An extended abstract of this work appears in *Topics in Cryptology – CT-RSA 2005*
(14 – 18 February 2005, San Francisco, CA)
A. J. Menezes Ed., Springer-Verlag, LNCS 3376, pages 191–208

Simple Password-Based Encrypted Key Exchange Protocols

Michel Abdalla and David Pointcheval

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

Abstract. Password-based encrypted key exchange are protocols that are designed to provide pair of users communicating over an unreliable channel with a secure session key even when the secret key or password shared between two users is drawn from a small set of values. In this paper, we present two simple password-based encrypted key exchange protocols based on that of Bellovin and Merritt. While one protocol is more suitable to scenarios in which the password is shared across several servers, the other enjoys better security properties. Both protocols are as efficient, if not better, as any of the existing encrypted key exchange protocols in the literature, and yet they only require a single random oracle instance. The proof of security for both protocols is in the random oracle model and based on hardness of the computational Diffie-Hellman problem. However, some of the techniques that we use are quite different from the usual ones and make use of new variants of the Diffie-Hellman problem, which are of independent interest. We also provide concrete relations between the new variants and the standard Diffie-Hellman problem.

Keywords. Password, encrypted key exchange, Diffie-Hellman assumptions.

1 Introduction

Background. Keys exchange protocols are cryptographic primitives used to provide a pair of users communicating over a public unreliable channel with a secure session key. In practice, one can find several flavors of key exchange protocols, each with its own benefits and drawbacks. An example of a popular one is the SIGMA protocol [18] used as the basis for the signature-based modes of the Internet Key Exchange (IKE) protocol. The setting in which we are interested in this paper is the 2-party symmetric one, in which every pair of users share a secret key. In particular, we consider the scenario in which the secret key is a password.

Password-based key exchange. Password-based key exchange protocols assume a more realistic scenario in which secret keys are not uniformly distributed over a large space, but rather chosen from a small set of possible values (a four-digit pin, for example). They also seem more convenient since human-memorable passwords are simpler to use than, for example, having additional cryptographic devices capable of storing high-entropy secret keys. The vast majority of protocols found in practice do not account, however, for such scenario and are often subject to so-called *dictionary* attacks. Dictionary attacks are attacks in which an adversary tries to break the security of a scheme by a brute-force method, in which it tries all possible combinations of secret keys in a given small set of values (i.e., the dictionary). Even though these attacks are not very effective in the case of high-entropy keys, they can be very damaging when the secret key is a password since the attacker has a non-negligible chance of winning.

To address this problem, several protocols have been designed to be secure even when the secret key is a password. The goal of these protocols is to restrict the adversary's success to on-line guessing attacks only. In these attacks, the adversary must be present and interact with the system in order to be able to verify whether its guess is correct. The security in these systems usually relies on a policy of invalidating or blocking the use of a password if a certain number of failed attempts has occurred.

Encrypted key exchange. The seminal work in the area of password-based key exchange is the encrypted key exchange (EKE) protocol of Bellovin and Merritt [7]. In their protocol, two users execute an encrypted version of the Diffie-Hellman key exchange protocol, in which each flow is encrypted using the password shared between these two users as the symmetric key. Due to the simplicity of their protocol, several other protocols were proposed in the literature based on it [6, 9, 10, 17, 20], each with its own instantiation of the encryption function. Our protocol is also a variation of their EKE protocol.

Minimizing the use of random oracles. One of our main goals is to provide schemes that are simple and efficient, but relying as little as possible on random oracles. Ideally, one would want to completely eliminate the need of random oracles as done in the KOY protocol [16]. However, such protocols tend to be less efficient than those based on the EKE protocol of Bellovin and Merritt [7].

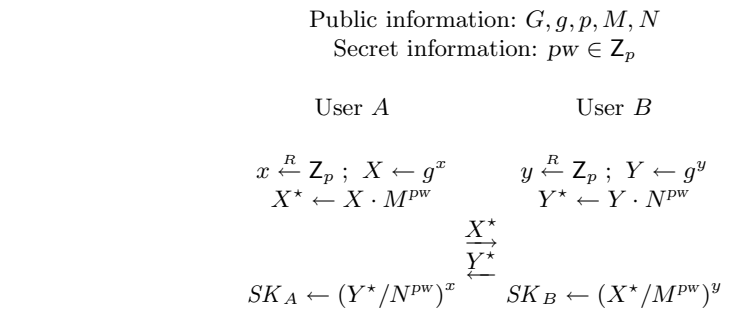

Fig. 1. An insecure password-based key exchange protocol.

To understand the difficulties involved in the design of protocols with few random oracles, let us consider the extreme case of the protocol in Figure 1 in which no random oracles are used. Despite being secure against passive attacks, this protocol can be easily broken by an active adversary performing a man-in-the-middle attack. Such an adversary can easily create two different sessions whose session keys are related in a predictable manner. For instance, an adversary can do so by multiplying $X^\star$ by g^r for a known value r. The relation between the underlying session keys SK_A and SK_B is $SK_B = SK_A \cdot Y^r$. Hence, if the adversary learns the value of these two keys, it can perform an off-line dictionary attack using $Y = (SK_B / SK_A)^{-r}$ and $Y^\star$ to recover the password. Moreover, since the adversary can use arbitrary values for r, we cannot detect such attacks.

Protecting against related key attacks. In order to fix the problem in the protocol presented in Figure 1 and prevent the adversary from altering the messages, one may be tempted to use message authentication code (MAC) algorithms for key derivation (e.g., by making the session key equal to $\mathsf{MAC}_{SK_A}(A, B, X^\star, Y^\star, 0)$) or key confirmation (e.g., by computing tags via $\mathsf{MAC}_{SK_A}(A, B, X^\star, Y^\star, 1)$). In fact, this is the approach used by Kobara and Imai in the construction of their password-authenticated key exchange protocol [17]. Unfortunately, this approach does not quite work.

Let us now explain the main problems with using MACs. First, the standard notion of security for MACs does not imply security against related key attacks. Hence, new and stronger security notions are required. Second, such new security notions may have to consider adversaries which are given access to a related-key tag-generation oracle. These are oracles that are capable of generating tags on messages under related keys, where the related key function is also passed

as a parameter. This is actually the approach used in [17]. However, it is not clear whether such MACs can even be built. Such security notion, for instance, may completely rule out the possibility of using block-cipher-based MAC algorithms since similar security requirements in the context of block ciphers are known to be impossible to achieve [2]. Perhaps, hash-based MAC algorithms may be able to meet these goals, but that does not seem likely without resorting to random oracles, which would defeat the purpose of using MACs in the first place.

Simple constructions. In this paper, we deal with the problem of related-key attacks by using a single instance of a random oracle in the key derivation process. We present two simple constructions, whose only difference to one another is the presence of the password in the key derivation function. The presence of the password in the key derivation function is an important aspect, for example, when one wants to consider extensions to the distributed case, where each server only holds a share of password (see [11]).

Surprisingly, the techniques that we use to prove the security of these two constructions are quite different and so are the exact security results. While we are able to provide a tight security reduction for the scheme which includes the password in the key derivation phase, the same cannot be said about the other scheme, for which we can only prove its security in the non-concurrent scenario. However, the techniques that we use to prove the security of the latter are quite interesting and make use of new variants of the Diffie-Hellman problem.

New Diffie-Hellman assumptions. The new variants of the Diffie-Hellman problem that we introduce are called 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 particular 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, we prove that this is *not* the case by providing concrete reductions to the computational Diffie-Hellman problem.

Contributions. In this paper, we address the issue of constructing efficient password-based encrypted key exchange protocols. Our main contributions are as follows.

Simple and efficient constructions in random oracle model. In this paper, we propose two new password-based encrypted key exchange protocols, called SPAKE1 and SPAKE2, both of which can be proven secure based on the hardness of the computational Diffie-Hellman problem in the random oracle model. Both protocols are comparable in efficiency to any of the existing EKE protocols, if not more efficient, and they only require one random oracle instance. This is contrast with existing EKE constructions, which require either a larger number of random oracle instances or additional ideal models, such as the ideal cipher model. Moreover, neither SPAKE1 nor SPAKE2 requires full domain hash functions or ideal ciphers onto a group, which are hard to implement efficiently. While one protocol is more suitable to scenarios in which the password is shared across several servers, the other enjoys better security properties.

New Diffie-Hellman assumptions. In proving the security of our protocols, we make use of new variations of the computational Diffie-Hellman assumption, called chosen-basis computational Diffie-Hellman assumptions. These new assumptions are of independent interest and we do expect to find new applications for it other than the ones in this paper. Reductions between the problems underlying the new assumptions and the standard computational Diffie-Hellman assumption are also provided.

Related work. Password-based authenticated key exchange has been extensively studied in the last few years [3, 8–10, 12–14, 16, 19, 21, 11] with the majority of them being submitted for

inclusion in the IEEE P1363.2 standard [15], a standard dealing with the issues of password-authenticated key agreement (e.g. EKE) and password-authenticated key retrieval. With the exception of [12, 13, 16], all of these protocols are only proven secure in the random oracle model.

