1999, pp. 63-72, ACM Press, 1999.

[11] V. Boyko, P. MacKenzie and S. Patel. *Provably Secure Password-authenticated Key Exchange using Diffie-Hellman*. In proceedings of Eurocrypt 2000, LNCS 1807, pp. 156-171, Springer-Verlag 2000.

[12] E. Bresson, O. Chevassut and D. Pointcheval. *New Security Results on Encrypted Key Exchange*. In proceedings of PKC 2004, LNCS 2947, pp. 145-158, Springer-Verlag, 2004.

[13] E. Bresson, O. Chevassut and D. Pointcheval. *Proof of Security for Password-based Key Exchange (IEEE P1363 AuthA Protocol and Extensions)*. In proceedings of ACM CCS 2003, pp. 241-250, ACM Press.

[14] E. Bresson, O. Chevassut and D. Pointcheval. *Group Diffie-Hellman Key Exchange Secure Against Dictionary Attack*. In proceedings of Asiacrypt 2002, LNCS 2501, pp. 497-514, Springer-Verlag, 2002

[15] E. Bresson, O. Chevassut, and D. Pointcheval. *Provably Authenticated Group Diffie-Hellman Key Exchange - The Dynamic Case*. In proceedings of Asiacrypt 2001, LNCS 2248, pp. 290-309, Springer-Verlag, 2001.

[16] E. Bresson, O. Chevassut, D. Pointcheval, and J. J. Quisquater. *Provably Authenticated Group Diffie-Hellman Key Exchange*. In proceedings of ACM CCS 2001, pp. 255-264, ACM Press, 2001.

[17] M. Burmester and Y. Desmedt. *A Secure and Efficient Conference Key Distribution System*. In proceedings of Eurocrypt 1994, LNCS 950, pp. 275-286, Springer-Verlag, 1995.

[18] 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 ICICS 2002, LNCS 2513, pp. 134-146, Springer-Verlag, 2002.

[19] Y. Ding and P. Horster. *Undetectable On-line Password Guessing Attacks*. In ACM SIGOPS Operating Systems Review, 29(4), pp. 77-86, 1995.

[20] R. Dutta and R. Barua. *Constant Round Dynamic Group Key Agreement*. In proceedings of ISC 2005, LNCS 3650, pp. 74-88, Springer-Verlag, 2005 Also available at http://eprint.iacr.org/2005/221.

[21] R. Dutta and R. Barua. *Password-Based Encrypted Group Key Agreement*. In the International Journal of Network Security (IJNS), Vol.3, No.1, pp. 23-34, July 2006 (to appear). Available at http://isrc.nchu.edu.tw/ijns.

[22] R. Gennaro and Y. Lindell. *A Framework for Password-based Authenticated Key Exchange*. In proceedings of Eurocrypt 2003, LNCS 2656, pp. 524-543, Springer-Verlag, May 2003. Also available at http://eprint.iacr.org/2003/032.

[23] O. Goldreich and Y. Lindell. *Session-key Generation using Human Memorable Passwords only*. In proceedings of Crypto 2001, LNCS 2139, pp. 408-432, Springer-Verlag, 2001. Also available at http://eprint.iacr.org/2000/057.

[24] L. Gong. *Optimal Authentication Protocols resistant to Password Guessing Attacks*. In CSFW'95: The 8th IEEE Computer Security Foundation Workshop, pp. 24-29, Kenmare, County Kerry, Ireland, March 13-15, 1995. IEEE Computer Society.

[25] L. Gong, T. M. A. Lomas, R. M. Needham and J. H. Saltzer. *Protecting Poorly Chosen Secrets from Guessing Attacks*. In IEEE JSAC, 11(5), pp. 648-656, June 1993.

[26] S. Halevi and H. Krawczyk. *Public Key Cryptography and Password Protocols*. In ACM Transactions on Information and System Security, pp. 524-543, ACM Press, 1999.

