The extended abstract of this work appears in Andrew Patrick and Moti Yung, editors, *Financial Cryptography and Data Security – FC 2005*, Lectures Notes in Computer Science Vol. 3570, pages 341–356, Roseau, The Commonwealth Of Dominica, Feb. 28 – Mar. 3, 2005. Springer-Verlag, Berlin, Germany.

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

Michel Abdalla 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

The area of password-based authenticated key exchange protocols has been the subject of a vast amount of work in the last few years due to its practical aspects. In these protocols, the goal is to enable users communicating over an unreliable channel to establish a secure session key even when the secret key that they share is drawn from a small set of values. Despite the attention given to it, it was only recently that this problem has been formally addressed in the three-party setting. In this setting, the users trying to establish a secret session key are only required to share a password with a trusted server and not directly among themselves. In this paper, we introduce a new three-party password-based authenticated key exchange protocol based on the two-party encrypted key exchange of Bellovin and Merritt. Our protocol is reasonably efficient and has a per-user computational cost that is comparable to that of the underlying two-party encrypted key exchange. 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, authenticated key exchange, Diffie-Hellman assumptions, multi-party protocols.

Contents

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 [2] 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 [2], 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 [**?**] using the MAC keys obtained in the previous phase.

EFFICIENT 3-PARTY PASSWORD-BASED PROTOCOLS. Though attractive and natural, the construction given in [2] 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 [8].

In order to understand our construction, let us first recall the example given in [2] of an insecure 3-party password-based key exchange protocol, which we reproduce in Figure 1. As noted in [2], this protocol is not secure because it allows one user in the system to perform an off-line dictionary attack against other users.

RE-ENCRYPTION WITH RANDOMIZATION. The main problem with the protocol in Figure 1 resides in

Public information: $G, g, p, \mathcal{E}, \mathcal{D}, H$

$$\begin{array}{ccccc}
\text{Client A} & & \text{Server} & & \text{Client B}\\
pw_A \in \mathcal{D} & & pw_A, pw_B \in \mathcal{D} & & pw_B \in \mathcal{D}\\
 & & & & \\
x \overset{R}{\leftarrow} \mathsf{Z}_p \;;\; X_A \leftarrow g^x & & & & y \overset{R}{\leftarrow} \mathsf{Z}_p \;;\; Y_B \leftarrow g^y\\
X_A^\star \leftarrow \mathcal{E}_{pw_A}(X_A) & & & & Y_B^\star \leftarrow \mathcal{E}_{pw_B}(Y_B)\\
 & \xrightarrow{X_A^\star} & & \xleftarrow{Y_B^\star} & \\
 & & X_S \leftarrow \mathcal{D}_{pw_A}(X_A^\star) & & \\
 & & Y_S \leftarrow \mathcal{D}_{pw_B}(Y_B^\star) & & \\
 & & Y_S^\star \leftarrow \mathcal{E}_{pw_A}(Y_S) & & \\
 & & X_S^\star \leftarrow \mathcal{E}_{pw_B}(X_S) & & \\
 & \xleftarrow{Y_S^\star} & & \xrightarrow{X_S^\star} & \\
Y_A \leftarrow \mathcal{D}_{pw_A}(Y_S^\star) & & & & X_B \leftarrow \mathcal{D}_{pw_B}(X_S^\star)\\
K_A \leftarrow Y_A^x & & & & K_B \leftarrow X_B^y\\
SK_A \leftarrow H(A \,\|\, B \,\|\, S \,\|\, K_A) & & & & SK_B \leftarrow H(A \,\|\, B \,\|\, S \,\|\, K_B)
\end{array}$$

Figure 1: An example of an insecure 3-party password-based encrypted key exchange protocol [2].

the fact that the same value is encrypted twice, once using pw_A and once using pw_B, thus allowing both users A and B to learn each other's password via an off-line dictionary attack. To overcome this problem, it is crucial that the server randomizes the value received from one participant before re-encrypting it using the password of the other participant.

Starting from this idea, we can design a provably-secure protocol, based on the encrypted key exchange of Bellovin and Merritt [8]. The new protocol, whose *simplified* description is given in Figure 2, 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 [2]. 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. [1]

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., [4, 9, 10, 15, 16, 18, **?**]), only a few take into account the 3-party scenario (e.g., [2, 11, 17, 20,

[1] This is no longer true, since in [25], Szydlo presents two simple and very efficient attacks against the two versions of the chosen-basis decisional Diffie-Hellman problem being introduced in this paper. As a result, the chosen-basis decisional Diffie-Hellman assumptions must no longer be considered to be valid assumptions. It is also important to point out that, in previous versions of this paper, lower bound proofs in the generic model for the chosen-basis decisional Diffie-Hellman assumptions were also presented. Unfortunately, those proofs contained mistakes (which were exploited in the attack by Szydlo) and are no longer included in the current version of the paper. More details will be included in future versions of this paper.

Public information: $G, g, p, \mathcal{E}, \mathcal{D}, H$

$$\begin{array}{ccccc}
\text{Client } A & & \text{Server } S & & \text{Client } B \\
pw_A \in \mathcal{D} & & pw_A, pw_B \in \mathcal{D} & & pw_B \in \mathcal{D} \\
 & & & & \\
x \overset{R}{\leftarrow} \mathsf{Z}_p \; ; \; X \leftarrow g^x & & r \overset{R}{\leftarrow} \mathsf{Z}_p & & y \overset{R}{\leftarrow} \mathsf{Z}_p \; ; \; Y \leftarrow g^y \\
X^\star \leftarrow \mathcal{E}_{pw_A}(X) & & & & Y^\star \leftarrow \mathcal{E}_{pw_B}(Y) \\
 & \xrightarrow{A, B, X^\star} & & \xleftarrow{B, A, Y^\star} & \\
 & & X \leftarrow \mathcal{D}_{pw_A}(X^\star) & & \\
 & & Y \leftarrow \mathcal{D}_{pw_B}(Y^\star) & & \\
 & & \overline{X} \leftarrow X^r & & \\
 & & \overline{Y} \leftarrow Y^r & & \\
 & & \overline{Y}^\star \leftarrow \mathcal{E}_{pw_A}(\overline{Y}) & & \\
 & & \overline{X}^\star \leftarrow \mathcal{E}_{pw_B}(\overline{X}) & & \\
 & \xleftarrow{S, B, \overline{Y}^\star} & & \xrightarrow{S, A, \overline{X}^\star} & \\
\overline{Y} \leftarrow \mathcal{D}_{pw_A}(\overline{Y}^\star) & & & & \overline{X} \leftarrow \mathcal{D}_{pw_B}(\overline{X}^\star) \\
K \leftarrow \overline{Y}^x & & & & K \leftarrow \overline{X}^y \\
SK \leftarrow H(\text{Transcript} \,\|\, K) & & & & SK \leftarrow H(\text{Transcript} \,\|\, K)
\end{array}$$

Figure 2: An efficient 3-party password-based encrypted key exchange protocol.

24, 26, 27]). Moreover, to the best of our knowledge, with the exception of the generic construction in [2], 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 [6], 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 [2].

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 [7, 21, 10]. 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 [2]. 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 [1].

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 4.3. We conclude our paper by presenting detailed security proofs for 3PAKE and for the several lemmas described in the paper, respectively, in Appendix A and Appendix B.

2 Definitions

We now recall the formal security model for 3-party password-authenticated key exchange protocols introduced in [2], which in turn builds upon those of Bellare and Rogaway [5, 6] and that of Bellare, Pointcheval, and Rogaway [4]. In doing so, we omit the definitions for forward security as the latter is out of the scope of the present paper.

2.1 Communication model

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 [2], 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 $b = 1$.

2.2 Security definitions

Notation. Following [2], which in turn follows [5, 6], 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:

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

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 [1]. 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 $\mathbf{Adv}^{\mathrm{ake}}_{P,\mathcal{D}}$ 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.

3.1 Definitions

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 $\mathbf{Exp}^{\mathrm{cdh}}_{\mathbb{G}}(\mathcal{A})$, 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 $\mathbf{Exp}^{\mathrm{cdh}}_{\mathbb{G}}(\mathcal{A})$ outputs 1 if $Z = g^{uv}$ and 0 otherwise. We define the *advantage* of $\mathcal{A}$ in violating the CDH assumption as $\mathbf{Adv}^{\mathrm{cdh}}_{\mathbb{G}}(\mathcal{A}) = \Pr[\,\mathbf{Exp}^{\mathrm{cdh}}_{\mathbb{G}}(\mathcal{A}) = 1\,]$ and the *advantage function* of the group, $\mathbf{Adv}^{\mathrm{cdh}}_{\mathbb{G}}(t)$, as the maximum value of $\mathbf{Adv}^{\mathrm{cdh}}_{\mathbb{G}}(\mathcal{A})$ 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, $\mathbf{Exp}^{\mathrm{ddh\text{-}real}}_{\mathbb{G}}(\mathcal{A})$ and $\mathbf{Exp}^{\mathrm{ddh\text{-}rand}}_{\mathbb{G}}(\mathcal{A})$. 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 $\mathbf{Exp}^{\mathrm{ddh\text{-}real}}_{\mathbb{G}}(\mathcal{A})$ and g^z for a random z in $\mathbf{Exp}^{\mathrm{ddh\text{-}rand}}_{\mathbb{G}}(\mathcal{A})$. 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, $\mathbf{Adv}^{\mathrm{ddh}}_{\mathbb{G}}(\mathcal{A})$, as $\Pr[\,\mathbf{Exp}^{\mathrm{ddh\text{-}real}}_{\mathbb{G}}(\mathcal{A}) = 1\,] - \Pr[\,\mathbf{Exp}^{\mathrm{ddh\text{-}rand}}_{\mathbb{G}}(\mathcal{A}) = 1\,]$. The *advantage function* of the group, $\mathbf{Adv}^{\mathrm{ddh}}_{\mathbb{G}}(t)$, 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 3.1 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 3.1 [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:

$$\begin{aligned}
\mathbf{Adv}^{\mathrm{cddh1}}_{\mathbb{G}}(\mathcal{A}, U, V, X, r_0, r_1) &= 2 \cdot \Pr[\,\mathbf{Exp}^{\mathrm{cddh1}}_{\mathbb{G},b}(\mathcal{A}, U, V, X, r_0, r_1) = b\,] - 1 \\
\mathbf{Adv}^{\mathrm{cddh1}}_{\mathbb{G}}(\mathcal{A}) &= \mathbf{E}_{U,V,X,r_0,r_1}\left[\mathbf{Adv}^{\mathrm{cddh1}}_{\mathbb{G}}(\mathcal{A}, U, V, X, r_0, r_1)\right] \\
\mathbf{Adv}^{\mathrm{cddh1}}_{\mathbb{G}}(t) &= \max_{\mathcal{A}}\{\,\mathbf{Adv}^{\mathrm{cddh1}}_{\mathbb{G}}(\mathcal{A})\,\},
\end{aligned}$$

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

Definition 3.2 [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 $d$
```

Now 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), the *advantage* of $\mathcal{A}$, and the *advantage function* of the group, respectively, as follows:

$$\begin{aligned}
\mathbf{Adv}^{\mathrm{cddh2}}_{\mathbb{G},\mathcal{A},U,V,r_0,r_1} &= 2 \cdot \Pr[\,\mathbf{Exp}^{\mathrm{cddh2}}_{\mathbb{G},b}(\mathcal{A}, U, V, r_0, r_1) = b\,] - 1 \\
\mathbf{Adv}^{\mathrm{cddh2}}_{\mathbb{G}}(\mathcal{A}) &= \mathbf{E}_{U,V,r_0,r_1}\left[\mathbf{Adv}^{\mathrm{cddh2}}_{\mathbb{G}}(\mathcal{A}, U, V, r_0, r_1)\right] \\
\mathbf{Adv}^{\mathrm{cddh2}}_{\mathbb{G}}(t) &= \max_{\mathcal{A}}\{\,\mathbf{Adv}^{\mathrm{cddh2}}_{\mathbb{G}}(\mathcal{A})\,\},
\end{aligned}$$

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

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

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

$$\begin{array}{rcl}
\mathbf{Adv}^{\mathrm{pcddh1}}_{\mathbb{G},n}(\mathcal{A}, \mathcal{P}, X, k, r_0, r_1) & = & 2 \cdot \Pr[\,\mathbf{Exp}^{\mathrm{pcddh1}}_{\mathbb{G},n,b}(\mathcal{A}, \mathcal{P}, X, k, r_0, r_1) = b\,] - 1 \\
\mathbf{Adv}^{\mathrm{pcddh1}}_{\mathbb{G},n}(\mathcal{A}, \mathcal{P}) & = & \mathbf{E}_{X,k,r_0,r_1}\left[\mathbf{Adv}^{\mathrm{pcddh1}}_{\mathbb{G},n}(\mathcal{A}, \mathcal{P}, X, k, r_0, r_1)\right] \\
\mathbf{Adv}^{\mathrm{pcddh1}}_{\mathbb{G},n}(t, \mathcal{P}) & = & \max_{\mathcal{A}}\{\, \mathbf{Adv}^{\mathrm{pcddh1}}_{\mathbb{G},n}(\mathcal{A}, \mathcal{P})\,\},
\end{array}$$

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

Definition 3.4 [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 $\mathbf{Exp}^{\mathrm{pcddh2}}_{\mathbb{G},n,b}(\mathcal{A}, \mathcal{P}, k, r_0, r_1)$
    $(X, Y, s) \stackrel{R}{\leftarrow} \mathcal{A}^{\mathcal{P}}(\mathsf{find})$
    $U \leftarrow \mathcal{P}(k)$
    $X' \leftarrow (X/U)^{r_b}$
    $Y' \leftarrow Y^{r_0}$
    $d \leftarrow \mathcal{A}^{\mathcal{P}}(\mathsf{guess}, s, X', Y', k)$
    return $d$
```

Now define the *advantage* of $\mathcal{A}$ in violating the password-based chosen-basis decisional Diffie-Hellman 2 assumption with respect to $(\mathcal{P}, k, r_0, r_1)$, the *advantage* of $\mathcal{A}$, and the *advantage function* of the group, respectively, as follows:

$$\begin{array}{rcl}
\mathbf{Adv}^{\mathrm{pcddh2}}_{\mathbb{G},n}(\mathcal{A}, \mathcal{P}, k, r_0, r_1) & = & 2 \cdot \Pr[\,\mathbf{Exp}^{\mathrm{pcddh2}}_{\mathbb{G},n,b}(\mathcal{A}, \mathcal{P}, k, r_0, r_1) = b\,] - 1 \\
\mathbf{Adv}^{\mathrm{pcddh2}}_{\mathbb{G},n}(\mathcal{A}, \mathcal{P}) & = & \mathbf{E}_{k,r_0,r_1}\left[\mathbf{Adv}^{\mathrm{pcddh2}}_{\mathbb{G},n}(\mathcal{A}, \mathcal{P}, k, r_0, r_1)\right] \\
\mathbf{Adv}^{\mathrm{pcddh2}}_{\mathbb{G},n}(t, \mathcal{P}) & = & \max_{\mathcal{A}}\{\, \mathbf{Adv}^{\mathrm{pcddh2}}_{\mathbb{G},n}(\mathcal{A}, \mathcal{P})\,\},
\end{array}$$

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

3.2 Relations

Relations between the PCDDH1 and CDDH1 problems. The following two lemmas, whose proofs are in Appendix B, 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 3.5 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 3.6 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 3.7 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 3.8 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 is also in Appendix B, 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 3.9 [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:

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

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 4.3 ways in which one can modify our protocol to make it concurrent.