Perhaps, the related work that is closest to ours is the pretty-simple password-authenticated key exchange protocol of Kobara and Imai [17], whose proof of security is claimed to be in the "standard" model. Their protocol consists of EKE phase that is similar to the one used in our protocols followed by an authentication phase based on message authentication code (MAC) algorithms. However, the security model which they use is different from the standard one and hence their result only applies to their specific model. Moreover, as we pointed out above, their protocol needs a stronger security notion for the MAC algorithm and it is not clear whether such MACs can be built without resorting to random oracles, which would contradict their claims.

Organization. In Section 2, we recall the security model for password-based authenticated key exchange. Next, in Section 3, we present our new variants of the Diffie-Hellman problem and their relations to the computational Diffie-Hellman problem. Section 4 then introduces the first of our password-based encrypted key exchange protocols, called SPAKE1, along with its proof of security. SPAKE1 is in fact based on one of the variants of the Diffie-Hellman problem introduced in Section 3. Our second protocol, SPAKE2, is then presented in Section 4 along with its security claims. In the appendix, we present proofs for several lemmas in Section 3 as well as the proof of security for SPAKE2.

2 Security model for password-based key exchange

We now recall the security model for password-based authenticated key exchange of Bellare et al. [3].

Protocol participants. Each participant in the password-based key exchange is either a client $C \in \mathcal{C}$ or a server $S \in \mathcal{S}$. The set of all users or participants $\mathcal{U}$ is the union $\mathcal{C} \cup \mathcal{S}$.

Long-lived keys. Each client $C \in \mathcal{C}$ holds a password pw_C. Each server $S \in \mathcal{S}$ holds a vector $pw_S = \langle pw_S[C] \rangle_{C \in \mathcal{C}}$ with an entry for each client, where $pw_S[C]$ is the transformed-password, as defined in [3]. In this paper, we only consider the symmetric model, in which $pw_S[C] = pw_C$, but they may be different in general. pw_C and pw_S are also called the long-lived keys of client C and server S.

Protocol execution. 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. During the execution, the adversary may create several instances of a participant. 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. The query types available to the adversary are as follows:

- *Execute*(C^i, S^j): This query models passive attacks in which the attacker eavesdrops on honest executions between a client instance C^i and a server instance S^j. The output of this query consists of the messages that were exchanged during the honest execution of the protocol.
- *Send*(U^i, m): This query models an active attack, in which the adversary may tamper with the message being sent over the public channel. The output of this query is the message that the participant instance U^i would generate upon receipt of message m.

- *Reveal*(U^i): This query models the misuse of session keys by a user. If a session key is not defined for instance U^i or if a *Test* query was asked to either U^i or to its partner, then return $\perp$. Otherwise, return the session key held by the instance U^i.
- *Test*(U^i): This query tries to capture the adversary's ability to tell apart a real session key from a random one. If no session key for instance U^i is defined, then return the undefined symbol $\perp$. Otherwise, return the session key for instance U^i if $b = 1$ or a random key of the same size if $b = 0$.

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

PARTNERING. The definition of partnering uses the notion of session identifications (*sid*). 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 session identifications; (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. In practice, the *sid* could be taken to be the partial transcript of the conversation between the client and the server instances before the acceptance.

FRESHNESS. The notion of freshness is defined to avoid cases in which adversary can trivially break the security of the scheme. The goal is to only allow the adversary to ask *Test* queries to *fresh* oracle instances. More specifically, we say an instance U^i is *fresh* if it has *accepted* and if both U^i and its partner are *unopened*.

SEMANTIC SECURITY. Consider an execution of the key exchange protocol P by an adversary $\mathcal{A}$, in which the latter is given access to the *Reveal*, *Execute*, *Send*, and *Test* oracles and asks a single *Test* query to a *fresh* instance, and outputs a guess bit b'. 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 is successful. The ake-**advantage** of an adversary $\mathcal{A}$ in violating the semantic security of the protocol P and the **advantage function** of the protocol P, when passwords are drawn from a dictionary $\mathcal{D}$, are respectively

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

where maximum is over all $\mathcal{A}$ with time-complexity at most t and using resources at most R (such as the number of queries to its oracles). The definition of time-complexity that we use henceforth is the usual one, which includes the maximum of all execution times in the experiments defining the security plus the code size [1].

3 Diffie-Hellman assumptions

In this section, we recall the definitions for the computational Diffie-Hellman assumption and introduce some new variants of it, which we use in the proof of security of simple password-based encrypted key exchange protocols. We also present some relations between these assumptions. In doing so, we borrow some of the notations in [1].

3.1 Definitions

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

Computational Diffie-Hellman: CDH. The CDH assumption states that given g^u and g^v, where u and v were drawn at random from Z_p, it is hard to compute g^{uv}. Under the computational Diffie-Hellman assumption it might be possible for the adversary to compute something interesting about g^{uv} given g^u and g^v.

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 an adversary $\mathcal{A}$. 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. Then, we define *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.

Chosen-basis computational Diffie-Hellman: CCDH. The chosen-basis computational Diffie-Hellman problem is a variation of the CDH problem. It considers an adversary that is given three random elements M, N and X in G and whose goal is to find a triple of values (Y, u, v) such that $u = \mathrm{CDH}(X, Y)$ and $v = \mathrm{CDH}(X/M, Y/N)$. The idea behind this assumption is that the adversary may be able to successfully compute either u (e.g., by choosing $Y = g$ and $u = X$) or v (e.g., by choosing $Y = g \cdot N$ and $v = X/M$), but not both. In fact, as we prove later, solving this problem is equivalent to solving the underlying computational Diffie-Hellman problem in $\mathbb{G}$. We now proceed with the formal definition.

Definition 1 (CCDH). Let $\mathbb{G} = (G, g, p)$ be a represented group and let $\mathcal{A}$ be an adversary. Consider the following experiment, where M, N and X are elements in G,

$$
\begin{array}{l}
\textbf{Experiment } \mathbf{Exp}^{\mathrm{ccdh}}_{\mathbb{G}}(\mathcal{A}, M, N, X) \\
\quad (Y, u, v) \leftarrow \mathcal{A}(M, N, X) \\
\quad u' \leftarrow \mathrm{CDH}(X, Y) \\
\quad v' \leftarrow \mathrm{CDH}(X/M, Y/N) \\
\quad \textbf{if } u = u' \textbf{ and } v = v' \textbf{ then } b \leftarrow 1 \textbf{ else } b \leftarrow 0 \\
\quad \textbf{return } b
\end{array}
$$

Now define the *advantage* of $\mathcal{A}$ in violating the CCDH assumption with respect to (M, N, X), the *advantage* of $\mathcal{A}$, and the *advantage function* of the group, respectively, as follows:

$$
\begin{aligned}
\mathbf{Adv}^{\mathrm{ccdh}}_{\mathbb{G}}(\mathcal{A}, M, N, X) &= \Pr[\,\mathbf{Exp}^{\mathrm{ccdh}}_{\mathbb{G}}(\mathcal{A}, M, N, X) = 1\,] \\
\mathbf{Adv}^{\mathrm{ccdh}}_{\mathbb{G}}(\mathcal{A}) &= \Pr_{M,N,X}\left[\,\mathbf{Adv}^{\mathrm{ccdh}}_{\mathbb{G}}(\mathcal{A}, M, N, X)\,\right] \\
\mathbf{Adv}^{\mathrm{ccdh}}_{\mathbb{G}}(t) &= \max_{\mathcal{A}}\{\,\mathbf{Adv}^{\mathrm{ccdh}}_{\mathbb{G}}(\mathcal{A})\,\},
\end{aligned}
$$

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

Password-based chosen-basis computational Diffie-Hellman: PCCDH. The password-based chosen-basis computational Diffie-Hellman problem is a variation of the chosen-basis computational Diffie-Hellman described above that is more appropriate to the password-based setting. The inputs to the problem and the adversarial goals are also somewhat different in this case so let us explain it.

Let $\mathcal{D} = \{1, \ldots, n\}$ be a dictionary containing n equally likely password values and let $\mathcal{P}$ be a public injective map $\mathcal{P}$ from $\{1, \ldots, n\}$ into Z_p. An example of an admissible map $\mathcal{P}$ is the one in which $\{1, \ldots, n\}$ is mapped into the subset $\{1, \ldots, n\}$ of Z_p. Now let us consider an adversary that runs in two stages. In the first stage, the adversary is given as input three random elements M, N and X in G as well as the public injective map $\mathcal{P}$ and it outputs a value Y in G. Next, we choose a random password $k \in \{1, \ldots, n\}$ and give it to the adversary. We also compute the mapping $r = \mathcal{P}(k)$ of the password k. The goal of the adversary in this second stage is to output a value K such that $K = \mathrm{CDH}(X/M^r, Y/N^r)$.

