The Relationship between Password-Authenticated Key Exchange and Other Cryptographic Primitives

Minh-Huyen Nguyen *

Harvard University, Cambridge, MA
mnguyen@eecs.harvard.edu

Abstract. We consider the problem of password-authenticated key exchange (PAK) also known as session-key generation using passwords: constructing session-key generation protocols that are secure against active adversaries (person-in-the-middle) and only require the legitimate parties to share a low-entropy password (e.g. coming from a dictionary of size poly(n)).
We study the relationship between PAK and other cryptographic primitives. The main result of this paper is that password-authenticated key exchange and public-key encryption are *incomparable* under black-box reductions. In addition, we strengthen previous results by Halevi and Krawczyk [14] and Boyarsky [5] and show how to build key agreement and semi-honest oblivious transfer from any PAK protocol that is secure for the Goldreich-Lindell (GL) definition [11].
We highlight the difference between two existing definitions of PAK, namely the indistinguishability-based definition of Bellare, Pointcheval and Rogaway (BPR) [1] and the simulation-based definition of Goldreich and Lindell [11] by showing that there exists a PAK protocol that is secure for the BPR definition and only assumes the existence of one-way functions in the case of exponential-sized dictionaries. Hence, unlike the GL definition, the BPR definition does not imply semi-honest oblivious transfer for exponental-sized dictionaries under black-box reductions.

1 Introduction

The problem of *password-authenticated key exchange* (PAK), also known as *session-key generation using passwords*, is to enable private communication between two legitimate parties over an insecure channel in the setting where the legitimate parties have only a small amount of shared information, i.e. a low-entropy key such as an ATM pin or a human-chosen password. In addition to its practical implications, the problem of session-key generation using passwords is quite natural as it focuses on the minimal amount of information that two parties must share in order to perform non-trivial cryptography. A recent series of works [1, 6, 11, 17, 8, 20, 7] has focused on our theoretical understanding of

* Supported by NSF grants CCR-0205423, CNS-0430336, and ONR grant N00014-04-1-0478

this PAK problem by proposing several definitions of security as well as secure protocols. Bellare, Pointcheval and Rogaway [1] proposed a definition based on the indistinguishability of the session key. Following the simulation paradigm for secure multi-party computation, Boyko, MacKenzie and Patel [6] and Goldreich and Lindell [11] gave their own simulation-based definitions. However, it is not clear how these existing definitions of security for PAK relate to one another.

The first protocols for the password-authenticated key exchange problem were proposed in the security literature, based on informal definitions and heuristic arguments (e.g. [4, 24]). The first rigorous proofs of security were given in the random oracle model by [1, 6]. Only recently were rigorous solutions without random oracles given, in independent works by Goldreich and Lindell [11] (under the assumption that trapdoor permutations exist) and Katz, Ostrovsky, and Yung [17] (under number-theoretic assumptions). Subsequently, the protocol of [11] was simplified in [20] and the protocol of [17] was generalized in [8, 7].

What is the minimal assumption needed to solve PAK? How does this problem relate to other basic cryptographic primitives such as key agreement and oblivious transfer? These are natural questions to ask when considering the problem of password-authenticated key exchange. The goal of this paper is to study the relationship between PAK and other cryptographic primitives as well as try to explain how the existing definitions of security for PAK relate to one another. Next, we informally describe the problem of password-authenticated key exchange.

Password-Authenticated Key Exchange. The problem of session-key generation using passwords suggested by Bellovin and Merritt [3] considers the situation where Alice and Bob share a password, i.e. an element chosen uniformly at random from a small dictionary $\mathcal{D} \subseteq \{0,1\}^n$. This dictionary can be very small, e.g. $|\mathcal{D}| = \text{poly}(n)$, and in particular it may be feasible for an adversary to exhaustively search it. The aim is to construct a protocol enabling Alice and Bob to generate a "random" session key $K \in \{0,1\}^n$ which they can subsequently use for standard private-key cryptography. We consider an active adversary that completely controls the communication channel between Alice and Bob and in particular can attempt to impersonate either party through a person-in-the-middle attack.

The goal of a PAK protocol is that, even after the adversary mounts such an attack, Alice and Bob will generate a session key that is indistinguishable from uniform even given the adversary's view. However, our ability to achieve this goal is limited by two unpreventable attacks. First, the adversary can block all communication, so it can prevent one or both of the parties from completing the protocol and obtaining a session key. Second, the adversary can choose a password $\tilde{w}$ uniformly at random from $\mathcal{D}$ and attempt to impersonate one of the parties. With probability $1/|\mathcal{D}|$, the guess equals the real password (i.e., $\tilde{w} = w$), and the adversary will succeed in the impersonation and therefore learn the session key. Thus, we revise the goal to effectively limit the adversary to these two attacks.

Our Results. Our goal in this paper is to understand the relationship between session-key generation using passwords and other well-known cryptographic primitives. Doing so will help us characterize the complexity of PAK and place this problem within our current view of cryptography. In this work we study the relationship of PAK to *public-key encryption* (PKE), *oblivious transfer* (OT) and *key agreement* (KA). We provide positive results, e.g. exhibit a reduction of KA to PAK, as well as negative results, e.g. prove that PAK does not imply PKE under black-box reductions.

Following the oracle separation paradigm of [15], we first separate PAK and PKE by constructing an oracle Γ relative to which PAK exists but PKE does not.

Theorem 1. *There is no "black-box" construction of PKE from PAK for the Goldreich-Lindell (GL) definition [11].*

Loosely speaking, a black-box construction of the primitive Q from the primitive P is a construction of Q out of P which does not use the code of the implementation of P [1]. We note that similarly to most separation results, Theorem 1 and our other separation results only apply to uniform adversaries. We actually prove Theorem 1 using a definition of PAK which is stronger than the GL definition in order to strengthen the result. This separation result can also be seen in a positive way since it provides a direction for proving implications. In order to prove that PAK implies PKE, one must use non-black-box techniques, for example by using the code of an adversary for the PKE protocol.

We then exhibit a reduction of *semi-honest OT* [2] to PAK for the GL definition.

Theorem 2. *The existence of a PAK-protocol that is secure for the GL definition implies semi-honest OT (via a black-box reduction). Moreover, this reduction does not depend on the size of the dictionary $\mathcal{D}$ and holds even for dictionaries of exponential size such as $\mathcal{D} = \{0,1\}^n$.*

The proof of Theorem 2 actually uses only the weaker definition of [20] where the security holds for a *specific* dictionary $\mathcal{D}$ and the probability of breaking is bounded by $\frac{1}{\omega(\log n)}$ instead of $O\left(\frac{1}{|\mathcal{D}|}\right)$, which strengthens the result.

