SPHF-Friendly Non-Interactive Commitments

Michel Abdalla[1], Fabrice Benhamouda[1], Olivier Blazy[2], Céline Chevalier[3], and David Pointcheval[1]

[1] École Normale Supérieure, CNRS-INRIA, Paris, France
[2] Ruhr-Universität Bochum, Germany
[3] Université Panthéon-Assas, Paris, France

Abstract. In 2009, Abdalla *et al.* proposed a reasonably practical password-authenticated key exchange (PAKE) secure against adaptive adversaries in the universal composability (UC) framework. It exploited the Canetti-Fischlin methodology for commitments and the Cramer-Shoup smooth projective hash functions (SPHFs), following the Gennaro-Lindell approach for PAKE. In this paper, we revisit the notion of non-interactive commitments, with a new formalism that implies UC security. In addition, we provide a quite efficient instantiation. We then extend our formalism to SPHF-friendly commitments. We thereafter show that it allows a blackbox application to one-round PAKE and oblivious transfer (OT), still secure in the UC framework against adaptive adversaries, assuming reliable erasures and a single global common reference string, even for multiple sessions. Our instantiations are more efficient than the Abdalla *et al.* PAKE in Crypto 2009 and the recent OT protocol proposed by Choi *et al.* in PKC 2013. Furthermore, the new PAKE instantiation is the first one-round scheme achieving UC security against adaptive adversaries.

1 Introduction

Commitment schemes are one of the most fundamental primitives in cryptography, serving as a building block for many cryptographic applications such as zero-knowledge proofs [22] and secure multi-party computation [21]. In a typical commitment scheme, there are two main phases. In a *commit* phase, the committer computes a commitment C for some message x and sends it to the receiver. Then, in an *opening* phase, the committer releases some information δ to the receiver which allows the latter to verify that C was indeed a commitment of x. To be useful in practice, a commitment scheme should satisfy two basic security properties. The first one is *hiding*, which informally guarantees that no information about x is leaked through the commitment C. The second one is *binding*, which guarantees that the committer cannot generate a commitment C that can be successfully opened to two different messages.

Smooth Projective Hash Functions (SPHFs) were introduced by Cramer and Shoup [17] as a means to design chosen-ciphertext-secure public-key encryption schemes. In addition to providing a more intuitive abstraction for their original public-key encryption scheme in [16], the notion of SPHF also enabled new efficient instantiations of their scheme under different complexity assumptions, such as quadratic residuosity. Due to its usefulness, the notion of SPHF was later extended to several other contexts, such as password-authenticated key exchange (PAKE) [20], oblivious transfer (OT) [27,15], and blind signatures [7,6].

Password-Authenticated Key Exchange (PAKE) protocols were proposed in 1992 by Bellovin and Merritt [5] where authentication is done using a simple password, possibly drawn from a small space subject to exhaustive search. Since then, many schemes have been proposed and studied. SPHFs have been extensively used, starting with the work of Gennaro and Lindell [20] which generalized an earlier construction by Katz, Ostrovsky, and Yung (KOY) [29], and followed by several other works [11,2]. More recently, a variant of SPHFs proposed by Katz and Vaikuntanathan even allowed the construction of one-round PAKE schemes [30,6].

The first ideal functionality for PAKE protocols in the UC framework [8,12] was proposed by Canetti *et al.* [11], who showed how a simple variant of the Gennaro-Lindell methodology [20] could lead to a secure protocol. Though quite efficient, their protocol was not known to be secure against adaptive adversaries, that are capable of corrupting players at any time, and learn their internal states. The first ones to propose an adaptively secure PAKE in the UC framework were Barak *et al.* [3] using general techniques from multi-party computation (MPC). Though conceptually simple, their solution results in quite inefficient schemes.

The first reasonably practical adaptively secure PAKE was proposed by Abdalla *et al.* [2], following the Gennaro-Lindell methodology with the Canetti-Fischlin commitment [10]. They had to build a complex SPHF to handle the verification of such a commitment. Thus, the communication complexity was high and the protocol required four rounds. No better adaptively secure scheme has been proposed so far.

Oblivious Transfer (OT) was introduced in 1981 by Rabin [34] as a way to allow a receiver to get exactly one out of k messages sent by another party, the sender. In these schemes, the receiver should be oblivious to the other values, and the sender should be oblivious to which value was received. Since then, several instantiations and optimizations of such protocols have appeared in the literature, including proposals in the UC framework [31,13].

More recently, new instantiations have been proposed, trying to reach round-optimality [26], or low communication costs [33]. The 1-out-of-2 OT scheme by Choi *et al.* [15] based on the DDH assumption seems to be the most efficient one among those that are secure against adaptive corruptions in the CRS model with erasures. But it does not scale to 1-out-of-k OT, for $k > 2$.

1.1 Properties of Commitment Schemes

Basic Properties. In addition to the binding and hiding properties, certain applications may require additional properties from a commitment scheme. One such property is *equivocability* [4], which guarantees that a commitment C can be opened in more than a single way when in possession of a certain trapdoor information. Another one is *extractability*, which allows the computation of the message x committed in C when in possession of a certain trapdoor information. Yet another property that may also be useful for cryptographic applications is *non-malleability* [18], which ensures that the receiver of a unopened commitment C for a message x cannot generate a commitment for a message that is related to x.