Note that an adversary that correctly guesses the password k in its first stage can easily solve this problem by computing $r = \mathcal{P}(k)$ and making, for instance, $Y = g \cdot N^r$ and $K = X/M^r$. Since we assume k to be chosen uniformly at random from the dictionary $\{1, \ldots, n\}$, an adversary that chooses to guess the password and follow this strategy can succeed with probability $1/n$.

The idea behind the password-based chosen-basis computational Diffie-Hellman assumption is that no adversary can do much better than the adversary described above. In fact, as we later prove, this should be the case as long as the computational Diffie-Hellman problem is hard in $\mathbb{G}$. We now proceed with the formal definition.

Definition 2 (PCCDH). Let $\mathbb{G} = (G, g, p)$ be a represented group and let $\mathcal{A}$ be an adversary. Consider the following experiment, where M and N are elements in G, and $\mathcal{P}$ is a public injective map from $\{1, \ldots, n\}$ into Z_p,

$$
\begin{array}{l}
\textbf{Experiment } \mathbf{Exp}^{\mathrm{pccdh}}_{\mathbb{G},n}(\mathcal{A}, M, N, X', \mathcal{P}) \\
\quad (Y', st) \leftarrow \mathcal{A}(\mathsf{find}, M, N, X', \mathcal{P}) \\
\quad k \xleftarrow{R} \{1, \ldots, n\} \ ; \ r \leftarrow \mathcal{P}(k) \\
\quad (K) \leftarrow \mathcal{A}(\mathsf{guess}, st, k) \\
\quad X \leftarrow X'/M^r \ ; \ Y \leftarrow Y'/N^r \\
\quad \textbf{if } K = \mathrm{CDH}(X, Y) \textbf{ then } b \leftarrow 1 \textbf{ else } b \leftarrow 0 \\
\quad \textbf{return } b
\end{array}
$$

Now define the *advantage* of $\mathcal{A}$ in violating the PCCDH assumption with respect to $(M, N, X', \mathcal{P})$, the *advantage* of $\mathcal{A}$, and the *advantage function* of the group, respectively, as follows:

$$
\begin{aligned}
\mathbf{Adv}^{\mathrm{pccdh}}_{\mathbb{G},n}(\mathcal{A}, M, N, X', \mathcal{P}) &= \Pr[\,\mathbf{Exp}^{\mathrm{pccdh}}_{\mathbb{G},n}(\mathcal{A}, M, N, X', \mathcal{P}) = 1\,] \\
\mathbf{Adv}^{\mathrm{pccdh}}_{\mathbb{G},n}(\mathcal{A}, \mathcal{P}) &= \Pr_{M,N,X'}\left[\,\mathbf{Adv}^{\mathrm{pccdh}}_{\mathbb{G},n}(\mathcal{A}, M, N, X', \mathcal{P})\,\right] \\
\mathbf{Adv}^{\mathrm{pccdh}}_{\mathbb{G},n}(t, \mathcal{P}) &= \max_{\mathcal{A}}\{\,\mathbf{Adv}^{\mathrm{pccdh}}_{\mathbb{G},n}(\mathcal{A}, \mathcal{P})\,\},
\end{aligned}
$$

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

Set password-based chosen-basis computational Diffie-Hellman: S-PCCDH. The set password-based chosen-basis computational Diffie-Hellman problem (S-PCCDH) is a multidimensional variation of the password-based chosen-basis computational Diffie-Hellman problem described above, in which the adversary is allowed to return not one key but a list of keys at the end of the second stage. In this case, the adversary is considered successful if the list of keys contains the correct value. We now proceed with the formal definition.

Definition 3 (S-PCCDH). Let $\mathbb{G} = (G, g, p)$ be a represented group and let $\mathcal{A}$ be an adversary. Consider the following experiment, where M and N are elements in G, and $\mathcal{P}$ is a public injective map from $\{1, \ldots, n\}$ into Z_p,

Experiment $\mathbf{Exp}^{\text{s-pccdh}}_{\mathbb{G},n,s}(\mathcal{A}, M, N, X', \mathcal{P})$
$(Y', st) \leftarrow \mathcal{A}(\mathsf{find}, M, N, X', \mathcal{P})$
$k \xleftarrow{R} \{1, \ldots, n\}$; $r \leftarrow \mathcal{P}(k)$
$(\mathcal{S}) \leftarrow \mathcal{A}(\mathsf{guess}, st, k)$
$X \leftarrow X'/M^r$; $Y \leftarrow Y'/N^r$
if $\mathrm{CDH}(X, Y) \in \mathcal{S}$ **and** $|\mathcal{S}| \leq s$ **then** $b \leftarrow 1$ **else** $b \leftarrow 0$
return b

As above, we define the *advantage* of $\mathcal{A}$ in violating the S-PCCDH assumption with respect to $(M, N, X', \mathcal{P})$, the *advantage* of $\mathcal{A}$, and the *advantage function* of the group, respectively, as follows:

$$\mathbf{Adv}^{\text{s-pccdh}}_{\mathbb{G},n,s}(\mathcal{A}, M, N, X', \mathcal{P}) = \Pr[\,\mathbf{Exp}^{\text{s-pccdh}}_{\mathbb{G},n,s}(\mathcal{A}, M, N, X', \mathcal{P}) = 1\,]$$
$$\mathbf{Adv}^{\text{s-pccdh}}_{\mathbb{G},n,s}(\mathcal{A}, \mathcal{P}) = \Pr_{M,N,X'}\left[\,\mathbf{Adv}^{\text{s-pccdh}}_{\mathbb{G},n,s}(\mathcal{A}, M, N, X', \mathcal{P})\,\right]$$
$$\mathbf{Adv}^{\text{s-pccdh}}_{\mathbb{G},n,s}(t, \mathcal{P}) = \max_{\mathcal{A}}\{\,\mathbf{Adv}^{\text{s-pccdh}}_{\mathbb{G},n,s}(\mathcal{A}, \mathcal{P})\,\},$$

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

3.2 Some relations

In this section, we first provide two relations between the above problems. The first result is meaningful for small n (polynomially bounded in the asymptotic framework). The second one considers larger dictionaries. Then, we show that these assumptions are implied by the classical computational Diffie-Hellman assumption. Finally, we also prove that the most general assumption is also implied by the classical computational Diffie-Hellman assumption.

Relations between the PCCDH and CCDH problems. The following two lemmas present relations between the PCCDH and CCDH problems. The first lemma, whose proof can be found in Appendix B, is oriented to the case of small dictionaries, for which n is polynomially-bounded. However, if n is large, super-polynomial in the asymptotic framework, or more concretely $n \geq 8/\epsilon$, then one should use the second lemma, whose proof can be easily derived from the proof of the first lemma (see Appendix B).

Lemma 4. *Let $\mathbb{G} = (G, g, p)$ be a represented group, let n be an integer, and let $\mathcal{P}$ be a public injective map from $\{1, \ldots, n\}$ into Z_p.*

$$\frac{2}{n} \geq \mathbf{Adv}^{\text{pccdh}}_{\mathbb{G},n}(t, \mathcal{P}) \geq \frac{1}{n} + \epsilon \Longrightarrow \mathbf{Adv}^{\text{ccdh}}_{\mathbb{G}}(2t + 3\tau) \geq \frac{n}{128} \times \epsilon^3.$$

Lemma 5. *Let $\mathbb{G} = (G, g, p)$ be a represented group, let n be an integer, and let $\mathcal{P}$ be a public injective map from $\{1, \ldots, n\}$ into Z_p.*

$$\mathbf{Adv}^{\text{pccdh}}_{\mathbb{G},n}(t, \mathcal{P}) \geq \epsilon \geq \frac{8}{n} \Longrightarrow \mathbf{Adv}^{\text{ccdh}}_{\mathbb{G}}(2t + 3\tau) \geq \frac{\epsilon^2}{32},$$

where τ denotes the time for an exponentiation in $\mathbb{G}$.

Relation between the CCDH and CDH problems. The following lemma, whose proof is in Appendix B, shows that the CCDH and CDH problems are indeed equivalent.

Lemma 6. *Let $\mathbb{G} = (G, g, p)$ be a represented group.*

$$\mathbf{Adv}^{\mathrm{ccdh}}_{\mathbb{G}}(t) \leq \mathbf{Adv}^{\mathrm{cdh}}_{\mathbb{G}}(t + 2\tau),$$

where τ denotes the time for an exponentiation in $\mathbb{G}$.

Relation between the S-PCCDH and CDH problems. The following lemma, whose proof is in Appendix B, gives a precise relation between the S-PCCDH and CDH problems.