[27] D. P. Jablon. *Strong Password-only Authenticated Key Exchange*. In SIGCOMM Computer Communication Review, 26(5), pp.5-26, 1996.

[28] M. Jakobsson and S. Wetzel. *Security Weaknesses in Bluetooth*. In proceedings CT-RSA 2001, LNCS 2020, pp. 176-191, Springer-Verlag, 2001.

[29] S. Jiang and G. Gong. *Password-based Key Exchange With Mutual Authentication*. In proceedings of SAC 2004, LNCS 3006, pp. 291-306, Springer-Verlag, 2004. Also available at http://www.iacr.org/2004/196.

[30] J. Katz and M. Yung. *Scalable Protocols for Authenticated Group Key Exchange*. In proceedings of Crypto 2003, LNCS 2729, pp. 110-125, Springer-Verlag, 2003.

[31] J. Katz, R. Ostrovsky and M. Yung. *Efficient Password-authenticated Key Exchange using Human-Memorable Passwords*. In proceedings of Eurocrypt 2001, LNCS 2045, pp. 475-494, Springer-Verlag, 2001.

[32] C. Kaufman, R. Perlman and M. Speciner. *Network Security*. Prentice Hall, 1997.

[33] H. J. Kim, S. M. Lee and D. H. Lee. *Constant-Round Authenticated Group Key Exchange for Dynamic Groups*. In proceedings of Asiacrypt 2004, LNCS 3329, pp. 245-259, Sringer-Verlag, 2004.

[34] J. Kim, S. Kim, J. Kwak and D. Won. *Cryptanalysis and Improvement of Password Authenticated Key Exchange Scheme between Clients with Different Passwords*. In ICCSA 2004, LNCS 3043, pp. 895-902, Springer-Verlag, 2004.

[35] H. Krawczyk. *SIGMA: The "SIGn-and-MAc" Approach to Authenticate Diffie-Hellman and its use in the Ike Protocols*. In proceedings of Crypto 2003, LNCS 2729, pp. 400-425, Springer-Verlag, 2003.

[36] S. M. Lee, J. Y. Hwang and D. H. Lee. *Efficient Password-Based Group Key Exchange*. In proceedings of TrustBus 2004, LNCS 3184, pp. 191-199, Springer-Verlag, 2004.

[37] C. L. Lin, H. M. Sun and T. Hwang. *Three-party Encrypted Key Exchange: Attacks and Solution*. In ACM SIGOPS Operating Systems Review, 34(4), pp. 12-20, 2000.

[38] C. L. Lin, H. M. Sun, M. Steiner and T. Hwang. *Three-party Encrypted Key Exchange without Server Public Keys*. In IEEE Communications Letters, 5(12), pp.497-499, 2001.

[39] S. Lucks. *Open Key Exchange: How to Defeat Dictionary Attacks without Encrypting Public Keys*. In proceedings of the Workshop of Security Protocols, LNCS 1361, pp. 79-90, Springer-Verlag, 1997.

[40] P. D. Mackenzie. *The PAK Suit : Protocols for Password-Authenticated Key Exchange*. Contributions to IEEE P1363.2, 2002.

[41] P. MacKenzie, S. Patel and R. Swaminathan. *Password-authenticated Key Exchange Based on RSA*. In proceedings of Asiacrypt 2000, LNCS 1976, pp. 599-613, Springer-Verlag, 2000.

[42] P. D. Mackenzie, T. Shrimpton and M. Jakobsson. *Threshold Password-authenticated Key Exchange*. In proceedings of Crypto 2002, LNCS 2442, pp. 385-400, Springer-Verlag, 2002.

[43] P. MacKenzie and R. Swaminathan. *Secure Network Authentication with Password Identification*. Submission to IEEE P1363a, August 1999. Available from http://grouper.ieee.org/groups/1363/.