4.1 Description

Our 3-party password-based protocol, 3PAKE, is based the on password-based key exchange protocols in [7, 10, 21], which in turn are based on the encrypted key exchange of Bellovin and Merritt [8]. The description of 3PAKE is given in Figure 3, where (G, g, p) is the represented group; ℓ_r and ℓ_k are security parameters; and $G_1 : \mathcal{U}^2 \times \mathcal{D} \rightarrow G$, $G_2 : \mathcal{U}^2 \times \{0,1\}^{\ell_r} \times \mathcal{D} \times G \rightarrow G$, and $H : \mathcal{U}^3 \times \{0,1\}^{\ell_r} \times G^4 \rightarrow \{0,1\}^{\ell_k}$ are random oracles.[2]

[2] In previous versions of this paper as well as in the extended abstract [3], there was a discrepancy between the scheme description being presented in the main body of the paper and the one being used in the proof of security in the appendix. More precisely, in the scheme description in the main body of the paper, the identities of the clients were incorrectly omitted from the input of the hash functions G_1 and G_2. As shown in [13], such an omission can lead to an attack when static corruptions of players are allowed in the model. In this new version, this discrepancy has been fixed.

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, his own identity, and the identity of his intended partner 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 3 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
$pw_A \in \mathcal{D}$		$pw_A, pw_B \in \mathcal{D}$		$pw_B \in \mathcal{D}$
$x \overset{R}{\leftarrow} \mathsf{Z}_p \ ; \ X \leftarrow g^x$ $\mathsf{pw}_{A,1} \leftarrow G_1(A, B, pw_A)$ $X^\star \leftarrow X \cdot \mathsf{pw}_{A,1}$		$r \overset{R}{\leftarrow} \mathsf{Z}_p \ ; \ R \overset{R}{\leftarrow} \{0,1\}^{\ell_r}$		$y \overset{R}{\leftarrow} \mathsf{Z}_p \ ; \ Y \leftarrow g^y$ $\mathsf{pw}_{B,1} \leftarrow G_1(A, B, pw_B)$ $Y^\star \leftarrow Y \cdot \mathsf{pw}_{B,1}$
	$\xrightarrow{X^\star}$		$\xleftarrow{Y^\star}$	
		$\mathsf{pw}_{A,1} \leftarrow G_1(A, B, pw_A)$ $\mathsf{pw}_{B,1} \leftarrow G_1(A, B, pw_B)$ $X \leftarrow X^\star / \mathsf{pw}_{A,1}$ $Y \leftarrow Y^\star / \mathsf{pw}_{B,1}$ $\overline{X} \leftarrow X^r$ $\overline{Y} \leftarrow Y^r$ $\mathsf{pw}_{A,2} \leftarrow G_2(A, B, R, pw_A, X^\star)$ $\mathsf{pw}_{B,2} \leftarrow G_2(A, B, R, pw_B, Y^\star)$ $\overline{Y}^\star \leftarrow \overline{Y} \cdot \mathsf{pw}_{A,2}$ $\overline{X}^\star \leftarrow \overline{X} \cdot \mathsf{pw}_{B,2}$		
	$\xleftarrow{R, \overline{Y}^\star}$		$\xrightarrow{R, \overline{X}^\star}$	
$\mathsf{pw}_{A,2} \leftarrow G_2(A, B, R, pw_A, X^\star)$ $\overline{Y} \leftarrow \overline{Y}^\star / \mathsf{pw}_{A,2} \ ; \ K \leftarrow \overline{Y}^x$ $T \leftarrow R, X^\star, Y^\star, \overline{X}^\star, \overline{Y}^\star$ $SK \leftarrow H(A, B, S, T, K)$				$\mathsf{pw}_{B,2} \leftarrow G_2(A, B, R, pw_B, Y^\star)$ $\overline{X} \leftarrow \overline{X}^\star / \mathsf{pw}_{B,2} \ ; \ K \leftarrow \overline{X}^y$ $T \leftarrow R, X^\star, Y^\star, \overline{X}^\star, \overline{Y}^\star$ $SK \leftarrow H(A, B, S, T, K)$

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

Correctness. In an honest execution of the protocol in Figure 3, 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 3 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.

4.2 Security

As the following theorem states, $\mathsf{3PAKE}$ 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 4.1 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 $\mathsf{3PAKE}$ describe the encrypted key exchange protocol associated with these primitives as defined in Figure 3. Then, for any numbers t, q_{server}, q_{start}, q_{exe}, q_{G_1}, q_{G_2}, and q_H,

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

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

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 Appendix A 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.

4.3 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 [**?**, 19] 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 [2].

Fourth, 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 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., [12]).

Acknowledgements

This work has been supported in part by the European Commission through the IST Program under Contract IST-2002-507932 ECRYPT. The information in this document reflects only the author's views, is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability

References

[1] Michel Abdalla, Mihir Bellare, and Phillip Rogaway. The oracle Diffie-Hellman assumptions and an analysis of DHIES. In David Naccache, editor, *Topics in Cryptology – CT-RSA 2001*, volume 2020 of *Lecture Notes in Computer Science*, pages 143–158, San Francisco, CA, USA, April 8–12, 2001. Springer-Verlag, Berlin, Germany.

[2] Michel Abdalla, Pierre-Alain Fouque, and David Pointcheval. Password-based authenticated key exchange in the three-party setting. In Serge Vaudenay, editor, *PKC 2005: 8th International Workshop on Theory and Practice in Public Key Cryptography*, volume 3386 of *Lecture Notes in Computer Science*, pages 65–84, Les Diablerets, Switzerland, January 23–26, 2005. Springer-Verlag, Berlin, Germany.

[3] Michel Abdalla and David Pointcheval. Interactive Diffie-Hellman assumptions with applications to password-based authentication. In Andrew Patrick and Moti Yung, editors, *Financial Cryptography 2005*, volume 3570 of *Lecture Notes in Computer Science*, pages 341–356, Roseau, The Commonwealth Of Dominica, February 28 – March 3, 2005. Springer-Verlag, Berlin, Germany.

[4] Mihir Bellare, David Pointcheval, and Phillip Rogaway. Authenticated key exchange secure against dictionary attacks. In Bart Preneel, editor, *Advances in Cryptology – EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 139–155, Bruges, Belgium, May 14–18, 2000. Springer-Verlag, Berlin, Germany.

[5] Mihir Bellare and Phillip Rogaway. Entity authentication and key distribution. In Douglas R. Stinson, editor, *Advances in Cryptology – CRYPTO'93*, volume 773 of *Lecture Notes in Computer Science*, pages 232–249, Santa Barbara, CA, USA, August 22–26, 1994. Springer-Verlag, Berlin, Germany.

[6] Mihir Bellare and Phillip Rogaway. Provably secure session key distribution — the three party case. In *28th Annual ACM Symposium on Theory of Computing*, pages 57–66, Philadephia, Pennsylvania, USA, May 22–24, 1996. ACM Press.

[7] Mihir Bellare and Phillip Rogaway. The AuthA protocol for password-based authenticated key exchange. Contributions to IEEE P1363, March 2000.

[8] Steven M. Bellovin and Michael Merritt. Encrypted key exchange: Password-based protocols secure against dictionary attacks. In *1992 IEEE Symposium on Security and Privacy*, pages 72–84, Oakland, California, USA, May 1992. IEEE Computer Society Press.

[9] Victor Boyko, Philip D. MacKenzie, and Sarvar Patel. Provably secure password-authenticated key exchange using Diffie-Hellman. In Bart Preneel, editor, *Advances in Cryptology – EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 156–171, Bruges, Belgium, May 14–18, 2000. Springer-Verlag, Berlin, Germany.

[10] Emmanuel Bresson, Olivier Chevassut, and David Pointcheval. New security results on encrypted key exchange. In Feng Bao, Robert Deng, and Jianying Zhou, editors, *PKC 2004: 7th International Workshop on Theory and Practice in Public Key Cryptography*, volume 2947 of *Lecture Notes in Computer Science*, pages 145–158, Singapore, March 1–4, 2004. Springer-Verlag, Berlin, Germany.

[11] Jin Wook Byun, Ik Rae Jeong, Dong Hoon Lee, and Chang-Seop Park. Password-authenticated key exchange between clients with different passwords. In Robert H. Deng, Sihan Qing, Feng Bao, and Jianying Zhou, editors, *ICICS 02: 4th International Conference on Information and Communication Security*, volume 2513 of *Lecture Notes in Computer Science*, pages 134–146, Singapore, December 9–12, 2002. Springer-Verlag, Berlin, Germany.

[12] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. In *30th Annual ACM Symposium on Theory of Computing*, pages 209–218, Dallas, Texas, USA, May 23–26, 1998. ACM Press.

[13] Kim-Kwang Raymond Choo, Colin Boyd, and Yvonne Hitchcock. Errors in computational complexity proofs for protocols. In Bimal K. Roy, editor, *Advances in Cryptology – ASIACRYPT 2005*, volume 3788 of *Lecture Notes in Computer Science*, pages 624–643, Chennai, India, December 4–8, 2005. Springer-Verlag, Berlin, Germany.

[14] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 13–25, Santa Barbara, CA, USA, August 23–27, 1998. Springer-Verlag, Berlin, Germany.

[15] Rosario Gennaro and Yehuda Lindell. A framework for password-based authenticated key exchange. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 524–543, Warsaw, Poland, May 4–8, 2003. Springer-Verlag, Berlin, Germany. `http://eprint.iacr.org/2003/032.ps.gz`.

[16] Oded Goldreich and Yehuda Lindell. Session-key generation using human passwords only. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 408–432, Santa Barbara, CA, USA, August 19–23, 2001. Springer-Verlag, Berlin, Germany. `http://eprint.iacr.org/2000/057`.

[17] Li Gong. Optimal authentication protocols resistant to password guessing attacks. In *CSFW'95: The 8th IEEE Computer Security Foundation Workshop*, pages 24–29, Kenmare, County Kerry, Ireland, March 13–15, 1995. IEEE Computer Society.

[18] Shai Halevi and Hugo Krawczyk. Public-key cryptography and password protocols. *ACM Transactions on Information and System Security*, 2(3):230–268, August 1999.

[19] Hugo Krawczyk. SIGMA: The "SIGn-and-MAc" approach to authenticated Diffie-Hellman and its use in the IKE protocols. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 400–425, Santa Barbara, CA, USA, August 17–21, 2003. Springer-Verlag, Berlin, Germany.

[20] Chun-Li Lin, Hung-Min Sun, and Tzonelih Hwang. Three-party encrypted key exchange: Attacks and a solution. *ACM SIGOPS Operating Systems Review*, 34(4):12–20, October 2000.

[21] Philip D. MacKenzie. The PAK suite: Protocols for password-authenticated key exchange. Contributions to IEEE P1363.2, 2002.

[22] Roger M. Needham and Michael D. Schroeder. Using encryption for authentication in large networks of computers. *Communications of the Association for Computing Machinery*, 21(21):993–999, December 1978.

[23] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000.

[24] Michael Steiner, Gene Tsudik, and Michael Waidner. Refinement and extension of encrypted key exchange. *ACM SIGOPS Operating Systems Review*, 29(3):22–30, July 1995.

[25] Michael Szydlo. A note on chosen-basis decisional Diffie-Hellman assumptions. In Giovanni Di Crescenzo and Avi Rubin, editors, *Financial Cryptography 2006*, Lecture Notes in Computer Science, Anguilla, British West Indies, February 27 – March 2, 2006. Springer-Verlag, Berlin, Germany.

[26] Shuhong Wang, Jie Wang, and Maozhi Xu. Weaknesses of a password-authenticated key exchange protocol between clients with different passwords. In Markus Jakobsson, Moti Yung, and Jianying Zhou, editors, *ACNS 04: 2nd International Conference on Applied Cryptography and Network Security*, volume 3089 of *Lecture Notes in Computer Science*, pages 414–425, Yellow Mountain, China, June 8–11, 2004. Springer-Verlag, Berlin, Germany.

[27] Her-Tyan Yeh, Hung-Min Sun, and Tzonelih Hwang. Efficient three-party authentication and key agreement protocols resistant to password guessing attacks. *Journal of Information Science and Engineering*, 19(6):1059–1070, November 2003.