Lemma 7. *Let $\mathbb{G} = (G, g, p)$ be a represented group, let n and s be integers, and let $\mathcal{P}$ be a public injective map from $\{1, \ldots, n\}$ into Z_p.*

$$\mathbf{Adv}^{\mathrm{s-pccdh}}_{\mathbb{G},n,s}(t, \mathcal{P}) \geq \frac{1}{n} + \epsilon \Longrightarrow \mathbf{Adv}^{\mathrm{cdh}}_{\mathbb{G}}(t') \geq \frac{n^2\epsilon^6}{2^{14}} - \frac{2s^4}{p},$$

where $t' = 4t + (18 + 2s)\tau$ and τ denotes the time for an exponentiation in $\mathbb{G}$. More concretely,

$$\mathbf{Adv}^{\mathrm{s-pccdh}}_{\mathbb{G},n,s}(t, \mathcal{P}) \geq \frac{1}{n} + \epsilon \geq \frac{1}{n} \times \left(1 + \frac{8(ns)^{2/3}}{p^{1/6}}\right) \Longrightarrow \mathbf{Adv}^{\mathrm{cdh}}_{\mathbb{G}}(t') \geq \frac{n^2\epsilon^6}{2^{15}}.$$

4 SPAKE1: a simple non-concurrent password-based encrypted key exchange

We now introduce our first protocol, SPAKE1, which is a *non-concurrent* password-based encrypted key exchange protocol, based on the multi-dimensional version of password-based chosen-basis computational Diffie-Hellman problem, S-PCCDH.

4.1 Description

SPAKE1 is a variation of the password-based encrypted key exchange protocol of Bellovin and Merritt [7], in which we replace the encryption function $\mathcal{E}_{pw}(.)$ with a simple one-time pad function. More specifically, whenever a user A wants to send the encryption of a value $X \in G$ to a user B, it does so by computing $X \cdot M^{pw}$, where M is an element in G associated with user A and the password pw is assumed to be in Z_p. The session identification is defined as the transcript of the conversation between A and B, and the session key is set to be the hash (random oracle) of the session identification, the user identities, and the Diffie-Hellman key. The password pw is *not* an input to the hash function. The full description of SPAKE1 is given in Figure 2.

Correctness. The correctness of our protocol follows from the fact that, in an honest execution of the protocol, $K_A = K_B = g^{xy}$.

4.2 Security

As Theorem 8 states, our non-concurrent password-based key exchange protocol is secure in the random oracle model as long as we believe that the S-PCCDH problem is hard in $\mathbb{G}$.

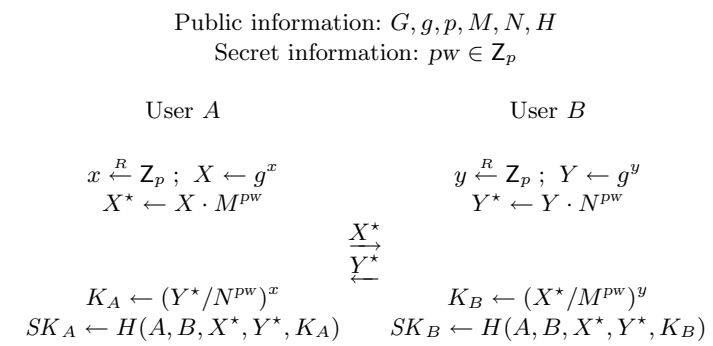

Fig. 2. SPAKE1: a simple non-concurrent password-based key exchange protocol.

Theorem 8. *Let $\mathbb{G}$ be a represent group and let $\mathcal{D}$ be a uniformly distributed dictionary of size $|\mathcal{D}|$. Let* SPAKE1 *describe the password-based encrypted key exchange protocol associated with these primitives as defined in Figure 2. Then, for any numbers t, q_{start}, q^A_{send}, q^B_{send}, q_H, q_{exe},*

$$\begin{aligned}
&\mathbf{Adv}^{\text{ake}}_{\mathsf{SPAKE},\mathcal{D}}(t, q_{\text{start}}, q^A_{\text{send}}, q^B_{\text{send}}, q_H, q_{\text{exe}}) \\
&\leq\ 2 \cdot (q^A_{\text{send}} + q^B_{\text{send}}) \cdot \mathbf{Adv}^{\text{s-pccdh}}_{\mathbb{G},|\mathcal{D}|,q_H}(t', \mathcal{P}) + \\
&\quad 2 \cdot \left(\frac{(q_{\text{exe}} + q_{\text{send}})^2}{2p} + q_H\ \mathbf{Adv}^{\text{cdh}}_{\mathbb{G}}(t + 2q_{\text{exe}}\tau + 3\tau) \right)\ ,
\end{aligned}$$

where q_H represents the number of queries to the H oracle; q_{exe} represents the number of queries to the Execute oracle; q_{start} and q^A_{send} represent the number of queries to the Send oracle with respect to the initiator A; q^B_{send} represents the number of queries to the Send oracle with respect to the responder B; $q_{\text{send}} = q^A_{\text{send}} + q^B_{\text{send}} + q_{\text{start}}$; $t' = t + O(q_{\text{start}}\tau)$; and τ is the time to compute one exponentiation in $\mathbb{G}$.

Since the S-PCCDH problem can be reduced to the CDH problem according to Lemma 7, it follows that SPAKE1 is a secure non-concurrent password-based key exchange protocol in the random oracle model as long as the CDH problem is hard in $\mathbb{G}$, as stated in Corollary 9.

Corollary 9. *Let $\mathbb{G}$ be a represent group and let $\mathcal{D}$ be a uniformly distributed dictionary of size $|\mathcal{D}|$. Let* SPAKE1 *describe the password-based encrypted key exchange protocol associated with these primitives as defined in Figure 2. Then, for any numbers t, q_{start}, q^A_{send}, q^B_{send}, q_H, q_{exe},*

$$\begin{aligned}
&\mathbf{Adv}^{\text{ake}}_{\mathsf{SPAKE},\mathcal{D}}(t, q_{\text{start}}, q^A_{\text{send}}, q^B_{\text{send}}, q_H, q_{\text{exe}}) \\
&\leq\ 2 \cdot \left(\frac{q^A_{\text{send}} + q^B_{\text{send}}}{|\mathcal{D}|} + \sqrt[6]{\frac{2^{14}}{|\mathcal{D}|^2}\mathbf{Adv}^{\text{cdh}}_{\mathbb{G}}(t') + \frac{2^{15} q_H^4}{|\mathcal{D}|^2 p}} \right) + \\
&\quad 2 \cdot \left(\frac{(q_{\text{exe}} + q_{\text{send}})^2}{2p} + q_H\ \mathbf{Adv}^{\text{cdh}}_{\mathbb{G}}(t + 2q_{\text{exe}}\tau + 3\tau) \right)\ ,
\end{aligned}$$

where $t' = 4t + O((q_{\text{start}} + q_H)\tau)$ and the other parameters are defined as in Theorem 8.

Proof idea. Let $\mathcal{A}$ be an adversary against the semantic security of SPAKE. The idea is to use $\mathcal{A}$ to build adversaries for each of the underlying primitives in such a way that if $\mathcal{A}$ succeeds in breaking the semantic security of SPAKE, then at least one of these adversaries succeeds in

H oracle	– 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 $r \in \{0,1\}^{l_k}$, add the record (q, r) to the list Λ_H (resp. Λ_H), and return r.

Fig. 3. Simulation of random oracles H and H'.

Send queries	– On a query $Send(A^i, \textbf{start})$, assuming A^i is in the correct state, we proceed as follows: **if** ActiveSessionIndex $\neq 0$ **then abort** $A^{\text{ActiveSessionIndex}}$ ActiveSessionIndex $= i$ $\theta \xleftarrow{R} \mathsf{Z}_p$; $\Theta \leftarrow g^{\theta}$; $\Theta^{\star} \leftarrow \Theta \cdot M^{pw}$ **return** $(A, \Theta^{\star})$ – On a query $Send(B^i, (A, \Theta^{\star}))$, assuming B^i is in the correct state, we proceed as follows: $\phi \xleftarrow{R} \mathsf{Z}_p$; $\Phi \leftarrow g^{\phi}$; $\Phi^{\star} \leftarrow \Phi \cdot N^{pw}$ $K \leftarrow (\Theta^{\star}/M^{pw})^{\phi}$ $SK \leftarrow H(A, B, \Theta^{\star}, \Phi^{\star}, K)$ **return** $(B, \Phi^{\star})$ – On a query $Send(A^i, (B, \Phi^{\star}))$, assuming A^i is in the correct state, we proceed as follows: $K \leftarrow (\Phi^{\star}/N^{pw})^{\theta}$ $SK \leftarrow H(A, B, \Theta^{\star}, \Phi^{\star}, K)$ ActiveSessionIndex $= 0$

Fig. 4. Simulation of *Send* oracle query.

breaking the security of an underlying primitive. Our proof consists of a sequence of hybrid experiments, starting with the real attack and ending in an experiment in which the adversary's advantage is 0, and for which we can bound the difference in the adversary's advantage between any two consecutive experiments.

Proof of Theorem 8. Our proof uses a sequence of hybrid experiments, the first of which corresponds to the actual attack. For each experiment $\mathbf{Exp}_n$, we define an event SUCC_n corresponding to the case in which the adversary correctly guesses the bit b involved in the *Test* query.