Combining Theorem 2 and the result of Gertner et al. [9] that there is no black-box construction of semi-honest OT from PKE, we obtain the following corollary:

Corollary 1. *There is no black-box construction of GL-secure PAK from PKE.*

[1] We refer the reader to Section 2.3 and [22] for a more formal definition of black-box reductions. In the taxonomy of [22], we are considering *semi* black-box reductions.

[2] In the honest (but curious) or semi-honest model, the parties Alice and Bob are guaranteed to follow the protocol but might use their views of the interaction in order to compute some additional information.

Putting Theorem 1 and Corollary 1 together, we obtain that PAK and PKE are *incomparable* under black-box reductions. This is similar to the result of [9] that OT and PKE are incomparable under black-box reductions, and thus provides an additional motivation to try and establish the equivalence of PAK and OT, as conjectured in [5]. Indeed, the protocol proposed by Goldreich and Lindell is actually based on the existence of oblivious transfer and one-way permutations. Theorem 2 shows that if one can bypass the use of one-way permutations (for example by using one-way functions instead of one-way permutations) and build a secure PAK protocol from oblivious transfer only, then PAK and OT are equivalent [3].

The question of the relationship between PAK to KA is particularly interesting as the PAK and KA problems are very similar in essence: both problems consider honest parties A and B who wish to generate a common random session key K. In the case of PAK, the honest parties have to withstand an active adversary and share a low-entropy password whereas in the case of KA, the honest parties have to withstand a passive adversary and share no prior information. Combining Theorem 2 and the previous result by Gertner et al. [9] that semi-honest OT is strictly stronger than KA under black-box reductions, we obtain the following corollary:

Corollary 2. *The existence of a PAK protocol that is secure for the GL definition implies KA (via a black-box reduction). Moreover, this reduction does not depend on the size of the dictionary $\mathcal{D}$ and holds even for dictionaries of exponential size such as $\mathcal{D} = \{0,1\}^n$.*

Combining Corollary 1 and the previous result by Gertner et al. [9] that PKE implies KA (via a black-box reduction), we obtain the following corollary:

Corollary 3. *There is no black-box construction of GL-secure PAK from KA.*

Again, Corollary 3 can be seen in a positive way: to prove that KA implies PAK, one must use non-black-box techniques.

Theorem 2 also enables us to understand the relationship between existing definitions of security and in particular to highlight a difference between the simulation-based definition of [11] and the indistinguishability-based definition of [1]. Indeed, we have the following result:

Theorem 3. *If one-way functions exist, there exists a PAK protocol that is secure for the Bellare-Pointcheval-Rogaway (BPR) definition [1] for the dictionary $\mathcal{D} = \{0,1\}^n$.*

Hence, unlike the GL definition, the BPR definition does not imply honest OT in the case of exponential-sized dictionaries under black-box reductions. However we conjecture that any PAK protocol that is secure for the BPR definition for *polynomial-sized dictionaries* implies semi-honest OT.

[3] This equivalence would be non-black-box as the known construction of OT from honest OT is non-black-box since it uses the zero-knowledge proofs of [12] (see [9]).

Related Work. Although the relationship between PAK and other cryptographic primitives has not been explicitly studied before, some results are known for the related problem of *password-based authentication*, where the legitimate parties only want to be convinced that they are talking to one another (but not generate a common session key). Assuming the existence of one-way functions, it is known that one can transform a PAK protocol into a protocol for password-based authentication using two additional messages [2, 1, 16, 11, 17].

Halevi and Krawczyk [14] showed that a secure protocol for password-based authentication[4] can be used to implement KA. We see Corollary 2 as a strengthening of their result, since our result holds even for dictionaries of exponential size whereas their reduction only holds for polynomial-sized dictionaries.

Boyarsky [5] states without proof that password-based authentication[5] implies OT, which is similar to Theorem 2 . However, [5] does not provide a formal definition of PAK for which this implication holds, and indeed, our results show that the relationship between PAK and OT *is* sensitive to the choice of definition. Moreover, our black-box construction of semi-honest OT from a secure PAK protocol holds even if we relax the security of the PAK protocol in two respects. First, it holds even if the PAK protocol is secure only for a fixed dictionary of exponential size, e.g. $\mathcal{D} = \{0,1\}^n$. Second, we only require that the probability of breaking the PAK protocol be bounded by $\frac{1}{\omega(\log n)}$ (on security parameter 1^n) instead of $O\left(\frac{1}{|\mathcal{D}|}\right)$.

2 Preliminaries

We denote by n the security parameter, by U_n the uniform distribution over strings of length n, by $\mathrm{neg}(n)$ a negligible function and write $x \stackrel{\mathrm{R}}{\leftarrow} S$ when x is chosen uniformly from the set S. We use the abbreviation "PPT" for probabilistic polynomial-time algorithms.

Since we will prove our results for uniform adversaries, our definitions are for the uniform model of computation. An ensemble $X = \{X_n\}_{n\in\mathbb{N}}$ is *(polynomial-time) samplable* if there exists a PPT algorithm M such that for every n, the random variables $M(1^n)$ and X_n are identically distributed.

Let S be a set of strings. For a function $\gamma : \mathbb{N} \to [0,1]$, we say that the probability ensembles $\{X_w\}_{w\in S}$ and $\{Y_w\}_{w\in S}$ are $(1-\gamma)$-*indistinguishable* (denoted by $\{X_w\} \stackrel{\gamma}{\equiv} \{Y_w\}$) if for every PPT algorithm D, for all sufficiently large n, for every $w \in \{0,1\}^n \cap S$,

$$|\Pr[D(X_w, w) = 1] - \Pr[D(Y_w, w) = 1]| < \gamma(n) + \mathrm{neg}(n)$$

In the proofs, we will slightly abuse notation when talking about a distribution's index w by writing "for every $w \in S$" and omitting the index w as an

[4] Their result is for password-based one-way authentication where one clients tries to authenticate itself to a server.

[5] This result is for password-based mutual authentication where two honest parties try to authenticate each other.

input to the distinguisher D. We say that $\{X_w\}$ and $\{Y_w\}$ are *computationally indistinguishable*, which we denote by $X_w \stackrel{c}{\equiv} Y_w$, if they are 1-indistinguishable.

2.1 Cryptographic Primitives

Two-party Protocols The following is an informal presentation of two-party computation which will suffice for our purposes. Recall that we are interested in protocols for *semi-honest* oblivious transfer and key-agreement for which we are guaranteed that the two parties follow the protocol. We refer the reader to [10] for more details.

