Interactive Diffie-Hellman Assumptions With Applications to Password-Based Authentication

Michel Abdalla and David Pointcheval

Departement d'Informatique
École normale supérieure
45 Rue d'Ulm, 75230 Paris Cedex 05, France
{Michel.Abdalla,David.Pointcheval}@ens.fr
http://www.di.ens.fr/users/{mabdalla,pointche}.

Abstract. Password-based authenticated key exchange are protocols that are designed to provide strong authentication for client-server applications, such as online banking, even when the users' secret keys are considered weak (e.g., a four-digit pin). In this paper, we address this problem in the three-party setting, in which the parties trying to authenticate each other and to establish a session key only share a password with a trusted server and not directly among themselves. This is the same setting used in the popular *Kerberos* network authentication system. More precisely, we introduce a new three-party password-based authenticated key exchange protocol. Our protocol is reasonably efficient and has a per-user computational cost that is comparable to that of the underlying two-party authenticated key exchange protocol. The proof of security is in the random oracle model and is based on new and apparently stronger variants of the decisional Diffie-Hellman problem which are of independent interest.

Keywords. Password-based authentication, Diffie-Hellman assumptions, multi-party protocols.

1 Introduction

Motivation. Key exchange protocols are cryptographic primitives that allow users communicating over an unreliable channel to establish secure sessions keys. They are widely used in practice and can be found in several different flavors. In this paper, we are interested in the setting in which the secret keys shared among the users are not uniformly distributed over a large space, but are rather drawn from a small set of values (e.g., a four-digit pin). This seems to be a more realistic scenario since, in practice, these keys are usually chosen by humans. Moreover, they also seem to be more convenient to use as they do not require the use of more specialized hardware for storing or generating secret keys.

Due to the low entropy of the secret keys, password-based protocols are always subject to password-guessing attacks. In these attacks, also known as dictionary attacks, the adversary tries to impersonate a user by simply guessing the value of his password. Since these attacks cannot be completely ruled out, the goal of password-based protocol is to limit the adversary's capability to the online case only. In an online attack, whose success probability is still non-negligible, the adversary needs be present and interact with the system during his attempt to impersonate a user. In other words, the adversary has no means of verifying off-line whether or not a given password guess is correct. The idea of restricting the adversary to the online case only is that we can limit the damage caused by such attacks by using other means, such as limiting the number of failed login attempts or imposing a minimum time interval between failed attempts.

PASSWORD-BASED PROTOCOLS IN THE 3-PARTY MODEL. Due to their practical aspects, password-based key exchange protocols have been the subject of extensive work in the recent years. But despite the attention given to them, it was only recently [1] that the problem has been formally addressed in the three-party model, where the server is considered to be a trusted third party (TTP). This is the same scenario used in the popular 3-party *Kerberos* authentication system. The main advantage of these systems is that users are only required to remember a single password, the one they share with a trusted server, while still being able to establish secure sessions with many users. The main drawback is the need of the trusted server during the establishment of these secure sessions.

In [1], the authors put forth a formal model of security for 3-party password-based authenticated key exchange (PAKE) and present a natural and generic construction of a 3-party password-based authenticated key exchange from any secure 2-party one. There are three phases in their generic construction. In the first phase, a high-entropy session key is generated between the server and each of the two clients using an instance of the 2-party PAKE protocol for each client. In the second phase, a message authentication code (MAC) key is distributed by the server to each client using a 3-party key distribution protocol. In the final phase, both clients execute an authenticated version of the Diffie-Hellman key exchange protocol [13] using the MAC keys obtained in the previous phase.

EFFICIENT 3-PARTY PASSWORD-BASED PROTOCOLS. Though attractive and natural, the construction given in [1] is not particularly efficient. Not only does it require a large amount of computation by the server and the clients, but it also has a large number of rounds. In this paper, we show how to improve both measures when the underlying 2-party password-based key exchange protocol is based on the encrypted key exchange protocol of Bellovin and Merritt [7].

The main idea behind our protocol is quite simple. In order to protect legitimate users from learning each other's password via an off-line dictionary attack, the server randomizes all the values that it receives from one participant before re-encrypting them using the password of another participant. Starting from this idea, we can design a provably-secure protocol, based on the encrypted key exchange of Bellovin and Merritt [7]. The new protocol is quite simple and elegant and, yet, we can prove its security (see Section 4). Moreover, it is also rather efficient, specially when compared to the generic construction in [1]. In particular, the costs for each participant of the new 3-party protocol are comparable to those of a 2-party key exchange protocol. The main drawback of the new 3-party protocol is that it relies on stronger assumptions than those used by the generic construction in addition to being in the random oracle model.

New Diffie-Hellman assumptions. Despite the simplicity of the protocol, its proof of security does not follow directly from the standard Diffie-Hellman assumptions and requires the introduction of some new variants of these standard assumptions. We call them chosen-basis Diffie-Hellman assumptions due to the adversary's capability to choose some of the bases used in the definition of the problem. These assumptions are particularly interesting when considered in the context of password-based protocols and we do expect to find applications for them other than the ones in this paper. Despite being apparently stronger than the standard Diffie-Hellman assumptions, no separations or reductions between these problems are known. Hence, to gain more confidence in these assumptions, we also provide lower bounds for them in the generic group model of Shoup [23].

Related work. Password-based authenticated key exchange has been quite extensively studied in recent years. While the majority of the work deals with different aspects of 2-party key exchange (e.g., [3, 8, 9, 14, 15, 17, 20]), only a few take into account the 3-party scenario (e.g., [1, 10, 16, 19, 24–26]). Moreover, to the best of our knowledge, with the exception of the generic construction in [1], none of the password-based schemes in the 3-party scenario enjoys provable security. Other protocols, such as the Needham and Schroeder protocol for authenticated key exchange [22] and the symmetric-key-based key distribution scheme of Bellare and Rogaway [5], do consider the 3-party setting, but not in the password-based scenario. As we mentioned above, the goal of the present work is to provide a more efficient and provably-secure alternative to the generic protocol of [1].

Contributions. We make two main contributions in this paper.

An efficient construction in Random Oracle model. We present a new construction of a 3-party password-based (implicitly) authenticated key exchange protocol, based on the encrypted key exchange protocols in [6, 21, 9]. The protocol is quite efficient, requiring only 2 exponentiations and a few multiplications from each of the parties involved in the protocol. This amounts to less than half of the computational cost for the server if the latter were to perform two separate key exchange protocols, as in the generic construction of [1]. The gain in efficiency, however, comes at the cost of stronger security assumptions. The security proof is in the Random Oracle model and makes use of new and stronger variations of the Decisional Diffie-Hellman assumption.