Experiment $\mathbf{Exp}_0$. This experiment corresponds to the real attack, which starts by choosing a random password pw. By definition, we have

$$\mathbf{Adv}^{\text{ake}}_{\mathsf{SPAKE}}(\mathcal{A}) = 2 \cdot \Pr[\,\text{SUCC}_0\,] - 1 \tag{1}$$

Experiment $\mathbf{Exp}_1$. In this experiment, we simulate the *Execute*, *Reveal*, and *Send* oracles as in the real attack (see Figure 4 and Figure 5), after having chosen a random password pw. One can easily see that this experiment is perfectly indistinguishable from the real experiment. Hence,

$$\Pr[\,\text{SUCC}_1\,] = \Pr[\,\text{SUCC}_0\,] \tag{2}$$

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 transcript $((A, X^{\star}), (B, Y^{\star}))$. Since either $X^{\star}$ or $Y^{\star}$ was simulated and thus chosen uniformly at random, the probability of collisions in the transcripts is at most $(q_{\text{send}}+q_{\text{exe}})^2/(2p)$, according to the birthday paradox. Consequently,

$$\left|\Pr[\,\text{SUCC}_2\,] - \Pr[\,\text{SUCC}_1\,]\right| \leq \frac{(q_{\text{exe}} + q_{\text{send}})^2}{2p} \tag{3}$$

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 $Execute(A^i, B^j)$, proceed as follows:
  $\theta \stackrel{R}{\leftarrow} \mathsf{Z}_p$; $\Theta \leftarrow g^\theta$; $\Theta^\star \leftarrow \Theta \cdot M^{pw}$
  $\phi \stackrel{R}{\leftarrow} \mathsf{Z}_p$; $\Phi \leftarrow g^\phi$; $\Phi^\star \leftarrow \Phi \cdot N^{pw}$
  $K \leftarrow \Theta^\phi$
  $SK_A \leftarrow H(A, B, \Theta^\star, \Phi^\star, K)$; $SK_B \leftarrow SK_A$
  return $((A, \Theta^\star), (B, \Phi^\star))$

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

Fig. 5. Simulation of *Execute*, *Reveal* and *Test* queries.

Experiment $\mathbf{Exp}_3$. In this experiment, we replace the random oracle H by a secret one, for computing SK_A and SK_B 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 CDH assumption holds. More precisely, we use a private random oracle H', and in the *Execute* oracle queries, one gets $SK_A, SK_B \leftarrow H'(A, B, \Theta^\star, \Phi^\star)$.

Lemma 10. $\left|\Pr[\,\textsc{Succ}_3\,] - \Pr[\,\textsc{Succ}_2\,]\right| \leq q_H \cdot \mathbf{Adv}^{\mathrm{cdh}}_{\mathbb{G}}(t + 2q_{\mathrm{exe}}\tau + 3\tau)$.

Proof. The proof of Lemma 10 uses the random self-reducibility of the Diffie-Hellman problem. Indeed, the only way for an execution to be altered by the above modification is if the adversary directly asks for $H(A, B, \Theta^\star, \Phi^\star, K)$, which will output something different from $H'(A, B, \Theta^\star, \Phi^\star)$, the answer of a *Reveal* query. But let us simulate the *Execute* oracle with a Diffie-Hellman instance (A, B), and thus $\Theta \leftarrow A \cdot g^\theta$ and $\Phi \leftarrow B \cdot g^\phi$. As a consequence, the above event means that $K = \mathrm{CDH}(\Theta, \Phi) = \mathrm{CDH}(A, B) \times A^\phi \times B^\theta \times g^{\phi\theta}$ is in the list of the queries asked to H: a random guess leads to $\mathrm{CDH}(A, B)$.

Experiment $\mathbf{Exp}_4$. The goal of this experiment is to bound the advantage of the adversary during active attacks, in which the adversary has possibly generated the input of a *Send* oracle. To achieve this goal, we change the simulation of the *Send* oracle so that its output is chosen uniformly at random and independently of the password. The session key associated with each oracle is a bit string of appropriate length chosen uniformly at random and independently of input being provided to the *Send* oracle. The exact simulation of the *Send* oracle is as follows:

– On a query of type $(A^i, \mathbf{start}\ )$, we reply with $(A, X^\star = g^{x^\star})$ for a random $x^\star \in \mathsf{Z}_p$, if A^i is in the correct state. If another concurrent session already exists for user A, then we also terminate that session.
– On a query of type $(B^i, (A, X^\star))$, we reply with $(B, Y^\star = g^{y^\star})$ for a random $y^\star \in \mathsf{Z}_p$ and we set the session key SK_B to $H'(A, B, X^\star, Y^\star)$, if B^i is in the correct state.
– On a query of type $(A^i, (B, Y^\star))$, we set the session key SK_A to $H'(A, B, X^\star, Y^\star)$, if A^i is in the correct state.

As the following lemma shows, the adversary cannot do much better than simply guessing the password when distinguishing the current experiment from the previous one.

Lemma 11. $\left|\Pr[\,\textsc{Succ}_4\,] - \Pr[\,\textsc{Succ}_3\,]\right| \leq (q^A_{\text{send}} + q^B_{\text{send}}) \cdot \mathbf{Adv}^{\text{s-pccdh}}_{\mathbb{G},|\mathcal{D}|,q_H}(t', \mathcal{P})$, *where* $t' = t + O(q_{\text{start}}\tau)$.

Proof. The proof of this lemma is based on a sequence of $q^A_{\text{send}} + q^B_{\text{send}} + 1$ hybrid experiments $\mathbf{Hybrid}_{3,j}$, where j is an index between 0 and $q_{AB} = q^A_{\text{send}} + q^B_{\text{send}}$. Let i be a counter for number of queries of the form $(B^k, (A, X^\star))$ or $(A^k, (B, Y^\star))$. That is, we do not count start queries (we do not increment this counter). We define Experiment $\mathbf{Hybrid}_{3,j}$ as follows:

- If $i \leq j$, then we processes the current *Send* query as in Experiment $\mathbf{Exp}_4$.
- If $i > j$, then we processes the current *Send* query as in Experiment $\mathbf{Exp}_3$.

It is clear from the above definition that experiments $\mathbf{Hybrid}_{3,0}$ and $\mathbf{Hybrid}_{3,q_{AB}}$ are equivalent to experiments $\mathbf{Exp}_3$ and $\mathbf{Exp}_4$, respectively. Now let P_j denote the probability of event Succ in Experiment $\mathbf{Hybrid}_{3,j}$. It follows that $\Pr[\,\textsc{Succ}_3\,] = P_0$ and $\Pr[\,\textsc{Succ}_4\,] = P_{q_{AB}}$. Moreover,

$$\left|\Pr[\,\textsc{Succ}_4\,] - \Pr[\,\textsc{Succ}_3\,]\right| \leq \sum_{j=1}^{q_{AB}} \left|P_j - P_{j-1}\right| .$$

The lemma will follow easily from bounding $\left|P_j - P_{j-1}\right|$. In order to so, consider the following algorithm D_j for the S-PCCDH problem in $\mathbb{G}$.

Algorithm D_j. Let $U = g^u$, $V = g^v$, and $W = g^w$ be random elements in G and let $\mathcal{P}$ be any injective map from $\{1, \ldots, n\}$ into Z_p. D_j starts running $\mathcal{A}$, simulating all its oracles. The *Reveal*, *Execute*, and *Test* oracles are simulated as in Experiment $\mathbf{Exp}_3$. The *Send* oracle is simulated as follows, Let i be the index of the current *Send* query.

- If the *Send* query is of the form $(A^k, \textbf{start}\ )$,
  - if $i \leq j$, then D_j replies with $(A, X^\star = Wg^{x^\star})$ for a random $x^\star \in \mathsf{Z}_p$, if A^k is in the correct state. If another concurrent session already exists for user A, then D_j also terminates that session.
  - if $i > j$, then D_j processes it as in Experiment $\mathbf{Exp}_3$.
- If the query is of the form $(B^k, (A, X^\star))$,
  - if $i < j$, then D_j processes it as in Experiment $\mathbf{Exp}_4$.
  - if $i = j$, then D_j replies with $(B, Y^\star = W)$. It also returns $(st, Y' = X^\star)$ as the output of its find stage and waits for the input (st, k) of the guess stage. It then sets the password pw shared between A and B to $\mathcal{P}(k)$ and the session key SK_B to $H(A, B, X^\star, Y^\star, K_B)$, where $K_B = (X^\star / V^{pw})^{w-u\ pw}$. We note that st should contain all the necessary information for D_j to continue the execution of $\mathcal{A}$ and the simulation of its oracles in the guess stage. Let this be *Case B*.
  - if $i > j$, then D_j processes it as in Experiment $\mathbf{Exp}_3$.
- If the *Send* query is of the form $(A^k, (B, Y^\star))$,
  - if $i < j$, then D_j processes it as in Experiment $\mathbf{Exp}_4$.
  - if $i = j$, and A^k is in the correct state, then it returns $(st, Y' = Y^\star)$ as the output of its find stage and waits for the input (st, k) of the guess stage. Then, it sets the password pw shared between A and B to $\mathcal{P}(k)$ and the session key SK_A to $H(A, B, X^\star, Y^\star, K_A)$, where $K_A = (Y^\star / V^{pw})^{w+x^\star-u\ pw}$. Let this be *Case A*.
  - if $i > j$, then D_j processes it as in Experiment $\mathbf{Exp}_3$.