A two-party protocol problem is defined by specifying a (possibly probabilistic) functionality $f : \{0,1\}^\star \times \{0,1\}^\star \to \{0,1\}^\star \times \{0,1\}^\star, (x,y) \to (f_1(x,y), f_2(x,y))$ which maps pairs of inputs to pairs of outputs. A two-party protocol is a pair of probabilistic polynomial-time algorithms (A,B) which represent the strategies of the two parties, i.e. functions that map a party's input, private randomness and the sequence of messages received so far to the next message to be sent. The view of a party consists of its input, its random-tape and the sequence of messages received. We measure the amount of interaction in a protocol by its number of rounds, where a round consists of a single message sent from one party to another. Whenever we consider a protocol for securely computing a functionality f, we assume that the protocol correctly computes f when both parties follow the protocol, i.e. the joint output distribution of the protocol played by parties following the protocol on input pair (x,y) equals the distribution of $f(x,y)$.

Semi-honest Oblivious Transfer In the semi-honest model, the two parties A and B are guaranteed to follow the protocol but might use their views of the interaction in order to learn some additional information. As noted in [9], in the semi-honest model, one can transform an OT protocol for bits into an OT protocol for strings without increasing the number of rounds. We will therefore focus on the version of OT where s_0 and s_1 are bits rather than strings. 1-out-of-2 oblivious transfer (OT) is the following two-party functionality:

– Inputs: A has the security parameter 1^n and two secret bits s_0 and s_1. B has the security parameter 1^n and a selection bit c.
– Outputs: A outputs nothing, B outputs s_c.

A protocol (A,B) for semi-honest OT is *secure* if there exists a pair of PPT $(\tilde{A}, \tilde{B})$ such that:

– *Receiver's privacy*: for every s_0, s_1, c, $\tilde{A}(1^n, s_0, s_1)$ is computationally indistinguishable from A's view of the interaction $(A(1^n, s_0, s_1), B(1^n, c))$
– *Sender's privacy*: for every s_0, s_1, c, $\tilde{B}(1^n, b, s_b)$ is computationally indistinguishable from B's view of the interaction $(A(1^n, s_0, s_1), B(1^n, c))$

Key Agreement Key agreement (KA) is the following two-party functionality:

- Inputs: A and B have the security parameter 1^n.
- Outputs: A and B output the same string K of length n

A KA protocol is *secure* if we have $(T, K) \stackrel{c}{\equiv} (T, U_n)$ where T is the transcript of the interaction $(A(1^n), B(1^n))$ and K is the common output of A and B in the interaction $(A(1^n), B(1^n))$. In other words, the session key K will be computationally indistinguishable from a truly random string given the view of a passive adversary.

2.2 Password-Authenticated Key Exchange

Password-authenticated key exchange (PAK) or session-key generation using passwords is similar to key agreement in that two honest parties A and B want to generate a session key K of length n that is indistinguishable from uniform even given the adversary's view. However, PAK differs from KA in two important respects. First, A and B have as input a shared password w which is chosen at random from a dictionary $\mathcal{D} \subseteq \{0,1\}^n$. Second, the adversary is not passive but completely controls the communication channel between A and B.

The Goldreich-Lindell Definition and its Variants The definition of PAK in [11] follows the standard paradigm for secure computation: define an ideal functionality (using a trusted third party) and require that every adversary attacking the real protocol can be simulated by an ideal adversary attacking the ideal functionality. In the real protocol, an active adversary can prevent one or both of the parties from completing the protocol. Thus, in the ideal model, we will allow C_{ideal} to specify an input bit dec_B, which determines whether B obtains a session key or not [6]. We can therefore cast PAK as a three-party functionality which is described in the ideal model as follows.

Ideal model

- Inputs: A and B receive a security parameter 1^n and a joint password $w \stackrel{R}{\leftarrow} \mathcal{D}$.
- A and B both send w to the trusted party. C_{ideal} sends a decision bit dec_B to the trusted party to indicate whether B's execution is successful or not.
- Outputs: The trusted party chooses $K \stackrel{R}{\leftarrow} \{0,1\}^n$ and sends it to A. If $\text{dec}_B = 1$, then the trusted party sends K to B; otherwise it sends $\perp$ to B.

The ideal distribution of inputs and outputs is defined by:

$$\text{IDEAL}_{C_{\text{ideal}}}(\mathcal{D}) = (w, \text{output}(A), \text{output}(B), \text{output}(C_{\text{ideal}}))$$

[6] We will adopt the convention that A always completes the protocol and accepts.

Real model Let A, B be the honest parties and let C be any PPT real adversary. In an initialization stage, A and B receive $w \xleftarrow{\text{R}} \mathcal{D}$. The real protocol is executed by A and B communicating via C. We will augment C's view of the protocol with B's decision bit, denoted by dec_B, where $\text{dec}_B = \texttt{reject}$ if $\text{output}(B) = \perp$, and $\text{dec}_B = \texttt{accept}$ otherwise (indeed in typical applications, the decision of B will be learned by the real adversary C: if B obtains a session key, then it will use it afterwards; otherwise, B will stop communication or try to re-initiate an execution of the protocol). C's augmented view is denoted by $\text{view}(C^{A(w),B(w)})$.

The real distribution of inputs and outputs is defined by:

$$\text{REAL}_C(\mathcal{D}) = (w, \text{output}(A), \text{output}(B), \text{view}(C^{A(w),B(w)}))$$

One might want to say that a PAK protocol is secure if the above ideal and real distributions are computationally indistinguishable. Unfortunately as mentioned above, an active adversary can guess the password and successfully impersonate one of the parties with probability $\frac{1}{|\mathcal{D}|}$. This implies that the real and ideal distributions are always distinguishable with probability at least $\frac{1}{|\mathcal{D}|}$ so we will only require that the distributions be distinguishable with probability at most $O\left(\frac{1}{|\mathcal{D}|}\right)$. In the case of a passive adversary, we require that the real and ideal distributions be computationally indistinguishable (for all subsequent definitions, this requirement will be implicit):

Definition 1. *[11] A protocol for password-based session-key generation is secure if for every samplable dictionary $\mathcal{D} \subseteq \{0,1\}^n$, for every real adversary C, there exists an ideal adversary C_{ideal} such that the ideal and real distributions are $\left(1 - O\left(\frac{1}{|\mathcal{D}|}\right)\right)$-indistinguishable* [7].

Although standard definitions of security for PAK protocols require that the security hold for *every* dictionary, we will consider two variants of the standard GL definition (Definition 1) where we change the security to hold for a *specific* dictionary $\mathcal{D}$ instead of every dictionary. Moreover, we will only require that the distributions be distinguishable with probability at most γ, where γ is a function of the dictionary size $|\mathcal{D}|$ and the security parameter n, and not necessarily $O\left(\frac{1}{|\mathcal{D}|}\right)$.

Although these variants of Definition 1 are weaker, a PAK protocol which is secure for a specific dictionary is still interesting since it corresponds to the setting where the honest parties are restricted to choose their passwords from a specific dictionary, such as in the case of ATM pin numbers[8]. Moreover, as noted

[7] As pointed out by Rackoff, this basic definition is actually not completely satisfactory and needs to be augmented to take into account any use of the key K by one party while the other party has not completed the protocol. Our results will hold for the augmented definition as well but we will not handle the augmented definition explicitly.