New Diffie-Hellman assumptions. The proof of security of our protocol makes use of new non-standard variations of the standard Diffie-Hellman assumptions. These assumptions are of independent interest as they deal with interesting relations between the computational and the decisional versions of the Diffie-Hellman assumption. We call them chosen-basis decisional Diffie-Hellman assumptions, given the adversary's capability to choose some of the bases used in the definition of the problem. Despite being apparently stronger than the standard Diffie-Hellman assumptions, no separations or reductions between these problems are known. Lower bounds in the generic group model are also provided for these new assumptions.

Organization. In Section 2, we recall the formal model of security for 3-party password-based authenticated key exchange. Next, in Section 3, we recall the definitions of the standard Diffie-Hellman assumptions and introduce some new variants of these assumptions, on which the security of our protocol is based. We also present some relations between these assumptions. Section 4 then presents our 3-party password-based key exchange protocol, called 3PAKE, along with its security claims. Some important remarks are also presented in Section 4.

2 Definitions

We now recall the formal security model for 3-party password-authenticated key exchange protocols introduced in [1], which in turn builds upon those of Bellare and Rogaway [4, 5] and that of Bellare, Pointcheval, and Rogaway [3].

Protocol participants. The distributed system we consider is made up of three disjoint sets: $\mathcal{S}$, the set of trusted servers; $\mathcal{C}$, the set of honest clients; and $\mathcal{E}$, the set of malicious clients. We also denote the set of all clients by $\mathcal{U}$. That is, $\mathcal{U} = \mathcal{C} \cup \mathcal{E}$. As in [1], we also assume $\mathcal{S}$ to contain only a single trusted server.

Long-lived keys. Each participant $U \in \mathcal{U}$ holds a password pw_U. The server S holds a vector $\mathsf{pw}_S = \langle pw_U \rangle_{U \in \mathcal{U}}$ with an entry for each client.

Execution of the protocol. The interaction between an adversary $\mathcal{A}$ and the protocol participants occurs only via oracle queries, which model the adversary capabilities in a real attack. While in a concurrent model, several instances may be active at any given time, only one active user instance is allowed for a given intended partner and password in a non-concurrent model. Let U^i denote the instance i of a participant U and let b be a bit chosen uniformly at random. These queries are as follows:

- *Execute*$(U_1^{i_1}, S^j, U_2^{i_2})$: This query models passive attacks in which the attacker eavesdrops on honest executions among client instances $U_1^{i_1}$ and $U_2^{i_2}$ and the server instance S^j. The output of this query consists of the messages that were exchanged during the honest execution of the protocol.
- *Reveal*(U^i): This query models the misuse of session keys by clients. It returns to the adversary the session key of client instance U^i, if the latter is defined.
- *SendClient*(U^i, m): This query models an active attack. It outputs the message that client instance U^i would generate upon receipt of message m.
- *SendServer*(S^j, m): This query models an active attack against a server. It outputs the message that server instance S^j would generate upon receipt of message m.
- *Test*(U^i): This query is used to measure the semantic security of the session key of client instance U^i, if the latter is defined. If the key is not defined, it returns $\perp$. Otherwise, it returns either the session key held by client instance U^i if $b = 0$ or a random of key of the same size if $b = 1$.

Notation. Following [1], which in turn follows [4, 5], an instance U^i is said to be *opened* if a query *Reveal*(U^i) has been made by the adversary. We say an instance U^i is *unopened* if it is not *opened*. We say an instance U^i has *accepted* if it goes into an accept mode after receiving the last expected protocol message.

Partnering. The definition of partnering uses the notion of session identifications (*sid*), which in our case is the partial transcript of the conversation between the clients and the server before the acceptance. More specifically, two instances U_1^i and U_2^j are said to be 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 instance U^i is considered *fresh* if that it has *accepted*, both U^i and its partner (as defined by the partner function) are *unopened* and they are both instances of honest clients.

AKE semantic security. Consider an execution of the key exchange protocol P by the adversary $\mathcal{A}$, in which the latter is given access to the *Execute*, *SendClient*, *SendServer*, and *Test* oracles and asks at most one *Test* query to a *fresh* instance of an honest client. Let b' be his output. Such an adversary is said to win the experiment defining the semantic security if $b' = b$, where b is the hidden bit used by the *Test* oracle. Let Succ denote the event in which the adversary wins this game.

The *advantage* of $\mathcal{A}$ in violating the AKE semantic security of the protocol P and the *advantage function* of the protocol P, when passwords are drawn from a dictionary $\mathcal{D}$, are defined, respectively, as follows:

$$\mathbf{Adv}_{P,\mathcal{D}}^{\mathrm{ake}}(\mathcal{A}) = 2 \cdot \Pr[\,\textsc{Succ}\,] - 1 \quad \text{and} \quad \mathbf{Adv}_{P,\mathcal{D}}^{\mathrm{ake}}(t, R) = \max_{\mathcal{A}}\{\, \mathbf{Adv}_{P,\mathcal{D}}^{\mathrm{ake}}(\mathcal{A})\,\},$$

where maximum is over all $\mathcal{A}$ with time-complexity at most t and using resources at most R (such as the number of oracle queries). The definition of time-complexity is the usual one, which includes the maximum of all execution times in the experiments defining the security plus the code size. The probability rescaling was added to make the advantage of an adversary that simply guesses the bit b equal to 0.

A 3-party password-based key exchange protocol P is said to be semantically secure if the advantage $\mathbf{Adv}_{P,\mathcal{D}}^{\mathrm{ake}}$ is only negligibly larger than $kn/|\mathcal{D}|$, where n is number of active sessions and k is a constant. Note that $k = 1$ is the best one can hope for since an adversary that simply guesses the password in each of the active sessions has an advantage of $n/|\mathcal{D}|$.

3 Diffie-Hellman Assumptions

In this section, we recall the definitions of standard Diffie-Hellman assumptions and introduce some new variants, which we use in the security proof of our protocol. We also present some relations between these assumptions.

Henceforth, we assume a finite cyclic group G of prime order p generated by an element g. We also call the tuple $\mathbb{G} = (G, g, p)$ a represented group.