Though commitment schemes satisfying stronger properties such as *non-malleability*, *equivocability*, and *extractability* may be useful for solving specific problems, they usually stop short of guaranteeing security when composed with arbitrary protocols. To address this problem, Canetti and Fischlin [10] proposed an ideal functionality for commitment schemes in the universal composability (UC) framework [8] which guarantees all these properties simultaneously and remain secure even under concurrent compositions with arbitrary protocols. Unfortunately, they also showed that such commitment schemes can only be realized if one makes additional setup assumptions, such as the existence of a common reference string (CRS) [10], random oracles [25], or secure hardware tokens [28].

Equivocable and Extractable Commitments. As the work of Canetti and Fischlin [10], this work also aims to build *non-interactive* commitment schemes which can simultaneously guarantee *non-malleability*, *equivocability*, and *extractability* properties. To this end, we first define a new notion of commitment scheme, called $\mathtt{E}^2$-commitments, for which there exists an alternative setup algorithm, whose output is computationally indistinguishable from that of a normal setup algorithm and which outputs a common trapdoor that allows for both equivocability and extractability: this trapdoor not only allows for the extraction of a committed message, but it can also be used to create simulated commitments which can be opened to any message.

To define the security of $\mathtt{E}^2$-schemes, we first extend the security notions of standard equivocable commitments and extractable commitments to the $\mathtt{E}^2$-commitment setting: Since the use of a common trapdoor for equivocability and extractability could potentially be exploited by an adversary to break the extractability or equivocability properties of an $\mathtt{E}^2$-commitment scheme, we define stronger versions of these notions, which account for the fact that the same trapdoor is used for both extractability or equivocability. In particular, in these stronger notions, the adversary is given oracle access to the simulated commitment and extractor algorithms.

Finally, after defining the security of $\mathtt{E}^2$-schemes, we further show that these schemes remain secure even under arbitrary composition with other cryptographic protocols. More precisely, we show that any $\mathtt{E}^2$–commitment scheme which meets the strong versions of the equivocability or extraction notions is a non-interactive UC-secure (multiple) commitment scheme in the presence of adaptive adversaries, assuming reliable erasures and a single global CRS.

SPHF-Friendly Commitments. In this work, we are interested in building non-interactive $\mathtt{E}^2$-commitments, to which smooth projective hash functions can be efficiently associated. Unfortunately, achieving this goal is not so easy due to the equivocability property of $\mathtt{E}^2$-commitments. To understand why, let X be the domain of an SPHF function and let L be some underlying NP language such that it is computationally hard to distinguish a random element in L from a random element in $X \setminus L$. A key property of these SPHF functions that makes them so useful for applications such as PAKE and OT is that, for words C in L, their values can be computed using either a *secret* hashing key hk or a *public* projected key hp together a witness w to the fact that C is indeed in L. A typical example of a language in which we are interested is the language L_x corresponding to the set of elements $\{C\}$ such that C is a valid commitment of x. Unfortunately, when commitments are equivocable, the language L_x containing the set of valid commitments of x may not be well defined since a commitment C could potentially be opened to any x. To get around this problem and be able to use SPHFs with $\mathtt{E}^2$-commitments, we show that it suffices for an $\mathtt{E}^2$-commitment scheme to satisfy two properties. The first one is the stronger version of the equivocability notion, which guarantees that equivocable commitments are computationally indistinguishable from normal commitments, even when given oracle access to the simulated commitment and extractor algorithms. The second one, which is called *robustness*, is new and guarantees that commitments generated by polynomially-bounded adversaries are perfectly binding. Finally, we say that a commitment scheme is *SPHF-friendly* if it satisfies both properties and if it admits an SPHF on the languages L_x.

1.2 Contributions

A new SPHF-friendly $\mathtt{E}^2$-commitment construction. First, we define the notion of SPHF-friendly $\mathtt{E}^2$-commitment together with an instantiation. The new construction, which is called $\mathcal{E}^2\mathcal{C}$ and described in Section 4, is inspired by the commitment schemes in [10,13,2]. Like the construction in [2], it combines a variant of the Cramer-Shoup encryption scheme (as an extractable commitment scheme) and an equivocable commitment scheme to be able to simultaneously achieve both equivocability and extractability. However, unlike the construction in [2], we rely on Haralambiev's perfectly hiding commitment [24, Section 4.1.4], instead of the Pedersen commitment [32].

Since the opening value of Haralambiev's scheme is a group element that can be encrypted in one ElGamal-like ciphertext to allow extractability, this globally leads to a better communication and computational complexity for the commitment. The former is linear in $m \cdot \mathfrak{K}$, where m is the bit-length of the committed value and $\mathfrak{K}$, the security parameter. This is significantly better than the extractable commitment construction in [2] which was linear in $m \cdot \mathfrak{K}^2$, but asymptotically worse than the two proposals in [19] that are linear in $\mathfrak{K}$, and thus independent of m. However, we point out the latter proposals in [19] are not SPHF-friendly since they are not robust.

We then show in Theorem 4 that a labeled $\mathtt{E}^2$-commitment satisfying stronger notions of equivocability and extractability is a non-interactive UC-secure commitment scheme in the presence of adaptive adversaries, assuming reliable erasures and a single global CRS, and we apply this result to our new construction.