A Proof of security for 3PAKE

Our proof uses a hybrid argument consisting of a sequence of experiments, the first of which corresponds to the actual attack. For each experiment $\mathbf{Exp}_n$, we define an event $\textsc{Succ}_n$ corresponding to the case in which the adversary correctly guesses the bit b involved in the *Test* query.

H and G_i oracles	– On hash query $H(q)$ (resp. $H'(q)$) for which there exists a record (q, r) in the list Λ_H (resp. Λ_H), return r. Otherwise, choose an element $r \in \{0,1\}^{\ell_k}$, add the record (q, r) to the list Λ_H (resp. Λ_H), and return r.
	– On hash query $G_i(q)$, for which there exists a record $(q, r, \star)$ in the list Λ_{G_i}, return r. Otherwise, choose an element $r \in G$, add the record $(q, r, \perp)$ to the list Λ_{G_i}, and return r.

Figure 4: Simulation of random oracles H, H', G_1, and G_2.

SendClient oracle

– On a query *SendClient*$(U_1^i, (U_2, \mathbf{start}))$, assuming U_1^i is in the correct state, we proceed as follows:

```
    $\theta \xleftarrow{R} \mathsf{Z}_p$ ; $\Theta \leftarrow g^{\theta}$
    $\mathsf{pw}_1 \leftarrow G_1(U_1, U_2, pw_{U_1})$
    $\Theta^{\star} \leftarrow \Theta \cdot \mathsf{pw}_{U_1,1}$
    $\mathbf{return}\ (U_1, U_2, \Theta^{\star})$
```

– On a query *SendClient*$(U_1^i, (U_2, S, R_S, \overline{\Phi}^{\star}))$, assuming U_1^i is in the correct state and U_2 is the intended partner, we proceed as follows:

```
    $\mathsf{pw}_{U_1,2} \leftarrow G_2(U_1, U_2, R_S, pw_{U_1}, \Theta^{\star})$
    $\overline{\Phi} \leftarrow \overline{\Phi}^{\star} / \mathsf{pw}_{U_1,2}$
    $K \leftarrow \overline{\Phi}^{\theta}$
    $SK_{U_1} \leftarrow H(A, B, S, R_S, \Theta^{\star}, \Phi^{\star}, \overline{\Theta}^{\star}, \overline{\Phi}^{\star}, K)$
```

Figure 5: Simulation of *SendClient* oracle query.

Experiment $\mathbf{Exp}_0$. This experiment corresponds to the real attack, in the random oracle model. By definition, we have

$$\mathbf{Adv}^{\mathrm{ake}}_{\mathsf{3PAKE},\mathcal{D}}(A) \quad = \quad 2 \cdot \Pr[\,\textsc{Succ}_0\,] - 1 \tag{1}$$

Experiment $\mathbf{Exp}_1$. In this experiment, we simulate the hash oracles G_1, G_2 and H as usual by maintaining hash lists Λ_{G_1}, Λ_{G_2}, and Λ_H (see Figure 4). In addition to these hash oracles, we also simulate a private hash oracle H' which we will be using later. The *Execute*, *Reveal*, *SendClient*, *SendServer* and *Test* oracles are also simulated as in the real attack (see Figure 5, Figure 6, and Figure 7). One can easily see that this experiment is perfectly indistinguishable from the real experiment. Hence,

$$\Pr[\,\textsc{Succ}_1\,] \quad = \quad \Pr[\,\textsc{Succ}_0\,] \tag{2}$$

Experiment $\mathbf{Exp}_2$. In this experiment, we simulate all oracles as in Experiment $\mathbf{Exp}_1$, except that we halt all executions in which a collision occurs in the output of the G_1 and G_2 oracles or in the transcript $((U_1, U_2, X^{\star}), (U_2, U_1, Y^{\star}), (S, U_2, R, \overline{Y}^{\star}), (S, U_1, R, \overline{X}^{\star}))$. According to the birthday paradox, the probability of collisions in the output of the G_i oracle is at most $q_{G_i}^2/(2p)$, for $i = 1, 2$. Similarly, the probability of collisions in the transcripts is at most $(q_{\mathrm{start}} + q_{\mathrm{exe}})^2/(2p)$, since either $X^{\star}$ or $Y^{\star}$ was simulated and thus chosen uniformly at random. Consequently,

$$\left|\Pr[\,\textsc{Succ}_2\,] - \Pr[\,\textsc{Succ}_1\,]\right| \quad \leq \quad \frac{q_{G_1}^2 + q_{G_2}^2 + (q_{\mathrm{exe}} + q_{\mathrm{start}})^2}{2p} \tag{3}$$

SendServer oracle

```
– On query $SendServer(S^i, ((U_1, U_2, \Theta^\star), (U_2, U_1, \Phi^\star)))$, proceed as follows:
    $r \xleftarrow{R} \mathsf{Z}_p$
    $R_S \xleftarrow{R} \{0,1\}^{\ell_r}$
    $\mathsf{pw}_{U_1,1} \leftarrow G_1(U_1, U_2, pw_{U_1})$ ;                $\mathsf{pw}_{U_2,1} \leftarrow G_1(U_2, U_1, pw_{U_2})$
    $\Theta \leftarrow \Theta^\star / \mathsf{pw}_{U_1,1}$ ;                            $\Phi \leftarrow \Phi^\star / \mathsf{pw}_{U_2,1}$
    $\overline{\Theta} \leftarrow \Theta^r$ ;                                          $\overline{\Phi} \leftarrow \Phi^r$
    $\mathsf{pw}_{U_1,2} \leftarrow G_2(U_1, U_2, R_S, pw_{U_1}, \Theta^\star)$ ; $\mathsf{pw}_{U_2,2} \leftarrow G_2(U_2, U_1, R_S, pw_{U_2}, \Phi^\star)$
    $\overline{\Phi}^\star \leftarrow \overline{\Phi} \cdot \mathsf{pw}_{U_1,2}$ ;                           $\overline{\Theta}^\star \leftarrow \overline{\Theta} \cdot \mathsf{pw}_{U_2,2}$
    return $((U_2, S, R_S, \overline{\Phi}^\star), (U_2, S, R_S, \overline{\Theta}^\star))$
```

Figure 6: Simulation of *SendServer* oracle query.

Experiment $\mathbf{Exp}_3$. In this experiment, we replace the Diffie-Hellman key K with a random element in G for all sessions generated via an *Execute* oracle query. As the following lemma shows, the difference between the current experiment and the previous one is negligible as long as the DDH assumption holds.

Lemma A.1 $\left|\Pr[\,\textsc{Succ}_3\,] - \Pr[\,\textsc{Succ}_2\,]\right| \le q_{\text{exe}} \cdot \mathbf{Adv}_{\mathbb{G}}^{\text{ddh}}(t)$.

Proof: The proof of Lemma A.1 uses a sequence of hybrid experiments $\mathbf{Hybrid}_{3,j}$, where j is an index between 0 and q_{exe}. Let i represent the i-th query to a *Execute* oracle. We define Experiment $\mathbf{Hybrid}_{3,j}$ as follows. If $i \le j$, then we compute the Diffie-Hellman key K with a random element in G for all sessions generated as we would in Experiment $\mathbf{Exp}_2$. Otherwise, we compute the Diffie-Hellman key K as g^k, where k is a random index in Z_p. Note that Experiments $\mathbf{Exp}_2$ and $\mathbf{Exp}_3$ are equivalent to Experiments $\mathbf{Hybrid}_{3,0}$ and $\mathbf{Hybrid}_{3,q_{\text{exe}}}$, respectively.

Let P_j be the probability of the event $\textsc{Succ}$ in Experiment $\mathbf{Hybrid}_{3,j}$. Then,

$$\Pr[\,\textsc{Succ}_3\,] = P_{q_{\text{exe}}} \quad \text{and} \quad \Pr[\,\textsc{Succ}_2\,] = P_0\ ,$$

and

$$\Big|\Pr[\,\textsc{Succ}_3\,] - \Pr[\,\textsc{Succ}_2\,]\Big| \;=\; \Big|\sum_{j=1}^{q_{\text{exe}}} P_j - P_{j-1}\Big| = \sum_{j=1}^{q_{\text{exe}}} \left|P_j - P_{j-1}\right| .$$

The lemma follows easily from the above by showing that $\left|P_j - P_{j-1}\right|$ is at most $\mathbf{Adv}_{\mathbb{G}}^{\text{ddh}}(t)$. To do so, consider the following algorithm D_j against the Diffie-Hellman problem in $\mathbb{G}$. Let X, Y, and W be the input for D_j. D_j starts running A, answering all queries as in Experiment $\mathbf{Hybrid}_{3,j-1}$, up until the j-th query to the *Execute* oracle. To answer to this query, D_j uses the values X and Y that it received as input in place of Θ and Φ, respectively. It also sets the Diffie-Hellman keys K_{U_1} and K_{U_2} relative to users U_1 and U_2 to W^r. All remaining *Execute* oracle queries are simulated as in Experiment $\mathbf{Exp}_2$. Finally, D_j outputs the same bit b outputted by A as its guess.

In order to analyze the advantage of D_j, first note that if $W = \text{CDH}(X, Y)$, then the probability of D_j outputting 1 equals the probability of A outputting 1 in Experiment $\mathbf{Hybrid}_{3,j-1}$. If W

Execute, *Reveal* and *Test* queries.

```
– On query $Reveal(U^i)$, proceed as follows:
    if session key $SK$ is defined for instance $U_i$
    then return $SK$,
    else return $\perp$.

– On query $Execute(U_1^i, U_2^j, S^k)$, proceed as follows:
    $(U_1, U_2, \Theta^\star) \leftarrow SendClient(U_1^i, (U_2, \mathbf{start}))$
    $(U_2, U_1, \Phi^\star) \leftarrow SendClient(U_2^j, (U_1, \mathbf{start}))$
    $((U_2, S, R_S, \overline{\Phi}^\star), (U_1, S, R_S, \overline{\Theta}^\star)) \leftarrow$
        $SendServer(S^k, ((U_1, U_2, \Theta^\star), (U_2, U_1, \Phi^\star)))$
    $SendClient(U_1^i, (U_2, S, R_S, \overline{\Phi}^\star))$
    $SendClient(U_2^j, (U_1, S, R_S, \overline{\Theta}^\star))$
    return $((U_1, U_2, \Theta^\star), (U_2, U_1, \Phi^\star), (U_2, S, R_S, \overline{\Phi}^\star), (U_1, S, R_S, \overline{\Theta}^\star))$

– On query $Test(U^i)$, proceed as follows:
    $SK \leftarrow Reveal(U^i)$
    if $SK = \perp$ then return $\perp$
    else
        $b \overset{R}{\leftarrow} \{0,1\}$
        if $b = 0$ then $SK' \leftarrow SK$ else $SK' \overset{R}{\leftarrow} \{0,1\}^{\ell_k}$
    return $SK'$
```

Figure 7: Simulation of *Execute*, *Reveal* and *Test* queries.

is a random element in G, then the probability of D_j outputting 1 equals the probability of A outputting 1 in Experiment $\mathbf{Hybrid}_{3,j}$. Hence,

$$\mathbf{Adv}_{\mathbb{G}}^{\mathrm{ddh}}(D_j) \quad = \quad \left| P_j - P_{j-1} \right| ,$$

and, given that D_j runs in time at most t,

$$\left| P_j - P_{j-1} \right| , \quad \leq \quad \mathbf{Adv}_{\mathbb{G}}^{\mathrm{ddh}}(t)$$

▌

Experiment $\mathbf{Exp}_4$. In this experiment, we once again change the simulation of queries to the *Execute* oracle. This time, we change the way we compute the values $\overline{\Theta}$ and $\overline{\Phi}$ so that a different value of r is used to compute each of them. That is, we make $\overline{\Theta} = \Theta^r$ and $\overline{\Phi} = \Phi^{r'}$ for random and independent values r and r' in Z_p. As the following lemma shows, the difference between the current experiment and the previous one is negligible as long as the DDH assumption holds.

Lemma A.2 $\left| \Pr[\,\textsc{Succ}_4\,] - \Pr[\,\textsc{Succ}_3\,] \right| \leq q_{\mathrm{exe}} \cdot \mathbf{Adv}_{\mathbb{G}}^{\mathrm{ddh}}(t)$.

Proof: The proof of Lemma A.2 is similar to that of Lemma A.1, so we only point out the differences here. First, the main difference between the two is that we now rely on the fact the adversary should not be able to distinguish the case where the same random index r is used from the case where different random indices r and r' are used. As shown in [14], this problem is equivalent to the DDH problem. The other difference is that the DDH problem is now embedded in the values of $\overline{\Phi}$ and $\overline{\Theta}$, and not in the Diffie-Hellman key K as in the proof of Lemma A.1. ▌

Experiment $\mathbf{Exp}_5$. In this experiment, we change for the last time the simulation of queries to the *SendClient* and *SendServer* oracles whenever we have a passive attack executed either via *Execute* queries so that the output of the *SendServer* oracle queries are independent of its input and of the passwords of honest users. More specifically, we now compute $\overline{\Theta}^\star$ and $\overline{\Phi}^\star$ as $g^{\overline{\theta}^\star}$ and $g^{\overline{\phi}^\star}$, respectively, where $\overline{\theta}^\star$ and $\overline{\phi}^\star$ are random indices in Z_p.

In order to understand the differences between the current experiment and the previous one, please note that in the previous experiment, the values $\overline{\Theta}^\star$ and $\overline{\Phi}^\star$ were both independent from the session key and uniformly distributed in G, as both values were computed using a different random index r in Z_p. Hence, no information on the password is leaked through these values. As a result, the current experiment and the previous one are perfectly indistinguishable.

$$\Pr[\,\textsc{Succ}_5\,] \;=\; \Pr[\,\textsc{Succ}_4\,]\,. \qquad (4)$$

Experiment $\mathbf{Exp}_6$. In this experiment, we change the way we compute the session keys of certain sessions, by using our *private* random oracle H' instead of H. The goal is to make the session key of those sessions not only independent of the password but also independent of the Diffie-Hellman secret K. This is achieved by changing the simulation of the *SendClient* oracle so that the session key SK is computed via $H'(U_1, U_2, S, R, \Theta^\star, \Phi^\star, \overline{\Theta}^\star, \overline{\Phi}^\star)$. That is, the Diffie-Hellman key K is no longer used.

