Efficient Password Authenticated Key Exchange via Oblivious Transfer

Ran Canetti[1,⋆], Dana Dachman-Soled[2], Vinod Vaikuntanathan[3], and Hoeteck Wee[4,⋆⋆]

[1] Tel Aviv University & Boston University
[2] Microsoft Research New England
[3] University of Toronto
[4] George Washington University

Abstract. We present a new framework for constructing efficient password authenticated key exchange (PAKE) protocols based on oblivious transfer (OT). Using this framework, we obtain:

- an efficient and simple UC-secure PAKE protocol that is secure against adaptive corruptions *without erasures.*
- efficient and simple PAKE protocols under the Computational Diffie-Hellman (CDH) assumption and the hardness of factoring. (Previous efficient constructions rely on hash proof systems, which appears to be inherently limited to decisional assumptions.)

All of our constructions assume a common reference string (CRS) but do not rely on random oracles.

Keywords: Password Authenticated Key Exchange, UC security, adaptive security, oblivious transfer, search assumptions.

1 Introduction

Password authenticated key-exchange (PAKE) allows two parties with a shared password to mutually authenticate each other and establish a shared key, without explicitly revealing the password in the process [BM93]. PAKE is well-suited for use in web authentication (in place of having the user input her password directly), as it resists phishing and other social engineering attacks; if a user mistakenly authenticates herself to a phisher via a PAKE protocol, the protocol will fail, but the user's password remains safe. For this application, it is important that the PAKE protocol remains secure even amidst concurrent executions, as is unavoidable on the Internet.

[⋆] Supported by the Check Point Institute for Information Security, Marie Curie grant PIRG03-GA-2008-230640, and ISF grant 0603805843.

[⋆⋆] Supported by NSF CAREER Award CNS-0953626.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 449–466, 2012.

Prior work. The study of PAKE was initiated by Bellovin and Merritt [BM93]. Formal models for PAKE were developed several years later in [BPR00, BMP00, GL01, CHK$^+$05], and solutions were first presented in the random oracle/ideal cipher models [BPR00, BMP00, MPS00]. Since then, there has been a large number of constructions in the standard model, without relying on random oracles. For instance, we now know how to achieve security in the "plain model" without any additional trusted set-up [GL01, NV04, BCL$^+$05, GJO10]; these constructions typically rely on general techniques for secure computation. However, these protocols are fairly inefficient in terms of communication, computation and round complexity and seem unlikely to lead to a practical instantiation.

In this work, we focus on efficient constructions in the common reference string (CRS) model, initiated by Katz, Ostrovsky and Yung [KOY01] and revisited in [GL03, JG04, CHK$^+$05, KMTG05, G08, ACP09, KV09, GK10, KV11]. Note that in practice, the CRS can be hard-coded into an implementation of the protocol. In addition to being computationally efficient and constant-round, these protocols remain secure even with adversarially coordinated concurrent executions. All of these works rely on the paradigm of smooth projective hashing [CS02, CS98] (either directly or indirectly). The most general and most recent is that of Groce and Katz [GK10], which building on [JG04], shows how to realize efficient PAKE with two building blocks: a CPA-secure encryption scheme supporting projective hashing, and a CCA-secure encryption scheme. This improves over previous works which require a CCA-secure scheme that supports smooth projective hashing.

The reliance on smooth projective hashing leads to two limitations on the ensuing protocols: first, all known instantiations of smooth projective hashing rely on decisional assumptions. e.g., the Decisional Diffie-Hellman (DDH) assumption or the quadratic residuosity assumption. In general, decisional assumptions are a much stronger class of assumptions than computational assumptions based on search problems, such as factoring, finding shortest vectors in lattices, or even the Computational Diffie-Hellman (CDH) problem. Indeed, there are groups, such as certain elliptic curve groups with bilinear pairing map, where the DDH assumption does not hold, but the Computational Diffie-Hellman (CDH) problem appears to be hard. As such, schemes based on search problems are generally preferred to those based on decisional assumptions. However, such schemes seem very hard to obtain.

Second, modifying the schemes based on smooth projective hashing to achieve security against adaptive corruptions (where an adversary may choose which parties to corrupt during the execution of the protocol) appears to be fairly challenging. This was first achieved in the recent work of Abdalla et al. [ACP09], under the additional assumption of secure erasures.

1.1 Our Contributions

We present the first construction of reasonably efficient PAKE protocols that bypass the "projective hashing" paradigm. Instead, we rely on oblivious transfer (OT) as the main cryptographic building block. We obtain new PAKE protocols that achieve various combinations of the following properties: (a) conceptual simplicity, (b) efficiency, (c) security against adaptive corruptions even without erasures, and (d) reliance on relatively weak hardness assumptions.

Before we outline our result, we first mention that there are two prevailing security notions for PAKE that achieve security under concurrent executions and in particular, guarantee resilience against man-in-the-middle attacks. The first and most basic notion is that of "concurent PAKE" put forth by Bellare et al. and Boyko et al. [BPR00, BMP00]. The second and stronger notion is that of "UC secure PAKE" [CHK+05, C01], which guarantee security amidst composition with arbitrary protocols, and with arbitrary, unknown and possibly correlated password distributions.

Our results. Specifically, we show:

- Two UC-secure PAKE protocols. The first only assumes an ideal OT functionality, and is secure against adaptive corruptions without erasures. Combined with the OT protocol with Garay et al. [GWZ09], we obtain a reasonably efficient UC-secure PAKE protocol in the CRS model that is secure against adaptive corruptions without erasures. (Prior protocols that achieve adaptive security are either in the Random Oracle model [ACCP08], or require secure erasures [ACP09] or are highly inefficient [BCL+05].)
  The second protocol builds on [GK10], is a more efficient variant of the first, and relies on a CCA2-secure PKE in addition to OT. It only tolerates static corruptions. We defer the details of this construction to the full version.
- New PAKE protocols under search assumptions, notably CDH and hardness of factoring. Previous efficient instantiations rely on hash proof systems, which appears to be inherently limited to decisional assumptions. This construction requires a special variant of OT. Here we also provide some constructions of this special OT variant.

1.2 Overview of Our Constructions

We proceed to provide an overview of our constructions.

The UC Constructions. The main novelty in our UC constructions are protocols that assume ideal authenticated channels as well as ideal "OT channels" and realize the following two party functionality, which we call randomized equality ($\mathcal{F}_{\mathsf{re}}$): If the inputs provided by the parties are equal, then both parties obtain the same fresh random key. If the parties provide different inputs, then each party obtains a special symbol $\perp$.

Given such a protocol, we use a generic transformation from [BCL+05] to obtain a protocol that realizes the "split version" of $\mathcal{F}_{\mathsf{re}}$, which turns out to be equivalent to $\mathcal{F}_{\mathsf{pwKE}}$, the ideal password-based key exchange functionality. The above transformation results in an additional cost of generating a key for a signature scheme, and then signing each message. Alternatively, we may rely on a more efficient transformation described in [CCGS10], that costs only a single key exchange protocol, plus a MAC creation and verification per message (although this transformation only achieves adaptive security *with erasures*).