One-round adaptively secure PAKE. Second, we provide a generic construction of a one-round UC-secure PAKE from any SPHF-friendly commitment. The UC-security holds against adaptive adversaries, assuming reliable erasures and a single global CRS, as shown in Section 6. In addition to being the first one-round adaptively secure PAKE, our new scheme also enjoys a much better communication complexity than previous adaptively secure PAKE schemes. For instance, in comparison to the PAKE in [2], which is currently the most efficient adaptively secure PAKE, the new scheme gains a factor of $\mathfrak{K}$ in the overall communication complexity, where $\mathfrak{K}$ is the security parameter. However, unlike their scheme, our new construction requires pairing-friendly groups.

Three-round adaptively secure 1-out-of-k OT. Third, we provide a generic construction of a three-round UC-secure 1-out-of-k OT from any SPHF-friendly commitment. The UC-security holds against adaptive adversaries, assuming reliable erasures and a single global CRS, as shown in Section 7. Besides decreasing the total number of rounds with respect to existing OT schemes with similar security levels, our resulting protocol also has a better communication complexity than the best known solution so far [15]. Moreover, our construction is more general and provides a solution for 1-out-of-k OT schemes while the solution in [15] only works for $k = 2$.

Due to space restrictions, complete proofs and some details were postponed to the full version [1].

2 Basic Notions for Commitments

We first review the basic definitions of non-interactive commitments, with some examples. Then, we consider the classical additional notions of equivocability and extractability. In this paper, the qualities of adversaries will be measured by their successes and advantages in certain experiments $\mathsf{Exp}^{\mathsf{sec}}$ or $\mathsf{Exp}^{\mathsf{sec}\text{-}b}$ (between the cases $b = 0$ and $b = 1$), denoted $\mathsf{Succ}^{\mathsf{sec}}(\mathcal{A}, \mathfrak{K})$ and $\mathsf{Adv}^{\mathsf{sec}}(\mathcal{A}, \mathfrak{K})$ respectively, while the security of a primitive will be measured by the maximal successes or advantages of any adversary running within a time bounded by some t in the appropriate experiments, denoted $\mathsf{Succ}^{\mathsf{sec}}(t)$ and $\mathsf{Adv}^{\mathsf{sec}}(t)$ respectively. Adversaries can keep state during the different phases. We denote $\overset{\$}{\leftarrow}$ the outcome of a probabilistic algorithm or the sampling from a uniform distribution.

2.1 Non-Interactive Labeled Commitments

A non-interactive labeled commitment scheme $\mathcal{C}$ is defined by three algorithms:

- $\mathsf{SetupCom}(1^{\mathfrak{K}})$ takes as input the security parameter $\mathfrak{K}$ and outputs the global parameters, passed through the CRS ρ to all other algorithms;
- $\mathsf{Com}^{\ell}(x)$ takes as input a label ℓ and a message x, and outputs a pair (C, δ), where C is the commitment of x for the label ℓ, and δ is the corresponding opening data (a.k.a. decommitment information). This is a probabilistic algorithm;

```
$\mathsf{Exp}_{\mathcal{A}}^{\mathtt{hid}\text{-}b}(\mathfrak{K})$
    $\rho \overset{\$}{\leftarrow} \mathsf{SetupCom}(1^{\mathfrak{K}})$
    $(\ell, x_0, x_1, \mathsf{state}) \overset{\$}{\leftarrow} \mathcal{A}(\rho)$
    $(C, \delta) \overset{\$}{\leftarrow} \mathsf{Com}^{\ell}(x_b)$
    return $\mathcal{A}(\mathsf{state}, C)$
```

```
$\mathsf{Exp}_{\mathcal{A}}^{\mathtt{bind}}(\mathfrak{K})$
    $\rho \overset{\$}{\leftarrow} \mathsf{SetupCom}(1^{\mathfrak{K}})$
    $(C, \ell, x_0, \delta_0, x_1, \delta_1) \overset{\$}{\leftarrow} \mathcal{A}(\rho)$
    if $\neg\mathsf{VerCom}^{\ell}(C, x_0, \delta_0)$ then return $0$
    if $\neg\mathsf{VerCom}^{\ell}(C, x_1, \delta_1)$ then return $0$
    return $x_0 \neq x_1$
```

Fig. 1. Hiding and Binding Properties

- $\mathsf{VerCom}^{\ell}(C, x, \delta)$ takes as input a commitment C, a label ℓ, a message x, and the opening data δ and outputs 1 (true) if δ is a valid opening data for C, x and ℓ. It always outputs 0 (false) on $x = \bot$.

Using the experiments $\mathsf{Exp}_{\mathcal{A}}^{\mathtt{hid}}(\mathfrak{K})$ and $\mathsf{Exp}_{\mathcal{A}}^{\mathtt{bind}}(\mathfrak{K})$ defined in Figure 1, one can state the basic properties:

- *Correctness*: for all correctly generated CRS ρ, all commitments and opening data honestly generated pass the verification VerCom test: for all ℓ, x, if $(C, \delta) \overset{\$}{\leftarrow} \mathsf{Com}^{\ell}(x)$, then $\mathsf{VerCom}^{\ell}(C, x, \delta) = 1$;
- *Hiding Property*: the commitment does not leak any information about the committed value. $\mathcal{C}$ is said (t, ε)-hiding if $\mathsf{Adv}_{\mathcal{C}}^{\mathtt{hid}}(t) \leq \varepsilon$.
- *Binding Property*: no adversary can open a commitment in two different ways. $\mathcal{C}$ is said (t, ε)-binding if $\mathsf{Succ}_{\mathcal{C}}^{\mathtt{bind}}(t) \leq \varepsilon$.