Computational Diffie-Hellman assumption: CDH. The CDH assumption in a represented group $\mathbb{G}$ states that given g^u and g^v, where u, v were drawn at random from Z_p, it is hard to compute g^{uv}. This can be defined more precisely by considering an Experiment $\mathbf{Exp}_{\mathbb{G}}^{\mathrm{cdh}}(\mathcal{A})$, in which we select two values u and v in Z_p, compute $U = g^u$, and $V = g^v$, and then give both U and V to $\mathcal{A}$. Let Z be the output of $\mathcal{A}$. Then, the Experiment $\mathbf{Exp}_{\mathbb{G}}^{\mathrm{cdh}}(\mathcal{A})$ outputs 1 if $Z = g^{uv}$ and 0 otherwise. We define the *advantage* of $\mathcal{A}$ in violating the CDH assumption as $\mathbf{Adv}_{\mathbb{G}}^{\mathrm{cdh}}(\mathcal{A}) = \Pr[\,\mathbf{Exp}_{\mathbb{G}}^{\mathrm{cdh}}(\mathcal{A}) = 1\,]$ and the *advantage function* of the group, $\mathbf{Adv}_{\mathbb{G}}^{\mathrm{cdh}}(t)$, as the maximum value of $\mathbf{Adv}_{\mathbb{G}}^{\mathrm{cdh}}(\mathcal{A})$ over all $\mathcal{A}$ with time-complexity at most t.

Decisional Diffie-Hellman assumption: DDH. Roughly, the DDH assumption states that the distributions (g^u, g^v, g^{uv}) and (g^u, g^v, g^w) are computationally indistinguishable when u, v, w are drawn at random from Z_p. As before, we can define the DDH assumption more formally by defining two experiments, $\mathbf{Exp}_{\mathbb{G}}^{\mathrm{ddh\text{-}real}}(\mathcal{A})$ and $\mathbf{Exp}_{\mathbb{G}}^{\mathrm{ddh\text{-}rand}}(\mathcal{A})$. In both experiments, we compute two values $U = g^u$ and $V = g^v$ as before. But in addition to that, we also provide a third input, which is g^{uv} in $\mathbf{Exp}_{\mathbb{G}}^{\mathrm{ddh\text{-}real}}(\mathcal{A})$ and g^z for a random z in $\mathbf{Exp}_{\mathbb{G}}^{\mathrm{ddh\text{-}rand}}(\mathcal{A})$. The goal of the adversary is to guess a bit indicating the experiment he thinks he is in. We define the *advantage* of $\mathcal{A}$ in violating the DDH assumption, $\mathbf{Adv}_{\mathbb{G}}^{\mathrm{ddh}}(\mathcal{A})$, as $\Pr[\,\mathbf{Exp}_{\mathbb{G}}^{\mathrm{ddh\text{-}real}}(\mathcal{A}) = 1\,] - \Pr[\,\mathbf{Exp}_{\mathbb{G}}^{\mathrm{ddh\text{-}rand}}(\mathcal{A}) = 1\,]$. The *advantage function* of the group, $\mathbf{Adv}_{\mathbb{G}}^{\mathrm{ddh}}(t)$, is then defined in a similar manner.

Chosen-basis Decisional Diffie-Hellman assumptions. The security of our protocol relies on two new variations of the DDH assumption, which we call *Chosen-basis Decisional Diffie-Hellman* assumptions 1 and 2, where 1 and 2 denote the number of values outputted by the adversary at the end of the first phase. So, let us start by motivating the first of these, the CDDH1 assumption. A similar argument can be used to justify our second assumption, CDDH2, and hence we only provide its formal definition.

The CDDH1 assumption considers an adversary running in two stages. In a find stage, the adversary is given three values $U = g^u$, $V = g^v$, and $X = g^x$, where u, v, and x are random elements in Z_p. The adversary should then select an element Y in G. Using Y, we then consider two games. In the first game $(b = 0)$, we pick a random bit b_0 and set another bit $b_1 = b_0$ to the same value. We then choose two secret random values r_0 and r_1, we compute two pairs of values (X_0, K_0) and (X_1, K_1) using bits r_{b_0} and r_{b_1} as in Definition 1 below and the value $Y' = Y^{r_0}$, and we give them to the adversary. In other words, in this game, we compute both pairs using the same exponent, which may or may not be the same used in the computation of Y' from Y, the value previously chosen by the adversary. The second game $(b = 1)$ is similar to the first one except that b_1 is set to $1 - b_0$ and hence the pairs (X_0, K_0) and (X_1, K_1) are computed using different exponents. The adversary wins if he guesses correctly the bit $b = b_0 \oplus b_1$.

To understand the subtlety of the assumption, let us consider the different strategies the adversary may take. First, if the adversary chooses $Y = g^y$ knowing its discrete log y, then he can compute $\mathrm{CDH}(X/U, Y)$ as well as g^{r_0}. He can also verify that each key K_i is in fact X_i^y. Hence, the keys K_i do not leak any additional information. Let $g_0 = X/U$ and $g_1 = X/V$. Then $X_i = g_i^{r_{b_i}}$. Thus, the adversary in this case needs to be able to tell whether the same exponent is used in X_i knowing only g^{r_0}. We believe this is not easy.

Now let us consider the case in which the adversary chooses Y as a function of the inputs that he was given at the find stage (hence not knowing y). In this case, the adversary should not be able to compute the CDH value and hence the values K_i are not of much help either. Consider the case where he chooses $Y = X/U$. Then, using Y', the adversary can easily know the value of b_0 by checking whether $X_0 = Y'$. However, that does not seem to be of much help since he now needs to tell whether $X_0 = g_0^{r_{b_0}}$ was computed using the same exponent as $X_1 = g_1^{r_{b_1}}$. Knowing b_0 does not seem of any help. We now proceed with the formal definitions.

Definition 1 (CDDH1)**.** *Let $\mathbb{G} = (G, g, p)$ be a represented group and let $\mathcal{A}$ be an adversary. Consider the following experiment, defined for $b = 0, 1$, where U, V, and X are elements in G and r_0 and r_1 are elements in Z_p.*

```
Experiment $\mathbf{Exp}^{\mathrm{cddh1}}_{\mathbb{G},b}(\mathcal{A}, U, V, X, r_0, r_1)$
    $(Y, s) \overset{R}{\leftarrow} \mathcal{A}(\mathsf{find}, U, V, X)$
    $b_0 \overset{R}{\leftarrow} \{0,1\}$ ; $b_1 = b \oplus b_0$
    $X_0 \leftarrow (X/U)^{r_{b_0}}$ ;  $K_0 \leftarrow \mathrm{CDH}(X/U, Y)^{r_{b_0}}$
    $X_1 \leftarrow (X/V)^{r_{b_1}}$ ;  $K_1 \leftarrow \mathrm{CDH}(X/V, Y)^{r_{b_1}}$
    $Y' \leftarrow Y^{r_0}$
    $d \leftarrow \mathcal{A}(\mathsf{guess}, s, X_0, K_0, X_1, K_1, Y')$
    return $d$
```