First Construction. Our first protocol for realizing $\mathcal{F}_{\mathsf{re}}$ is extremely simple. Assume for now that we have an ideal 1-out-of-$|\mathcal{D}|$ OT functionality. The first party acts as the OT receiver and uses as input his password. The second party acts as the OT sender and picks $|\mathcal{D}|$ random strings $r_1, \ldots, r_{|\mathcal{D}|}$ as input. The first party uses the OT output as his session key, and the second party uses the string indexed by his password. Indeed, if both parties are honest, they agree on the same random key, and if the first party is corrupted, he learns nothing about the session key unless he guesses the right password. There are two issues with the protocol as described:

- The protocol only handles dictionaries of polynomial size. To fix this, we observe that we only require that the $|\mathcal{D}|$ random strings be pairwise independent. In particular, we can replace the 1-out-of-$|\mathcal{D}|$ OT functionality with $\log |\mathcal{D}|$ copies of 1-out-of-2 OT, where the second party now picks $\log |\mathcal{D}|$ pairs of random string, and the first party outputs the XOR of the $\log |\mathcal{D}|$ OT outputs. (In the overview of the remaining constructions, we omit this optimization for simplicity.)
- The protocol does not tolerate corruptions of the second party; for instance, the second party could set all $|\mathcal{D}|$ strings to be equal thereby learning the session key. To fix this, we repeat the basic protocol one more time, with the roles of the parties reversed, and the final session key is the XOR of the two session keys. By running the basic protocol in reverse, we guarantee that the second party also learns nothing about the session key unless it guesses the right password. (This idea of running a basic protocol with reversed roles appears in the early works of Katz et al. [KOY01, GL03] too.)

Combining this construction with the adaptively secure OT given in [GWZ09], we obtain the following result:

> **Proposition 1 (informal).** *There exists a constant-round UC-secure PAKE protocol in the CRS model that is secure against an adaptive adversary without erasures and without authenticated channels. The protocol may be based on DDH or DCR and both parties exchange a constant number of group elements.*

Second Construction. To motivate our second construction, which is inspired by that of Groce and Katz [GK10, JG04], consider again our basic protocol based on an ideal 1-out-of-$|\mathcal{D}|$ OT functionality. Instead of running the basic protocol a second time in order to handle corruptions of the second party, we have the second party send an encryption of her password. The advantage over the first protocol is that the computation costs for a CCA2-secure encryption is typically lower than that of running another OT protocol. In more detail, we assume in addition a common reference string (CRS), and handle corruptions of the second party as follows:

- Both parties run the basic protocol. Let $r_1, \ldots, r_{|\mathcal{D}|}$ denote $|\mathcal{D}|$ random strings chosen by the second party, and let π denote her password. She then

> parses r_π as a pair of random strings $\mathsf{skey}\|\mathsf{rand}$, sends along an encryption C of π (and her identifier) using randomness rand, under a public key for a CCA2-secure encryption scheme that is part of the CRS. The first party encrypts her password with randomness determined by the output from the basic protocol. If the ciphertext matches C, both parties output skey as the session key.

If the first party is corrupted and fails to guess the right password, then both skey and rand are truly random from her point of view, and the ciphertext C reveals no information about the second party's password via semantic security. On the other hand, if the second party is corrupted and fails to guess the right password, then C will not match the first party's password by (perfect) correctness of the underlying encryption. In the proof of security, the simulator will decrypt C to extract the password of the second party.

The Concurrently-Secure PAKE. Our concurrently-secure PAKE is essentially the same as our second UC-secure construction, except we replace the underlying UC-secure OT with an OT protocol that achieves much weaker guarantees. Roughly speaking, we relax the security guarantee for corrupted senders to an indistinguishability-based notion, and moreover, we no longer require that the OT guarantee non-malleability. The resulting construction may also be viewed as an abstraction of the Groce-Katz protocol [GK10, JG04], where we use an OT primitive in lieu of the CPA-secure encryption with projective hashing. We provide two different approaches towards realizing the underlying OT primitive.

Concurrent PAKE from Lossiness. Our first approach is based on dual-mode cryptosystems, a "lossy" primitive introduced by Peikert et al. [PVW08]. Combined with our general framework, we obtain the following result:

> **Proposition 2 (informal).** *There exists a three-message PAKE protocol in the CRS model that relies on black-box access to a dual-mode cryptosystem and a CCA-secure encryption scheme and achieves concurent security with mutual authentication (against a static adversary).*

We stress that this construction should be viewed mainly as a feasibility result on black-box constructions of PAKE protocols in the CRS model based on general assumptions. The work of Peikert and Waters [PW08] introduced the notion of lossy trapdoor functions, and showed that they also yield CCA-secure encryption schemes. This raised the natural question of understanding connections between smooth projective hashing and "lossy" primitives. Our work demonstrates that for concurrently-secure PAKE protocol, it is indeed possible to avoid the use of smooth projective hashing and rely solely on "lossy" primitives (notably the dual mode encryption scheme in [PVW08] and lossy trapdoor functions) in a black-box way.

Concurrent PAKE from Search Assumptions. Our second approach starts with the Bellare-Micali OT protocol based on CDH. Combined with our general framework, we obtain the following result:

> **Proposition 3 (informal).** *There exists a constant-round PAKE protocol in the CRS model based on hardness of factoring or CDH (computational Diffie-Hellman assumption) that achieves concurrent security with mutual authentication (against a static adversary). Moreover, each party sends a quadratic number of group elements.*

Password Based Group Key Exchange. The second UC construction and the concurrently secure construction have the following additional attractive property: The generated session key is determined exclusively by one of the parties. Furthermore, this key can be chosen by this party in advance, before the protocol begins. This property allows for a natural extension of these PAKE protocols to efficient password based group key exchange protocols: One party exchanges a key with each one of the other parties, using the above property to ensure that all parties agree on the same key.

This approach to group key exchange is indeed different than the approach in prior works on this problem, e.g. [ABCP06, AP06], which concentrate on "contributory protocols" where all parties "contribute" to the group key. Still, it arguably provides an adequate level of security. This approach is particularly suitable to groups where there is one special party (either the group manager or the multi-caster of the data): here this party is the only one that does work that's proportional to the size of the group. The work done by all other parties is independent of the size of the group.

2 UC-Secure PAKE from Oblivious Transfer

We present a UC-secure PAKE protocol from Oblivious Transfer. An alternative construction appears in the full version.

Definitions. For simplicity and clarity, we begin by realizing single-session PAKE, and we extend all of these definitions and results to multi-sessions in the full version[1]. We present the functionality $\mathcal{F}_{\mathsf{pwKE}}$ for password-based key exchange. The description of the functionality is a modified version of the description in [GK10] (which is itself a modification of [CHK+05]). In particular, $\mathcal{F}_{\mathsf{pwKE}}$ captures PAKE protocols which achieve *explicit mutual authentication.* We refer the reader to [CHK+05, GK10] for motivating discussion regarding the particular choices made in this formulation of the functionality.