Let K be the part of the input of H that is not present in H' and let $K_1, \ldots, K_{q_H}$ be the list of all such elements. When in Case A, D_j sets $K'_i = K_i / (Y'/V^{pw})^{x^\star}$ for $i = 1, \ldots, q_H$, where $x^\star$ is

the value used to compute $X^\star$ in the crucial query. When in Case B, D_j simply sets $K'_i = K_i$. Finally, D_j outputs $K'_1, \ldots, K'_{q_H}$.

We note that in the above, the password is only defined at the j-th step and it is not used before that. Due to the non-concurrency, we do not need to know the password for simulating flows in Experiment $\mathbf{Exp}_4$. We only need it in Experiment $\mathbf{Exp}_3$.

Using the knowledge of u, v, and w in the above, it is clear that the processing of the *Send* queries matches that of Experiment $\mathbf{Hybrid}_{3,j-1}$. However, in the actual description of the S-PCCDH problem, we do not have access to these values. For this reason, the actual algorithm D_j replaces the random oracle H by a secret random oracle H' in the computation of SK_A and SK_B during the processing of the j-th *Send* query. More precisely, it computes SK_A and SK_B as $H'(A, B, X^\star, Y^\star)$. Moreover, we note that in this new scenario, the processing of the *Send* queries matches that of Experiment $\mathbf{Hybrid}_{3,j}$.

Probability analysis. Let AskH represent the event in which the adversary asks for $H(A, B, X^\star, Y^\star, K)$, where $K = \mathrm{CDH}(X^\star/U^{pw}, Y^\star/V^{pw})$ and either $X^\star$ or $Y^\star$ is involved in the crucial j-th query. We first observe that experiments $\mathbf{Hybrid}_{3,j-1}$ and $\mathbf{Hybrid}_{3,j}$ are identical if event AskH does not happen. Therefore, it follows that the probability difference $\left|P_j - P_{j-1}\right|$ is at most $\Pr[\,\textsc{AskH}\,]$.

However, whenever event AskH happens, we know that the list of queries asked to H contains the key $K = \mathrm{CDH}(X^\star/U^{pw}, Y^\star/V^{pw})$ involved in the crucial query, and thus D_j will be able to successfully use $\mathcal{A}$ to help it solve the S-PCCDH problem. This is because K_A (Case A) or K_B (Case B) can be used to compute the solution $\mathrm{CDH}(W/U^{pw}, Y'/V^{pw})$ for the S-PCCDH problem as follows:

$$K_A = \mathrm{CDH}(Y^\star/V^{pw}, Wg^{x^\star}/U^{pw}) = \mathrm{CDH}(W/U^{pw}, Y'/V^{pw}) \times (Y'/V^{pw})^{x^\star}$$
$$K_B = \mathrm{CDH}(X^\star/V^{pw}, W/U^{pw}) = \mathrm{CDH}(W/U^{pw}, Y'/V^{pw})$$

Therefore, the list of candidates $K'_1, \ldots, K'_{q_H}$ outputted by D_j should contain the solution for the S-PCCDH problem whenever AskH happens. Hence, $\Pr[\,\textsc{AskH}\,]$ is less than or equal to the success probability of D_j. The lemma follows easily from the fact that D_j has time-complexity at most t'. ∎

5 SPAKE2: a simple concurrent password-based encrypted key exchange

We now introduce our second protocol, SPAKE2, which is a *concurrent* password-based encrypted key exchange protocol, based on the computational Diffie-Hellman problem, CDH.

5.1 Description

SPAKE2 is a also variation of the password-based encrypted key exchange protocol of Bellovin and Merritt[7] and is almost exactly like SPAKE1. The only difference between the two is in the key derivation function, which also includes the password pw. More specifically, the session key in SPAKE2 is set to be the hash (random oracle) of the session identification, the user identities, the Diffie-Hellman key, and the password. In other words, $SK \leftarrow H(A, B, X^\star, Y^\star, pw, K)$. The session identification is still defined as the transcript of the conversation between A and B.

5.2 Security

As the following theorem states, our concurrent password-based key exchange protocol is secure in the random oracle model as long as the CDH problem is hard in G. The proof of Theorem 12 can be found in Appendix C.

Theorem 12. *Let $\mathbb{G}$ be a represent group and let $\mathcal{D}$ be a uniformly distributed dictionary of size $|\mathcal{D}|$. Let* SPAKE2 *describe the password-based encrypted key exchange protocol associated with these primitives as defined in Section 5.1. Then, for any numbers t, q_{start}, q^A_{send}, q^B_{send}, q_H, q_{exe},*

$$\begin{aligned}
&\mathbf{Adv}^{\text{ake}}_{\mathsf{SPAKE2},\mathcal{D}}(t, q_{\text{start}}, q^A_{\text{send}}, q^B_{\text{send}}, q_H, q_{\text{exe}}) \\
&\leq\ 2 \cdot \left(\frac{q^A_{\text{send}} + q^B_{\text{send}}}{n} + \frac{(q_{\text{exe}} + q_{\text{send}})^2}{2p} \right) + \\
&\quad 2 \cdot \left(q_H\ \mathbf{Adv}^{\text{cdh}}_{\mathbb{G}}(t + 2q_{\text{exe}}\tau + 3\tau) + q_H^2\ \mathbf{Adv}^{\text{cdh}}_{\mathbb{G}}(t + 3\tau) \right),
\end{aligned}$$

where the parameters are defined as in Theorem 8.

Acknowledgments

The work described in this document has been supported in part by the European Commission through the IST Programme 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. M. Abdalla, M. Bellare, and P. Rogaway. The oracle Diffie-Hellman assumptions and an analysis of DHIES. In D. Naccache, editor, *CT-RSA 2001*, volume 2020 of *LNCS*, pages 143–158. Springer-Verlag, Apr. 2001.
2. M. Bellare and T. Kohno. A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In E. Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 491–506. Springer-Verlag, May 2003.
3. M. Bellare, D. Pointcheval, and P. Rogaway. Authenticated key exchange secure against dictionary attacks. In B. Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 139–155. Springer-Verlag, May 2000.
4. M. Bellare and P. Rogaway. Entity authentication and key distribution. In D. R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*. Springer-Verlag, Aug. 1994.
5. M. Bellare and P. Rogaway. Provably secure session key distribution — the three party case. In *28th ACM STOC*, pages 57–66. ACM Press, May 1996.
6. M. Bellare and P. Rogaway. The AuthA protocol for password-based authenticated key exchange. Contributions to IEEE P1363, Mar. 2000.
7. S. M. Bellovin and M. Merritt. Encrypted key exchange: Password-based protocols secure against dictionary attacks. In *1992 IEEE Symposium on Security and Privacy*, pages 72–84. IEEE Computer Society Press, May 1992.
8. V. Boyko, P. MacKenzie, and S. Patel. Provably secure password-authenticated key exchange using Diffie-Hellman. In B. Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 156–171. Springer-Verlag, May 2000.
9. E. Bresson, O. Chevassut, and D. Pointcheval. Security proofs for an efficient password-based key exchange. In *ACM CCS 03*. ACM Press, Oct. 2003.
10. E. Bresson, O. Chevassut, and D. Pointcheval. New security results on encrypted key exchange. In F. Bao, R. Deng, and J. Zhou, editors, *PKC 2004*, volume 2947 of *LNCS*, pages 145–158. Springer-Verlag, Mar. 2004.
11. M. Di Raimondo and R. Gennaro. Provably secure threshold password-authenticated key exchange. In E. Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 507–523. Springer-Verlag, May 2003.
12. R. Gennaro and Y. Lindell. A framework for password-based authenticated key exchange. In E. Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 524–543. Springer-Verlag, May 2003. `http://eprint.iacr.org/2003/032.ps.gz`.
13. O. Goldreich and Y. Lindell. Session-key generation using human passwords only. In J. Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 408–432. Springer-Verlag, Aug. 2001. `http://eprint.iacr.org/2000/057.ps.gz`.