Now define the advantage of $\mathcal{A}$ in violating the chosen-basis decisional Diffie-Hellman 1 assumption with respect to (U, V, X, r_0, r_1), the advantage of $\mathcal{A}$, and the advantage function of the group, respectively, as follows:

$$\mathbf{Adv}^{\mathrm{cddh1}}_{\mathbb{G}}(\mathcal{A}, U, V, X, r_0, r_1) = 2 \cdot \Pr[\,\mathbf{Exp}^{\mathrm{cddh1}}_{\mathbb{G},b}(\mathcal{A}, U, V, X, r_0, r_1) = b\,] - 1$$

$$\mathbf{Adv}^{\mathrm{cddh1}}_{\mathbb{G}}(\mathcal{A}) = \mathbf{E}_{U,V,X,r_0,r_1}\left[\mathbf{Adv}^{\mathrm{cddh1}}_{\mathbb{G}}(\mathcal{A}, U, V, X, r_0, r_1)\right]$$

$$\mathbf{Adv}^{\mathrm{cddh1}}_{\mathbb{G}}(t) = \max_{\mathcal{A}}\{\,\mathbf{Adv}^{\mathrm{cddh1}}_{\mathbb{G}}(\mathcal{A})\,\},$$

where the maximum is over all $\mathcal{A}$ with time-complexity at most t. ◇

Definition 2 (CDDH2)**.** *Let $\mathbb{G} = (G, g, p)$ be a represented group and let $\mathcal{A}$ be an adversary. Consider the following experiment, defined for $b = 0, 1$, where U and V are elements in G and r_0 and r_1 are elements in Z_p.*

```
Experiment $\mathbf{Exp}^{\mathrm{cddh2}}_{\mathbb{G},b}(\mathcal{A}, U, V, r_0, r_1)$
    $(X, Y, s) \overset{R}{\leftarrow} \mathcal{A}(\mathsf{find}, U, V)$
    $b_0 \overset{R}{\leftarrow} \{0,1\}$ ; $b_1 = b \oplus b_0$
    $X_0 \leftarrow (X/U)^{r_{b_0}}$ ;  $X_1 \leftarrow (X/V)^{r_{b_1}}$ ;  $Y' \leftarrow Y^{r_0}$
    $d \leftarrow \mathcal{A}(\mathsf{guess}, s, X_0, X_1, Y')$
    return $d$
```

We define the advantage of $\mathcal{A}$ in violating the chosen-basis decisional Diffie-Hellman 2 assumption with respect to (U, V, r_0, r_1), $\mathbf{Adv}^{\mathrm{cddh2}}_{\mathbb{G},\mathcal{A},U,V,r_0,r_1}$, the advantage of $\mathcal{A}$, $\mathbf{Adv}^{\mathrm{cddh2}}_{\mathbb{G}}(\mathcal{A})$, and the advantage function of the group, $\mathbf{Adv}^{\mathrm{cddh2}}_{\mathbb{G}}(t)$, as in Definition 1. ♢

Password-based Chosen-basis Decisional Diffie-Hellman assumptions. The actual proof of security of our protocol uses password-related versions of the chosen-basis decisional Diffie-Hellman assumptions, which we call *password-based chosen-basis decisional Diffie-Hellman* assumptions 1 and 2.

Definition 3 (PCDDH1)**.** *Let $\mathbb{G} = (G, g, p)$ be a represented group and let $\mathcal{A}$ be an adversary. Consider the following experiment, defined for $b = 0, 1$, where $\mathcal{P}$ is a random function from $\{1, \ldots, n\}$ into G, X is an element in G, k is a password in $\{1, \ldots, n\}$, and r_0 and r_1 are elements in Z_p.*

```
Experiment $\mathbf{Exp}^{\mathrm{pcddh1}}_{\mathbb{G},n,b}(\mathcal{A}, \mathcal{P}, X, k, r_0, r_1)$
    $(Y, s) \xleftarrow{R} \mathcal{A}^{\mathcal{P}}(\mathsf{find}, X)$
    $U \leftarrow \mathcal{P}(k)$ ; $X' \leftarrow (X/U)^{r_b}$ ; $K \leftarrow \mathrm{CDH}(X/U, Y)^{r_b}$ ; $Y' \leftarrow Y^{r_0}$
    $d \leftarrow \mathcal{A}(\mathsf{guess}, s, X', Y', K, k)$
    return $d$
```

We define the advantage of $\mathcal{A}$ in violating the password-based chosen-basis decisional Diffie-Hellman 1 assumption with respect to $(\mathcal{P}, X, k, r_0, r_1)$, $\mathbf{Adv}^{\mathrm{pcddh1}}_{\mathbb{G},n}(\mathcal{A}, \mathcal{P}, X, k, r_0, r_1)$, the advantage of $\mathcal{A}$, $\mathbf{Adv}^{\mathrm{pcddh1}}_{\mathbb{G},n}(\mathcal{A}, \mathcal{P})$, and the advantage function of the group, $\mathbf{Adv}^{\mathrm{pcddh1}}_{\mathbb{G},n}(t, \mathcal{P})$, as in Definition 1. ♢

Definition 4 (PCDDH2)**.** *Let $\mathbb{G} = (G, g, p)$ be a represented group and let $\mathcal{A}$ be an adversary. Consider the following experiment, defined for $b = 0, 1$, where $\mathcal{P}$ is a random function from $\{1, \ldots, n\}$ into G, k is a password in $\{1, \ldots, n\}$, and r_0 and r_1 are elements in Z_p.*

```
Experiment $\mathbf{Exp}^{\mathrm{pcddh2}}_{\mathbb{G},n,b}(\mathcal{A}, \mathcal{P}, k, r_0, r_1)$
    $(X, Y, s) \xleftarrow{R} \mathcal{A}^{\mathcal{P}}(\mathsf{find})$
    $U \leftarrow \mathcal{P}(k)$ ; $X' \leftarrow (X/U)^{r_b}$ ; $Y' \leftarrow Y^{r_0}$
    $d \leftarrow \mathcal{A}^{\mathcal{P}}(\mathsf{guess}, s, X', Y', k)$
    return $d$
```

We define the advantage of $\mathcal{A}$ in violating the password-based chosen-basis decisional Diffie-Hellman 2 assumption with respect to $(\mathcal{P}, k, r_0, r_1)$, $\mathbf{Adv}^{\mathrm{pcddh2}}_{\mathbb{G},n}(\mathcal{A}, \mathcal{P}, k, r_0, r_1)$, the advantage of $\mathcal{A}$, $\mathbf{Adv}^{\mathrm{pcddh2}}_{\mathbb{G},n}(\mathcal{A}, \mathcal{P})$, and the advantage function of the group, $\mathbf{Adv}^{\mathrm{pcddh2}}_{\mathbb{G},n}(t, \mathcal{P})$, as in Definition 1. ♢

Relations between the PCDDH1 **and** CDDH1 **problems.** The following two lemmas, whose proofs can be found in the full version of this paper [2], present relations between the PCDDH1 and CDDH1 problems. The first result is meaningful for small n (polynomially bounded in the asymptotic framework). The second one considers larger dictionaries.