Correctness is always perfectly required, and one can also require either the binding or the hiding property to be perfect.

The reader can remark that labels are actually useless in the hiding and the binding properties. But they will become useful in $\mathtt{E}^2$-commitment schemes introduced in the next section. This is somehow similar to encryption scheme: labels are useless with encryption schemes which are just `IND-CPA`, but are very useful with `IND-CCA` encryption schemes.

2.2 Perfectly Binding Commitments: Public-Key Encryption

To get perfectly binding commitments, classical instantiations are public-key encryption schemes, which additionally provide extractability (see below). The encryption algorithm is indeed the commitment algorithm, and the random coins become the opening data that allow to check the correct procedure of the commit phase. The hiding property relies on the indistinguishability (`IND-CPA`), which is computationally achieved, whereas the binding property relies on the correctness of the encryption scheme and is perfect.

Let us define the ElGamal-based commitment scheme:

- $\mathsf{SetupCom}(1^{\mathfrak{K}})$ chooses a cyclic group $\mathbb{G}$ of prime order p, g a generator for this group and a random scalar $z \overset{\$}{\leftarrow} \mathbb{Z}_p$. It sets the CRS $\rho = (\mathbb{G}, g, h = g^z)$;
- $\mathsf{Com}(M)$, for $M \in \mathbb{G}$, chooses a random element $r \overset{\$}{\leftarrow} \mathbb{Z}_p$ and outputs the pair $(C = (u = g^r, e = h^r \cdot M), \delta = r)$;

- $\mathsf{VerCom}(C = (u, e), M, \delta = r)$ checks whether $C = (u = g^r, e = h^r \cdot M)$.

This commitment scheme is hiding under the DDH assumption and perfectly binding. It is even extractable using the decryption key z: $M = e/u^z$. However, it is not labeled. The Cramer-Shoup encryption scheme [16] admits labels and is extractable and non-malleable, thanks to the `IND-CCA` security level.

2.3 Perfectly Hiding Commitments

The Pedersen scheme [32] is the most famous perfectly hiding commitment: $\mathsf{Com}(m) = g^m h^r$ for a random scalar $r \overset{\$}{\leftarrow} \mathbb{Z}_p$ and a fixed basis $h \in \mathbb{G}$. The binding property relies on the DL assumption. Unfortunately, the opening value is the scalar r, which makes it hard to encrypt/decrypt efficiently, as required in our construction below. Haralambiev [24, Section 4.1.4] recently proposed a new commitment scheme, called TC4 (without label), with a group element as opening value:

- $\mathsf{SetupCom}(1^{\mathfrak{K}})$ chooses an asymmetric pairing-friendly setting $(\mathbb{G}_1, g_1, \mathbb{G}_2, g_2, \mathbb{G}_T, p, e)$, with an additional independent generator $T \in \mathbb{G}_2$. It sets the CRS $\rho = (\mathbb{G}_1, g_1, \mathbb{G}_2, g_2, T, \mathbb{G}_T, p, e)$;
- $\mathsf{Com}(x)$, for $x \in \mathbb{Z}_p$, chooses a random element $r \overset{\$}{\leftarrow} \mathbb{Z}_p$ and outputs the pair $(C = g_2^r T^x, \delta = g_1^r)$;
- $\mathsf{VerCom}(C, x, \delta)$ checks whether $e(g_1, C/T^x) = e(\delta, g_2)$.

This commitment scheme is clearly perfectly hiding, since the groups are cyclic, and for any $C \in \mathbb{G}_2$, $x \in \mathbb{Z}_p$, there exists $\delta \in \mathbb{G}_1$ that satisfies $e(g_1, C/T^x) = e(\delta, g_2)$. More precisely, if $C = g_2^u$ and $T = g_2^t$, then $\delta = g_1^{u-tx}$ opens C to any x. The binding property holds under the DDH assumption in $\mathbb{G}_2$, as proven in [24, Section 4.1.4].

2.4 Equivocable Commitments

An equivocable commitment scheme $\mathcal{C}$ extends on the previous definition, with $\mathsf{SetupCom}$, Com, VerCom, and a second setup $\mathsf{SetupComT}(1^{\mathfrak{K}})$ that additionally outputs a trapdoor τ, and

- $\mathsf{SimCom}^{\ell}(\tau)$ that takes as input the trapdoor τ and a label ℓ and outputs a pair (C, eqk), where C is a commitment and eqk an equivocation key;
- $\mathsf{OpenCom}^{\ell}(\mathsf{eqk}, C, x)$ that takes as input a commitment C, a label ℓ, a message x, and an equivocation key eqk for this commitment, and outputs an opening data δ for C and ℓ on x.