[8] By restricting our attention to a specific dictionary, it may be possible to obtain a more efficient protocol, such as the [20] simplification of [11].

in [20], such a PAK protocol can be converted into one for arbitrary dictionaries in the common reference string model (using the common reference string as the seed of a randomness extractor [21]).

Definition 2. *[20] Let $\mathcal{D} \subseteq \{0,1\}^n$ be a samplable dictionary. A protocol for password-based session-key generation is* $(1-\gamma)$-GL-secure *for the dictionary $\mathcal{D}$ (where γ is a function of the dictionary size $|\mathcal{D}|$ and n) if for every real adversary C, there exists an ideal adversary C_{ideal} such that the ideal and real distributions are $(1-\gamma)$-indistinguishable.*

Our goal is to make γ as small as possible. Ideally, we would like $\gamma = O\left(\frac{1}{|\mathcal{D}|}\right)$. Note that Definition 2 guarantees that the password w is $(1-\gamma)$ indistinguishable from a random password $\tilde{w} \stackrel{\text{R}}{\leftarrow} \mathcal{D}$ since C_{ideal} learns nothing about the password w which is explicitly in the ideal distribution.

Security with respect to password guesses A stronger definition of security can be obtained by allowing the ideal adversary some number of password guesses but requiring that the ideal and real distributions be computationally indistinguishable. We will therefore modify the ideal model by adding the following steps after A and B receive their inputs:

- C_{ideal} sends its (possibly adaptive) guesses for the password $w_1, \cdots, w_\alpha$ to the trusted party. The trusted party answers whether the guesses are correct or not.
- If the adversary C_{ideal} guesses the password correctly, C_{ideal} can force the outputs of A and B to be whatever it wants.

The modified ideal distribution for α password guesses is defined by:

$$\text{IDEAL}^{\text{Guess}}_{C_{\text{ideal}}}(\mathcal{D}) = (w, \text{output}(A), \text{output}(B), \text{output}(C_{\text{ideal}}))$$

Definition 3 (Security with respect to α password guesses). *Let $\mathcal{D} \subseteq \{0,1\}^n$ be a samplable dictionary. A protocol for password-based session-key generation is* secure with respect to α password guesses *for the dictionary $\mathcal{D}$ if for every real adversary C, there exists an ideal adversary C_{ideal} making at most α password guesses such that the ideal and real distributions are computationally indistinguishable.*

Note that the ideal model in the definition of security with respect to password guesses can be simulated by the ideal model in Definition 2 with probability $\left(1 - \frac{\alpha}{|\mathcal{D}|}\right)$. Hence we obtain that the definition of security with respect to password guesses is stronger than Definition 2:

Proposition 1. *Security with respect to α password guesses implies GL-security with $\gamma = \frac{\alpha}{|\mathcal{D}|}$.*

In Section 3, we will show that even the *stronger* definition of security with respect to password guesses does not imply PKE under black-box reductions. In Section 4, we will show that the *weaker* GL definition (Definition 2) implies semi-honest OT.

Other Definitions Bellare, Pointcheval and Rogaway [1] introduced a definition based on the *indistinguishability* of the session key. In this model, there are not just two honest parties as in the previous definitions but rather a set of honest parties (called principals) that are either a *client* or a *server*. Each client has some password $w \xleftarrow{R} \mathcal{D}$ and each server has the passwords of the clients.

The interaction of the adversary with the principals is modeled using oracle queries. Each principal is modeled by a collection of oracles that represent all possible actions, such as passive eavesdropping (the adversary sees the transcript of a protocol execution between a client and a server), corruption of a party (the adversary obtains the client's password), loss of session keys (the adversary learns the session key generated by a protocol execution) and person-in-the-middle attack (the adversary sends messages of its choosing to a principal). The adversary is allowed to make these oracle queries to any principal and there might be several instances of the same principal U that model concurrent executions.

The adversary chooses a *test* concerning the instance i of an uncorrupted principal U: a bit b is chosen uniformly from $\{0, 1\}$. If $b = 0$, then the adversary is given the session key output by the instance i of the principal U. If $b = 1$, the adversary is given a truly random key. A PAK protocol is secure for the dictionary $\mathcal{D}$ according to the BPR definition if after mounting at most q person-in-the-middle attacks, the adversary has advantage at most $O\left(\frac{q}{|\mathcal{D}|}\right) + \mathrm{neg}(n)$ in distinguishing the true session key from a random key in this test.

Boyko, MacKenzie and Patel [6] proposed a *simulation-based* definition which allows the ideal adversary to make a constant number of password guesses to the trusted party. The BMP definition is similar to the definition of security with respect to password guesses (in fact, the definition of security with respect to password guesses was inspired by the BMP definition) but their model differs from ours in two important respects. First, there are not just two honest parties executing the protocol but rather a set of honest users. Each user may have several instances that model concurrent executions of the protocol. Second, the ideal and real distributions in this model *do not include the passwords*. Loosely speaking, a PAK protocol is secure according to the BMP definition if for every real adversary, there exists an ideal adversary such that the ideal and real distributions are computationally indistinguishable.

Both the BPR and BMP definitions present some advantages over the GL definition because they handle concurrent executions easily. However, unlike the GL definition, these definitions do not explicitly guarantee that the password w remain pseudorandom after an execution. For example, the first bit of the password w could be revealed during an execution. This distinction is important as we will show that unlike the GL definition, the BPR definition does not imply semi-honest OT for exponential-sized dictionaries. Indeed, in Section 5 we exhibit a PAK protocol that is secure according to the BPR definition for the dictionary $\mathcal{D} = \{0, 1\}^n$ but only assumes the existence of one-way functions. In particular, the password w does not remain pseudorandom after an execution of this protocol.

2.3 Black-box Reductions

We give an informal presentation of black-box reductions that will suffice for our purposes. For more details, we refer the reader to [22]. The function (or algorithm) $f : \{0,1\}^\star \to \{0,1\}^\star$ is an implementation of a primitive P if it satisfies the structural requirements of the primitive (for example, in the case of one-way permutations, we require that f be a length-preserving permutation). We do not require that the implementation f satisfy some security requirements.

A black-box reduction of Q to P is the construction of two PPT oracle machines G and S such that:

- If f is an implementation of P (not necessarily efficient), then G^f is an implementation of Q.
- For every adversary A (not necessarily efficient) that breaks the implementation G^f, $S^{A,f}$ breaks the implementation f.

A black-box reduction relativizes, hence to show that there are no black-box reductions of Q to P, it suffices to construct an oracle relative to which P exists but Q does not.