[44] A. Menezes, P. C. Van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, 1997. Also available at http://cacr.math.uwaterloo.ca/hac.

[45] NIST. AES, December 2000. Available at http://www.nist.gov/aes.

[46] K. Obraczka, G. Tsudik and K. Viswanath. *Pushing the Limits of Multi-cast in Ad Hoc Networks*. In International Conference on Distributed Computing System, April 2001, pp. 719-722.

[47] S. Patel. *Number Theoretic Attacks on Secure Password Schemes*. In proceedings of the 1997 IEEE Symposium on Security and Privacy, pp. 236-247, 1997.

[48] C. E. Perkins. *Ad Hoc Networking*. Addition Wesley, 2001.

[49] M. D. Raimondo and R. Gennaro. *Provably Secure Threshold Password-authenticated Key Exchange*. In proceedings of Eurocrypt 2003, LNCS 2656, pp. 507-523, Springer-Verlag, 2003.

[50] V. Shoup. *On Formal Models for Secure Key Exchange*. In IBM Technical Report RZ 3120, 1999. Also available at http://shoup.net/papers.

[51] J. G. Steiner, B. C. Neuman and J. L. Schiller. *Kerberos: An Authentication Service for Open Networks*. In proceedings of the USENIX Winter Conference, pp. 191-202, Dallas, 1998.

[52] M. Steiner, G. Tsudik and M. Waidner. *Refinement and Extension of Encrypted Key Exchange*. In ACM SIGOPS Operating Systems Review, 29(3), pp.22-30, 1995.

[53] G. Tsudik and E. Van Herreweghen. *Some Remarks on Protecting Weak Key and Poorly-chosen Secrets from Guessing Attacks*. In SRDS 1993: The 12th Symposium on Reliable Distributed Systems, pp. 136-142, Princeton, New Jersey, USA, October 6-8, 1993, IEEE Computer Society.

[54] 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 3089, pp. 414-425, Springer-Verlag, 2004.

[55] T. Wu. *The Secure Remote Password Protocol*. In proceedings of the 1998 Internet Society Symposium on Network and Distributed System Security, pp. 97-111, 1998.

[56] H. T. Yeh, H. M. Sun and T. Hwang. *Efficient Three-party Authentication and Key Agreement Protocols resistant to Password Guessing Attacks*. In Journals of Information Science and Engineering, 19(6), pp. 1059-1070, November 2003.

[57] M. Zhang. *Password Authenticated Key Exchange Using Quadratic Residues*. In proceedings of ACNS 2004, LNCS 3089, pp. 233-247, Springer-Verlag, 2004.

[58] L. Zhou and Z. J. Hass. *Securing Ad Hoc Networks.* In IEEE Network Magazine, 13(6), pp. 24-30, 1999.