Let us denote SCom the algorithm that takes as input the trapdoor τ, a label ℓ and a message x and which outputs $(C, \delta) \overset{\$}{\leftarrow} \mathsf{SCom}^{\ell}(\tau, x)$, computed as $(C, \mathsf{eqk}) \overset{\$}{\leftarrow} \mathsf{SimCom}^{\ell}(\tau)$ and $\delta \leftarrow \mathsf{OpenCom}^{\ell}(\mathsf{eqk}, C, x)$. Three additional properties are then associated: a *correctness* property, and two *indistinguishability* properties, which all together imply the *hiding* property.

```
$\mathsf{Exp}_{\mathcal{A}}^{\texttt{sim-ind-}b}(\mathfrak{K})$
    $(\rho, \tau) \overset{\$}{\leftarrow} \mathsf{SetupComT}(1^{\mathfrak{K}})$
    $(\ell, x, \mathsf{state}) \overset{\$}{\leftarrow} \mathcal{A}^{\mathsf{SCom}^{\cdot}(\tau,\cdot)}(\rho)$
    $\textbf{if } b = 0 \textbf{ then } (C, \delta) \overset{\$}{\leftarrow} \mathsf{Com}^{\ell}(x)$
    $\textbf{else } (C, \delta) \overset{\$}{\leftarrow} \mathsf{SCom}^{\ell}(\tau, x)$
    $\textbf{return } \mathcal{A}^{\mathsf{SCom}^{\cdot}(\tau,\cdot)}(\mathsf{state}, C, \delta)$
```

```
$\mathsf{Exp}_{\mathcal{A}}^{\texttt{bind-ext}}(\mathfrak{K})$
    $(\rho, \tau) \overset{\$}{\leftarrow} \mathsf{SetupComT}(1^{\mathfrak{K}})$
    $(C, \ell, x, \delta) \overset{\$}{\leftarrow} \mathcal{A}^{\mathsf{ExtCom}^{\cdot}(\tau,\cdot)}(\rho)$
    $x' \leftarrow \mathsf{ExtCom}^{\ell}(\tau, C)$
    $\textbf{if } x' = x \textbf{ then return } 0$
    $\textbf{else return } \mathsf{VerCom}^{\ell}(C, x, \delta)$
```

Fig. 2. Simulation Indistinguishability and Binding Extractability

- *Trapdoor Correctness*: all simulated commitments can be opened on any message: for all ℓ, x, if $(C, \mathsf{eqk}) \overset{\$}{\leftarrow} \mathsf{SimCom}^{\ell}(\tau)$ and $\delta \leftarrow \mathsf{OpenCom}^{\ell}(\mathsf{eqk}, C, x)$, then $\mathsf{VerCom}^{\ell}(C, x, \delta) = 1$;
- *Setup Indistinguishability*: one cannot distinguish the CRS ρ generated by SetupCom from the one generated by SetupComT. $\mathcal{C}$ is said (t, ε)-setup-indistinguishable if the two distributions for ρ are (t, ε)-computationally indistinguishable. We denote $\mathsf{Adv}_{\mathcal{C}}^{\texttt{setup-ind}}(t)$ the distance between the two distributions.
- *Simulation Indistinguishability*: one cannot distinguish a real commitment (generated by Com) from a fake commitment (generated by SCom), even with oracle access to fake commitments. $\mathcal{C}$ is said (t, ε)-simulation-indistinguishable if $\mathsf{Adv}_{\mathcal{C}}^{\texttt{sim-ind}}(t) \leq \varepsilon$ (see the experiments $\mathsf{Exp}_{\mathcal{A}}^{\texttt{sim-ind-}b}(\mathfrak{K})$ in Figure 2).

More precisely, when the trapdoor correctness is satisfied, since commitments generated by SimCom are perfectly hiding (they can be opened in any way using OpenCom), $\mathsf{Adv}_{\mathcal{C}}^{\texttt{hid}}(t) \leq \mathsf{Adv}_{\mathcal{C}}^{\texttt{setup-ind}}(t) + \mathsf{Adv}_{\mathcal{C}}^{\texttt{sim-ind}}(t)$.

Definition 1 (Equivocable Commitment). *A commitment scheme $\mathcal{C}$ is said (t, ε)-equivocable if, first, the basic commitment scheme satisfies the correctness property and is both (t, ε)-binding and (t, ε)-hiding, and, secondly, the additional algorithms guarantee the trapdoor correctness and make it both (t, ε)-setup-indistinguishable and (t, ε)-simulation-indistinguishable.*

2.5 Extractable Commitments

An extractable commitment scheme $\mathcal{C}$ also extends on the initial definition, with SetupCom, Com, VerCom, as well as the second setup $\mathsf{SetupComT}(1^{\mathfrak{K}})$ that additionally outputs a trapdoor τ, and

- $\mathsf{ExtCom}^{\ell}(\tau, C)$ which takes as input the trapdoor τ, a commitment C, and a label ℓ, and outputs the committed message x, or $\perp$ if the commitment is invalid.