[1] Note that for single-session PAKE we may require an independent common reference string for each concurrent PAKE session; however, realizing multi-session PAKE allows us to have a single global common reference string for an unbounded number of concurrent PAKE sessions.

Functionality $\mathcal{F}_{\mathsf{pwKE}}$

The functionality $\mathcal{F}_{\mathsf{pwKE}}$ is parameterized by a security parameter λ. It interacts with an adversary $\mathcal{S}$ and a set of parties via the following queries:

Upon receiving a query $(\mathsf{NewSession}, \mathsf{sid}, I, R\pi_I)$ **from party** I**:**
Record (I, R, π_I), mark this record fresh, and send a message (sid, I, R) to $\mathcal{S}$. Ignore all future messages from I with the same ssid.
Upon receiving a query $(\mathsf{sid}, \mathsf{ok})$ **from** $\mathcal{S}$**:**
Send a message $(\mathsf{NewSession}, \mathsf{sid}, I, R)$ to R. Ignore all future (ok) messages.
Upon receiving a query $(\mathsf{Respond}, \mathsf{sid}, I, R, \pi_R)$ **from** R**:** Record (R, I, π) and mark this record fresh.

Upon receiving a query $(\mathsf{TestPwd}, \mathsf{sid}, P, \pi')$ **from the adversary** $\mathcal{S}$**:**
If $P \in \{I, R\}$, there is a record of the form $(P, *, \pi)$ which is fresh, then do: If $\pi' = \pi$, mark the record $\mathsf{compromised}$ and reply to $\mathcal{S}$ with "correct guess". If $\pi \neq \pi'$, mark the record $\mathsf{interrupted}$ and reply to $\mathcal{S}$ with "wrong guess". fresh,
Upon receiving a query $(\mathsf{NewKey}, \mathsf{sid}, P, \mathsf{skey})$ **from** $\mathcal{S}$**, where** $|\mathsf{skey}| = \lambda$**:**

If there is a record of the form $(P, *, \pi)$ that is not marked $\mathsf{completed}$, do:
- If this record is $\mathsf{compromised}$, or either I or R is corrupted, then output $(\mathsf{sid}, \mathsf{skey})$ to player P.
- else, if there is a record $(*, P, \pi', \mathsf{server}, \mathsf{skey}')$ with $\pi' = \pi$, then send skey' to player P.

Fig. 1. The password-based key-exchange functionality $\mathcal{F}_{\mathsf{pwKE}}$

Constructions. The construction of both protocols proceeds in three steps. First, in Section 2.1, we define a (randomized) equality-testing functionality $\mathcal{F}_{\mathsf{re}}$ which, informally speaking, captures PAKE in the authenticated channels model. In Section 2.2, we show a protocol that securely implements $\mathcal{F}_{\mathsf{re}}$ in the OT-hybrid model, tolerating adaptive corruptions (a second protocol that implements $\mathcal{F}_{\mathsf{re}}$ is presented in the full version). These protocols assume built-in authenticated channels whereas our end goal, of course, is to implement PAKE without any authenticated channels. Thus, our second step is to transform these protocols into ones that do not assume authenticated channels, but implement a "split version" of $\mathcal{F}_{\mathsf{re}}$ (See Section 2.3 for more details) using the transformation of Barak, Canetti, Lindell, Pass and Rabin [BCL$^+$05]. Together with the adaptively secure OT protocol of Garay, Wichs and Zhou [GWZ09], this gives us a protocol implementing the split $\mathcal{F}_{\mathsf{re}}$ functionality in the common reference string model, tolerating adaptive corruptions. Finally, we show (in Proposition 4) that the split $\mathcal{F}_{\mathsf{re}}$ functionality already captures UC-secure PAKE. We note that this three step method of constructing UC PAKE protocols was pointed out in the work of Barak et al. [BCL$^+$05].

2.1 The Randomized Equality-Testing Functionality

We define a (randomized) equality-testing functionality $\mathcal{F}_{\mathsf{re}}$ that, roughly speaking, takes inputs from two parties and does the following:

– if the inputs are equal, sends both parties the same random session key; moreover, if either party is corrupted, the adversary is allowed to set the key.
– if the inputs are unequal, send both parties the special symbol $\perp$.

More precisely, $\mathcal{F}_{\mathsf{re}}$ captures a protocol between two players – an "initiator" I and a "responder" R. The initiator starts the protocol by sending a message to the functionality $\mathcal{F}_{\mathsf{re}}$ that includes his input π_I. The functionality then allows the adversary $\mathcal{S}$ to determine when to "wake up" the responder R into starting the protocol. Once woken up, R sends his input x_R to the functionality. If the inputs match, then the functionality assigns the same random key to both parties. Otherwise, it assigns a special symbol $\perp$ to both of them. Thus, this definition corresponds to achieving *explicit mutual authentication.* We allow the ideal-model adversary two special powers. First, we allow him to set the shared key if one of the parties is corrupted and both the parties have the same input (jumping ahead, we note that this corresponds to his ability to set the key in case he guessed one of the parties' password correctly). Furthermore, he controls the delivery of messages to the parties. This is an ability that he inevitably has in the real world.

Functionality $\mathcal{F}_{\mathsf{re}}$

The functionality $\mathcal{F}_{\mathsf{re}}$ is parameterized by a security parameter λ and a "dictionary" $\mathcal{D}$. It interacts with an *initiator* I, a *responder* R, and the adversary $\mathcal{S}$ via the following messages:

Upon receiving a query $(\mathsf{Init}, \mathsf{sid}, I, R, \pi_I), \pi_I \in \mathcal{D}$ **from party** I**:**
Record (I, π_I) and send a message (sid, I, R) to $\mathcal{S}$. Ignore all future messages from I with the same ssid.

Upon receiving a query $(\mathsf{sid}, \mathsf{ok})$ **from** $\mathcal{S}$**:**
Send a message $(\mathsf{wakeup}, \mathsf{sid}, I, R)$ to R. Ignore all future (ok) messages.

Upon receiving a query $(\mathsf{Respond}, \mathsf{sid}, I, R, \pi_R)$ **from** R**:**
– If $\pi_R = \pi_I$, then choose $\mathsf{skey} \leftarrow \{0,1\}^\lambda$ and store $\mathsf{out}_I = \mathsf{out}_R = \mathsf{skey}$.
– If $\pi_R \neq \pi_I$, then set $\mathsf{out}_I = \mathsf{out}_R = \perp$.
In both cases, ignore subsequent inputs from R.

Upon receiving a query $(\mathsf{Corrupt}, \mathsf{sid}, I, R, (P, K))$ **from** $\mathcal{S}$**, where** $P \in \{I, R\}$**:**

If $\pi_R = \pi_I$, then set $\mathsf{out}_I = \mathsf{out}_R = K$. Output the message $(\mathsf{corrupted})$ to P.