A Algorithm for the protocol MDB-PWD

procedure PwdKeyAgree ($U[1,\ldots,n]$)
  (Round 1):
  $U_0 = U_n, U_{n+1} = U_1$;
1. **for** $i = 1$ to n **do in parallel**
2. $U_i (= \Pi_{U_i}^{d_i})$ chooses $x_i \in_R Z_q^*$ and nonce $k_i \in_R \{0,1\}^l$;
3. U_i computes $X_i = g^{x_i}$, $\mathsf{str}_i^{(1)} = U_i|d_i|1$ and $Y_i = \mathcal{E}_{\mathsf{pw}|U_i}(\mathsf{str}_i^{(1)}|X_i)$;
4. U_i sends $\mathsf{str}_i^{(1)}|Y_i$ to U_{i-1} and U_{i+1};
5. **end for**
  (Round 2):
  $Y_0 = Y_n, Y_{n+1} = Y_1$;
6. **for** $i = 1$ to $n-1$ **do in parallel**
7. U_i on receiving $\mathsf{str}_{i-1}^{(1)}|Y_{i-1}$ from U_{i-1} and $\mathsf{str}_{i+1}^{(1)}|Y_{i+1}$ from U_{i+1}, computes $\mathcal{D}_{\mathsf{pw}|U_{i-1}}(Y_{i-1})$, $\mathcal{D}_{\mathsf{pw}|U_{i+1}}(Y_{i+1})$ to recover $\mathsf{str}_{i-1}^{(1)}|X_{i-1}$ and $\mathsf{str}_{i+1}^{(1)}|X_{i+1}$ respectively;
8. U_i aborts the protocol if any two of X_{i-1}, X_i, X_{i+1} are same; else executes the following steps;
9. U_i computes $K_i^L = \mathcal{H}(X_{i-1}^{x_i})$, $K_i^R = \mathcal{H}(X_{i+1}^{x_i})$, $\overline{X}_i = K_i^R \oplus K_i^L$, $\mathsf{str}_i^{(2)} = U_i|d_i|2$ and $\overline{Y}_i = \mathcal{E}'_{\mathsf{pw}|U_i}(\mathsf{str}_i^{(2)}|(k_i|\overline{X}_i))$;
10. U_i sends $\mathsf{str}_i^{(2)}|\overline{Y}_i$ to the rest of the users;
11. **end for**
12. U_n on receiving $\mathsf{str}_{n-1}^{(1)}|Y_{n-1}$ from U_{n-1} and $\mathsf{str}_1^{(1)}|Y_1$ from U_1, computes $\mathcal{D}_{\mathsf{pw}|U_{n-1}}(Y_{n-1})$, $\mathcal{D}_{\mathsf{pw}|U_1}(Y_1)$ and recovers $\mathsf{str}_{n-1}^{(1)}|X_{n-1}$, $\mathsf{str}_1^{(1)}|X_1$ respectively;
13. U_n computes $K_n^L = \mathcal{H}(X_{n-1}^{x_n})$, $K_n^R = \mathcal{H}(X_1^{x_n})$, $\overline{X}_n = k_n \oplus K_n^R$, $\mathsf{str}_n^{(2)} = U_n|d_n|2$ and $\overline{Y}_n = \mathcal{E}''_{\mathsf{pw}|U_n}\left(\mathsf{str}_n^{(2)}|\overline{X}_n\right)$;
14. U_n sends $\mathsf{str}_n^{(2)}|\overline{Y}_n$ to the rest of the users;
  Note that $K_i^R = K_{i+1}^L$ for $1 \le i \le n-1$ and $K_n^R = K_1^L$;
  (Key Computation):
15. **for** $i = 1$ to n **do in parallel**
16.   **for** $j = 1$ to $n-1$, $j \neq i$ **do**
17. U_i computes $\mathcal{D}'_{\mathsf{pw}|U_j}(\overline{Y}_j)$ and extracts $\overline{X}_j$, k_j, d_j;
18.   **end for**
19. U_i computes $\mathcal{D}''_{\mathsf{pw}|U_n}(\overline{Y}_n)$ and extracts $\overline{X}_n$, d_n;
20. **end for**
21. **for** $i = 1$ to n **do in parallel**
22.   **for** $j = 1$ to $i-1$ **do**
23. U_i computes $K_{i-j}^L = K_{i-j+1}^L \oplus \overline{X}_{i-j}$;
    Note that $K_{i-j}^L = K_{i-j-1}^R$ and $K_1^L = K_n^R$.
24.   **end for**
    Thus U_i has recovered K_n^R.
25. **end for**

for $i = 1$ to n **do in parallel**

  U_i computes $k_n = \overline{X}_n \oplus K_n^R$, the session key $\mathsf{sk}_{U_i}^{d_i} = \mathcal{H}(k_1|k_2|\ldots|k_n)$ and the session identity $\mathsf{sid}_{U_i}^{d_i} = \{(U_1, d_1), \ldots, (U_n, d_n)\}$;

end for

end $\mathsf{PwdKeyAgree}$