As above, three additional properties are then associated: a *correctness* property, and the *setup indistinguishability*, but also an *extractability* property, which implies, together with the setup indistinguishability, the *binding* property:

- *Trapdoor Correctness*: all commitments honestly generated can be correctly extracted: for all ℓ, x, if $(C, \delta) \overset{\$}{\leftarrow} \mathsf{Com}^{\ell}(x)$ then $\mathsf{ExtCom}^{\ell}(C, \tau) = x$;

- *Setup Indistinguishability*: as above;
- *Binding Extractability*: one cannot fool the extractor, *i.e.*, produce a commitment and a valid opening data to an input x while the commitment does not extract to x. $\mathcal{C}$ is said (t, ε)-binding-extractable if $\mathsf{Succ}_{\mathcal{C}}^{\mathtt{bind\text{-}ext}}(t) \leq \varepsilon$ (see the experiment $\mathsf{Exp}_{\mathcal{A}}^{\mathtt{bind\text{-}ext}}(\mathfrak{K})$ in Figure 2).

More precisely, when one breaks the binding property with $(C, \ell, x_0, \delta_0, x_1, \delta_1)$, if the extraction oracle outputs $x' = x_0$, then one can output (C, ℓ, x_1, δ_1), otherwise one can output (C, ℓ, x_0, δ_0). In both cases, this breaks the binding-extractability: $\mathsf{Adv}_{\mathcal{C}}^{\mathtt{bind}}(t) \leq \mathsf{Adv}_{\mathcal{C}}^{\mathtt{setup\text{-}ind}}(t) + \mathsf{Succ}_{\mathcal{C}}^{\mathtt{bind\text{-}ext}}(t)$.

Definition 2 (Extractable Commitment). *A commitment scheme $\mathcal{C}$ is said (t, ε)-extractable if, first, the basic commitment scheme satisfies the correctness property and is both (t, ε)-binding and (t, ε)-hiding, and, secondly, the additional algorithms guarantee the trapdoor correctness and make it both (t, ε)-setup-indistinguishable and (t, ε)-binding-extractable.*

3 Equivocable and Extractable Commitments

3.1 $\mathtt{E}^2$-Commitments: Equivocable and Extractable

Public-key encryption schemes are perfectly binding commitments that are additionally extractable. The Pedersen and Haralambiev commitments are perfectly hiding commitments that are additionally equivocable. But none of them have the two properties at the same time. This is now our goal.

Definition 3 ($\mathtt{E}^2$-Commitment). *A commitment scheme $\mathcal{C}$ is said (t, ε)-$\mathtt{E}^2$ (equivocable and extractable) if the indistinguishable setup algorithm outputs a common trapdoor that allows both equivocability and extractability. If one denotes $\mathsf{Adv}_{\mathcal{C}}^{\mathtt{e}^2}(t)$ the maximum of $\mathsf{Adv}_{\mathcal{C}}^{\mathtt{setup\text{-}ind}}(t)$, $\mathsf{Adv}_{\mathcal{C}}^{\mathtt{sim\text{-}ind}}(t)$, and $\mathsf{Succ}_{\mathcal{C}}^{\mathtt{bind\text{-}ext}}(t)$, then it should be upper-bounded by ε.*

But with such a common trapdoor, the adversary could exploit the equivocation queries to break extractability and extraction queries to break equivocability. Stronger notions can thus be defined, using the experiments $\mathsf{Exp}_{\mathcal{A}}^{\mathtt{s\text{-}sim\text{-}ind\text{-}}b}(\mathfrak{K})$ and $\mathsf{Exp}_{\mathcal{A}}^{\mathtt{s\text{-}bind\text{-}ext}}(\mathfrak{K})$ in Figure 3, in which SCom is supposed to store each query/answer (ℓ, x, C) in a list Λ and ExtCom-queries on such an SCom-output (ℓ, C) are answered by x (as it would be when using Com instead of SCom).

- *Strong Simulation Indistinguishability*: one cannot distinguish a real commitment (generated by Com) from a fake commitment (generated by SCom), even with oracle access to the extraction oracle (ExtCom) and to fake commitments (using SCom). $\mathcal{C}$ is said (t, ε)-strongly-simulation-indistinguishable if $\mathsf{Adv}_{\mathcal{C}}^{\mathtt{s\text{-}sim\text{-}ind}}(t) \leq \varepsilon$;
- *Strong Binding Extractability* (informally introduced in [13] as "simulation extractability"): one cannot fool the extractor, *i.e.*, produce a commitment and a valid opening data (not given by SCom) to an input x while the

```
$\mathsf{Exp}_{\mathcal{A}}^{\texttt{s-sim-ind-}b}(\mathfrak{K})$
    $(\rho, \tau) \overset{\$}{\leftarrow} \mathsf{SetupComT}(1^{\mathfrak{K}})$;
    $(\ell, x, \mathsf{state}) \overset{\$}{\leftarrow} \mathcal{A}^{\mathsf{SCom}^{\cdot}(\tau,\cdot),\mathsf{ExtCom}^{\cdot}(\tau,\cdot)}(\rho)$
    **if** $b = 0$ **then** $(C, \delta) \overset{\$}{\leftarrow} \mathsf{Com}^{\ell}(x)$
    **else** $(C, \delta) \overset{\$}{\leftarrow} \mathsf{SCom}^{\ell}(\tau, x)$
    **return** $\mathcal{A}^{\mathsf{SCom}^{\cdot}(\tau,\cdot),\mathsf{ExtCom}^{\cdot}(\tau,\cdot)}(\mathsf{state}, C, \delta)$
```

```
$\mathsf{Exp}_{\mathcal{A}}^{\texttt{s-bind-ext}}(\mathfrak{K})$
    $(\rho, \tau) \overset{\$}{\leftarrow} \mathsf{SetupComT}(1^{\mathfrak{K}})$
    $(C, \ell, x, \delta) \overset{\$}{\leftarrow} \mathcal{A}^{\mathsf{SCom}^{\cdot}(\tau,\cdot),\mathsf{ExtCom}^{\cdot}(\tau,\cdot)}(\rho)$
    $x' \leftarrow \mathsf{ExtCom}^{\ell}(\tau, C)$
    **if** $(\ell, x', C) \in \Lambda$ **then return** $0$
    **if** $x' = x$ **then return** $0$
    **else return** $\mathsf{VerCom}^{\ell}(C, x, \delta)$
```