Upon receiving a query $(\mathsf{sid}, \mathsf{Out}, P), P \in \{I, R\}$ **from** $\mathcal{S}$**:**
Send $(\mathsf{output}, \mathsf{sid}, I, R, \mathsf{out}_P)$ to the player P. Ignore all subsequent (Out, P) queries for the same player P.

Fig. 2. The Randomized Equality-Testing functionality $\mathcal{F}_{\mathsf{re}}$

Connection to $\mathcal{F}_{\mathsf{pwKE}}$. Let $\mathsf{s}\mathcal{F}_{\mathsf{re}}$ be the functionality obtained by applying the "split functionality" transformation of [BCL$^+$05] to the functionality $\mathcal{F}_{\mathsf{re}}$. We show that $\mathsf{s}\mathcal{F}_{\mathsf{re}}$ is already powerful enough to capture the password-authenticated key exchange functionality $\mathcal{F}_{\mathsf{pwKE}}$. More formally, we show the following proposition whose proof is deferred to the full version.

Proposition 4. *There is a protocol* Π_{REtoPAKE} *that securely implements the* $\mathcal{F}_{\mathsf{pwKE}}$ *functionality in the* $\mathsf{s}\mathcal{F}_{\mathsf{re}}$*-hybrid model, tolerating adaptive corruptions and without assuming authenticated channels.*

2.2 Randomized Equality Testing Protocol 1

We now describe our first randomized equality testing protocol Π_{REfromOT} in the $\mathcal{F}_{\mathsf{OT}}$-hybrid model. We show that the protocol is secure against *adaptive corruptions* in a model with built-in authenticated channels.

UC Randomized Equality Testing Protocol Π_{REfromOT} in the $\mathcal{F}_{\mathsf{OT}}$-Hybrid Model

The protocol is between two players I and R. Assume that the dictionary $\mathcal{D} \subseteq \{0,1\}^{\ell}$.

Code for Player P_b interacting with P_{1-b}, where $b \in \{0,1\}$ and $P_0, P_1 \in \{I, R\}$.

1. P_b, on input $\pi \in \mathcal{D}$ does the following. Let $\pi = \pi_1, \ldots, \pi_\ell$, where $\pi_i \in \{0,1\}$.
   - **(Run OT as the Receiver)** For every $i \in [1 \ldots \ell]$, send $(\mathsf{Receiver}, \mathsf{sid}||i, \pi_i)$ to $\mathcal{F}_{\mathsf{OT}}$.
   - **(Run OT as the Sender)** For every $i \in [1 \ldots \ell]$, choose a pair of random strings $(w^b_{i,0}, w^b_{i,1}) \in \{0,1\}^{3\lambda}$ and send the message $(\mathsf{Sender}, \mathsf{sid}||i, (w^b_{i,0}, w^b_{i,1}))$ to $\mathcal{F}_{\mathsf{OT}}$.
2. P_b waits to receive messages $(\mathsf{Output}, \mathsf{sid}||i, (w')^b_i)$ from $\mathcal{F}_{\mathsf{OT}}$ for all $i \in [1 \ldots \ell]$. It then computes $K' = \bigoplus_{i=1}^{\ell} w'_i = \mathsf{skey}'||\mathsf{test}'_0||\mathsf{test}'_1$.
3. P_b computes the value
$$K = \bigoplus_{i=1}^{\ell} w^b_{i,\pi_i} \overset{\Delta}{=} \mathsf{skey}||\mathsf{test}_0||\mathsf{test}_1 \quad (\text{where } \mathsf{skey}, \mathsf{test}_0, \mathsf{test}_1 \in \{0,1\}^{\lambda})$$
and sends $(\mathsf{test}_b \oplus \mathsf{test}'_b)$ to P_{1-b}.
4. P_b waits to receive $\mathsf{test} \oplus \mathsf{test}' \in \{0,1\}^{\lambda}$ from P_{1-b}, and checks if $\mathsf{test} \oplus \mathsf{test}'$ matches $\mathsf{test}_{1-b} \oplus \mathsf{test}'_{1-b}$.
   - If the check does not pass, then output $\perp$.
   - If the check passes, output $(\mathsf{sid}, \mathsf{skey}' \oplus \mathsf{skey})$.

   In either case, terminate the session.

Fig. 3. Randomized Equality Testing Protocol Π_{REfromOT}

Theorem 1. *The protocol Π_{REfromOT} in Figure 3 securely realizes the randomized equality testing functionality $\mathcal{F}_{\mathsf{re}}$ in the $\mathcal{F}_{\mathsf{OT}}$-hybrid model, in the presence of adaptive corruptions, and assuming authenticated channels.*

Proof. Let $\mathcal{A}$ be an *adaptive* adversary interacting with a pair of parties I and R running the protocol Π_{REfromOT}. We show that for every such $\mathcal{A}$, there is an ideal-world adversary (simulator) $\mathcal{S}$ interacting with dummy parties and the ideal functionality $\mathcal{F}_{\mathsf{re}}$ such that no environment $\mathcal{Z}$ can distinguish between an interaction with $\mathcal{A}$ in the protocol Π_{REfromOT} and an interaction with $\mathcal{S}$ in the ideal world.

Description of the Simulator. The simulator $\mathcal{S}$ starts by invoking a copy of $\mathcal{A}$ and running a simulated interaction of $\mathcal{A}$ with the environment $\mathcal{Z}$ and the parties running the protocol. $\mathcal{S}$ proceeds as follows:

Simulating the Communication with $\mathcal{Z}$: Every message that $\mathcal{S}$ receives from the environment $\mathcal{Z}$ is written to $\mathcal{A}$'s input tape. In the same vein, every output value that $\mathcal{A}$ writes to its output tape is copied to $\mathcal{S}$'s own output tape (to be read later by $\mathcal{Z}$).

Simulating the Case when the Initiator I is Corrupted: $\mathcal{S}$ does the following.

- Upon receiving a message $(\mathsf{Sender}, \mathsf{sid}||i, (\omega_0, \omega_1))$ from $\mathcal{A}$ in session $\mathsf{sid}, \mathsf{ssid}$, record $w^b_{i,0} = \omega_0$ and $w^b_{i,1} = \omega_1$.
- Upon receiving a message $(\mathsf{Receiver}, \mathsf{sid}||i, \beta)$ from $\mathcal{A}$, record $\pi_i = \beta$. Choose a uniformly random string $(w')^b_{i,\pi_i} \leftarrow \{0,1\}^\lambda$ and send it to $\mathcal{A}$.
- As soon as all the bits π_i are received, let $\pi = \pi_1 \ldots \pi_\ell$, and write the message $(\mathsf{Init}, \mathsf{sid}, \mathsf{ssid}, I, R, \pi)$ on the outgoing communication tape of the corrupted (ideal model) I (to be sent to the functionality $\mathcal{F}_{\mathsf{re}}$). Also send (ok) to the ideal functionality $\mathcal{F}_{\mathsf{re}}$.
- As soon as all the pairs $(w^b_{i,0}, w^b_{i,1})$ have been recorded (for all $i \in [\ell]$), compute the key