3 There is no black-box construction of PKE from PAK

3.1 Overview of the result

Theorem 4. *There exists an oracle Γ relative to which PAK exists but PKE does not.*

The oracle Γ we will use is composed of the following parts:

- f_1, f_2 and f_3 are three uniformly distributed length-tripling injective functions.
- The function R is defined to satisfy $R(w, s, \alpha) = K$ whenever $\alpha = f_3(w, K, r, f_2(w, s, f_1(w, K, r)))$ for some $|K| = |r| = |s| = |w|$, $\perp$ otherwise (R is well-defined since the f_i's are injective).
- a **PSPACE**-complete oracle

We now describe a PAK protocol using Γ:

Protocol 1. 1. Inputs: A and B have a security parameter 1^n and a joint password $w \in \mathcal{D} \subseteq \{0,1\}^n$, where $\mathcal{D}$ is samplable.
2. A chooses two n-bit strings $K_A, r_A \overset{\text{R}}{\leftarrow} \{0,1\}^n$ and sends $\alpha_1 \overset{\text{def}}{=} f_1(w, K_A, r_A)$. B receives β_1.
3. B chooses $r_B \overset{\text{R}}{\leftarrow} \{0,1\}^n$ and sends $\beta_2 \overset{\text{def}}{=} f_2(w, r_B, \beta_1)$. A receives α_2.
4. A sends $\alpha_3 \overset{\text{def}}{=} f_3(w, K_A, r_A, \alpha_2)$. B receives β_3.

5. Outputs: A outputs K_A. B outputs $R(w, r_B, \beta_3)$.

Note that in this protocol A always accepts and B accepts iff $R(w, r_B, \beta_3) \neq \perp$.

We prove Theorem 4 via the following two lemmas. The first lemma establishes that relative to Γ, PAK exists.

Lemma 1. *Protocol 1 is secure with respect to 2 password guesses for the dictionary $\mathcal{D}$, i.e. for every real adversary C, there exists an ideal adversary C_{ideal} with 2 password guesses such that $\text{REAL}_C(\mathcal{D}) \stackrel{c}{\equiv} \text{IDEAL}^{\text{Guess}}_{C_{\text{ideal}}}(\mathcal{D})$, where the probabilities are also taken over the random choice of Γ.*

The proof of Lemma 1 is quite involved and can be found in the full version of the paper [19]. We try to give the main idea of the proof in the section below.

It is known that PKE and 2-round KA are equivalent [9]. Thus, to prove Theorem 4, it suffices to prove that relative to Γ, there is no secure 2-round KA protocol.

Lemma 2. *For every 2-round KA protocol (A, B), for every polynomial p, there exists a passive adversary E such that the probability over Γ and the random tapes of A, B and E that E outputs the session key is at least $1 - \frac{1}{n^2 p(n)}$.*

In other words, with overwhelming probability, any 2-round KA protocol is not secure since there exists a passive adversary E that is able to distinguish the session key from a truly random string. The proof of Lemma 2 is very similar to that of [15, 23] and can be found in the full version of the paper [19]. Using Lemmas 1 and 2, we show that with probability 1 over the random choice of Γ, Protocol 1 is secure with respect to 2 password guesses and there exists no secure 2-round KA protocol. This establishes Theorem 4.

3.2 Relative to Γ, PAK exists

We give some intuition on how to prove that Protocol 1 is secure with respect to 2 password guesses. For every real adversary C, we need to exhibit an ideal adversary with 2 password guesses which simulates C's view. We will follow the paradigm of [6] and show how to transform some of the real adversary's queries to the oracle Γ into password guesses for the ideal adversary.

The ideal adversary C_{ideal} will run the real adversary C and simulate the honest parties A and B. Using the queries made by C to the oracle Γ and the messages sent by C, C_{ideal} will determine if password guesses need to be made and if so, forward these password guesses to the trusted party. The output of C_{ideal} will be C's view of this simulated execution and we show that C's view of this execution simulated by C_{ideal} produces a view which is computationally indistinguishable from C's view of a real execution with A and B.

- As long as no password guess has been successful, C_{ideal} will simulate the honest parties by sending random strings of appropriate length. Intuitively, in this case, the messages sent by the honest parties A and B in a real execution are computationally indistinguishable from random strings with respect to the real adversary's view.

– If a password guess has been successful, C_{ideal} will have the password w and intuitively C will simulate the honest parties A and B perfectly.

We now show how to transform some of the real adversary's queries to Γ into password guesses for the ideal adversary. When C makes a query to the oracle Γ, C_{ideal} makes this query to Γ and records the query/answer pair. Recall that an active adversary C can mount a person-in-the-middle attack that effectively gives two concurrent executions of the PAK protocol, one between A and C and one between C and B. We denote by α_i the ith message in the (A, C) interaction and by β_i the ith message in the (C, B) interaction. We define *password guesses in a real interaction* of C with A and B as follows.

– Password guess in the (C, B) interaction: C impersonates A on input $w' \in \mathcal{D}$ by sending to B the messages $\beta_1 = f_1(w', K_C, r_C)$ for some pair $(K_C, r_C) \in \{0,1\}^n \times \{0,1\}^n$ and $\beta_3 = f_3(w', K_C, r_C, \beta_2)$. C's guess w' is correct if $dec_B = 1$.
– Password guess in the (A, C) interaction: C impersonates B on input $w'' \in \mathcal{D}$ by sending to A the message $\alpha_2 = f_2(w'', r_C, \alpha_1)$ for some string $r_C \in \{0,1\}^n$. C's guess w'' is correct if $R(w'', r_C, \alpha_3) \neq \perp$

We can turn these cases in the real model into *password guesses in the ideal model*:

– Password guess in the simulated (C, B) interaction:
If C sends $\beta_1 = f_1(w', K_C, r_C)$ for some previous query (w', K'_C, r_C) made by C to f_1, C_{ideal} sends its guess w' to the trusted party.
– Password guess in the simulated (A, C) interaction:
If C sends $\alpha_2 = f_2(w'', r_C, \alpha_1)$ for some previous query (w'', r_C, α_1) made by C to f_2, C_{ideal} sends its guess w'' to the trusted party.

4 GL-security for PAK implies semi-honest OT

Theorem 5. *Let $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ be a samplable ensemble such that $\mathcal{D}_n \subseteq \{0,1\}^{\text{poly}(n)}$. The existence of a $(1-\gamma)$-GL-secure PAK protocol for the dictionary $\mathcal{D}_n$ on security parameter 1^n such that $\gamma \leq \frac{1}{5f(n)}\left(1 - \frac{1}{t(n)}\right)$ for some function $f(n) = \omega(\log n)$ and some polynomial t implies semi-honest OT (via a black-box reduction).*

Note that we do not require many dictionaries for a single security parameter 1^n, but rather a single fixed dictionary $\mathcal{D}_n$ for a given security parameter 1^n.