The sessions in which we modify the simulation of a *SendClient* oracle are all of those for which one of the following conditions is met.

- Both U_1 and U_2 are honest players and U_2 is the intended partner of instance U_1^i, and the input query to the *SendClient* oracle for instance U_1^i does not match the output of any *SendServer* oracle query. In other words, the input query was generated by the adversary;

- Both U_1 and U_2 are honest players, U_2 is the intended partner of instance U_1^i, and the input query to *SendClient* oracle for instance U_1^i matches the output a *SendServer* oracle S_k, the input to which contains one part matching the output of *SendClient* query $(U_1^i, (U_2, \mathbf{start}))$ and the other part not matching the output of any *SendClient* query.

Please note that we can test the occurrence of any of these events by looking up the list of inputs and outputs of each simulated oracle instance.

Let $\textsc{AskH}_n$ denote the event in which the adversary asks a query $(U_1, U_2, S, R, X^\star, Y^\star, \overline{X}^\star, \overline{Y}^\star, K_{U_1})$ or $(U_1, U_2, S, R, X^\star, Y^\star, \overline{X}^\star, \overline{Y}^\star, K_{U_2})$ to the random oracle H for some execution transcript $((U_1, U_2, X^\star), (U_2, U_1, Y^\star), (S, U_2, R, \overline{Y}^\star), (S, U_1, R, \overline{X}^\star))$ when in Experiment $\mathbf{Exp}_n$. That is, $\textsc{AskH}_n$ denotes the event in Experiment $\mathbf{Exp}_n$ that either the oracle query $(U_1, U_2, S, R, X^\star, Y^\star, \overline{X}^\star, \overline{Y}^\star, \mathrm{CDH}(X^\star/\mathsf{pw}_{U_1,1}, \overline{Y}^\star/\mathsf{pw}_{U_1,2}))$ was asked to H and $X^\star$ was simulated or the oracle query $(U_1, U_2, S, R, X^\star, Y^\star, \overline{X}^\star, \overline{Y}^\star, \mathrm{CDH}(Y^\star/\mathsf{pw}_{U_2,1}, \overline{X}^\star/\mathsf{pw}_{U_2,2}))$ was asked to H and $Y^\star$ was simulated.

There are some important observations to be made regarding the differences between the current experiment and the previous one. First, experiments $\mathbf{Exp}_5$ and $\mathbf{Exp}_6$ can only be told apart if event $\textsc{AskH}_5$ or event $\textsc{AskH}_6$ occurs since this is the only scenario in which the answers to the hash query H may differ. Therefore,

$$\Pr\left[\,\textsc{Succ}_6 \mid \overline{\textsc{AskH}_6}\,\right] \;=\; \Pr\left[\,\textsc{Succ}_5 \mid \overline{\textsc{AskH}_5}\,\right]. \qquad (5)$$

Second, the probability of the events $\textsc{AskH}_5$ and $\textsc{AskH}_6$ are the same as the adversary has equal chance in both experiments of asking a *crucial* query. Thus,

$$\Pr[\,\textsc{AskH}_5\,] \;=\; \Pr[\,\textsc{AskH}_6\,], \tag{6}$$

which, combined with Equation 5, leads to

$$\big|\Pr[\,\textsc{Succ}_6\,] - \Pr[\,\textsc{Succ}_5\,]\big| \;\leq\; \Pr[\,\textsc{AskH}_6\,]. \tag{7}$$

Third, and lastly, the replacement of the random oracle H by the private random oracle H' in the current experiment together with the fact that the session key of passive attacks were already made independent of the Diffie-Hellman key in previous experiments makes it impossible for the adversary to tell apart the real session key from a random one in any valid *Test* query. This is so because a random value is returned in all scenarios for which a valid *Test* query can be made and because we removed transcripts collisions. Note that a transcript collision could have leaked the session key to the adversary via a reveal query to the other session in which the transcript appears. Therefore, the success probability of the adversary is exactly 1/2 in the current experiment.

$$\Pr[\,\textsc{Succ}_6\,] \;=\; \frac{1}{2} \tag{8}$$

As the adversary can no longer tell apart real session keys from random ones, we will not consider the success probability of the adversary in the remaining experiments. Instead, we will concentrate on the event $\textsc{AskH}$ whose probability we still need to evaluate in order to determine an upper bound on the adversary's success probability in the real attack (Experiment $\mathbf{Exp}_0$).

Experiment $\mathbf{Exp}_7$. The goal of this experiment is to bound the advantage of the adversary in those cases where the latter is performing an active attack against the server. Such attacks occur when at least one or both parts of the input of a *SendServer* oracle are generated by the adversary, thus not matching the output of a previous *SendClient* oracle query. In this scenario, an active adversary may try to use the server to learn information about the password of an honest user and later use it to impersonate that user. To achieve our goal, we change the simulation of the *SendServer* oracle so that any output value corresponding to an honest user is computed using a random value and not the input provided by the *SendServer* oracle input.

Let $(S^k, (U_1, U_2, \Theta^\star), (U_2, U_1, \Phi^\star))$ be a *SendServer* oracle query to server instance S^k. If either $(U_1, U_2, \Theta^\star)$ or $(U_2, U_1, \Phi^\star)$ or both match the output of previous *SendClient* queries, then we change the simulation of the *SendServer* oracle query as follows. If U_1 is an honest user, then we compute $\overline{\Phi}^\star$ as $g^{\overline{\phi}} \cdot \mathsf{pw}_{U_1,2}$, where $\overline{\phi}$ is a random index in Z_p. Otherwise, we compute $\overline{\Phi}^\star$ as we would in Experiment $\mathbf{Exp}_6$. Likewise, If U_2 is an honest user, then we compute $\overline{\Theta}^\star$ as $g^{\overline{\theta}} \cdot \mathsf{pw}_{U_2,2}$, where $\overline{\theta}$ is a random index in Z_p. Otherwise, we compute $\overline{\Theta}^\star$ as we would in Experiment $\mathbf{Exp}_6$.

As the following lemma shows, the adversary cannot do much better than simply guessing the password.

Lemma A.3 $\big|\textsc{AskH}_7 - \textsc{AskH}_6\big| \leq q_{\mathrm{server}} \cdot \max\{\, 2 \cdot \mathbf{Adv}^{\mathrm{pcddh1}}_{\mathbb{G},|\mathcal{D}|}(q_{\mathrm{start}} \cdot t)\;,\; \mathbf{Adv}^{\mathrm{pcddh2}}_{\mathbb{G},|\mathcal{D}|}(t)\,\}$.

Proof: The proof of this lemma is based on a sequence of $q_{\mathrm{server}} + 1$ hybrid experiments $\mathbf{Hybrid}_{7,j}$, where j is an index between 0 and q_{server}. Let i represent the i-th query to a *SendServer* oracle for which both parts of the input do not match the output of any *SendClient* oracle query, and let $(S^k, ((U_1, U_2, \Theta^\star), (U_2, U_1, \Phi^\star)))$ be this query. We define Experiment $\mathbf{Hybrid}_{7,j}$ as follows. If $i \leq j$, then we check whether U_1 and U_2 are honest users. If U_1 is an honest user, then we compute

$\overline{\Phi}^\star$ as $g^{\overline{\phi}} \cdot \mathsf{pw}_{U_1,2}$, where $\overline{\phi}$ is a random index in Z_p. Otherwise, we compute $\overline{\Phi}^\star$ as we would in Experiment $\mathbf{Exp}_6$. If U_2 is an honest user, then we compute $\overline{\Theta}^\star$ as $g^{\overline{\theta}} \cdot \mathsf{pw}_{U_1,2}$, where $\overline{\theta}$ is a random index in Z_p. Otherwise, we compute $\overline{\Theta}^\star$ as we would in Experiment $\mathbf{Exp}_6$.

From the definition of the hybrid experiments, one can see that no changes are made to the simulation when $j = 0$ and, thus, experiments $\mathbf{Hybrid}_{7,0}$ and $\mathbf{Exp}_6$ are equivalent. Moreover, since there are at most q_{server} such queries, Experiment $\mathbf{Hybrid}_{7,q_{\mathrm{server}}}$ corresponds to the case where we modify the simulation of all *SendServer* oracle queries with inputs coming from the adversary and, thus, experiments $\mathbf{Hybrid}_{7,j}$ and $\mathbf{Exp}_7$ are also equivalent. Let us define $\textsc{AskH}_{7,j}$ to be the event $\textsc{AskH}$ in Experiment $\mathbf{Hybrid}_{7,j}$. Then,

$$\textsc{AskH}_6 = \textsc{AskH}_{7,0} \quad \text{and} \quad \textsc{AskH}_6 = \textsc{AskH}_{7,q_{\mathrm{server}}} \ , \tag{9}$$

and

$$\left|\Pr[\,\textsc{AskH}_7\,] - \Pr[\,\textsc{AskH}_6\,]\right| \le \sum_{j=1}^{q_{\mathrm{server}}} \left|\Pr[\,\textsc{AskH}_{7,j}\,] - \Pr[\,\textsc{AskH}_{7,j-1}\,]\right| \ . \tag{10}$$

We now claim that

$$\left|\Pr[\,\textsc{AskH}_{7,j}\,] - \Pr[\,\textsc{AskH}_{7,j-1}\,]\right| \le \max\{\, 2 \cdot \mathbf{Adv}^{\mathrm{pcddh1}}_{\mathbb{G},|\mathcal{D}|}(q_{\mathrm{start}} \cdot t)\ ,\ \mathbf{Adv}^{\mathrm{pcddh2}}_{\mathbb{G},|\mathcal{D}|}(t)\,\} \ . \tag{11}$$

Lemma A.3 follows easily from above claim by substituting Equation 11 in Equation 10.

Let us now prove the claim in Equation 11. Let D be a distinguisher for the event $\textsc{AskH}$ in both experiments $\mathbf{Hybrid}_{7,j-1}$ and $\mathbf{Hybrid}_{7,j}$. Using D, we will build two distinguishers M_1 and M_2 for the PCDDH1 and PCDDH2 problems, respectively. M_1 will be used whenever only one part of the input for the j-th *SendServer* oracle query comes from the adversary ($\textsc{Case1}$). M_2 will be used whenever both parts of the input for the j-th *SendServer* oracle query comes from the adversary ($\textsc{Case2}$). There is no need for us to guess which case we are in as this information is available from the simulation. $\textsc{Case1}$ and $\textsc{Case2}$ are mutually exclusive.

We now define our distinguisher M_1 for the PCDDH1 problem. Let X be the input to M_1's find stage. M_1 starts its find stage by choosing a random index k between 1 and q_{start}. Next, M_1 starts running D, the distinguisher for the event $\textsc{AskH}$ in experiments $\mathbf{Exp}_7$ and $\mathbf{Exp}_6$. M_1 simulates all oracles as it normally would in Experiment $\mathbf{Exp}_6$, with the exception of the *SendServer* oracle and *SendClient* oracles. The simulation of the *SendClient* oracle is modified as follows. All queries to this oracle are answered as in Experiment $\mathbf{Exp}_6$, except for the k-th query of the form $(U_1, U_2, \mathbf{start})$. To answer this query, we use the input X that we received and output (U_1, U_2, X). The simulation of the *SendServer* oracle is as follows. Let i represent the i-th query to the *SendServer* oracle and let $(S^k, ((U_1, U_2, \Theta^\star), (U_2, U_1, \Phi^\star)))$ be this query. If $i < j$, then we check whether U_1 and U_2 are honest users. If U_1 is an honest user, then we compute $\overline{\Phi}^\star$ as $g^{\overline{\phi}} \cdot \mathsf{pw}_{U_1,2}$, where $\overline{\phi}$ is a random index in Z_p. Otherwise, we compute $\overline{\Phi}^\star$ as we would in Experiment $\mathbf{Exp}_6$. If U_2 is an honest user, then we compute $\overline{\Theta}^\star$ as $g^{\overline{\theta}} \cdot \mathsf{pw}_{U_1,2}$, where $\overline{\theta}$ is a random index in Z_p. Otherwise, we compute $\overline{\Theta}^\star$ as we would in Experiment $\mathbf{Exp}_6$. If $i = j$, then let us assume wlog that $(U_1, U_2, \Theta^\star)$ is the input that comes from the simulated oracle and that $(U_2, U_1, \Phi^\star)$ is the input that comes from the adversary. At this point, M_1 should check whether the tuple $(U_1, U_2, \Theta^\star)$ matches the output of the k-th start query. If there is no match, then M_1 should restart D using fresh coins up to q_{start} times. If the tuple $(U_1, U_2, \Theta^\star)$ matches the output of the k-th start query, then M_1 returns (s, Y) as the output of its find stage, where s contains all the necessary information that M_1 may need to continue running the simulation of D in the guess stage.