$$K' = \bigoplus_{i=1}^{\ell} (w')^b_{i,\pi_i} \stackrel{\Delta}{=} \mathsf{skey}'||\mathsf{test}' \qquad \text{and} \qquad K = \bigoplus_{i=1}^{\ell} w^b_{i,\pi_i} \stackrel{\Delta}{=} \mathsf{skey}||\mathsf{test}$$

  where $\mathsf{skey}, \mathsf{skey}', \mathsf{test}, \mathsf{test}' \in \{0,1\}^\lambda$. Send a message $(\mathsf{Corrupt}, \mathsf{sid}, I, R, \mathsf{skey} \oplus \mathsf{skey}')$ to the functionality $\mathcal{F}_{\mathsf{re}}$.
- Send the messages (out, I) and (out, R) to $\mathcal{F}_{\mathsf{re}}$, and receive out_I from $\mathcal{F}_{\mathsf{re}}$. (*Remark:* Note that in case the inputs of I and R match, $\mathsf{out}_I = \mathsf{skey} \oplus \mathsf{skey}'$, otherwise $\mathsf{out}_I = \perp$. Thus, given out_I, $\mathcal{S}$ can tell if the inputs of I and R are the same or not.)
- If $\mathsf{out}_I \neq \perp$, send test' to $\mathcal{A}$. Otherwise send a uniformly random string $\mathsf{test}'' \leftarrow \{0,1\}^\lambda$ to $\mathcal{A}$.

Simulating the case when the Responder R is Corrupted: Since the protocol is completely symmetric between the two parties, the simulation is identical to that for a corrupted initiator I, except that $\mathcal{S}$ runs the following pre-amble phase:

- Wait to receive a message $(\mathsf{sid}, \mathsf{ssid}, I, R)$ from the functionality $\mathcal{F}_{\mathsf{re}}$. Send $(\mathsf{sid}, \mathsf{ssid}, \mathsf{ok})$ to $\mathcal{F}_{\mathsf{re}}$ and receive a message $(\mathsf{wakeup}, \mathsf{sid}, \mathsf{ssid}, I, R)$ from $\mathcal{F}_{\mathsf{re}}$.

The simulation from this point on is identical to the simulation for a corrupted I.

Simulating the case when both or neither of the parties is Corrupted: When both parties are corrupted, the simulator simply runs $\mathcal{A}$ internally (who itself generates all the messages). When neither party is corrupted, $\mathcal{S}$ produces uniformly random strings $\mathsf{test}, \mathsf{test}' \leftarrow \{0,1\}^\ell$ and forwards them to $\mathcal{A}$.

Dealing with Corruptions: Upon receiving a "Corrupt P_b" message from $\mathcal{A}$, where $P_b \in \{I, R\}$, corrupt the ideal-model $\tilde{P}_b \in \{\tilde{I}, \tilde{R}\}$, and obtain its input π_b and output out_{P_b}. When party P_b is corrupted by $\mathcal{A}$, $\mathcal{S}$ must produce both an input (and output) as well as random tape and private view for party P_b in the simulation. The random tape of party P_b consists of the pairs $(w^b_{i,0}, w^b_{i,1})$ for every $i \in [1 \dots \ell]$ and the private view of party P_b consists of the strings $(w')^b_i$ for every $i \in [1 \dots \ell]$. Thus, upon corruption of party P_b $\mathcal{S}$ will return to $\mathcal{A}$ the input π_b and output out_{P_b} obtained by corrupting the ideal-model $\tilde{P}_b$ as well as the values $w^b_{i,0}, w^b_{i,1}) \in \{0,1\}^{3\lambda}, (w')^b_i$ for every $i \in [1 \dots \ell]$. There are several cases to consider:

Corruption of party P_b before messages have been exchanged in Stage 3. $\mathcal{S}$ corrupts the ideal-model $\tilde{P}_b \in \{\tilde{I}, \tilde{R}\}$, and obtains its input π_{P_b}.

- If party P_{1-b} is not yet corrupted then $\mathcal{S}$ chooses $w^b_{i,0}, w^b_{i,1}, (w')^b_i$ for every $i \in [1 \dots \ell]$ uniformly at random and returns these values to $\mathcal{A}$. $\mathcal{S}$ continues the simulation for the case that party P_b is corrupted.
- If party P_{1-b} has already been corrupted then the values $w^{1-b}_{i,0}, w^{1-b}_{i,1}, (w')^{1-b}_i$ for every $i \in [1 \dots \ell]$ are already known and so $\mathcal{S}$ must ensure that the values of $w^b_{i,0}, w^b_{i,1}, (w')^b_i$ for every $i \in [1 \dots \ell]$ are consistent with these values.

  Thus, $\mathcal{S}$ does the following: For every $i \in [1 \dots \ell]$, $\mathcal{S}$ sets $w^b_{i,\pi_{1-b,i}} = (w')^{1-b}_i$ and chooses $w^b_{i,1-\pi_{1-b,i}}$ uniformly at random. For every $i \in [1 \dots \ell]$, $\mathcal{S}$ sets $(w')^b_i = w^{1-b}_{i,\pi_{b,i}}$. $\mathcal{S}$ returns these values to $\mathcal{A}$ and continues the simulation for the case that both parties are corrupted.

Corruption of party P_b after messages have been exchanged in Stage 3. $\mathcal{S}$ corrupts the ideal-model $\tilde{P}_b \in \{\tilde{I}, \tilde{R}\}$, obtains its input π_b, and output of either skey or $\perp$.

- If party P_{1-b} is not yet corrupted then $\mathcal{S}$ does the following: If the output is skey then $\mathcal{S}$ chooses $w^b_{i,0}, w^b_{i,1}, (w')^b_i$ for every $i \in [1 \dots \ell]$ uniformly at random conditioned on $K \oplus K'$ being consistent with $\mathsf{test}_b \oplus \mathsf{test}'_b$, $\mathsf{test}_{1-b} \oplus \mathsf{test}'_{1-b}$ and returns these values to $\mathcal{A}$. If the output is $\perp$ $\mathcal{S}$ chooses $w^b_{i,0}, w^b_{i,1}, (w')^b_i$ for every $i \in [1 \dots \ell]$ uniformly at random conditioned on $K \oplus K'$ being consistent with $\mathsf{test}_b \oplus \mathsf{test}'_b$ and returns these values to $\mathcal{A}$. $\mathcal{S}$ continues the simulation for the case that party P_b is corrupted.

- If party P_{1-b} has already been corrupted then the values $w_{i,0}^{1-b}, w_{i,1}^{1-b}, (w')_i^{1-b}$ for every $i \in [1 \ldots \ell]$ are already known and so $\mathcal{S}$ must ensure that the values of $w_{i,0}^{b}, w_{i,1}^{b}, (w')_i^{b}$ for every $i \in [1 \ldots \ell]$ are consistent with these values.