In order to prove Theorem 5, we consider a protocol for "*Weak OR*" (WOR). A WOR protocol (A, B) computes the functionality of the standard OR but its security is weak. More formally, a protocol (A, B) for weak OR is $(1-\eta)$-*secure* if

– *B's privacy*: A's view of $(A(1^n, 1), B(1^n, 0))$ is $(1-\eta)$-indistinguishable from $(A(1^n, 1), B(1^n, 1))$.

– *A's privacy*: B's view of $(A(1^n, 0), B(1^n, 1))$ is $(1-\eta)$ indistinguishable from $(A(1^n, 1), B(1^n, 1))$.

As suggested by Boyarsky [5], we establish Theorem 5 in two steps:

1. We first prove that a PAK protocol that is $(1-\gamma)$-GL-secure can be used to build a WOR protocol that is $(1-5\gamma)$-secure.
2. We then show that a WOR protocol that is $(1-\eta)$-secure for $\eta \leq \frac{1}{f(n)}\left(1 - \frac{1}{t(n)}\right)$ can be used to build a secure protocol for semi-honest OT. Our proof that WOR implies semi-honest OT is similar to the proof of Kilian [18] that OR implies OT but the two results are incomparable since we restrict our focus to the semi-honest setting but are given a weaker OR primitive.

4.1 GL-security for PAK implies Weak OR

Given a $(1-\gamma)$-GL-secure PAK protocol (A_P, B_P) for the samplable dictionary $\mathcal{D} \subseteq \{0,1\}^n$, we build the following WOR protocol.

Protocol 2. 1. Inputs: A has a bit a, B has a bit b.
2. A chooses $w, w' \xleftarrow{\text{R}} \mathcal{D}$ and sends w to B. B chooses $w'' \xleftarrow{\text{R}} \mathcal{D}$. (This is where we use the assumption that $\mathcal{D}$ is samplable.)
3. A and B run the PAK protocol (A_P, B_P) on inputs w_A and w_B respectively, where w_A and w_B are defined as follows:
   – If $a = 0$, A_P sets its password w_A to be w. Otherwise $w_A = w'$.
   – If $b = 0$, B_P sets its password w_B to be w. Otherwise $w_B = w''$.

   At the end of the PAK protocol, B sends its decision bit dec_B to A.
4. Outputs: If $a = 1$ and $\mathrm{dec}_B = 1$, then A sends a message to B to set the output to be 1. Similarly, if $b = 1$ and $\mathrm{dec}_B = 1$, then B sends a message to A to set the output to be 1. Otherwise, the common output of the execution is set to be $(1 - \mathrm{dec}_B)$.

Analysis Sketch. Note that Protocol 2 computes the OR functionality correctly. If $a = b = 0$, then the passwords w_A and w_B are both equal to w and by definition of the PAK protocol, B will accept and the common output of A and B will be $1 - \mathrm{dec}_B = 0$. If $a = 1$ or $b = 1$, we know that either B rejects (and the common output will be $1 - \mathrm{dec}_B = 1$) or one of the parties will send an additional message and set the output to be 1.

If $a = 1$, then $\mathrm{OR}(a, b) = 1$ regardless of the value of b so A should not learn B's input. Indeed, we will show that A's view of the interaction $(A(1), B(0))$ is $(1 - 5\gamma)$-indistinguishable from A's view of the interaction $(A(1), B(1))$ (the reasoning for B is similar).

We first consider A's view of the interaction $(A(1), B(b))$ when the possible additional message sent by B in Step 4 is not included. This possibly truncated view of A of the interaction $(A(1), B(b))$ is $(w, A_P(w')^{B_P(w_B)})$ where the second component refers to A_P's view of the PAK protocol and w_B is either w or w''. Because w' is independent of w_B, we can think of $A_P(w')$ as a real adversary

C for the PAK protocol that interacts with the honest party $B_P(w_B)$. Since the PAK protocol is $(1-\gamma)$-GL-secure, we can show that even if the adversary C is given w, C cannot distinguish the case $w_B = w$ from the case $w_B = w''$ with probability greater than 2γ. This is because in the ideal model, an ideal adversary learns nothing about the password w_B.

If B sends an additional message after the execution of the PAK protocol, then we know that B's input is $b = 1$, which makes A's views of $(A(1), B(0))$ and $(A(1), B(1))$ distinguishable. Recall that B sends an additional message iff B_P accepts in an execution of the PAK protocol where A_P has input $w' \stackrel{\mathrm{R}}{\leftarrow} \mathcal{D}$ and B_P has input $w'' \stackrel{\mathrm{R}}{\leftarrow} \mathcal{D}$. Because w' is independent of w'', we can think of $A_P(w')$ as a real adversary C for the PAK protocol that interacts with the honest party $B_P(w'')$. Since the PAK protocol is $(1-\gamma)$-GL-secure, we can show that an adversary C makes B accept (and B sends an additional message after the execution of the PAK protocol) with probability at most 3γ.

4.2 Weak OR implies semi-honest OT

Lemma 3. *The existence of a WOR protocol that is $(1-\eta)$-secure for $\eta \leq \frac{1}{f(n)}\left(1 - \frac{1}{t(n)}\right)$ for some function $f(n) = \omega(\log n)$ and some polynomial t (where 1^n is the security parameter) implies honest OT (via a black-box reduction).*

In order to prove Lemma 3, we introduce a two-party functionality called "*Weak OT*" (WOT). A protocol (A, B) for weak OT is similar to a protocol for OT except that

- B does not choose which secret bit it will obtain. That is, B has no input except for the security parameter 1^n and when interacting with $A(1^n, s_0, s_1)$, B's output is (c, s_c) for a random bit c.
- For every s_0, s_1 and a random bit c, $\tilde{B}(1^n, c, s_c)$ is $(1-\epsilon)$ indistinguishable from B's view of the interaction $(A(1^n, s_0, s_1), B(1^n))$, where $\epsilon \leq 1 - \frac{1}{t(n)}$ for some polynomial t. In other words, the sender's privacy only holds with probability $(1-\epsilon)$.

Kilian [18] showed how to build a protocol for OT from a secure protocol for OR in two steps:

1. Using a secure protocol for OR, we first build a protocol for weak OT
2. Using a protocol for weak OT, we then build a protocol for OT

To prove Lemma 3, we adapt these two steps to our weak OR primitive. We strengthen the first step of [18] and show how to build a protocol for weak OT given a weak OR protocol that is $(1-\eta)$-secure. More precisely, we first show how to use a weak OR protocol to build a protocol for a functionality called "very weak OT" and then we show how use a protocol for very weak OT to implement weak OT. For the second step, we can use Kilian's result:

Lemma 4. *[18] The existence of a protocol for weak OT implies (honest) OT*[9].