14. S. Halevi and H. Krawczyk. Public-key cryptography and password protocols. In *ACM Transactions on Information and System Security*, pages 524–543. ACM, 1999.
15. IEEE draft standard P1363.2. Password-based public key cryptography. `http://grouper.ieee.org/groups/1363/passwdPK`, May 2004. Draft Version 15.
16. J. Katz, R. Ostrovsky, and M. Yung. Efficient password-authenticated key exchange using human-memorable passwords. In B. Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 475–494. Springer-Verlag, May 2001.
17. K. Kobara and H. Imai. Pretty-simple password-authenticated key-exchange under standard assumptions. *IEICE Transactions*, E85-A(10):2229–2237, Oct. 2002. Also available at `http://eprint.iacr.org/2003/038/`.
18. H. Krawczyk. SIGMA: The "SIGn-and-MAc" approach to authenticated Diffie-Hellman and its use in the IKE protocols. In D. Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 400–425. Springer-Verlag, Aug. 2003.
19. P. MacKenzie, S. Patel, and R. Swaminathan. Password-authenticated key exchange based on RSA. In T. Okamoto, editor, *ASIACRYPT 2000*, volume 1976 of *LNCS*, pages 599–613. Springer-Verlag, Dec. 2000.
20. P. D. MacKenzie. The PAK suite: Protocols for password-authenticated key exchange. Contributions to IEEE P1363.2, 2002.
21. P. D. MacKenzie, T. Shrimpton, and M. Jakobsson. Threshold password-authenticated key exchange. In M. Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 385–400. Springer-Verlag, Aug. 2002.
22. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000.
23. V. Shoup. Lower bounds for discrete logarithms and related problems. In W. Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer-Verlag, May 1997.

A The splitting lemma

For simplicity, we reproduce here the splitting lemma presented in [22].

Lemma 13 (Splitting Lemma). *Let $A \subset X \times Y$ such that $\Pr[(x,y) \in A] \geq \epsilon$. For any $\alpha < \epsilon$, define*

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

then the following statements hold:

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

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

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

This implies a contradiction, hence the result. Statement (ii) is a straightforward consequence of the definition. We finally turn to the last assertion, using Bayes' law:

$$\Pr[B \mid A] = 1 - \Pr[\bar{B} \mid A] = 1 - \Pr[A \mid \bar{B}] \cdot \Pr[\bar{B}] / \Pr[A] \geq 1 - (\epsilon - \alpha)/\epsilon = \alpha/\epsilon.$$

B Proof of lemmas

B.1 Proof of Lemma 4

By definition of $\mathbf{Adv}^{\mathrm{pccdh}}_{\mathbb{G},n}(\mathcal{A},\mathcal{P})$, we have

$$\Pr[\, b = 1 \,] \geq \frac{1}{n} + \epsilon,$$

where all the variables follow the distribution defined in **Experiment** $\mathbf{Exp}^{\mathrm{pccdh}}_{\mathbb{G},n}(\mathcal{A}, M, N, X', \mathcal{P})$, for uniformly distributed $M, N, X' \in G$ and $b \in \{0,1\}$. The probability space is thus defined by the variables ω, ρ, the random tapes of the adversary in the find-stage and the guess-stage respectively; M, N, X', the group elements; and k, the password:

$$\Omega_0 = \left\{ (\omega, \rho, M, N, X', k) \;\middle|\; \omega, \rho \xleftarrow{R} \{0,1\}^\star;\ (M, N, X') \xleftarrow{R} G^3;\ k \xleftarrow{R} \{1, \ldots, n\} \right\}.$$

By definition, we have

$$\Pr_{\Omega_0}[\, b = 1 \,] \geq \frac{1}{n} + \epsilon.$$

Applying the *splitting lemma* 13, we get that if we split Ω_0 into $\Omega'_1 \times \Omega_1$:

$$\Omega'_1 = \left\{ \omega \;\middle|\; \omega \xleftarrow{R} \{0,1\}^\star \right\}$$
$$\Omega_1 = \left\{ (\rho, M, N, X', k) \;\middle|\; \rho \xleftarrow{R} \{0,1\}^\star\ ;\ (M, N, X') \xleftarrow{R} G^3\ ;\ k \xleftarrow{R} \{1, \ldots, n\} \right\},$$

there exists a set $\mathcal{S}_1 \subseteq \Omega'_1$ such that

$$\Pr_{\Omega'_1}[\, \mathcal{S}_1 \,] \geq \frac{\epsilon}{2}, \qquad \text{for any } \omega \in \mathcal{S}_1, \Pr_{\Omega_1}[\, b = 1 \,] \geq \frac{1}{n} + \frac{\epsilon}{2},$$

$$\Pr_{\Omega_0}[\, \omega \in \mathcal{S}_1 \mid b = 1 \,] \geq \frac{n\epsilon}{2 + 2n\epsilon}.$$

Applying again the *splitting lemma* 13, we get that if we split Ω_1 into $\Omega'_2 \times \Omega_2$:

$$\Omega'_2 = \left\{ (M, N, X') \;\middle|\; (M, N, X') \xleftarrow{R} G^3 \right\}$$
$$\Omega_2 = \left\{ (\rho, k) \;\middle|\; \rho \xleftarrow{R} \{0,1\}^\star\ ;\ k \xleftarrow{R} \{1, \ldots, n\} \right\},$$

for each random tape $\omega \in \mathcal{S}_1$, there exists a set $\mathcal{S}_2(\omega) \subseteq \Omega'_2$ such that

$$\Pr_{\Omega'_2}[\, \mathcal{S}_2(\omega) \,] \geq \frac{\epsilon}{4}, \qquad \text{for any } (M, N, X') \in \mathcal{S}_2(\omega), \Pr_{\Omega_2}[\, b = 1 \,] \geq \frac{1}{n} + \frac{\epsilon}{4},$$

$$\Pr_{\Omega_1}\left[\, (M, N, X') \in \mathcal{S}_2(\omega) \mid b = 1 \,\right] \geq \frac{n\epsilon}{4 + 2n\epsilon}.$$

Let (U, V, X) be a random instance of the CCDH problem. We choose a random tape $\omega \xleftarrow{R} \{0,1\}^\star$, and two random indices $i < j \in \{1, \ldots, n\}$. Then, we define $r = \mathcal{P}(i)$ and $r' = \mathcal{P}(j)$, and then set $\delta = r' - r$, $M = U^{1/\delta}$, $N = V^{1/\delta}$, as well as $X' = X \times M^r$. Since (U, V, X) is a random triple, (M, N, X') is also uniformly distributed (and thus independently of r and r', and thus of i and j): with probability greater than $1/n + \epsilon$, $\mathcal{A}$ wins the **Experiment** $\mathbf{Exp}^{\mathrm{pccdh}}_{\mathbb{G},n}(\mathcal{A}, M, N, X', \mathcal{P})$, with $k = i$, and the random tape (ω, ρ). In the favorable case, we have $\omega \in \mathcal{S}_1$ and $(M, N, X') \in \mathcal{S}_2(\omega)$ with probability

$$\frac{n\epsilon}{2 + 2n\epsilon} \times \frac{n\epsilon}{4 + 2n\epsilon} \geq \frac{1}{8} \times \left(\frac{n\epsilon}{1 + n\epsilon} \right)^2.$$

Thereafter, with probability greater than $\epsilon/4$, $\mathcal{A}$ wins the **Experiment** $\mathbf{Exp}^{\mathrm{pccdh}}_{\mathbb{G},n}(\mathcal{A}, M, N, X', \mathcal{P})$, with $k = j$, and the random tape (ω, ρ'):

$$\Pr[\, K = \mathrm{CDH}(X'/M^r, Y'/N^r) \wedge K' = \mathrm{CDH}(X'/M^{r'}, Y'/N^{r'}) \,]$$

$$\geq \left(\frac{1}{n}+\epsilon\right) \times \frac{1}{8} \times \left(\frac{n\epsilon}{1+n\epsilon}\right)^2 \times \frac{\epsilon}{4}.$$

Since we assumed, in the theorem, that $2/n \geq 1/n + \epsilon$, then $1 + n\epsilon \leq 2$. Furthermore,

$$\begin{aligned} K &= \mathrm{CDH}(X'/M^r, Y'/N^r) = \mathrm{CDH}(X, Y) \\ K' &= \mathrm{CDH}(X'/M^{r'}, Y'/N^{r'}) = \mathrm{CDH}(X/M^\delta, Y/N^\delta) = \mathrm{CDH}(X/U, Y/V) \end{aligned}$$

This concludes the proof of this lemma. ▮

B.2 Proof of Lemma 5

The proof of this lemma is similar to that of Lemma 4 and is hence omitted in this version of the paper.