  Thus, if the output is skey, $\mathcal{S}$ does the following: For every $i \in [1 \ldots \ell]$, $\mathcal{S}$ sets $w_{i,\pi_{1-b,i}}^{b} = (w')_i^{1-b}$ and chooses $w_{i,1-\pi_{1-b,i}}^{b}$ uniformly at random. For every $i \in [1 \ldots \ell]$, $\mathcal{S}$ sets $(w')_i^{b} = w_{i,\pi_{b,i}}^{1-b}$. $\mathcal{S}$ returns these values to $\mathcal{A}$. If the output is $\perp$ then there must be some $i^* \in [1 \ldots \ell]$ such that $\pi_{b,i^*} \neq \pi_{1-b,i^*}$. Thus, $\mathcal{S}$ does the following: For every $i \in [1 \ldots \ell]$, $\mathcal{S}$ sets $w_{i,\pi_{1-b,i}}^{b} = (w')_i^{1-b}$ and chooses $w_{i,1-\pi_{1-b,i}}^{b}$ uniformly at random conditioned on $K \oplus K'$ being consistent with $\mathsf{test}_b \oplus \mathsf{test}'_b$. (note that this is always possible since we can set $w_{i^*,\pi_{b,i^*}}^{b}$ to be whatever we want. For every $i \in [1 \ldots \ell]$, $\mathcal{S}$ sets $(w')_i^{b} = w_{i,\pi_{b,i}}^{1-b}$. $\mathcal{S}$ returns these values to $\mathcal{A}$ and continues the simulation for the case that both parties are corrupted.

Proof of Indistinguishability. We show that $\text{IDEAL}_{\mathcal{F}_{\mathsf{re}},\mathcal{S},\mathcal{Z}} \equiv \text{REAL}_{\Pi_{\mathsf{REfromOT}},\mathcal{A},\mathcal{Z}}$. The main idea of the proof is this: Let π_I and π_R be the inputs of I and R. (In case one or both of them are corrupted, then set π_I, resp. π_R, to be the string that the simulator extracts from I, resp. R) If $\pi_I = \pi_R$, it is easy to see that the simulation is perfect. If $\pi_I \neq \pi_R$, then we claim that the key K_R that the responder R computes is uniformly random from the view of $\mathcal{A}$. This is because the adversary $\mathcal{A}$ receives $w'_{i,\pi_{I,i}}$ for all $i \in [\ell]$ and K_R is computed as

$$K_R = \bigoplus_{i=1}^{\ell} w'_{i,\pi_{R,i}}$$

Without loss of generality, say $\pi_{R,1} \neq \pi_{I,1}$. Then, $(w')_{1,\pi_{R,1}}^{b}$ is uniformly random from the view of $\mathcal{A}$. In particular, this means that K_R is uniformly random from $\mathcal{A}$'s view, and thus, the message test' that it gets is correctly distributed. Furthermore, the simulated distribution is identical to the distribution generated by executing Π_{REfromOT} except for this. Thus, it follows that $\text{IDEAL}_{\mathcal{F}_{\mathsf{re}},\mathcal{S},\mathcal{Z}} \equiv \text{REAL}_{\Pi_{\mathsf{REfromOT}},\mathcal{A},\mathcal{Z}}$.

2.3 Implementing the Split $\mathcal{F}_{\mathsf{re}}$ Functionality without Authenticated Channels

The protocol Π_{REfromOT} in Section 2.2 implements the randomized equality testing functionality $\mathcal{F}_{\mathsf{re}}$ in the *authenticated channels* model. In this section, we use the results of Barak et al. [BCL+05] together with a specific implementation of the $\mathcal{F}_{\mathsf{OT}}$ functionality from Garay, Wichs and Zhou [GWZ09] to show that the protocol can be transformed into a protocol $\mathsf{s}\Pi_{\mathsf{REfromOT}}$ that implements the "split version" of the equality-testing functionality (called $\mathsf{s}\mathcal{F}_{\mathsf{re}}$). The new protocol does not assume authenticated channels, and yet, retain security against adaptive corruptions. For completeness, we define $\mathsf{s}\mathcal{F}_{\mathsf{re}}$ in the full version, and state the result of this transformation in Theorem 2.

Theorem 2. *There is a protocol* $\mathsf{s}\Pi_{\mathsf{REfromOT}}$ *that securely implements the split functionality* $\mathsf{s}\mathcal{F}_{\mathsf{re}}$ *in the* $\mathcal{F}_{\mathsf{crs}}$*-hybrid model, tolerating adaptive corruptions without erasures and without authenticated channels. The protocol is based on either DDH or the decisional composite residuosity (DCR) assumption, runs in a constant number of rounds and exchanges a constant number of group elements per session key.*

Proof. First, we note that the multi-session version of $\mathcal{F}_{\mathsf{re}}$ can be implemented using access to the multi-session version of $\mathcal{F}_{\mathsf{OT}}$ – essentially each new session of $\mathcal{F}_{\mathsf{re}}$ utilizes new invocation of the OT protocol. Then, using the result of Garay et al., the multi-session version of $\mathcal{F}_{\mathsf{OT}}$ can be implemented in the $\mathcal{F}_{\mathsf{crs}}$-hybrid model under either the DDH or DCR assumption. Put together, we have a protocol that implements the multi-session version of $\mathcal{F}_{\mathsf{re}}$ in the $\mathcal{F}_{\mathsf{crs}}$-hybrid model. Now, a theorem of Barak et al. [BCL+05] shows that any such protocol can be converted into a protocol for the split functionality $\mathsf{s}\mathcal{F}_{\mathsf{re}}$.

3 Concurrent PAKE from OT

We present a framework for concurrent PAKE based on OT, and show how to instantiate the underlying building blocks from search assumptions.

Definitions. We begin with an overview of the security definition for concurrent PAKE given in [GK10, BPR00] (detailed definitions are presented in the full version). Informally, an adversary interacts with various instances in the following ways:

- it can initiate and interact in an instance with any honest party;
- it can ask for the session key for some completed instance;
- it can passively eavesdrop on an instance between two honest parties;

The first two modes of interaction constitute a so-called “on-line attack”; the third one does not. Informally, a secure PAKE protocol guarantees secrecy of the session keys even in the presence of an active adversary. That is, we say that an adversary succeeds if it manages to distinguish the session key for some fresh instance from random (where an instance is “fresh” if the adversary has not previous asked for its session key). We use $\mathbf{AdvPAKE}_{\mathcal{A}}(\lambda)$ to denote the success probability of an adversary $\mathcal{A}$. Now, an adversary can always succeed with probability 1 by trying all passwords in the dictionary one-by-one. Informally, a protocol is secure if this is the best an adversary can do. Formally, we say that an instance represents an *on-line attack* if the adversary participated in the instance. The number of on-line attacks is a bound on the number of passwords the adversary could have tested in an on-line fashion.

We say that a PAKE protocol is *concurrently secure with explicit mutual authentication* if for all dictionaries $\mathcal{D}_\lambda$ and for all PPT adversaries $\mathcal{A}$ making at most $Q(\lambda)$ online attacks, the quantity $\mathbf{AdvPAKE}_{\mathcal{A}}(\lambda) - Q(\lambda)/|\mathcal{D}_\lambda|$ is bounded by a negligible function.