Lemma 1. *Let $\mathbb{G} = (G, g, p)$ be a represented group and let n be an integer. If there exists a distinguisher $\mathcal{A}$ such that $\mathbf{Adv}^{\mathrm{pcddh1}}_{\mathbb{G},n}(\mathcal{A}) \geq \frac{2}{n} + \epsilon$, then there exists a distinguisher $\mathcal{B}$ and a subset S of $G^3 \times \mathsf{Z}_p^2$ of probability greater than $\epsilon/8n^2$ such that for any $(U, V, X, r_0, r_1) \in S$, $\mathbf{Adv}^{\mathrm{cddh1}}_{\mathbb{G},n}(\mathcal{B}, U, V, X, r_0, r_1) \geq \frac{\epsilon^2}{8}$.*

Lemma 2. *Let $\mathbb{G} = (G, g, p)$ be a represented group and let n be an integer. If there exists a distinguisher $\mathcal{A}$ such that $\mathbf{Adv}^{\mathrm{pcddh1}}_{\mathbb{G},n}(\mathcal{A}) \geq \epsilon \geq \frac{16}{n}$, then there exists a distinguisher $\mathcal{B}$ and a subset S of $G^3 \times \mathsf{Z}_p^2$ of probability greater than $\epsilon^3/2^{10}$ such that for any $(U, V, X, r_0, r_1) \in S$, $\mathbf{Adv}^{\mathrm{cddh1}}_{\mathbb{G},n}(\mathcal{B}, U, V, X, r_0, r_1) \geq \frac{\epsilon^2}{8}$.*

Relations between the PCDDH2 **and** CDDH2 **problems.** The following two lemmas, whose proofs can be easily derived from the proofs of the previous two lemmas, present relations between the PCDDH2 and CDDH2 problems. While the first result is meaningful for small values of n, the second one considers larger values.

Lemma 3. *Let $\mathbb{G} = (G, g, p)$ be a represented group and let n be an integer. If there exists a distinguisher $\mathcal{A}$ such that $\mathbf{Adv}^{\mathrm{pcddh2}}_{\mathbb{G},n}(\mathcal{A}) \geq \frac{2}{n} + \epsilon$, then there exists a distinguisher $\mathcal{B}$ and a subset S of $G^2 \times \mathsf{Z}_p^2$ of probability greater than $\epsilon/8n^2$ such that for any $(U, V, r_0, r_1) \in S$ $\mathbf{Adv}^{\mathrm{cddh2}}_{\mathbb{G},n}(\mathcal{B}, U, V, r_0, r_1) \geq \frac{\epsilon^2}{8}$.*

Lemma 4. *Let $\mathbb{G} = (G, g, p)$ be a represented group and let n be an integer. If there exists a distinguisher $\mathcal{A}$ such that $\mathbf{Adv}^{\mathrm{pcddh1}}_{\mathbb{G},n}(\mathcal{A}) \geq \epsilon \geq \frac{16}{n}$, then there exists a distinguisher $\mathcal{B}$ and a subset S of $G^2 \times \mathsf{Z}_p^2$ of probability greater than $\epsilon^3/2^{10}$ such that for any $(U, V, r_0, r_1) \in S$ $\mathbf{Adv}^{\mathrm{cddh1}}_{\mathbb{G},n}(\mathcal{B}, U, V, r_0, r_1) \geq \frac{\epsilon^2}{8}$.*

Distinguishers. In all of the above relations, we show that if there exists an adversary against the password version of the chosen-basis decisional problem that is capable of doing better than just guessing the password, then we can construct a distinguisher for underlying chosen-basis decisional problem, whose success probability is non-negligible over a non-negligible subset of the probability space. Even though these results provide enough evidence of the hardness of breaking the original password-based problem, one may want a more concrete result that works for the most of the probability space. The next lemma, whose proof can be found in the full version of this paper [2], proves just that. More precisely, it shows that if a good distinguisher exists for a non-negligible portion of the probability space, then the same distinguisher is a good distinguisher either for the entire probability space or for at least half of it.

Lemma 5 (Amplification Lemma). *Let $E^b(x)$ be an experiment for $b \in \{0, 1\}$ and $x \in S$. Let D be a distinguisher between two experiments $E^0(x)$ and $E^1(x)$ with advantage ϵ for $x \in S'$, where $S' \subset S$ is of measure $\mu = |S'|/|S|$:*

$$\Pr_x[x \in S'] = \mu; \qquad \Pr_{b,x}[E^b(D, x) = b \mid x \in S'] \geq \frac{1}{2} + \frac{\epsilon}{2}.$$

Then either D is a good distinguisher on the whole set S:

$$\Pr_{b,x}[E^b(D,x)=b] \geq \frac{1}{2}+\frac{\mu\epsilon}{4},$$

or D is a good distinguisher for S' and $S\backslash S'$, one of which is a subset of measure greater than or equal to one half:

$$\begin{aligned} \Pr_x[x\in S'] &= \mu \qquad \Pr_{b,x}[E^b(D,x)=b \mid x\in S'] \geq \frac{1}{2}+\frac{\epsilon}{2}; \\ \Pr_x[x\in S\backslash S'] &= 1-\mu \qquad \Pr_{b,x}[E^b(D,x)=b \mid x\in S\backslash S'] \leq \frac{1}{2}-\frac{\mu\epsilon}{4}. \end{aligned}$$

Lower bounds. The following lemma, whose proof can be found in the full version of this paper [2], gives an upper bound on the advantage of any adversary for the CDDH1 or CDDH2 problem in the generic model of Shoup [23]. From it, it follows that any generic algorithm capable of solving the CDDH1 or CDDH2 problem with success probability bounded away from 1/2 has to perform at least $\Omega(p^{1/2})$ group operations. Please refer to [23] for a description of the model.