[9] This lemma uses the reduction of OT to weak OT given in [18], Section 2.4. The analysis is slightly different from the original analysis: we use the uniform version

Weak OR implies very weak OT A protocol (A, B) for *very weak OT* is similar to a protocol for OT except that both the sender's privacy and the receiver's privacy hold with low probability. More formally, a protocol for very weak OT is $(1-\eta)$-secure if the following conditions hold:

- *Receiver's privacy*: If $s_0 = s_1 = 1$, A's view of $(A(1^n, s_0, s_1), B(1^n, 0))$ is $(1-2\eta)$-indistinguishable from A's view of $(A(1^n, s_0, s_1), B(1^n, 1))$.
- *Sender's privacy*: For every s_0, s_1, c, B's view of $(A(1^n, s_c, s_{\bar{c}}), B(1^n, c))$ is $(1-\eta)$-indistinguishable from B's view of $(A(1^n, s_c, \overline{s_{\bar{c}}}), B(1^n, c))$

Given a WOR protocol that is $(1-\eta)$-secure, we build the following protocol for very weak OT.

Protocol 3. 1. Inputs: A has the security parameter 1^n and two secret bits s_0 and s_1. B has the security parameter 1^n and a selection bit c.
2. A sets $a_0 = s_0$ and $a_1 = s_1$. B sets $b_c = 0$ and $b_{\bar{c}} = 1$.
3. A and B run the WOR protocol to obtain $c_j = OR(a_j, b_j)$ for $j \in \{0, 1\}$. [10]
4. Outputs: B computes the secret bit $s_c = OR(a_c, b_c)$, A outputs nothing.

Analysis Sketch. Note that Protocol 3 computes the OT functionality correctly: B obtains the secret bit a_c because $a_c = OR(a_c, b_c) = OR(a_c, 0)$.

Since $b_{\bar{c}} = 1$, the security of the WOR protocol implies that B has advantage at most η in distinguishing the case $a_{\bar{c}} = 0$ from the case $a_{\bar{c}} = 1$.

If $s_0 = s_1 = 1$, i.e. $a_c = a_{\bar{c}} = 1$, the security of the WOR protocol implies that A has advantage at most 2η in distinguishing the case $c = 0$ from the case $c = 1$ (by the security of the WOR protocol). Note that in Protocol 3 if $a_0 = 0$ or $a_1 = 0$, then A learns B's selection bit c.

Very weak OT implies weak OT Given a $(1-\eta)$-secure protocol for very weak OT, where $\eta \leq \frac{1}{f(n)}\left(1 - \frac{1}{t(n)}\right)$, $f(n) = \omega(logn)$ and t is a polynomial, we build a protocol for weak OT. In order to amplify the receiver's privacy, we will repeat the protocol for very weak OT $f(n)$ times and apply a secret sharing scheme to B's selection bit.

Protocol 4. 1. Inputs: A has the security parameter 1^n and two secret bits s_0 and s_1. B has 1^n.
2. For $1 \leq i \leq f(n)$, A uniformly chooses $a_0^i, a_1^i \in \{0, 1\}$ and B uniformly chooses $c^i \in \{0, 1\}$.
3. For $1 \leq i \leq f(n)$, A and B execute the protocol for very weak OT on (a_0^i, a_1^i, c^i).

of Yao's XOR lemma to guarantee the sender's privacy and a hybrid argument to guarantee the receiver's privacy.

[10] The executions of the WOR protocol can be done in parallel. Indeed, the executions of the WOR protocol are independent because the parties are assumed to be honest. Hence the privacy condition still holds with probability $1-\eta$ for each execution of the WOR protocol (otherwise an adversary could violate the privacy condition for a single execution of the protocol by simulating an independent execution on its own).

4. A uniformly chooses $z^1, \cdots, z^{f(n)} \in \{0,1\}$ and sends to B the following values
   - for $1 \leq i \leq f(n)$, $q_0^i = z^i \oplus a_0^i$
   - for $1 \leq i \leq f(n)$, $q_1^i = z^i \oplus a_1^i \oplus (s_0 \oplus s_1)$
   - $Q = s_0 \oplus (\oplus_{i=1}^{f(n)} z^i)$
5. B computes for every $i \in [f(n)]$, $v^i = q_{c^i}^i \oplus a_{c^i}^i$, $c = \oplus_{i=1}^{f(n)} c^i$ and outputs $s_c = Q \bigoplus \left(\oplus_{i=1}^{f(n)} v^i\right)$.

Analysis Sketch. By the correctness of the protocol for very weak OT, we know that for every i, B learns the value of $a_{c^i}^i$. Thus we can show that B computes the secret s_c correctly in Protocol 4.

Intuitively, in order to know c, A needs to know the values of all the c^i's. By the security of the protocol for very weak OT, we know that for every i, the probability that A distinguishes the case $c^i = 1$ from the case $c^i = 0$ is at most $3/4 + 2\eta$. Using the uniform version of Yao's XOR Lemma [13], we can show that the probability that A distinguishes the case $c = 0$ from the case $c = 1$ is negligible.

Intuitively, in order to know $s_{\bar{c}}$, B needs to know the value of one of the $a_{\bar{c^i}}^i$. Using a hybrid argument, we can show that the probability that B distinguishes the case $s_{\bar{c}} = 0$ from the case $s_{\bar{c}} = 1$ is at most $\eta \cdot f(n) \leq 1 - \frac{1}{t(n)}$ for some polynomial t.

5 On the Different Definitions of PAK

We highlight the difference between the indistinguishability-based definition of [1] with the simulation-based definition of [11] by showing that unlike the GL definition, the BPR definition does not imply semi-honest OT in the case of exponential-sized dictionaries. Bellare, Pointcheval and Rogaway started with the model and definition of [2] for authenticated key exchange and modified them appropriately to take into account passwords instead of high-entropy keys. In particular, the definition of security of [1] for password-authenticated key exchange for the dictionary $\mathcal{D} = \{0,1\}^n$ (when we do not guarantee forward secrecy) is *exactly* the original definition of [2] for plain (=non-password-based) authenticated key exchange.

Consider the following protocol that was proposed in [2]. The legitimate parties share a password $w \xleftarrow{\text{R}} \mathcal{D} = \{0,1\}^n$ that we can see as two $n/2$-bit strings (w_1, w_2). The first part is taken as the key to a pseudorandom function family $\mathcal{F} = \{f_{w_1} : \{0,1\}^{2n} \to \{0,1\}^n\}_{w_1 \in \{0,1\}^{n/2}}$. The second part is taken as the key to a pseudorandom permutation family $\mathcal{G} = \{g_{w_2} : \{0,1\}^n \to \{0,1\}^n\}_{w_2 \in \{0,1\}^{n/2}}$.

Protocol 5. 1. Inputs: A and B have a security parameter 1^n and a joint password $w = (w_1, w_2) \in \mathcal{D} = \{0,1\}^n$.
2. A chooses $r_A \xleftarrow{\text{R}} \{0,1\}^n$ and sends $\alpha_1 = r_A$ to B. B receives β_1.
3. B chooses $r_B \xleftarrow{\text{R}} \{0,1\}^n$ and sends $\beta_2 \stackrel{\text{def}}{=} (r_A, r_B, f_{w_1}(r_A, r_B))$ to A. A receives α_2.