3.1 A General Framework

We present our general framework for concurrent PAKE (a variant of the Groce-Katz protocol) in Fig 4. The ingredients are a labeled CCA-secure encryption $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, and an OT protocol $(\mathbf{S}, \mathbf{R})$ in the CRS model that is (1) computationally hiding against $\mathbf{S}^*$ and (2) straight-line extractable and statistically hiding against $\mathbf{R}^*$

Overview. Here is an overview of the construction, assuming 1-out-of-$|\mathcal{D}|$ OT for simplicity:

- Both parties U and U' run the basic protocol: U acts as the OT receiver and uses as input his password $\pi_{U,U'}$. U' acts as the OT sender and picks $|\mathcal{D}|$ random strings $r_1, \ldots, r_{|\mathcal{D}|}$ as input. U parses the OT output as $\mathsf{skey}\|\mathsf{rand}$ and U' parses $r_{\pi_{U,U'}}$ as $\mathsf{skey}'\|\mathsf{rand}'$.
- U' sends an encryption C of $\pi_{U,U'}$ using randomness rand' and as label the transcript of the basic protocol (plus the identities), under a public key for a CCA2-secure encryption scheme that is part of the CRS.
- U checks if C is computed with the same password by encrypting $\pi_{U,U'}$ with randomness rand. If the ciphertext matches C, both parties output skey as the session key.

See Figure 4 for a description of the protocol. We establish the following:

Proposition 5. *Suppose* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is a labeled CCA-secure encryption scheme and* $(\mathbf{S}, \mathbf{R})$ *is an OT protocol in the CRS model that is (1) computationally hiding against* $\mathbf{S}^*$ *and (2) straight-line extractable and statistically hiding against* $\mathbf{R}^*$*. Then, the protocol in Fig 4 is a secure PAKE protocol with explicit mutual authentication.*

Proof Overview. We begin with a brief argument of security for the case where there is a single instance on the left and on the right:

- First, we want to argue that by OT security against senders, the LHS Stage 1 hides U's input π (which we extract) and so $\mathcal{A}$'s input $\tilde{\pi}$ to the RHS Stage 1 must be "independent" of π. This would imply that with probability $1 - 1/|\mathcal{D}|$, we have $\tilde{\pi} \neq \pi$ and thus U''s challenge test is statistically hidden from $\mathcal{A}$. Thus we bound the probability $\mathcal{A}$ wins on the right.
- Next, observe that if $\mathcal{A}$ plays a relaying strategy for Stages 1 on the left and the right, then it must continue to play a relaying strategy for U or U' to accept (since the transcript of Stage 1 uniquely determines an accepting transcript for the protocol). Otherwise, the labels for the CCA encryptions in Stage 2 on the left and right must differ. We may then argue that U''s encryption of π on the right does not help $\mathcal{A}$ provide a valid encryption of π on the left. Thus, we bound the probability $\mathcal{A}$ wins on the left.

The main subtlely lies in the first step: as stated, we require the underlying OT protocol to hide the receiver's input against a cheating sender that has access to an extraction trapdoor (which would require that the underlying OT protocol non-malleable). We bypass this issue via a more refined analysis. We defer the formal proof to the full version.

Concurrent PAKE

Common Reference String: The CRS for $(\mathbf{S}, \mathbf{R})$ and a public key $\textsc{pk}$ for $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

Inputs: Parties U and U' participating in instances Π_U^i and $\Pi_{U'}^j$, respectively, hold joint password $\pi = \pi_{U,U'} \in \mathcal{D}$, where $\mathcal{D} \subseteq \{0,1\}^\ell$.

PAKE phase:

Stage 1. U and U' engage in ℓ executions of $(\mathbf{S}, \mathbf{R})$ in parallel. In the i'th execution of $(\mathbf{S}, \mathbf{R})$:

- U' chooses a pair of random strings $(w_i^0, w_i^1) \leftarrow_{\mathrm{R}} \{0,1\}^{3\lambda}$ and runs $\mathbf{S}$ with input (w_i^0, w_i^1).
- U runs $\mathbf{R}$ with input $\pi_i \in \{0,1\}$ and receives output $w_i' := w_i^{\pi_i}$.

Stage 2. U' computes

$$\mathsf{rand}||\mathsf{test}||\mathsf{skey} := \bigoplus_{i=1}^{\ell} w_i^{\pi_i} \qquad (\text{where } \mathsf{rand}, \mathsf{skey}, \mathsf{test} \in \{0,1\}^\lambda)$$

and sends $C := \mathsf{Enc}_{\textsc{pk}}^{U||U'||\mathsf{trans}}(\pi; \mathsf{rand})$ to U where trans is the concatenation of the transcripts of all ℓ executions of $(\mathbf{S}, \mathbf{R})$.

Stage 3. U computes

$$\mathsf{rand}'||\mathsf{test}'||\mathsf{skey}' := \bigoplus_{i=1}^{\ell} w_i' \qquad (\text{where } \mathsf{rand}', \mathsf{skey}', \mathsf{test}' \in \{0,1\}^\lambda)$$

and sends test' and sets its session key to skey' if $C = \mathsf{Enc}_{\textsc{pk}}^{U||U'||\mathsf{trans}}(\pi; \mathsf{rand}')$ and aborts otherwise.

Stage 4. U' sets its session key to skey if $\mathsf{test}' = \mathsf{test}$ and aborts otherwise.

Fig. 4. Concurrent PAKE

3.2 Instantiating the Underlying OT

We present two approaches for instantiating the underlying OT in our general framework for concurrent PAKE. Recall that we require an OT protocol $(\mathbf{S}, \mathbf{R})$ in the CRS model that is (1) computationally hiding against $\mathbf{S}^*$ and (2) straight-line extractable and statistically hiding against $\mathbf{R}^*$.

Instantiations from Dual-Mode Encryption. In [PVW08], Peikert, Vaikuntanathan and Waters present a novel abstraction called "dual-mode cryptosystems" and show how to construct UC-secure OT from any dual-mode cryptosystem in the CRS model (where every pair of parties share a CRS). Moreover, in the so-called "messy mode", the ensuing OT protocol achieves statistical security against a corrupted receiver. We observe that the same protocol also achieves the security guarantees that we require. Combined with our general framework, we obtain the result stated in Proposition 2.

Instantiations from CDH and hardness of factoring. We start with a two-message bit-OT protocol in the CRS model that is (1) computationally hiding against $\mathbf{R}^*$ and (2) straight-line extractable and statistically hiding against $\mathbf{S}^*$ (note these are the "opposite" properties of what we need). Indeed, the Bellare-Micali OT protocol [BM89] based on CDH satisfies these properties. To obtain an instantiation based on hardness of factoring, we use the fact that CDH over $\mathbb{Z}_N^*$ is as hard as factoring [HK09, M88, S85]. We note that 2-message OT protocols were given by Halevi and Kalai [HK07]; however, their constructions are based on hash proof systems and thus are limited to decisional assumptions.