B.3 Proof of Lemma 6

Let us consider an instance A, B for the Diffie-Hellman problem. We choose a random $b \in \mathsf{Z}_p$. Then we set $M \leftarrow A$, $N \leftarrow B$ and $X \leftarrow A^b$. We run an adversary $\mathcal{A}$ on this triple (M, N, X), and get (Y, u, v), which in case of success $u = \mathrm{CDH}(X, Y)$ and $v = \mathrm{CDH}(X/M, Y/N)$:

$$\begin{aligned} u &= \mathrm{CDH}(X, Y) = \mathrm{CDH}(A^b, Y) = \mathrm{CDH}(A, Y)^b \\ v &= \mathrm{CDH}(X/M, Y/N) = \mathrm{CDH}(A^{b-1}, Y/B) = \mathrm{CDH}(A, Y/B)^{b-1} \\ &= \mathrm{CDH}(A, Y)^{b-1}/\mathrm{CDH}(A, B)^{b-1} \\ u^{b-1} &= \mathrm{CDH}(A, Y)^{b(b-1)} \\ v^b &= \mathrm{CDH}(A, Y)^{b(b-1)}/\mathrm{CDH}(A, B)^{b(b-1)} = u^{b-1}/\mathrm{CDH}(A, B)^{b(b-1)} \end{aligned}$$

Thus,

$$\mathrm{CDH}(A, B)^{b(b-1)} = u^{b-1}/v^b \quad \text{and} \quad \mathrm{CDH}(A, B) = u^{1/b} v^{-1/(b-1)}. \blacksquare$$

B.4 Proof of Lemma 7

In this proof, we use a general technique presented in [23] for computing the solution of a given instance of a problem by finding a collision between the list of candidates for two related instances.

On a given A, B instance of the Diffie-Hellman problem, we use exactly the same argument as in the proof of Lemma 4, running the S-PCCDH algorithm to break two instances of the CCDH problem: $U_1 = Ag^{u_1}$, $V_1 = Bg^{v_1}$ and $X_1 = A^{b_1}$, and $U_2 = Ag^{u_2}$, $V_2 = Bg^{v_2}$ and $X_2 = A^{b_2}$, for random values $u_1, u_2, v_1, v_2, b_1, b_2 \in \mathsf{Z}_p$. We then get four sets, $\mathcal{S}_1$, $\mathcal{S}'_1$, $\mathcal{S}_2$ and $\mathcal{S}'_2$, (instead of values) which contain candidates for $\mathrm{CDH}(X_1, Y_1)$, $\mathrm{CDH}(X_1/U_1, Y_1/V_1)$, $\mathrm{CDH}(X_2, Y_2)$ and $\mathrm{CDH}(X_2/U_2, Y_2/V_2)$ respectively. Therefore, with probability greater than $(n\epsilon^3/128)^2$, each set contains the correct value. Note that

$$\begin{aligned} K_i &= \mathrm{CDH}(X_i, Y_i) = \mathrm{CDH}(A, Y_i)^{b_i} \\ K'_i &= \mathrm{CDH}(X_i/U_i, Y_i/V_i) \\ &= \mathrm{CDH}(X_i, Y_i) \times \mathrm{CDH}(U_i, V_i)/\mathrm{CDH}(X_i, V_i) \times \mathrm{CDH}(U_i, Y_i) \\ &= \mathrm{CDH}(X_i, Y_i) \times \mathrm{CDH}(Ag^{u_i}, Bg^{v_i})/\mathrm{CDH}(A^{b_i}, Bg^{v_i}) \times \mathrm{CDH}(Ag^{u_i}, Y_i) \\ &= \mathrm{CDH}(X_i, Y_i) \times \mathrm{CDH}(A, B) A^{v_i} B^{u_i} g^{u_i v_i}/\mathrm{CDH}(A, B)^{b_i} A^{v_i b_i} \times \\ &\quad \mathrm{CDH}(A, Y_i) Y_i^{u_i} \\ &= \mathrm{CDH}(A, Y_i)^{b_i - 1} \times \mathrm{CDH}(A, B)^{1-b_i} \times A^{v_i(1-b_i)} (B/Y_i)^{u_i} g^{u_i v_i} \end{aligned}$$

$$\begin{aligned}\mathrm{CDH}(A,B)^{b_i-1} &= \mathrm{CDH}(A,Y_i)^{b_i-1} \times A^{v_i(1-b_i)}(B/Y_i)^{u_i} g^{u_i v_i}/K_i' \\ \mathrm{CDH}(A,B) &= \mathrm{CDH}(A,Y_i) \times A^{-v_i}(B/Y_i)^{u_i/(b_i-1)} g^{u_i v_i/(b_i-1)}/{K_i'}^{1/(b_i-1)} \\ &= \left({K_i}^{1/b_i}/{K_i'}^{1/(b_i-1)}\right) \times A^{-v_i}(B/Y_i)^{w_i} g^{w_i v_i},\end{aligned}$$

where $w_i = u_i/(b_i - 1) \bmod p$.

With all the pairs in $\mathcal{S}_1 \times \mathcal{S}_1'$, we build the s^2 candidates for $\mathrm{CDH}(A,B)$, and we do the same with all the pairs in $\mathcal{S}_2 \times \mathcal{S}_2'$. The first collision in the two lists is outputted as the value for $\mathrm{CDH}(A,B)$. But for each pair $(k_i, k_i') \in \mathcal{S}_1 \times \mathcal{S}_1'$, the candidate is

$$\alpha = \left({k_i}^{1/b_i}/{k_i'}^{1/(b_i-1)}\right) \times A^{-v_i}(B/Y_i)^{w_i} g^{w_i v_i} = \left(\frac{k_i}{K_i}\right)^{1/b_i} \times \left(\frac{K_i'}{k_i'}\right)^{1/(b_i-1)}.$$

If $k_i \neq K_i$, or $k_i' \neq K_i'$, it is totally random in G, because of the randomness of b_i. The probability of a wrong collision is thus bounded by $2s^4/p$. ∎

C Proof of Theorem 12

The proof of security for SPAKE2 also uses a hybrid argument consisting of a sequence of experiments, the first of which corresponds to the actual attack. Since the proof of security for SPAKE2 is similar to that of SPAKE1, we only state the differences here.

In $\mathbf{Exp}_1$, we need to change the simulation of oracles in order to allow concurrent executions and to account for the presence of the password in the key derivation function. The claims remain unchanged.

In $\mathbf{Exp}_2$, no changes are required and thus the claims remain unchanged.

In $\mathbf{Exp}_3$, the only change comes from the fact the password is part of the input of H but not of H'. This change, however, does not affect the claims.

In $\mathbf{Exp}_4$, we also need to account for the password being part of the input of H but not of H'. In this case, however, the claims are different. In order to bound the difference in the adversary's success probability in experiments $\mathbf{Exp}_3$ and $\mathbf{Exp}_4$, we use a technique similar to the one used in the proof of security of the protocol MDHKE in [10].

Let $\textsc{AskH}_4$ be the event in which the adversary directly asks the query $H(A, B, \Theta^\star, \Phi^\star, K)$ in experiment $\mathbf{Exp}_4$, where either $\Theta^\star$ or $\Phi^\star$ has been simulated during an active attack. Clearly, this is the only case in which the adversary can distinguish experiment $\mathbf{Exp}_4$ from experiment $\mathbf{Exp}_3$.

In order to upper-bound the probability of event $\textsc{AskH}_4$, let us first define $\textsc{Coll}$ to be the event in which there exist two different values pw_1 and pw_2 such that the tuples $(X^\star, Y^\star, pw_i, \mathrm{CDH}(X^\star/M^{pw_i}, Y^\star/N^{pw_i}))$. are present in the list of queries to H, for $i = 1, 2$. Then, by using a technique similar to that used in Lemma 5 in [10], one can show that

$$\Pr[\textsc{Coll}] \leq q_H^2 \cdot \mathbf{Adv}^{\mathrm{cdh}}_{\mathbb{G}}(t + 3\tau) .$$

Now, let us consider the event $\textsc{AskH}_4$ given that the event $\textsc{Coll}$ does not happen. That is, for each pair $(X^\star, Y^\star)$ involved in an active attack, there is at most one value of pw such that $(X^\star, Y^\star, pw, \mathrm{CDH}(X^\star/M^{pw}, Y^\star/N^{pw}))$ is in the list of queries to H. Since the password pw is

not needed in the simulation of the oracles, we can postpone choosing its value until the very end of the simulation, at which moment we can detect whether the event $\textsc{AskH}_4$ has happened. Hence, the probability that the event $\textsc{AskH}_4$ happens given that the event $\textsc{Coll}$ did not happen can be upper-bound by $\frac{q^A_{\text{send}}+q^B_{\text{send}}}{n}$. That is,

$$\Pr\left[\,\textsc{AskH}_4 \mid \overline{\textsc{Coll}}\,\right] \leq \frac{q^A_{\text{send}} + q^B_{\text{send}}}{n}\ .$$

As a result,

$$\begin{aligned}\Pr[\,\textsc{AskH}_4\,] &\leq \Pr[\,\textsc{AskH}_4 \wedge \textsc{Coll}\,] + \Pr[\,\textsc{AskH}_4 \wedge \overline{\textsc{Coll}}\,] \\ &\leq \Pr[\,\textsc{Coll}\,] + \Pr\left[\,\textsc{AskH}_4 \mid \overline{\textsc{Coll}}\,\right] \\ &\leq q_H^2 \cdot \mathbf{Adv}^{\text{cdh}}_{\mathbb{G}}(t+3\tau) + \frac{q^A_{\text{send}} + q^B_{\text{send}}}{n}\ .\end{aligned}$$

The proof of theorem follows immediately by noticing that the adversary's success probability is exactly $1/2$ in this last experiment. ▮