Fig. 3. Strong Simulation Indistinguishability and Strong Binding Extractability

commitment does not extract to x, even with oracle access to the extraction oracle (ExtCom) and to fake commitments (using SCom). $\mathcal{C}$ is said (t, ε)-strongly-binding-extractable if $\mathsf{Succ}_{\mathcal{C}}^{\texttt{s-bind-ext}}(t) \leq \varepsilon$.

They both imply the respective weaker notions since they just differ by giving access to the ExtCom-oracle in the former game, and to the SCom oracle in the latter. We insist that ExtCom-queries on SCom-outputs are answered by the related SCom-inputs. Otherwise, the former game would be void. In addition, VerCom always rejects inputs with $x = \perp$, which is useful in the latter game.

3.2 UC-Secure Commitments

The security definition for commitment schemes in the UC framework was presented by Canetti and Fischlin [10], refined by Canetti [9]. The ideal functionality is presented in Figure 4, where a *public delayed output* is an output first sent to the adversary $\mathcal{S}$ that eventually decides if and when the message is actually delivered to the recipient. In case of corruption of the committer, if this is before the `Receipt`-message for the receiver, the adversary chooses the committed value, otherwise it is provided by the ideal functionality, according to the `Commit`-message. Note this is actually the multiple-commitment functionality that allows multiple executions of the commitment protocol (multiple ssid's) for the same functionality instance (one sid). This avoids the use of joint-state UC [14].

Theorem 4. *A labeled $\texttt{E}^2$-commitment scheme $\mathcal{C}$, that is in addition strongly-simulation-indistinguishable or strongly-binding-extractable, is a non-interactive UC-secure commitment scheme in the presence of adaptive adversaries, assuming reliable erasures and authenticated channels.*

4 A Construction of Labeled $\texttt{E}^2$-Commitment Scheme

4.1 Labeled Cramer-Shoup Encryption on Vectors

For our construction we use a variant of the Cramer-Shoup encryption scheme for vectors of messages. Let $\mathbb{G}$ be a cyclic group of order p, with two independent generators g and h. The secret decryption key is a random vector $\mathsf{sk} =$

The functionality $\mathcal{F}_{\text{com}}$ is parametrized by a security parameter k. It interacts with an adversary $\mathcal{S}$ and a set of parties $P_1,\ldots,P_n$ via the following queries:

Commit phase: Upon receiving a query $(\texttt{Commit}, \textbf{sid}, \textbf{ssid}, \boldsymbol{P_i}, \boldsymbol{P_j}, \boldsymbol{x})$ from party $\boldsymbol{P_i}$: record the tuple $(\text{sid}, \text{ssid}, P_i, P_j, x)$ and generate a *public delayed output* $(\texttt{Receipt}, \text{sid}, \text{ssid}, P_i, P_j)$ to P_j. Ignore further Commit-message with the same ssid from P_i.

Decommit phase. Upon receiving a query $(\texttt{Reveal}, \textbf{sid}, \textbf{ssid}, \boldsymbol{P_i}, \boldsymbol{P_j})$ from party $\boldsymbol{P_i}$: ignore the message if $(\text{sid}, \text{ssid}, P_i, P_j, x)$ is not recorded; otherwise mark the record $(\text{sid}, \text{ssid}, P_i, P_j)$ as revealed and generate a *public delayed output* $(\texttt{Revealed}, \text{sid}, \text{ssid}, P_i, P_j, x)$ to P_j. Ignore further Reveal-message with the same ssid from P_i.

Fig. 4. Ideal Functionality for Commitment Scheme $\boldsymbol{\mathcal{F}}_{\text{com}}$

$(x_1, x_2, y_1, y_2, z) \xleftarrow{\$} \mathbb{Z}_p^5$ and the public encryption key is $\mathsf{pk} = (g, h, c = g^{x_1} h^{x_2}, d = g^{y_1} h^{y_2}, f = g^z, H)$, where H is randomly chosen in a collision-resistant hash function family $\mathcal{H}$ (actually, second-preimage resistance is enough). For a message-vector $\boldsymbol{M} = (M_i)_{i=1,\ldots,m} \in \mathbb{G}^m$, the multi-Cramer-Shoup encryption is defined as $m\text{-}\mathsf{MCS}^{\ell}_{\mathsf{pk}}(\boldsymbol{M}; (r_i)_i) = (\mathsf{CS}^{\ell}_{\mathsf{pk}}(M_i, \theta; r_i) = (u_i = g^{r_i}, v_i = h^{r_i}, e_i = f^{r_i} \cdot M_i, w_i = (cd^{\theta})^{r_i}))_i$, where $\theta = H(\ell, (u_i, v_i, e_i)_i)$ is the same for all the w_i's to ensure non-malleability contrary to what we would have if we had just concatenated Cramer-Shoup ciphertexts of the M_i's. Such a ciphertext $C = (u_i, v_i, e_i, w_i)_i$ is decrypted by $M_i = e_i / u_i^z$, after having checked the validity of the ciphertext, $w_i \stackrel{?}{=} u_i^{x_1 + \theta y_1} v_i^{x_2 + \theta y_2}$, for $i = 1, \ldots, m$. This multi-Cramer-Shoup encryption scheme, denoted MCS, is `IND-CCA` under the DDH assumption. It even verifies a stronger property `VIND-PO-CCA` (for Vector-Indistinguishability with Partial Opening under Chosen-Ciphertext Attacks), useful for the security proof of our commitment $\mathcal{E}^2\mathcal{C}$.