4. If $\alpha_2 \neq (r_A, r_B, f_{w_1}(r_A, r_B))$ (which A can check using its password), then A chooses $K_A \xleftarrow{\text{R}} \{0,1\}^n$. Otherwise, A sends $\alpha_3 \stackrel{\text{def}}{=} (r_B, f_{w_1}(r_B))$ to B. B receives β_3.
5. Outputs: If α_2 was of the form $(r_A, r_B, f_{w_1}(r_A, r_B))$, A outputs $g_{w_2}(r_B)$; otherwise, it outputs K_A. If $\beta_3 \neq (r_B, f_{w_1}(r_B))$, B rejects. Otherwise, B outputs $g_{w_2}(r_B)$.

Lemma 5. *[2] If one-way functions exist, Protocol 5 is a secure authenticated key exchange protocol for the definition of [2]. In other words, Protocol 5 is a secure PAK protocol for the dictionary $\mathcal{D} = \{0,1\}^n$ for the BPR definition.*

Hence, unlike the GL definition, the BPR definition does not imply semi-honest OT for dictionaries of exponential size under black-box reductions ([15, 9]). Intuitively, the difference is that the BPR definition does not guarantee that the password w remain pseudorandom after an execution of a secure PAK protocol. Indeed, we can see that the password w will not remain pseudorandom even with respect to a passive adversary's view of an execution of Protocol 5 since the adversary learns the pair $(r_B, f_{w_1}(r_B))$. The pseudorandomness property required by the GL definition makes a secure PAK protocol a strong enough primitive to imply semi-honest OT, even in the case of dictionaries of exponential size (which corresponds to plain authenticated key exchange). The guarantee that the password w remain pseudorandom after an execution of a PAK protocol is indeed important if one intends to also use the password in a protocol other than the PAK protocol.

This does not necessarily mean a PAK protocol that is secure for the BPR definition is not a "strong" primitive. Indeed, we conjecture that one can implement semi-honest OT using a PAK protocol that is secure for the BPR definition for *all dictionaries* (including poly-sized dictionaries which is the case of most interest). Another open question is the relationship between the simulation-based definition of Boyko, MacKenzie and Patel [6] and the BPR and GL definitions.

As noted in [7], "*settling on a "good" definition of security for password-based authentication has been difficult and remains a challenging problem*". However, a study of the relationship between each definition of security for PAK and other cryptographic primitives provides a better understanding of the tradeoffs and advantages offered by one definition of security over another.

Acknowledgments. We thank Salil Vadhan for suggesting this problem and for many helpful discussions and detailed comments. Many thanks to Yehuda Lindell for his help in reconstructing [5]. We are grateful to Alex Healy and Omer Reingold for helpful conversations on this subject and to the anonymous reviewers for their insightful comments.

References

1. Bellare, M., Pointcheval, D., Rogaway, P.: Authenticated Key Exchange Secure against Dictionary Attacks. Lecture Notes in Computer Science **1807** (2000) 139–155

2. Bellare, M., Rogaway, P.: Entity Authentication and Key Distribution. Lecture Notes in Computer Science **773** (1994) 232–249
3. Bellovin, S., Merritt, M.: Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks. ACM/IEEE Symposium on Research in Security and Privacy (1992) 72–84
4. Bellovin, S., Merritt, M.: Augmented Encrypted Key Exchange: A Password-Based Protocol Secure against Dictionary Attacks and Password File Compromise. ACM Conference on Computer and Communications Security (1993) 244–250
5. Boyarsky, M.: Public-Key Cryptography and Password Protocols: The Multi-User Case. ACM Conference on Computer and Communications Security (1999) 63–72
6. Boyko, V., MacKenzie, P., Patel, S.: Provably Secure Password-Authenticated Key Exchange Using Diffie-Hellman. Lecture Notes in Computer Science **1807** (2000) 156–171
7. Canetti, R., Halevi, S., Katz, J., Lindell, Y., MacKenzie, P.: Universally Composable Password-Based Key Exchange. Unpublished manuscript (2004)
8. Gennaro, R., Lindell, Y.: A Framework for Password-Based Authenticated Key Exchange. Lecture Notes in Computer Science **2656** (2003) 524–543
9. Gertner, Y., Kannan, S., Malkin, T., Reingold, O., Viswanathan, M.: The Relationship between Public-Key Encryption and Oblivious Transfer. IEEE Symposium on the Foundations of Computer Science (2001) 325–335
10. Goldreich, O.: Foundations of Cryptography, Volume 2. Cambridge University Press (2004)
11. Goldreich, O., Lindell, Y.: Session-Key Generation Using Human Passwords Only. Lecture Notes in Computer Science **2139** (2001) 408–432
12. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proofs. Journal of the ACM **38:3** (1991) 691–729
13. Goldreich, O., Nisan, N., Wigderson, A.: On Yao's XOR Lemma. Electronic Colloquium on Computational Complexity (1995) TR95-050
14. Halevi, S., Krawczyk, H.: Public-Key Cryptography and Password Protocols. ACM Conference on Computer and Communications Security (1998) 122–131
15. Impagliazzo, R., Rudich, S.: Limits on the Provable Consequences of One-way Permutations. ACM Symposium on Theory of Computing (1989) 44–61
16. Katz, J.: Efficient Cryptographic Protocols Preventing 'Man-in-the-Middle' Attacks. Ph.D. Thesis. Columbia University (2002)
17. Katz, J., Ostrovsky, R., Yung, M.: Efficient Password-Authenticated Key Exchange Using Human-Memorable Passwords. Lecture Notes in Computer Science **2045** (2001) 475–494
18. Kilian, J.: A General Completeness Theorem for Two-Party Games. ACM Symposium on Theory of Computing (1991) 553–560
19. Full version of this paper at http://www.people.fas.harvard.edu/˜ mnguyen
20. Nguyen, M.-H., Vadhan, S.: Simpler Session-Key Generation from Short Random Passwords. Lecture Notes in Computer Science **2951** (2004) 428–445
21. Nisan, N., Zuckerman, D.: Randomness is Linear in Space. Journal of Computer and System Sciences **52:1** (1996) 43–52
22. Reingold, O., Trevisan, L., Vadhan, S.: Notions of Reducibility between Cryptographic Primitives. Lecture Notes in Computer Science **2951** (2004) 1–20
23. Rudich, S.: The Use of Interaction in Public Cryptosystems. Lecture Notes in Computer Science **576** (1992) 242–251
24. Steiner, M., Tsudik, G., Waidner, M.: Refinement and Extension of Encrypted Key Exchange. Operating Systems Review **29:3** (1995) 22–30