Lemma 6. *Let p be a prime number and let σ represent a random injective mapping of Z_p into a set S of bit strings of cardinality at least p. Let $\mathcal{A}$ be a distinguisher for the* CDDH1 *or the* CDDH2 *problem in the generic setting making at most m queries to the group oracle associated with σ. Then, the advantage of $\mathcal{A}$ is at most $O(m^2/p)$.*

4 Our 3-party Password-based Protocol

In this section, we introduce our new protocol, a non-concurrent 3-party password-based authenticated key exchange protocol called 3PAKE, whose security proof is in the random oracle model. It assumes that the clients willing to establish a common secret session key share passwords with a common server. Even though the proof of security assumes a non-concurrent model, we outline in Section 4 ways in which one can modify our protocol to make it concurrent.

Description. Our 3-party password-based protocol, 3PAKE, is based the on password-based key exchange protocols in [6, 9, 21], which in turn are based on the encrypted key exchange of Bellovin and Merritt [7]. The description of 3PAKE is given in Figure 1, where (G, g, p) is the represented group; ℓ_r and ℓ_k are security parameters; and $G_1 : \mathcal{D}\to G$, $G_2 : \{0,1\}^{\ell_r} \times \mathcal{D} \times G \to G$, and $H : \mathcal{U}^3 \times \{0,1\}^{\ell_r} \times G^4 \to \{0,1\}^{\ell_k}$ are random oracles.

The protocol consists of two rounds of message. First, each client chooses an ephemeral public key by choosing a random element in Z_p and raising g to the that power, encrypts it using the output of the hash function G_1 with his password as the input, and sends it to the server. Upon receiving a message from each client, the server decrypts these messages to recover each client's ephemeral public key, chooses a random index $r \in \mathsf{Z}_p$ and a random element

$R \in \{0,1\}^{\ell_r}$, exponentiates each of the ephemeral public keys to the r-th power, and re-encrypts them using the output of the hash function G_2, with R and the appropriate first-round message and password as input.

In the second round of messages, the server sends to each client the encrypted value of the randomized ephemeral public key of their partner along with the messages that the server exchanged with that partner, which are omitted in Figure 1 for clarity. Upon receiving a message from the server, each client recovers the randomized ephemeral public key of his partner, computes the Diffie-Hellman key K, and the session key SK via a hash function H using as input K and the transcript of the conversation among the clients and the server. The session identification is defined to be the transcript $T = (R, X^\star, Y^\star, \overline{X}^\star, \overline{Y}^\star)$ of the conversation among the server and clients, along with their identity strings.

Public information: $G, g, p, \ell_r, \ell_k, G_1, G_2, H$

Client A		Server S		Client B
$pw_A \in \mathcal{D}$		$pw_A, pw_B \in \mathcal{D}$		$pw_B \in \mathcal{D}$
$x \overset{R}{\leftarrow} \mathsf{Z}_p$; $X \leftarrow g^x$ $\mathsf{pw}_{A,1} \leftarrow G_1(pw_A)$ $X^\star \leftarrow X \cdot \mathsf{pw}_{A,1}$		$r \overset{R}{\leftarrow} \mathsf{Z}_p$; $R \overset{R}{\leftarrow} \{0,1\}^{\ell_r}$		$y \overset{R}{\leftarrow} \mathsf{Z}_p$; $Y \leftarrow g^y$ $\mathsf{pw}_{B,1} \leftarrow G_1(pw_B)$ $Y^\star \leftarrow Y \cdot \mathsf{pw}_{B,1}$
	$\xrightarrow{X^\star}$		$\xleftarrow{Y^\star}$	
		$\mathsf{pw}_{A,1} \leftarrow G_1(pw_A)$ $\mathsf{pw}_{B,1} \leftarrow G_1(pw_B)$ $X \leftarrow X^\star / \mathsf{pw}_{A,1}$ $Y \leftarrow Y^\star / \mathsf{pw}_{B,1}$ $\overline{X} \leftarrow X^r$ $\overline{Y} \leftarrow Y^r$ $\mathsf{pw}_{A,2} \leftarrow G_2(R, pw_A, X^\star)$ $\mathsf{pw}_{B,2} \leftarrow G_2(R, pw_B, Y^\star)$ $\overline{Y}^\star \leftarrow \overline{Y} \cdot \mathsf{pw}_{A,2}$ $\overline{X}^\star \leftarrow \overline{X} \cdot \mathsf{pw}_{B,2}$		
	$\xleftarrow{R, \overline{Y}^\star}$		$\xrightarrow{R, \overline{X}^\star}$	
$\mathsf{pw}_{A,2} \leftarrow G_2(R, pw_A, X^\star)$ $\overline{Y} \leftarrow \overline{Y}^\star / \mathsf{pw}_{A,2}$; $K \leftarrow \overline{Y}^x$ $T \leftarrow R, X^\star, Y^\star, \overline{X}^\star, \overline{Y}^\star$ $SK \leftarrow H(A, B, S, T, K)$				$\mathsf{pw}_{B,2} \leftarrow G_2(R, pw_B, Y^\star)$ $\overline{X} \leftarrow \overline{X}^\star / \mathsf{pw}_{B,2}$; $K \leftarrow \overline{X}^y$ $T \leftarrow R, X^\star, Y^\star, \overline{X}^\star, \overline{Y}^\star$ $SK \leftarrow H(A, B, S, T, K)$

Fig. 1. 3PAKE: A provably-secure 3-party password-based authenticated key exchange protocol.

Correctness. In an honest execution of the protocol in Figure 1, we have $\overline{Y} = Y^r = g^{yr}$ and $\overline{X} = X^r = g^{xr}$. Hence, $K = \overline{Y}^x = \overline{X}^y = g^{xyr}$.

Efficiency. 3PAKE is quite efficient, not requiring much computational power from the server. Note that the amount of computation performed by the server in this case is comparable to that of each user. That is at least half the amount of computation that it would be required if the server were to perform a separate 2-party password-based encrypted key exchange with each user.

Rationale for the scheme. As pointed out in the introduction, the random value r is used by the server to hide the password of one user with respect to other users. For this same reason, it is also crucial that the server rejects any value $X^\star$ or $Y^\star$ whose underlying value X or Y is equal to 1. This is omitted in Figure 1 for clarity reasons only.

The reason for using two different masks $\mathsf{pw}_{A,1}$ and $\mathsf{pw}_{A,2}$ in each session, on the other hand, is a little more intricate and is related to our proof technique. More precisely, in our proof of security, we embed instances of the CDDH1 and CDDH2 problems in $\mathsf{pw}_{A,1}$ and $\mathsf{pw}_{A,2}$ and we hope to get an answer for these problems from the list of queries that the adversary makes to the G_1 and G_2 oracles. Unfortunately, this does not appear to be possible when the values of $\mathsf{pw}_{A,1}$ and $\mathsf{pw}_{A,2}$ are fixed for all sessions since a powerful adversary could be able to learn the values of $\mathsf{pw}_{A,1}$ and $\mathsf{pw}_{A,2}$ and break the semantic security of the scheme without querying the oracles for G_1 and G_2.