Let (s, X', Y', K, k) be the input to the guess stage of M_1. We then choose a random value for R in $\{0,1\}^{\ell_r}$ and return $((R, S, U_2, Y'), (R, S, U_1, X'))$ as the the answer to the j-th *SendServer* oracle query. The rest of the simulation of all oracles is done as in Experiment $\mathbf{Exp_6}$. The only difference is that, from now on, we define K as the Diffie-Hellman key associated with *SendClient* oracle. Hence, we can check for the AskH event associated with this session using the Diffie-Hellman key K and the other values that we used in the simulation of oracle U_1. If our guess for the k-th start query is correct, then one can see that the only difference between experiments $\mathbf{Hybrid}_{7,j-1}$ and $\mathbf{Hybrid}_{7,j}$ stems from the hidden bit associated with the PCDDH1 problem. Hence, we can use the event AskH to guess the hidden bit used in the computation of X' and K given to us at the input of the guess stage. If AskH occurs, then we output 0 else 1. Since we run this experiment up to q_{start} times, the probability that one of our guess for the index k is right is at least $1-1/e \geq 1/2$, where e is the base of the natural logarithm. Hence, given that we are in the correct scenario, we know that probability of success of M_1 is at least the difference between the probability of the event AskH in experiments $\mathbf{Hybrid}_{7,j-1}$ and $\mathbf{Hybrid}_{7,j}$. Using the fact that the running time of M_1 is at most $q_{\text{start}} \cdot t$, we have

$$\begin{aligned}\left|\Pr[\,\textsc{AskH}_{7,j} \mid \textsc{Case1}\,] - \Pr[\,\textsc{AskH}_{7,j-1} \mid \textsc{Case1}\,]\right| &\leq 2 \cdot \mathbf{Adv}^{\text{pcddh1}}_{\mathbb{G},|\mathcal{D}|}(M_1) \\ &\leq 2 \cdot \mathbf{Adv}^{\text{pcddh1}}_{\mathbb{G},|\mathcal{D}|}(q_{\text{start}} \cdot t)\ .\end{aligned}$$

We now define the distinguisher M_2 for the PCDDH2 problem. M_2 starts its find stage by running D, the distinguisher for the event AskH in experiments $\mathbf{Exp_7}$ and $\mathbf{Exp_6}$. M_2 simulates all oracles as it normally would in experiment $\mathbf{Exp_6}$, with the exception of the *SendServer* oracle. The simulation of the *SendServer* oracle is as follows. Let i represent the i-th query to the *SendServer* oracle and let $(S^k, ((U_1, U_2, \Theta^\star), (U_2, U_1, \Phi^\star)))$ be this query. We only consider cases where at least one of the users U_1 or U_2 is honest, since the bound in the claim follows trivially otherwise (experiments $\mathbf{Hybrid}_{7,j}$ and $\mathbf{Hybrid}_{7,j}$ are perfectly indistinguishable when both U_1 and U_2 are dishonest in the j-th query). If $i < j$, then we check whether U_1 and U_2 are honest users. If U_1 is an honest user, then we compute $\overline{\Phi}^\star$ as $g^{\overline{\phi}} \cdot \mathsf{pw}_{U_1,2}$, where $\overline{\phi}$ is a random index in Z_p. Otherwise, we compute $\overline{\Phi}^\star$ as we would in Experiment $\mathbf{Exp_6}$. If U_2 is an honest user, then we compute $\overline{\Theta}^\star$ as $g^{\overline{\theta}} \cdot \mathsf{pw}_{U_1,2}$, where $\overline{\theta}$ is a random index in Z_p. Otherwise, we compute $\overline{\Theta}^\star$ as we would in Experiment $\mathbf{Exp_6}$. If $i = j$, then let us assume wlog that U_1 represents a honest player and we let the hidden k used in the definition of the PCDDH2 problem be associated with the password of U_1. Using the password for the user U_2 and the input $\Phi^\star$, we then choose a random value for R in $\{0,1\}^{\ell_r}$ and compute the masks $\mathsf{pw}_{U_2,1}$ and $\mathsf{pw}_{U_2,2}$ and the value Φ. Next, we return $(s, \Theta^\star, \Phi)$ as the output of our find stage, where s contains all the necessary information that we may need to continue running the simulation of D in the guess stage.

Let (s, X', Y', k) be the input to the guess stage of M_2. We set $\overline{\Theta}^\star = X' \cdot \mathsf{pw}_{U_2,2}$ and $\overline{\Phi}^\star = Y'$ and return $((R, S, U_2, \overline{\Phi}^\star), (R, S, U_1, \overline{\Theta}^\star))$ as the the answer to the j-th *SendServer* oracle query. We also query the oracle $\mathcal{P}$ on input k and use $U = \mathcal{P}(k)$ as the mask $\mathsf{pw}_{U_1,1}$ for user U_1 when computing answers to future queries to *SendClient* and *SendServer* oracles with respect to U_1. $\mathsf{pw}_{U_1,1}$ is also used to check for event AskH with respect to oracles associated with user U_1. The rest of the simulation proceeds as in Experiment $\mathbf{Exp_6}$. As one can see, the only difference between experiments $\mathbf{Hybrid}_{7,j-1}$ and $\mathbf{Hybrid}_{7,j}$ stems from the hidden bit associated with the PCDDH2 problem. Hence, we can use the event AskH to guess the hidden bit used in the computation of X' and Y' given to us at the input of the guess stage. If AskH occurs, then we output 0 else 1. Given that we are in the correct scenario, we know that probability of success of M_2 is at least the difference between the probability of the event AskH in experiments $\mathbf{Hybrid}_{7,j-1}$ and $\mathbf{Hybrid}_{7,j}$.

Using the fact that the running time of M_2 is at most t, we have

$$\left|\Pr\left[\,\textsc{AskH}_{7,j} \;\middle|\; \textsc{Case2}\,\right] - \Pr\left[\,\textsc{AskH}_{7,j-1} \;\middle|\; \textsc{Case2}\,\right]\right| \le \mathbf{Adv}^{\mathrm{pcddh2}}_{\mathbb{G},|\mathcal{D}|}(M_2) \le \mathbf{Adv}^{\mathrm{pcddh2}}_{\mathbb{G},|\mathcal{D}|}(t)\ . \qquad (12)$$

Since $\textsc{Case1}$ and $\textsc{Case2}$ are mutually exclusive,

$$\Pr[\,\textsc{Case1}\,] + \Pr[\,\textsc{Case2}\,] = 1\ ,$$

and

$$\begin{aligned}
&\left|\Pr[\,\textsc{AskH}_{7,j}\,] - \Pr[\,\textsc{AskH}_{7,j-1}\,]\right| \\
&\quad\le \Big|\Pr\left[\,\textsc{AskH}_{7,j} \mid \textsc{Case1}\,\right]\cdot\Pr[\,\textsc{Case1}\,] + \Pr\left[\,\textsc{AskH}_{7,j} \mid \textsc{Case2}\,\right]\cdot\Pr[\,\textsc{Case2}\,] - \\
&\qquad \Pr\left[\,\textsc{AskH}_{7,j-1} \mid \textsc{Case1}\,\right]\cdot\Pr[\,\textsc{Case1}\,] - \Pr\left[\,\textsc{AskH}_{7,j-1} \mid \textsc{Case2}\,\right]\cdot\Pr[\,\textsc{Case2}\,]\Big| \\
&\quad\le \delta\mathbf{Adv}^{\mathrm{pcddh1}}_{\mathbb{G},|\mathcal{D}|}(q_{\mathrm{start}}\cdot t)\cdot\Pr[\,\textsc{Case1}\,] + \mathbf{Adv}^{\mathrm{pcddh2}}_{\mathbb{G},|\mathcal{D}|}(t)\cdot\Pr[\,\textsc{Case2}\,] \\
&\quad\le \max\{\,\delta\mathbf{Adv}^{\mathrm{pcddh1}}_{\mathbb{G},|\mathcal{D}|}(q_{\mathrm{start}}\cdot t)\ ,\ \mathbf{Adv}^{\mathrm{pcddh2}}_{\mathbb{G},|\mathcal{D}|}(t)\,\}\cdot\Pr[\,\textsc{Case1}\,] + \\
&\qquad \max\{\,\delta\mathbf{Adv}^{\mathrm{pcddh1}}_{\mathbb{G},|\mathcal{D}|}(q_{\mathrm{start}}\cdot t)\ ,\ \mathbf{Adv}^{\mathrm{pcddh2}}_{\mathbb{G},|\mathcal{D}|}(t)\,\}\cdot\Pr[\,\textsc{Case2}\,] \\
&\quad\le \max\{\,\delta\mathbf{Adv}^{\mathrm{pcddh1}}_{\mathbb{G},|\mathcal{D}|}(q_{\mathrm{start}}\cdot t)\ ,\ \mathbf{Adv}^{\mathrm{pcddh2}}_{\mathbb{G},|\mathcal{D}|}(t)\,\}\ .
\end{aligned}$$

▮

Remark A.4 This is the only part of the proof that does not work in the concurrent model. The reason for that is that in order to be able convert an adversary against our protocol into an adversary against the PCDDH1 problem, we must be able to detect the event AskH. However, when multiple concurrent sessions are allowed, we may not be able to detect the event AskH associated with each of the concurrent sessions. More specifically, consider the values k and $U = \mathcal{P}(k)$ used in the experiment defining the PCDDH1 problem. In our proof, we associate these values to pw_{U_1} and $\mathsf{pw}_{U_1,1}$, where U_1 is the user whose password the adversary is trying to obtain. Hence, in order to be able to detect the AskH event associated with sessions of user U_1, we must be able to compute $\mathrm{CDH}(X^\star/\mathsf{pw}_{U_1,1}, \overline{Y}^\star)$. In the case where we have only one session associated with U_1, we can do so using the key K given to us in the input of the guess stage. However, when multiple concurrent sessions are allowed, we are only able to do so for sessions which are started after we learn the value $U = \mathsf{pw}_{U_1,1}$ (by making $X^\star = g^x\cdot\mathsf{pw}_{U_1,1}$). Unfortunately, this is not possible for sessions that started prior to the moment in which we learn k and U, since for those sessions, we need to be able to compute $\mathrm{CDH}(X^\star/\mathsf{pw}_{U_1,1}, \overline{Y}^\star)$ only knowing the discrete logarithm of $X^\star$.

Experiment $\mathbf{Exp}_8$. In this experiment, we modify the simulation of the *SendServer* oracle in cases where only one part of its input comes from a previous simulated *SendClient* oracle so that we no longer use the password when computing the response to be sent to the simulated oracle. in fact, we want to make this answer independent of the input value provided in the query.

Let $(S^k, (U_1, U_2, \Theta^\star), (U_2, U_1, \Phi^\star))$ be a *SendServer* oracle query to server instance S^k so that either $(U_1, U_2, \Theta^\star)$ or $(U_2, U_1, \Phi^\star)$ matches the output of previous *SendClient* queries. Let us assume wlog that $(U_1, U_2, \Theta^\star)$ is the part of the input that came from a simulated oracle for U_1 (the other case is equivalent) and that the latter is an honest user. Then, we compute $\overline{\Phi}^\star$ as $g^{\overline{\phi}^\star}$, where $\overline{\phi}^\star$ is a random index in Z_p. In order to maintain the consistency of the simulation, we also change the computation of the Diffie-Hellman key K_{U_1}, setting it to $X^{\overline{\phi}^\star}$, where $X = \mathsf{pw}_{U_2,1}$. No change is made to the computation of $\overline{\Theta}^\star$.

We claim that the current experiment and the previous one are indistinguishable. To see why this is the case, first note that, in the previous experiment, the output being sent to the simulated oracle was already computed using a different random value r than the one used in the part of the output. This is still the case in this experiment. Second, also note that the relationship between the output being sent to the simulated oracle and its Diffie-Hellman key used to detect the AskH event is still preserved in the current experiment. Finally, no change was made to the part of the output that do not correspond to an oracle. Therefore,

$$\Pr[\,\textsc{AskH}_8\,] = \Pr[\,\textsc{AskH}_7\,]. \tag{13}$$

Experiment Exp$_9$. In this experiment, we modify the simulation of the *SendServer* oracle once again in cases where only one part of its input comes from a previous simulated *SendClient* oracle so that the part of the output being sent to the non-simulated but honest party no longer uses the password in its computation.

Let $(S^k, (U_1, U_2, \Theta^\star), (U_2, U_1, \Phi^\star))$ be a *SendServer* oracle query to server instance S^k so that either $(U_1, U_2, \Theta^\star)$ or $(U_2, U_1, \Phi^\star)$ matches the output of previous *SendClient* queries. Let us assume wlog that $(U_1, U_2, \Theta^\star)$ is the part of the input that came from a simulated oracle for U_1 (the other case is equivalent) and that both U_1 and U_2 are honest players. Then, we compute $\overline{\Theta}^\star$ as $g^{\overline{\theta}^\star}$, where $\overline{\phi}^\star$ is a random index in Z_p.

We claim that the current experiment and the previous one are indistinguishable. To see why this is the case, just notice that, in the previous experiment, this output was already uniformly distributed in G and already independent from the output being sent to the simulated oracle and from the Diffie-Hellman keys, and hence, independent of the password. It follows that

$$\Pr[\,\textsc{AskH}_9\,] = \Pr[\,\textsc{AskH}_8\,]\;. \tag{14}$$

Experiment Exp$_{10}$. In this experiment, we change the simulation of *SendClient* oracles so that we no longer use the password when answering to a $(U_1^i, (U_2, \textbf{start}))$ query, where U_1 and U_2 are honest users. Such change does not change any of the probabilities associated with the previous experiments since the passwords associated with these users were no longer being used in the simulation of the *SendServer* oracle or in the computation of the session key. Thus, we have

$$\Pr[\,\textsc{AskH}_{10}\,] \;=\; \Pr[\,\textsc{AskH}_9\,]\;. \tag{15}$$

Moreover, since the passwords of honest users are no longer used anywhere else, we can postpone choosing them until the very end of the simulation and only then use them to evaluate the probability of the event AskH$_{10}$. This is given by the following lemma.

Lemma A.5 $\Pr[\,\textsc{AskH}_{10}\,] \leq q_{\mathrm{start}}/|\mathcal{D}| + q_{G_1}^2 \cdot q_{G_2}^2 \cdot q_H^2 \cdot \mathbf{Adv}_{\mathbb{G}}^{\mathrm{cdh}}(t + 3\tau_e) + \frac{q_{G_1}+q_{G_2}}{p} + 2\;\frac{q_H}{p}$.