Next, we apply the "OT reversal" transformation of Wolf and Wullschleger [WW06] to obtain a three-message bit-OT protocol. We show that the ensuing bit-OT protocol has the properties we need, namely computationally hiding against $\mathbf{S}^*$ and straight-line extractable and statistically hiding against $\mathbf{R}^*$. Finally, we apply the bit OT to string OT transformation of Brassard, et. al [BCR86] (which is round-preserving) to obtain a string OT protocol with the properties we need. We defer details to the full version.

References

[ABCP06] Abdalla, M., Bresson, E., Chevassut, O., Pointcheval, D.: Password-Based Group Key Exchange in a Constant Number of Rounds. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 427–442. Springer, Heidelberg (2006)

[ACCP08] Abdalla, M., Catalano, D., Chevalier, C., Pointcheval, D.: Efficient Two-Party Password-Based Key Exchange Protocols in the UC Framework. In: Malkin, T. (ed.) CT-RSA 2008. LNCS, vol. 4964, pp. 335–351. Springer, Heidelberg (2008)

[ACP09] Abdalla, M., Chevalier, C., Pointcheval, D.: Smooth Projective Hashing for Conditionally Extractable Commitments. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 671–689. Springer, Heidelberg (2009)

[AP06] Abdalla, M., Pointcheval, D.: A Scalable Password-Based Group Key Exchange Protocol in the Standard Model. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 332–347. Springer, Heidelberg (2006)

[BCL$^+$05] Barak, B., Canetti, R., Lindell, Y., Pass, R., Rabin, T.: Secure Computation Without Authentication. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 361–377. Springer, Heidelberg (2005)

[BCR86] Brassard, G., Crépeau, C., Robert, J.M.: All-or-Nothing Disclosure of Secrets. In: Brassard, G. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 234–238. Springer, Heidelberg (1987)

[BM89] Bellare, M., Micali, S.: Non-interactive Oblivious Transfer and Applications. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 547–557. Springer, Heidelberg (1990)

[BM93] Bellovin, S.M., Merritt, M.: Augmented encrypted key exchange: A password-based protocol secure against dictionary attacks and password file compromise. In: Ashby, V. (ed.) 1st ACM Conference on Computer and Communications Security, pp. 244–250. ACM Press (November 1993)

[BMP00] Boyko, V., MacKenzie, P., Patel, S.: Provably Secure Password-Authenticated Key Exchange Using Diffie-Hellman. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 156–171. Springer, Heidelberg (2000)

[BPR00] Bellare, M., Pointcheval, D., Rogaway, P.: Authenticated Key Exchange Secure against Dictionary Attacks. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 139–155. Springer, Heidelberg (2000)

[C01] Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: 42nd Annual Symposium on Foundations of Computer Science (FOCS), pp. 136–145. IEEE Computer Society Press (2001)

[CCGS10] Camenisch, J., Casati, N., Gross, T., Shoup, V.: Credential Authenticated Identification and Key Exchange. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 255–276. Springer, Heidelberg (2010)

[CHK$^+$05] Canetti, R., Halevi, S., Katz, J., Lindell, Y., MacKenzie, P.: Universally Composable Password-Based Key Exchange. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 404–421. Springer, Heidelberg (2005)

[CS98] Cramer, R., Shoup, V.: A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998)

[CS02] Cramer, R., Shoup, V.: Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002)

[G08] Gennaro, R.: Faster and Shorter Password-Authenticated Key Exchange. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 589–606. Springer, Heidelberg (2008)

[GJO10] Goyal, V., Jain, A., Ostrovsky, R.: Password-Authenticated Session-Key Generation on the Internet in the Plain Model. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 277–294. Springer, Heidelberg (2010)

[GK10] Groce, A., Katz, J.: A new framework for efficient password-based authenticated key exchange. In: ACM Conference on Computer and Communications Security, pp. 516–525 (2010)

[GL01] Goldreich, O., Lindell, Y.: Session-Key Generation Using Human Passwords Only. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 408–432. Springer, Heidelberg (2001), http://eprint.iacr.org/2000/057

[GL03] Gennaro, R., Lindell, Y.: A Framework for Password-Based Authenticated Key Exchange (Extended Abstract). In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 524–543. Springer, Heidelberg (2003), http://eprint.iacr.org/2003/032.ps.gz

[GWZ09] Garay, J.A., Wichs, D., Zhou, H.-S.: Somewhat Non-committing Encryption and Efficient Adaptively Secure Oblivious Transfer. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 505–523. Springer, Heidelberg (2009)

[HK07] Kalai, Y.T.: Smooth Projective Hashing and Two-Message Oblivious Transfer. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 78–95. Springer, Heidelberg (2005); Cryptology ePrint Archive, Report 2007/118 (2007)

[HK09] Hofheinz, D., Kiltz, E.: The Group of Signed Quadratic Residues and Applications. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 637–653. Springer, Heidelberg (2009)

[JG04] Jiang, S., Gong, G.: Password Based Key Exchange with Mutual Authentication. In: Handschuh, H., Hasan, M.A. (eds.) SAC 2004. LNCS, vol. 3357, pp. 267–279. Springer, Heidelberg (2004)

[KMTG05] Katz, J., MacKenzie, P., Taban, G., Gligor, V.D.: Two-Server Password-Only Authenticated Key Exchange. In: Ioannidis, J., Keromytis, A.D., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 1–16. Springer, Heidelberg (2005)

[KOY01] Katz, J., Ostrovsky, R., Yung, M.: Efficient Password-Authenticated Key Exchange Using Human-Memorable Passwords. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 475–494. Springer, Heidelberg (2001)

[KV09] Katz, J., Vaikuntanathan, V.: Smooth Projective Hashing and Password-Based Authenticated Key Exchange from Lattices. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 636–652. Springer, Heidelberg (2009)

[KV11] Katz, J., Vaikuntanathan, V.: Round-Optimal Password-Based Authenticated Key Exchange. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 293–310. Springer, Heidelberg (2011)

[M88] McCurley, K.S.: A key distribution system equivalent to factoring. Journal of Cryptology 1(2), 95–105 (1988)

[MPS00] MacKenzie, P., Patel, S., Swaminathan, R.: Password-Authenticated Key Exchange Based on RSA. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 599–613. Springer, Heidelberg (2000)

[NV04] Nguyen, M.-H., Vadhan, S.P.: Simpler Session-Key Generation from Short Random Passwords. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 428–445. Springer, Heidelberg (2004)

[PVW08] Peikert, C., Vaikuntanathan, V., Waters, B.: A Framework for Efficient and Composable Oblivious Transfer. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 554–571. Springer, Heidelberg (2008)

[PW08] Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: Ladner, R.E., Dwork, C. (eds.) 40th Annual ACM Symposium on Theory of Computing (STOC), pp. 187–196. ACM Press (May 2008)

[S85] Shmuely, Z.: Composite diffie-hellman public-key generating systems are hard to break. Technical Report 356, Technion (1985)

[WW06] Wolf, S., Wullschleger, J.: Oblivious Transfer Is Symmetric. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 222–232. Springer, Heidelberg (2006)