To see how, let us assume two fixed but random values for $\mathsf{pw}_{A,1}$ and $\mathsf{pw}_{A,2}$ and that we are dealing with an adversary that knows the password of a legitimate but malicious user. Let us also assume that the adversary is capable of breaking the computational Diffie-Hellman inversion (CDHI) problem, in which the goal is to compute g^y from g, g^x, and g^{xy}. Since in the security model, the adversary is allowed to intercept and replay messages, he can play the role of the partner of A and ask a given query $(A, g^x \cdot \mathsf{pw}_{A,1})$ twice to the server. From the answers to these queries, the adversary would be able to compute two sets of values $(g^x \cdot \mathsf{pw}_{A,1}, g^y, g^{xr}, g^{yr} \cdot \mathsf{pw}_{A,2})$ and $(g^x \cdot \mathsf{pw}_{A,1}, g^y, g^{xr'}, g^{yr'} \cdot \mathsf{pw}_{A,2})$ based on different values r and r'. By dividing the last two terms of each set, the adversary can compute $g^{(r'-r)x}$ and $g^{(r'-r)y}$. Moreover, since the adversary plays the role of the partner of A and knows y, he can also compute $g^{r'-r}$. Hence, the adversary can learn the values of g, $g^{r'-r}$, and $g^{(r'-r)x}$ as well as $g^x \cdot \mathsf{pw}_{A,1}$. By solving the CDHI problem, he can also learn the value of g^x from g, $g^{r'-r}$, and $g^{(r'-r)x}$. Thus, he can recover $\mathsf{pw}_{A,1}$ without querying the oracle G_1 on various inputs pw. Moreover, since such adversary is capable of computing g^r from g, g^x, and g^{rx}, and hence capable of computing g^{ry}, he can also learn the value of $\mathsf{pw}_{A,2}$ without querying the oracle G_2.

Security. As the following theorem states, 3PAKE is a secure non-concurrent 3-party password-based key exchange protocol as long as the CDH, DDH, PCDDH1, and PCDDH2 problems are hard in $\mathbb{G}$. As shown in Section 3, the latter two problems are hard as long as CDDH1 and CDDH2 are hard in $\mathbb{G}$. Please note that the proof of security assumes $\mathcal{D}$ to be a uniformly distributed dictionary.

Theorem 1. *Let $\mathbb{G} = (G, g, p)$ be a represent group of prime order p and let $\mathcal{D}$ be a uniformly distributed dictionary of size $|\mathcal{D}|$. Let* 3PAKE *describe the encrypted key exchange protocol associated with these primitives as defined in Figure 1. Then, for any numbers t, $q_{\rm server}$, $q_{\rm start}$, $q_{\rm exe}$, q_{G_1}, q_{G_2}, and q_H,*

$$\mathbf{Adv}^{\rm ake}_{\mathsf{3PAKE},\mathbb{G},\mathcal{D}}(t, q_{\rm server}, q_{\rm start}, q_{\rm exe}, q_{G_1}, q_{G_2}, q_H) \;\le$$

$$\frac{2\; q_{\rm start}}{|\mathcal{D}|} + \frac{q_{G_1}^2 + q_{G_2}^2 + (q_{\rm exe} + q_{\rm start})^2}{p} + 4\; q_{\rm exe}\; \mathbf{Adv}^{\rm ddh}_{\mathbb{G}}(t) +$$
$$2 \cdot q_{\rm server} \cdot \max\{\, 2 \cdot \mathbf{Adv}^{\rm pcddh1}_{\mathbb{G},|\mathcal{D}|}(q_{\rm start} \cdot t)\;,\; \mathbf{Adv}^{\rm pcddh2}_{\mathbb{G},|\mathcal{D}|}(t)\,\} +$$
$$2\; q_{G_1}^2\; q_{G_2}^2\; q_H^2\; \mathbf{Adv}^{\rm cdh}_{\mathbb{G}}(t + 3\tau_e) + 2\; \frac{q_{G_1} + q_{G_2}}{p} + 4\; \frac{q_H}{p}\;,$$

where q_H, q_{G_1}, and q_{G_2} represent the number of queries to the H, G_1 and G_2 oracles, respectively; q_{exe} represents the number of queries to the Execute oracle; q_{start} represents the number of queries to the SendClient oracle used to initiate an client oracle instance; q_{server} represents the number of queries to the SendServer oracle; and τ_e denotes the exponentiation computational time in $\mathbb{G}$.

Proof idea. Here we only present a brief sketch of the proof. We refer the reader to the full version of this paper [2] for the full proof of security. The proof for 3PAKE defines a sequence of hybrid experiments, starting with the real attack and ending in an experiment in which the adversary has no advantage. Each experiment addresses a different security aspect.

Experiments 1 through 5 show that the adversary gains no information from passive attacks. They do so by showing that keys generated in these sessions can be safely replaced by random ones as long as the DDH assumption holds in $\mathbb{G}$.

In Experiment 6, we change the simulation of the random oracle H in all those situations for which the adversary may ask a valid test query. Such a change implies that the output of the test query is random and hence the advantage of the adversary in this case is 0. However, the difference between this experiment and previous still cannot be computed since it depends on the event AskH that the adversary asks certain queries to the random oracle H. Our goal at this point shifts to computing the probability of the event AskH.

In experiments 7 through 9, we deal with active attacks against the server. First, in Experiment 7, we show that the output values $\overline{X}^{\star}$ and $\overline{Y}^{\star}$ associated with honest users can be computed using random values and independently of each other as long as the PCDDH1 and PCDDH2 assumptions hold in $\mathbb{G}$. More precisely, we show how to upper-bound the difference in the probability of the event AskH using the PCDDH1 and PCDDH2 assumptions. Then, in the next two experiments, we show that for those cases in which we replaced $\overline{X}^{\star}$ and $\overline{Y}^{\star}$ with random values, the password is no longer used and that the Diffie-Hellman keys K used to compute the session keys for these users are indistinguishable from random.

Finally, in Experiment 10, we consider active attacks against a user. More precisely, we show that we can answer all *SendClient* queries with respect to honest users using random values, without using the password of these users, and without changing the probability of the event AskH. Moreover, at this moment, we also show how to bound the probability of the event AskH based on the hardness of the CDH problem in $\mathbb{G}$ and on the probability that the adversary successfully guesses the password of an honest user during an active attack against that user.