Proof: Please recall that AskH$_n$ denotes the event that, for some transcript $((U_1, U_2, X^\star), (U_2, U_1, Y^\star), (S, U_2, R, \overline{Y}^\star), (S, U_1, R, \overline{X}^\star))$ in Experiment $\mathbf{Exp}_n$, either one of the *crucial* oracle queries $(U_1, U_2, S, R, X^\star, Y^\star, \overline{X}^\star, \overline{Y}^\star, \mathrm{CDH}(X^\star/\mathsf{pw}_{U_1,1}, \overline{Y}^\star/\mathsf{pw}_{U_1,2}))$ or $(U_1, U_2, S, R, X^\star, Y^\star, \overline{X}^\star, \overline{Y}^\star, \mathrm{CDH}(Y^\star/\mathsf{pw}_{U_2,1}, \overline{X}^\star/\mathsf{pw}_{U_2,2}))$ lies in the list Λ_H in one of following two cases:

AskH1$_{10}$**:** U_1 and U_2 are honest players, U_2 is the intended partner of instance U_1^i, and the input query to the *SendClient* oracle for instance U_1^i matches the output a *SendServer* oracle S_k, whose input only partially comes from a simulated oracle.

$\textsc{AskH2}_{10}$**:** U_1 and U_2 are honest players, U_2 is the intended partner of instance U_1^i, and the input query $(S, U_2, R, \overline{X})$ to the *SendClient* oracle with respect to instance U_1^i was generated by the adversary.

Before proceeding with the probability analyses of the events $\textsc{AskH1}_{10}$ and $\textsc{AskH2}_{10}$, let us make two observations.

First, notice that we can disregard those cases for which the queries $(U_1, U_2, pw_{U_1}, R, X^\star)$ and $(U_2, U_1, pw_{U_2}, R, \overline{Y}^\star)$ were not asked to the G_2 oracle since the probability of the event $\textsc{AskH}$ is negligible in this case as stated by the following claim.

Claim A.6 Let $(X^\star, \overline{Y}^\star, \overline{X}^\star, Y^\star) \in G^4$, $R \in \{0,1\}^{\ell_r}$, S, U_1, and U_2 be a set of values involved in the communication either with an instance i of a participant U_1 in its role as an initiator or an instance j of a participant U_1 in its role as a responder, and let pw_{U_1} and pw_{U_2} represent the passwords associated with U_1 and U_2. Let $\textsc{AskG2}$ denote the event in which either the query (U_1, $U_2, pw_{U_1,2}, R, X^\star$) was asked to G_2 when communicating with U_1^i or the query ($U_2, U_1, pw_{U_2,2}, R$, $Y^\star$) was asked to G_2 when communicating with U_2^j. Then,

$$\Pr[\, \textsc{AskH}_{10} \wedge \overline{\textsc{AskG2}} \,] \leq 2 \, \frac{q_H}{p},$$

Proof: The proof is straight-forward. Let us consider the case where the communication is with instance U_1^i ($\textsc{CaseL}$). The other case is equivalent ($\textsc{CaseR}$). Let $K_{U_1} = \mathrm{CDH}(X^\star/\mathsf{pw}_{U_1,1}, \overline{Y}^\star/\mathsf{pw}_{U_1,2})$ be the key associated with values $\mathsf{pw}_{U_1,1} = G_1(U_1, U_2, pw_{U_1})$ and $\mathsf{pw}_{U_1,2} = G_2(U_1, U_2, pw_{U_1}, R, X^\star)$. Since the query $(U_1, U_2, pw_{U_1}, R, X^\star)$ has not been asked to the G_2 oracle, both $\mathsf{pw}_{U_1,2}$ and $\mathsf{pw}_{U_2,2}$ can take any value in G. Thus, the possible values for K_{U_1} are also uniformly distributed in G and the probability that a H query contains the value K_{U_1} in it is exactly $1/p$. Therefore,

$$\begin{aligned}
\Pr[\, \textsc{AskH}_{10} \wedge \overline{\textsc{AskG2}} \wedge \textsc{CaseL} \,] &\leq \Pr\left[\, \textsc{AskH}_{10} \wedge \textsc{CaseL} \;\middle|\; \overline{\textsc{AskG2}} \,\right] \\
&\leq \sum_{i=1}^{q_H} \Pr\left[\, K_i' = K_{U_1} \;\middle|\; \overline{\textsc{AskG2}} \,\right] \\
&\leq \sum_{i=1}^{q_H} \frac{1}{p} \;=\; \frac{q_H}{p}.
\end{aligned}$$

Similarly, we have

$$\Pr[\, \textsc{AskH}_{10} \wedge \overline{\textsc{AskG2}} \wedge \textsc{CaseR} \,] \;\leq\; \frac{q_H}{p}.$$

Since $\textsc{AskH}_{10}$ is only defined for one of these cases, it follows that

$$\Pr[\, \textsc{AskH}_{10} \wedge \overline{\textsc{AskG2}} \,] \;\leq\; 2 \, \frac{q_H}{p}.$$

∎

Second, as the following two claims show, we do not need to consider cases in which there are two pairs of elements $(\mathsf{pw}_{1,1}, \mathsf{pw}_{1,2})$ and $(\mathsf{pw}_{2,1}, \mathsf{pw}_{2,2})$ outputted by the G_1 and G_2 oracles such that:

- the queries $(U_1, U_2, S, R, X^\star, Y^\star, \overline{X}^\star, \overline{Y}^\star, K_j = \mathrm{CDH}(X^\star/\mathsf{pw}_{j,1}, \overline{Y}^\star/\mathsf{pw}_{j,2}))$, for $j = 1, 2$, were asked of H and $(X^\star, \overline{Y}^\star, \overline{X}^\star, Y^\star) \in G^4$, $R \in \{0,1\}^{\ell_r}$, S, U_1, and U_2 be a set of values involved in the communication with an instance i of a participant U_1 in its role as an initiator; or

- the queries $(U_1, U_2, S, R, X^\star, Y^\star, \overline{X}^\star, \overline{Y}^\star, K_j = \mathrm{CDH}(X^\star/\mathsf{pw}_{j,1}, \overline{Y}^\star/\mathsf{pw}_{j,2}))$, for $j = 1, 2$, were asked of H and $(X^\star, \overline{Y}^\star, \overline{X}^\star, Y^\star) \in G^4$, $R \in \{0,1\}^{\ell_r}$, S, U_1, and U_2 be a set of values involved in the communication with an instance i of a participant U_2 in its role as a responder.

Claim A.7 Let $(X^\star, \overline{Y}^\star, \overline{X}^\star, Y^\star) \in G^4$, $R \in \{0,1\}^{\ell_r}$, S, U_1, and U_2 be a set of values involved in the communication with an instance i of a participant U_1 in its role as an initiator, where both U_1 and U_2 are honest and the latter is the intended partner. Let $\textsc{Coll}$ denote the event in which there exist two different pairs of elements $(\mathsf{pw}_{1,1}, \mathsf{pw}_{1,2})$ and $(\mathsf{pw}_{2,1}, \mathsf{pw}_{2,2})$ outputted by the G_1 and G_2 oracles such that the queries $(U_1, U_2, S, R, X^\star, Y^\star, \overline{X}^\star, \overline{Y}^\star, K_j = \mathrm{CDH}(X^\star/\mathsf{pw}_{j,1}, \overline{Y}^\star/\mathsf{pw}_{j,2}))$, for $j = 1, 2$, were asked of H. Then,

$$\Pr[\,\textsc{Coll}\,] \le q_{G_1}^2 \cdot q_{G_2}^2 \cdot q_G^2 \cdot \mathbf{Adv}_{\mathbb{G}}^{\mathrm{cdh}}(t + 3\tau_e) + \frac{q_{G_1} + q_{G_2}}{p},$$

q_{G_1}, q_{G_2}, and q_H are, respectively, the number of queries asked to the G_1, G_2, and H oracles.

Proof: The proof parallels the proof of Lemma 5 in [10]. Our goal is to use the event $\textsc{Coll}$ in the simulation of A to help us solve the CDH problem in $\mathbb{G}$. Let Q_1 and Q_2 be the inputs to CDH problem. Let us assume that Q_1 and Q_2 are different from 1 (this case is trivial).

We start running A simulating its oracles as in the current experiment except for the G_1 and G_2 oracles. These last two oracles are simulated as follows. In order to answer to a query (U_1, U_2, pw) to the G_1 oracle, we first pick an element $k_1 \in \mathsf{Z}_p^\star$ uniformly at random and set pw_1, the output of G_1, to $Q_1^{k_1}$. Similarly, a G_2 oracle query $(U_1, U_2, R_S, pw, X^\star)$ is answered by first picking an element $k_2 \in \mathsf{Z}_p^\star$ uniformly at random and setting pw_2, the output of G_2, to $Q_2^{k_i}$.

One can see that such change in the simulation is indistinguishable from Experiment $\mathbf{Exp}_9$, except when one of the outputs of the G_1 or G_2 oracles in the original experiment is 1. This event occurs with probability at most $\frac{q_{G_1}+q_{G_2}}{p}$. Everything else remains the same.

Next, we notice that $(X^\star, \overline{Y}^\star, \overline{X}^\star, Y^\star)$ being involved in the communication with an instance i of a participant U_1 in its role as an initiator implies that we simulated that instance. Hence, we know $x^\star$ such that $X^\star = g^{x^\star}$. As $(\mathsf{pw}_{1,1}, \mathsf{pw}_{1,2})$ and $(\mathsf{pw}_{2,1}, \mathsf{pw}_{2,2})$ were outputted by the G_1 and G_2 oracles, we also know $k_{1,1}$, $k_{1,2}$, $k_{2,1}$, $k_{2,2}$ in Z_p such that $\mathsf{pw}_{j,1} = Q_1^{k_{j,1}}$ and $\mathsf{pw}_{j,2} = Q_2^{k_{j,1}}$ for $j = 1, 2$. Then, in case K_1 and K_2 lies in Λ_H, we have

$$\begin{aligned}
K_j &= \mathrm{CDH}(X^\star/\mathsf{pw}_{j,1}, \overline{Y}^\star/\mathsf{pw}_{j,2}) \\
&= \mathrm{CDH}(X^\star \cdot Q_1^{k_{j,1}}, \overline{Y}^\star \cdot Q_2^{k_{j,2}}) \\
&= \mathrm{CDH}(X^\star, \overline{Y}^\star) \cdot \mathrm{CDH}(X^\star, Q_2^{k_{j,2}}) \cdot \mathrm{CDH}(Q_1^{k_{j,1}}, \overline{Y}^\star) \cdot \mathrm{CDH}(Q_1^{k_{j,1}}, Q_2^{k_{j,2}}) \\
&= \mathrm{CDH}(X^\star, \overline{Y}^\star) \cdot \mathrm{CDH}(X^\star, Q_2)^{k_{j,2}} \cdot \mathrm{CDH}(Q_1, \overline{Y}^\star)^{k_{j,1}} \cdot \mathrm{CDH}(Q_1, Q_2)^{k_{j,1}k_{j,2}} \\
&= \overline{Y}^{\star x^\star} \cdot Q_2^{x^\star k_{j,2}} \cdot \mathrm{CDH}(Q_1, \overline{Y}^\star)^{k_{j,1}} \cdot \mathrm{CDH}(Q_1, Q_2)^{k_{j,1}k_{j,2}}.
\end{aligned}$$

Let $Z_j = K_j \cdot \overline{Y}^{\star -x^\star} \cdot Q_2^{-x^\star k_{j,2}}$. It follows that

$$Z_1^{k_{2,1}} / Z_2^{k_{1,1}} \;=\; \mathrm{CDH}(Q_1, Q_2)^{k_{1,1} k_{2,1} (k_{1,2} - k_{2,2})},$$

and

$$\mathrm{CDH}(Q_1, Q_2) \;=\; \left(Z_1^{k_{2,1}} / Z_2^{k_{1,1}} \right)^u,$$

where u is the inverse of $k_{1,1} k_{2,1} (k_{1,2} - k_{2,2})$ in Z_p, guaranteed to exist because $\mathsf{pw}_{2,2} \neq \mathsf{pw}_{1,2}$. The claim follows easily by guessing the two H queries, the two G_1 queries, and the two G_2 queries. ▌

Claim A.8 Let $(X^\star, \overline{Y}^\star, \overline{X}^\star, Y^\star) \in G^4$, $R \in \{0,1\}^{\ell_r}$, S, U_1, and U_2 be a set of values involved in the communication with an instance i of a participant U_2 in its role as a responder, where both U_1 and U_2 are honest players and U_1 is the intended partner of U_2. Let COLL denote the event in which there exist two different pairs of elements $(\mathsf{pw}_{1,1}, \mathsf{pw}_{1,2})$ and $(\mathsf{pw}_{2,1}, \mathsf{pw}_{2,2})$ outputted by the G_1 and G_2 oracles such that the queries $(U_1, U_2, S, R, X^\star, Y^\star, \overline{X}^\star, \overline{Y}^\star, K_j = \mathrm{CDH}(Y^\star / \mathsf{pw}_{j,1}, \overline{X}^\star / \mathsf{pw}_{j,2}))$, for $j = 1, 2$, were asked of H. Then,

$$\Pr[\,\textsc{Coll}\,] \leq q_{G_1}^2 \cdot q_{G_2}^2 \cdot q_G^2 \cdot \mathbf{Adv}_{\mathbb{G}}^{\mathrm{cdh}}(t + 3\tau_e) + \frac{q_{G_1} + q_{G_2}}{p},$$

q_{G_1}, q_{G_2}, and q_H are, respectively, the number of queries asked to the G_1, G_2, and H oracles.

Proof: The proof of this claim is similar to that of Claim A.7 and, hence, skipped here. ▌

Let us now consider the event ASKH1$_{10}$, in which both U_1 and U_2 are honest players, U_2 is the intended partner of instance U_1^i, and the input query to the *SendClient* oracle for instance U_1^i matches the output a *SendServer* oracle S_k, whose input only partially comes from a simulated oracle. This corresponds to the case where on one side of the *SendServer* oracle we have an oracle instance and on the other side we have the adversary, playing the role of a honest user.

Let $X^\star$, $\overline{Y}^\star$, $\overline{X}^\star$, $Y^\star$, R, S, U_1, and U_2 be a set of values involved in the communication with an instance U_1^i in its role as an initiator, where U_2 is the intended partner (the symmetric case, in which the communication is with the responder, is similar). Note that since online guessing attacks are always possible in this scenario, there is a non-negligible probability that event $AskH1_{10}$ occurs.

If the event COLL, defined in Claim A.7 and in Claim A.8, does not happen, then for each $(X^\star, \overline{Y}^\star, \overline{X}^\star, Y^\star, R, S, U_1, U_2$ set of values involved in the communication with an instance U_1^i in its role as an initiator, where U_2 is the honest intended partner, there is at most one pair of values $(\mathsf{pw}_1, \mathsf{pw}_2)$ such that $K_j = \mathrm{CDH}(X^\star / \mathsf{pw}_1, \overline{Y}^\star / \mathsf{pw}_2)$ lies in the list Λ_H. Since collisions in the output of G_1 and G_2 oracles were removed, the latter implies that there is one unique pw such that $\mathsf{pw}_1 = G_1(U_1, U_2, pw)$ and $\mathsf{pw}_2 = G_2(U_1, U_2, R, pw, X^\star)$. Since we only choose pw at the very end of the simulation, the probability that the latter collides with the ones chosen by the adversary is at most $q_{\mathrm{start}} / |\mathcal{D}|$. In other words,

$$\Pr\left[\,\textsc{AskH1}_{10} \wedge \textsc{AskG2} \mid \overline{\textsc{Coll}}\,\right] \;\leq\; q_{\mathrm{start}} / |\mathcal{D}|\,,$$

and

$$\begin{aligned}
\Pr[\,\textsc{AskH1}_{10} \wedge \textsc{AskG2}\,] \;&\leq\; \Pr\left[\,\textsc{AskH1}_{10} \wedge \textsc{AskG2} \mid \overline{\textsc{Coll}}\,\right] \cdot \Pr[\,\overline{\textsc{Coll}}\,] + \\
&\;\; \Pr\left[\,\textsc{AskH1}_{10} \wedge \textsc{AskG2} \mid \textsc{Coll}\,\right] \cdot \Pr[\,\textsc{Coll}\,] \\
&\leq\; \Pr\left[\,\textsc{AskH1}_{10} \wedge \textsc{AskG2} \mid \overline{\textsc{Coll}}\,\right] + \Pr[\,\textsc{Coll}\,] \\
&\leq\; q_{\mathrm{start}} / |\mathcal{D}| + \Pr[\,\textsc{Coll}\,]\,.
\end{aligned}$$

Next, let us consider the event $\textsc{AskH2}_{10}$, in which U_1 and U_2 are honest players, U_2 is the intended partner of instance U_1^i, and the input query $(S, U_2, R, \overline{X})$ to the *SendClient* oracle with respect to instance U_1^i in its role as an initiator (the responder case is similar) was generated by the adversary. Like in the previous case, if the event $\textsc{Coll}$ does not happen, then for each $(X^\star, \overline{Y}^\star, \overline{X}^\star, Y^\star, R, S, U_1, U_2$ set of values involved in the communication with an instance U_1^i in its role as an initiator, where U_2 is the honest intended partner, there is at most one pair of values $(\mathsf{pw}_1, \mathsf{pw}_2)$ such that $K_j = \mathrm{CDH}(X^\star/\mathsf{pw}_1, \overline{Y}^\star/\mathsf{pw}_2)$ lies in the list Λ_H. Using a similar argument, we have

$$\Pr\left[\ \textsc{AskH2}_{10} \wedge \textsc{AskG2}\ \middle|\ \overline{\textsc{Coll}}\ \right] \ \leq\ q_{\mathrm{start}}/|\mathcal{D}|\ ,$$

and

$$\Pr[\,\textsc{AskH2}_{10} \wedge \textsc{AskG2}\,] \ \leq\ q_{\mathrm{start}}/|\mathcal{D}| + \Pr[\,\textsc{Coll}\,]\ .$$

Finally, we can compute the probability of the event $\textsc{AskH}_{10}$ as follows.

$$\begin{array}{rcl}
\Pr[\,\textsc{AskH}_{10}\,] & \leq & \Pr[\,\textsc{AskH}_{10} \wedge \textsc{AskG2}\,] \cdot \Pr[\,\textsc{AskG2}\,] + \\
& & \Pr[\,\textsc{AskH}_{10} \wedge \overline{\textsc{AskG2}}\,] \cdot \Pr[\,\overline{\textsc{AskG2}}\,] \\
& \leq & \Pr[\,\textsc{AskH}_{10} \wedge \textsc{AskG2}\,] + \Pr[\,\textsc{AskH}_{10} \wedge \overline{\textsc{AskG2}}\,] \\
& \leq & \Pr[\,\textsc{AskH1}_{10} \wedge \textsc{AskG2}\,] \cdot \Pr[\,\textsc{Case1}\,] + \\
& & \Pr[\,\textsc{AskH2}_{10} \wedge \textsc{AskG2}\,] \cdot \Pr[\,\textsc{Case2}\,] + \Pr[\,\textsc{AskH}_{10} \wedge \overline{\textsc{AskG2}}\,] \\
& \leq & (q_{\mathrm{start}}/|\mathcal{D}| + \Pr[\,\textsc{Coll}\,]) \cdot \Pr[\,\textsc{Case1}\,] + \\
& & (q_{\mathrm{start}}/|\mathcal{D}| + \Pr[\,\textsc{Coll}\,]) \cdot \Pr[\,\textsc{Case2}\,] + \Pr[\,\textsc{AskH}_{10} \wedge \overline{\textsc{AskG2}}\,] \\
& \leq & q_{\mathrm{start}}/|\mathcal{D}| + \Pr[\,\textsc{Coll}\,] + \Pr[\,\textsc{AskH}_{10} \wedge \overline{\textsc{AskG2}}\,] \\
& \leq & q_{\mathrm{start}}/|\mathcal{D}| + q_{G_1}^2 \cdot q_{G_2}^2 \cdot q_H^2 \cdot \mathbf{Adv}_{\mathbb{G}}^{\mathrm{cdh}}(t + 3\tau_e) + \dfrac{q_{G_1} + q_{G_2}}{p} + 2\ \dfrac{q_H}{p}\ .
\end{array}$$

▌ ▌

B Proof of lemmas

B.1 The splitting lemma

For simplicity, we reproduce here the splitting lemma presented in [23].

Lemma B.1 [Splitting Lemma] Let $A \subset X \times Y$ such that $\Pr[(x, y) \in A] \geq \epsilon$. For any $\alpha < \epsilon$, define

$$B = \left\{ (x, y) \in X \times Y \ \middle|\ \Pr_{y' \in Y}[(x, y') \in A] \geq \epsilon - \alpha \right\} \ \text{ and } \ \bar{B} = (X \times Y) \backslash B,$$

then the following statements hold:

(i) $\Pr[B] \geq \alpha$

(ii) $\forall (x, y) \in B, \Pr_{y' \in Y}[(x, y') \in A] \geq \epsilon - \alpha$.

Proof: In order to prove statement (i), we argue by contradiction. Assume that $\Pr[B] < \alpha$. Then

$$\epsilon \ \leq\ \Pr[B] \cdot \Pr[A \,|\, B] + \Pr[\bar{B}] \cdot \Pr[A \,|\, \bar{B}] \ <\ \alpha \cdot 1 + 1 \cdot (\epsilon - \alpha) \ =\ \epsilon.$$

This implies a contradiction, hence the result. Statement (ii) is a straightforward consequence of the definition. ▌

B.2 Proof of Lemma 3.5

Let $\mathcal{R}$ represent the set of all random functions from $\{1, \ldots, n\}$ to G and let $\mathcal{R}[(k_0, U_0), \ldots, (k_s, U_s)]$ denote the subset of $\mathcal{R}$ such that k_i is mapped to U_i, for $i = 0, \ldots, s$. Let $\mathcal{A}$ be an adversary against the password-based chosen-basis decisional Diffie-Hellman 1 assumption with an advantage greater than $2/n + \varepsilon$. By the definition of $\mathbf{Adv}^{\mathrm{pcddh1}}_{\mathbb{G},n}(\mathcal{A}, \mathcal{P}, X, k, r_0, r_1)$, we have

$$\Pr[\, \mathbf{Exp}^{\mathrm{pcddh1}}_{\mathbb{G},n,b}(\mathcal{A}, \mathcal{P}, X, k, r_0, r_1) = b \,] \geq 1/2 + 1/n + \varepsilon/2 \,,$$

where the probability space is on $\Omega_0 = \{(\omega_1, \omega_2, \mathcal{P}, X, k, b, r_0, r_1)\}$ (ω_1 and ω_2 are the random tapes of $\mathcal{A}$ in the first step and second steps, respectively).

By applying the splitting lemma on the product probability space $\Omega'_1 \times \Omega_1$, where $\Omega'_1 = \{(\omega_1, \mathcal{P}, X, r_0, r_1)\}$ and $\Omega_1 = \{(\omega_2, k, b)\}$, one can show that there exists a subset S_1 of Ω'_1 with probability measure greater than $\varepsilon/4$ such that, for any $(\omega_1, \mathcal{P}, X, r_0, r_1) \in S_1$,

$$\Pr_{\Omega_1}\left[\mathbf{Exp}^{\mathrm{pcddh1}}_{\mathbb{G},b}(\mathcal{A}, \mathcal{P}, X, k, r_0, r_1 \;;\; \omega_1) = b\right] \geq 1/2 + 1/n + \varepsilon/4 \,,$$

where the probability space is now on $\Omega_1 = \{(\omega_2, k, b)\}$. In this game, since $(\omega_1, \mathcal{P}, X)$ is fixed, so is the output (Y, s) at the end of the first stage. Furthermore, since r_0 is also fixed, then so is Y'.

If we apply the splitting lemma once again on the product probability space $\Omega'_2 \times \Omega_2$, where $\Omega'_2 = \{k\}$ and $\Omega_2 = \{(\omega_2, b)\}$, one can show that there exists a subset $S_2(\omega_1, \mathcal{P}, X, r_0, r_1)$ of Ω'_2 with probability measure greater than $1/n + \varepsilon/8 > 1/n$ such that, for any k in $S_2(\omega_1, \mathcal{P}, X, r_0, r_1)$ (if $(\omega_1, \mathcal{P}, X, r_0, r_1)$ is in S_1),

$$\Pr_{\Omega_2}\left[\mathbf{Exp}^{\mathrm{pcddh1}}_{\mathbb{G},b}(\mathcal{A}, \mathcal{P}, X, k, r_0, r_1 \;;\; \omega_1) = b\right] \geq 1/2 + \varepsilon/8 \,,$$

where the probability space is now on $\Omega_2 = \{(\omega_2, b)\}$.

Here, one sees that $S_2(\omega_1, \mathcal{P}, X, r_0, r_1)$ is a subset of $\{1, \ldots, n\}$ of measure strictly greater than $1/n$. Hence, it is at least $2/n$. Therefore, there exist two values k_0 and k_1, and thus U_0 and U_1, for which this adversary can decide b with advantage greater than $\varepsilon/4$ in the following experiment, for $i = 0, 1$:

```
**Experiment** $\mathbf{Exp}^{\mathrm{pcddh1}}_{\mathbb{G},b}(\mathcal{A}, \mathcal{P}, X, k_i, r_0, r_1 \;;\; \omega_1)$
    $(Y, s) \leftarrow \mathcal{A}^{\mathcal{P}}(\mathsf{find}, X \;;\; \omega_1)$
    $U_i \leftarrow \mathcal{P}(k_i)$
    $X \leftarrow (X/U_i)^{r_b} \;;\; K \leftarrow \mathrm{CDH}(X/U_i, Y)^{r_b}$
    $Y' \leftarrow Y^{r_0}$
    $d \leftarrow \mathcal{A}(\mathsf{guess}, s, X, K, Y', k_i)$
    **return** $d$
```

In other words, if one randomly choose $k_0, k_1 \overset{R}{\leftarrow} \{1, \ldots, n\}$, $X \overset{R}{\leftarrow} G$, $r_0, r_1 \overset{R}{\leftarrow} \mathsf{Z}_p$, $U_i \overset{R}{\leftarrow} G$, for $i = 0, 1$ and $\mathcal{P} \overset{R}{\leftarrow} \mathcal{R}[(k_0, U_0), (k_1, U_1)]$, as well as a tape ω_1, with probability greater than $\varepsilon/4n^2$, the above adversary can decide b with advantage greater than $\varepsilon/4$, for both $i = 0$ and $i = 1$.

Let us now use the splitting lemma one last time on the product probability space $\{(\omega_1, k_0, k_1, r_0, r_1, \mathcal{P}')\} \times \{(X, U_0, U_1)\}$, where $\mathcal{P}'$ is randomly drawn from $\mathcal{R}[(k_0, 1), (k_1, g)]$, and when U_0 and U_1 are defined, we set $\mathcal{P}$ to be equal to $\mathcal{P}'$ except that $\mathcal{P}(k_i)$ is set to U_i. One can thus show that there exists a subset S of $\{(\omega_1, k_0, k_1, r_0, r_1, \mathcal{P}')\}$ with probability measure greater than $\varepsilon/8n^2$ such that, for any $(\omega_1, k_0, k_1, r_0, r_1, \mathcal{P}')$ in S, with probability greater than $\varepsilon/8n^2$ over (X, U_0, U_1), the above adversary can decide b with advantage greater than $\varepsilon/4$, for both $i = 0$ and $i = 1$.