Concluding remarks. First, the main reason for assuming an underlying group G of prime order p is to ensure that the exponentiation of an element in the group other than the unit yields a generator. For the same reason, it is crucial for the server to check whether the elements to which it applies the randomization step are different from the unit element. Both these assumptions are implicitly made in several parts of the proof and they are essential for the security of our protocol.

Second, the proof of security for 3PAKE assumes a non-concurrent model, in which only one instance of each player can exist at a time. One can argue that such proof is not worth much as it rules out most interesting attack scenarios or makes the scheme too restrictive to be used in practice. To address the first of these concerns, we argue that, even though the non-concurrent scenario rules out a significant class of attacks, it still allows many interesting ones. For example, the identity-misbinding attacks in [12, 18] still work in the non-concurrent scenario. To address the second concern, we point out that several applications found in practice do not require concurrency. And even when they do require concurrent sessions, it is usually between different pairs of users. A simple modification is enough to make our protocol work in the latter case, by including the users' identification in the input of the G_1 and G_2 hash functions.

Third, if full concurrency is required, then one could modify 3PAKE to make it work in this new scenario by adding two extra flows at the beginning of the protocol going from the server to each of the two users. Such flows would include nonces in the input of the G_1 and G_2 hash functions. Each user would also have to add its own nonce to the input of the G_1 and G_2 hash functions, and send it to the server along with $X^\star$ or $Y^\star$. Moreover, the protocol's efficiency would remain almost the same, except for the number of rounds, but would still be significantly better than the round complexity of the generic construction in [1].

Finally, some of the problems that we found in our proof may be avoidable in the "ideal-cipher model," in which the encryption function is considered to be a truly random permutation. The reason for that is that non-linear properties of the ideal cipher model naturally remove the algebraic properties existent in the "one-time pad" version of the encryption function. Nonetheless, we opted to rely only on a single idealized model, the random oracle model, which is already a strong assumption as other papers have shown (e.g., [11]).

Acknowledgements

This work has been supported in part by the European Commission through the IST Program under Contract IST-2002-507932 ECRYPT.

References

1. M. Abdalla, P.-A. Fouque, and D. Pointcheval. Password-based authenticated key exchange in the three-party setting. In *PKC 2005*, Springer-Verlag LNCS, 2005.
2. M. Abdalla and D. Pointcheval. Interactive Diffie-Hellman assumptions with applications to password-based authentication. Full version of current paper. Available from Authors's web pages.
3. M. Bellare, D. Pointcheval, and P. Rogaway. Authenticated key exchange secure against dictionary attacks. In *EUROCRYPT 2000*, Springer-Verlag LNCS 1807, 2000.
4. M. Bellare and P. Rogaway. Entity authentication and key distribution. In *CRYPTO'93*, Springer-Verlag LNCS 773, 1994.

5. M. Bellare and P. Rogaway. Provably secure session key distribution — the three party case. In *28th ACM STOC*. ACM Press, 1996.
6. M. Bellare and P. Rogaway. The AuthA protocol for password-based authenticated key exchange. Contributions to IEEE P1363, 2000.
7. S. M. Bellovin and M. Merritt. Encrypted key exchange: Password-based protocols secure against dictionary attacks. In *1992 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, 1992.
8. V. Boyko, P. MacKenzie, and S. Patel. Provably secure password-authenticated key exchange using Diffie-Hellman. In *EUROCRYPT 2000*, Springer-Verlag LNCS 1807, 2000.
9. E. Bresson, O. Chevassut, and D. Pointcheval. New security results on encrypted key exchange. In *PKC 2004*, Springer-Verlag LNCS 2947, 2004.
10. J. W. Byun, I. R. Jeong, D. H. Lee, and C.-S. Park. Password-authenticated key exchange between clients with different passwords. In *ICICS 02*, Springer-Verlag LNCS 2513, 2002.
11. R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. In *30th ACM STOC*. ACM Press, 1998.
12. R. Canetti and H. Krawczyk. Security analysis of IKE's signature-based key-exchange protocol. In *CRYPTO 2002*, Springer-Verlag LNCS 2442, 2002.
13. W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22:644–654, 1978.
14. R. Gennaro and Y. Lindell. A framework for password-based authenticated key exchange. In *EUROCRYPT 2003*, Springer-Verlag LNCS 2656, 2003.
15. O. Goldreich and Y. Lindell. Session-key generation using human passwords only. In *CRYPTO 2001*, Springer-Verlag LNCS 2139, 2001.
16. L. Gong. Optimal authentication protocols resistant to password guessing attacks. In *CSFW'95*, pages 24–29, 1995. IEEE Computer Society.
17. S. Halevi and H. Krawczyk. Public-key cryptography and password protocols. In *ACM Transactions on Information and System Security*, pages 524–543. 1999.
18. H. Krawczyk. SIGMA: The "SIGn-and-MAc" approach to authenticated Diffie-Hellman and its use in the IKE protocols. In *CRYPTO 2003*, Springer-Verlag LNCS 2729, 2003.
19. C.-L. Lin, H.-M. Sun, and T. Hwang. Three-party encrypted key exchange: Attacks and a solution. *ACM SIGOPS Operating Systems Review*, 34(4):12–20, Oct. 2000.
20. P. MacKenzie, S. Patel, and R. Swaminathan. Password-authenticated key exchange based on RSA. In *ASIACRYPT 2000*, Springer-Verlag LNCS 1976, 2000.
21. P. D. MacKenzie. The PAK suite: Protocols for password-authenticated key exchange. Contributions to IEEE P1363.2, 2002.
22. R. Needham and M. Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21(21):993–999, Dec. 1978.
23. V. Shoup. Lower bounds for discrete logarithms and related problems. In *EUROCRYPT'97*, Springer-Verlag LNCS 1233, 1997.
24. M. Steiner, G. Tsudik, and M. Waidner. Refinement and extension of encrypted key exchange. *ACM SIGOPS Operating Systems Review*, 29(3):22–30, July 1995.
25. S. Wang, J. Wang, and M. Xu. Weaknesses of a password-authenticated key exchange protocol between clients with different passwords. In *ACNS 04*, Springer-Verlag LNCS 3089, 2004.
26. H.-T. Yeh, H.-M. Sun, and T. Hwang. Efficient three-party authentication and key agreement protocols resistant to password guessing attacks. *Journal of Information Science and Engineering*, 19(6):1059–1070, Nov. 2003.