We now define an adversary against the CDDH1 problem as follows, for randomly chosen $(\omega_1, k_0, k_1, r_0, r_1, \mathcal{P}')$, that we now assume to be in S:

```
Algorithm $B(\mathsf{find}, U, V, X)$
    $\mathcal{P} \leftarrow \mathcal{P}'$ ; $\mathcal{P}(k_0) \leftarrow U$ ; $\mathcal{P}(k_1) \leftarrow V$
    $(Y, s) \leftarrow \mathcal{A}^{\mathcal{P}}(\mathsf{find}, X)$
    $\widetilde{s} \leftarrow (U, V, s)$
    return $(Y, \widetilde{s})$
```

```
Algorithm $B(\mathsf{guess}, \widetilde{s}, X_0, K_0, X_1, K_1, Y')$
    parse $\widetilde{s}$ as $(U, V, s)$
    $d_0 \leftarrow \mathcal{A}^{\mathcal{P}}(\mathsf{guess}, s, X_0, Y', K_0)$
    $d_1 \leftarrow \mathcal{A}^{\mathcal{P}}(\mathsf{guess}, s, X_1, Y', K_1)$
    return $d = d_0 \oplus d_1$
```

If one denotes by $\varepsilon_{i,b}$ the probability that $d_i = 1$ when $b_i = b$ in the game with experiment $\mathbf{Exp}^{\mathrm{pcddh1}}_{\mathbb{G},n,b}(\mathcal{A}, \mathcal{P}, X, k_i, r_0, r_1 \ ; \ \omega_1)$, since the two games are independent (with independent bits b), one gets

$$\begin{aligned}
&\Pr[d = 1 | b = 1] \\
&= \Pr[d_0 = 1 \wedge d_1 = 0 | (b_0 = 1 \wedge b_1 = 0) \vee (b_0 = 0 \wedge b_1 = 1)] \\
&\quad + \Pr[d_0 = 0 \wedge d_1 = 1 | (b_0 = 1 \wedge b_1 = 0) \vee (b_0 = 0 \wedge b_1 = 1)] \\
&= 2 \times \Pr[d_0 = 1 \wedge d_1 = 0 \wedge ((b_0 = 1 \wedge b_1 = 0) \vee (b_0 = 0 \wedge b_1 = 1))] \\
&\quad + 2 \times \Pr[d_0 = 0 \wedge d_1 = 1 \wedge ((b_0 = 1 \wedge b_1 = 0) \vee (b_0 = 0 \wedge b_1 = 1))] \\
&= 2 \Pr[d_0 = 1 \wedge d_1 = 0 \wedge b_0 = 1 \wedge b_1 = 0] + 2 \Pr[d_0 = 1 \wedge d_1 = 0 \wedge b_0 = 0 \wedge b_1 = 1] \\
&\quad + 2 \Pr[d_0 = 0 \wedge d_1 = 1 \wedge b_0 = 1 \wedge b_1 = 0] + 2 \Pr[d_0 = 0 \wedge d_1 = 1 \wedge b_0 = 0 \wedge b_1 = 1] \\
&= 2 \Pr[d_0 = 1 \wedge b_0 = 1] \Pr[d_1 = 0 \wedge b_1 = 0] + 2 \Pr[d_0 = 1 \wedge b_0 = 0] \Pr[d_1 = 0 \wedge b_1 = 1] \\
&\quad + 2 \Pr[d_0 = 0 \wedge b_0 = 1] \Pr[d_1 = 1 \wedge b_1 = 0] + 2 \Pr[d_0 = 0 \wedge b_0 = 0] \Pr[d_1 = 1 \wedge b_1 = 1] \\
&= \Pr[d_0 = 1 | b_0 = 1] \Pr[d_1 = 0 | b_1 = 0] + \Pr[d_0 = 1 | b_0 = 0] \Pr[d_1 = 0 | b_1 = 1] \\
&\quad + \Pr[d_0 = 0 | b_0 = 1] \Pr[d_1 = 1 | b_1 = 0] + \Pr[d_0 = 0 | b_0 = 0] \Pr[d_1 = 1 | b_1 = 1] \\
&= \varepsilon_{0,1}(1 - \varepsilon_{1,0}) + \varepsilon_{0,0}(1 - \varepsilon_{1,1}) + (1 - \varepsilon_{0,1})\varepsilon_{1,0} + (1 - \varepsilon_{0,0})\varepsilon_{1,1} \\
&= \varepsilon_{0,0} + \varepsilon_{0,1} + \varepsilon_{1,0} + \varepsilon_{1,1} - 2\varepsilon_{0,1}\varepsilon_{1,0} - 2\varepsilon_{0,0}\varepsilon_{1,1}
\end{aligned}$$

The same way, one gets

$$\Pr[d = 1 | b = 0] = \varepsilon_{0,0} + \varepsilon_{0,1} + \varepsilon_{1,0} + \varepsilon_{1,1} - 2\varepsilon_{0,0}\varepsilon_{1,0} - 2\varepsilon_{0,1}\varepsilon_{1,1}.$$

Then, the advantage is

$$\begin{aligned}
\Pr[d = 1 | b = 1] - \Pr[d = 1 | b = 0] &= 2(\varepsilon_{0,0}\varepsilon_{1,0} + \varepsilon_{0,1}\varepsilon_{1,1} - \varepsilon_{0,1}\varepsilon_{1,0} - \varepsilon_{0,0}\varepsilon_{1,1}) \\
&= 2\varepsilon_{0,0}(\varepsilon_{1,0} - \varepsilon_{1,1}) + 2\varepsilon_{0,1}(\varepsilon_{1,1} - \varepsilon_{1,0}) \\
&= 2(\varepsilon_{0,1} - \varepsilon_{0,0})(\varepsilon_{1,1} - \varepsilon_{0,1}) \\
&\geq 2 \times \varepsilon/4 \times \varepsilon/4 = \varepsilon^2/8 \ .
\end{aligned}$$

∎

B.3 Proof of Lemma 3.6

The proof of this lemma is similar to that of Lemma 3.5. Let $\mathcal{A}$ be an adversary against the password-based chosen-basis decisional Diffie-Hellman 1 assumption with an advantage greater than $\varepsilon \geq 16/n$. By the definition of $\mathbf{Adv}^{\mathrm{pcddh1}}_{\mathbb{G},n}(\mathcal{A}, \mathcal{P}, X, k, r_0, r_1)$, we have

$$\Pr[\,\mathbf{Exp}^{\mathrm{pcddh1}}_{\mathbb{G},n,b}(\mathcal{A}, \mathcal{P}, X, k, r_0, r_1) = b\,] \geq 1/2 + \varepsilon/2 \ ,$$

where the probability space is on $\Omega_0 = \{(\omega_1, \omega_2, \mathcal{P}, X, k, b, r_0, r_1)\}$ (ω_1 and ω_2 are the random tapes of $\mathcal{A}$ in the first step and second steps, respectively).

By applying the splitting lemma on the product probability space $\Omega_1' \times \Omega_1$, where $\Omega_1' = \{(\omega_1, \mathcal{P}, X, r_0, r_1)\}$ and $\Omega_1 = \{(\omega_2, k, b)\}$, one can show that there exists a subset S_1 of Ω_1' with probability measure greater than $\varepsilon/4$ such that, for any $(\omega_1, \mathcal{P}, X, r_0, r_1) \in S_1$,

$$\Pr_{\Omega_1}\left[\mathbf{Exp}^{\text{pcddh1}}_{\mathbb{G},b}(\mathcal{A}, \mathcal{P}, X, k, r_0, r_1\ ;\ \omega_1) = b\right] \geq 1/2 + \varepsilon/4\ ,$$

where the probability space is now on $\Omega_1 = \{(\omega_2, k, b)\}$. In this game, since $(\omega_1, \mathcal{P}, X)$ is fixed, so is the output (Y, s) at the end of the first stage. Furthermore, since r_0 is also fixed, then so is Y'.

If we apply the splitting lemma once again on the product probability space $\Omega_2' \times \Omega_2$, where $\Omega_2' = \{k\}$ and $\Omega_2 = \{(\omega_2, b)\}$, one can show that there exists a subset $S_2(\omega_1, \mathcal{P}, X, r_0, r_1)$ of Ω_2' with probability measure greater than $\varepsilon/8 \geq 2/n$ such that, for any k in $S_2(\omega_1, \mathcal{P}, X, r_0, r_1)$ (if $(\omega_1, \mathcal{P}, X, r_0, r_1)$ is in S_1),

$$\Pr_{\Omega_2}\left[\mathbf{Exp}^{\text{pcddh1}}_{\mathbb{G},b}(\mathcal{A}, \mathcal{P}, X, k, r_0, r_1\ ;\ \omega_1) = b\right] \geq 1/2 + \varepsilon/8\ ,$$

where the probability space is now on $\Omega_2 = \{(\omega_2, b)\}$.

Here, one sees that $S_2(\omega_1, \mathcal{P}, X, r_0, r_1)$ is a subset of $\{1, \ldots, n\}$ of size at least $2/n$. Therefore, there exist two values k_0 and k_1, and thus U_0 and U_1, for which this adversary can decide b with advantage greater than $\varepsilon/4$ in the following experiment, for $i = 0, 1$:

```
Experiment $\mathbf{Exp}^{\text{pcddh1}}_{\mathbb{G},b}(\mathcal{A}, \mathcal{P}, X, k_i, r_0, r_1\ ;\ \omega_1)$
    $(Y, s) \leftarrow \mathcal{A}^{\mathcal{P}}(\mathsf{find}, X\ ;\ \omega_1)$
    $U_i \leftarrow \mathcal{P}(k_i)$
    $X \leftarrow (X/U_i)^{r_b}\ ;\ K \leftarrow \text{CDH}(X/U_i, Y)^{r_b}$
    $Y' \leftarrow Y^{r_0}$
    $d \leftarrow \mathcal{A}(\mathsf{guess}, s, X, K, Y', k_i)$
    return $d$
```

In other words, if one randomly choose $k_0, k_1 \overset{R}{\leftarrow} \{1, \ldots, n\}$, $X \overset{R}{\leftarrow} G$, $r_0, r_1 \overset{R}{\leftarrow} \mathsf{Z}_p$, $U_i \overset{R}{\leftarrow} G$, for $i = 0, 1$ and $\mathcal{P} \overset{R}{\leftarrow} \mathcal{R}[(k_0, U_0), (k_1, U_1)]$, as well as a tape ω_1, with probability greater than $\varepsilon/4 \cdot \varepsilon/8 \cdot (\varepsilon/8 - 1/n) \geq \varepsilon^3/2^9$, the above adversary can decide b with advantage greater than $\varepsilon/4$, for both $i = 0$ and $i = 1$.

Let us now use the splitting lemma one last time on the product probability space $\{(\omega_1, k_0, k_1, r_0, r_1, \mathcal{P}')\} \times \{(X, U_0, U_1)\}$, where $\mathcal{P}'$ is randomly drawn from $\mathcal{R}[(k_0, 1), (k_1, g)]$, and when U_0 and U_1 are defined, we set $\mathcal{P}$ to be equal to $\mathcal{P}'$ except that $\mathcal{P}(k_i)$ is set to U_i. One can thus show that there exists a subset S of $\{(\omega_1, k_0, k_1, r_0, r_1, \mathcal{P}')\}$ with probability measure greater than $\varepsilon^3/2^{10}$ such that, for any $(\omega_1, k_0, k_1, r_0, r_1, \mathcal{P}')$ in S, with probability greater than $\varepsilon^3/2^{10}$ over (X, U_0, U_1), the above adversary can decide b with advantage greater than $\varepsilon/4$, for both $i = 0$ and $i = 1$.

Using the above facts, one can then build an adversary for for CDDH1 problem exactly as in the proof of Lemma 3.5. Moreover, by using similar arguments, one can also show that the advantage of this adversary would be at least $\varepsilon^2/8$. The bound claimed in Lemma 3.6 then easily follows. ▌

B.4 Proof of Lemma 3.9

Let us assume that D is not a good distinguisher:

$$\Pr_{b,x}[E^b(D, x) = b] \leq \frac{1}{2} + \frac{\alpha}{2},$$

for $\alpha \leq \mu\epsilon/2$. Then

$$\begin{aligned}
\frac{1}{2}+\frac{\alpha}{2} &\geq \Pr_{b,x}[E^b(D,x)=b] \\
&\geq \Pr_{b,x}[E^b(D,x)=b \wedge x \in S'] + \Pr_{b,x}[E^b(D,x)=b \wedge x \notin S'] \\
&\geq \left(\frac{1}{2}+\frac{\epsilon}{2}\right)\times\mu + \Pr_{b,x}[E^b(D,x)=b \mid x \notin S'] \times (1-\mu) \\
&\geq \left(\frac{1}{2}+\frac{\epsilon}{2}\right)\times\mu + \left(1-\Pr_{b,x}[E^b(D,x)\neq b \mid x \notin S']\right) \times (1-\mu) \\
&\geq \frac{\mu}{2}+\frac{\mu\epsilon}{2}+1-\mu-\Pr_{b,x}[E^b(D,x)\neq b \mid x \notin S'] \times (1-\mu) \\
&\geq 1-\frac{\mu}{2}+\frac{\mu\epsilon}{2}-\Pr_{b,x}[E^b(D,x)\neq b \mid x \notin S'] \times (1-\mu) \\
&\geq \frac{1}{2}+\frac{\mu\epsilon}{2}+\left(\frac{1}{2}-\Pr_{b,x}[E^b(D,x)\neq b \mid x \notin S']\right) \times (1-\mu) \\
\frac{\alpha-\mu\epsilon}{2} &\geq \left(\frac{1}{2}-\Pr_{b,x}[E^b(D,x)\neq b \mid x \notin S']\right) \times (1-\mu)
\end{aligned}$$

As a consequence,

$$\Pr_{b,x}[E^b(D,x)\neq b \mid x \notin S'] \geq \frac{1}{2}+\frac{\mu\epsilon-\alpha}{2(1-\mu)} \geq \frac{1}{2}+\frac{\mu\epsilon-\alpha}{2} \geq \frac{1}{2}+\frac{\mu\epsilon}{4}$$

■