4.2 Construction

In this section, we provide a concrete construction $\mathcal{E}^2\mathcal{C}$, inspired from [10,13,2], with the above multi-Cramer-Shoup encryption (as an extractable commitment scheme) and the TC4 Haralambiev's equivocable commitment scheme [24, Section 4.1.4]. The latter will allow equivocability while the former will provide extractability:

- $\mathsf{SetupComT}(1^{\mathfrak{K}})$ generates a pairing-friendly setting $(\mathbb{G}_1, g_1, \mathbb{G}_2, g_2, \mathbb{G}_T, p, e)$, with another independent generator h_1 of $\mathbb{G}_1$. It then generates the parameters of a Cramer-Shoup-based commitment in $\mathbb{G}_1$: $x_1, x_2, y_1, y_2, z \xleftarrow{\$} \mathbb{Z}_p$ and $H \xleftarrow{\$} \mathcal{H}$, and sets $\mathsf{pk} = (g_1, h_1, c = g_1^{x_1} h_1^{x_2}, d = g_1^{y_1} h_1^{y_2}, f_1 = g_1^z, H)$. It then chooses a random scalar $t \xleftarrow{\$} \mathbb{Z}_p$, and sets $T = g_2^t$. The CRS ρ is set as (pk, T) and the trapdoor τ is the decryption key (x_1, x_2, y_1, y_2, z) (a.k.a. extraction trapdoor) together with t (a.k.a. equivocation trapdoor). For $\mathsf{SetupCom}(1^{\mathfrak{K}})$, the CRS is generated the same way, but forgetting the scalars, and thus without any trapdoor;

- $\mathsf{Com}^{\ell}(\boldsymbol{M})$, for $\boldsymbol{M} = (M_i)_i \in \{0,1\}^m$ and a label ℓ, works as follows:
  - For $i = 1, \ldots, m$, it chooses a random scalar $r_{i,M_i} \stackrel{\$}{\leftarrow} \mathbb{Z}_p$, sets $r_{i,1-M_i} = 0$, and commits to M_i, using the TC4 commitment scheme with r_{i,M_i} as randomness: $a_i = g_2^{r_{i,M_i}} T^{M_i}$, and sets $d_{i,j} = g_1^{r_{i,j}}$ for $j = 0, 1$, which makes d_{i,M_i} the opening value for a_i to M_i; Let us also write $\boldsymbol{a} = (a_1, \ldots, a_m)$, the tuple of commitments.
  - For $i = 1, \ldots, m$ and $j = 0, 1$, it gets $\boldsymbol{b} = (b_{i,j})_{i,j} = 2m\text{-}\mathsf{MCS}_{\mathsf{pk}}^{\ell'}(\boldsymbol{d}; \boldsymbol{s})$, that is $(u_{i,j}, v_{i,j}, e_{i,j}, w_{i,j})_{i,j}$, where $\boldsymbol{d} = (d_{i,j})_{i,j}$ computed above, $\boldsymbol{s} = (s_{i,j})_{i,j} \stackrel{\$}{\leftarrow} \mathbb{Z}_p^{2m}$, and $\ell' = (\ell, \boldsymbol{a})$.

  The commitment is $C = (\boldsymbol{a}, \boldsymbol{b})$, and the opening information is the m-tuple $\delta = (s_{1,M_1}, \ldots, s_{m,M_m})$.
- $\mathsf{VerCom}^{\ell}(C, \boldsymbol{M}, \delta)$ checks the validity of the ciphertexts b_{i,M_i} with s_{i,M_i} and θ computed on the full ciphertext C, extracts d_{i,M_i} from b_{i,M_i} and s_{i,M_i}, and checks whether $e(g_1, a_i/T^{M_i}) = e(d_{i,M_i}, g_2)$, for $i = 1, \ldots, m$.
- $\mathsf{SimCom}^{\ell}(\tau)$ takes as input the equivocation trapdoor, namely t, and outputs $C = (\boldsymbol{a}, \boldsymbol{b})$ and $\mathsf{eqk} = \boldsymbol{s}$, where
  - For $i = 1, \ldots, m$, it chooses a random scalar $r_{i,0} \stackrel{\$}{\leftarrow} \mathbb{Z}_p$, sets $r_{i,1} = r_{i,0} - t$, and commits to both 0 and 1, using the TC4 commitment scheme with $r_{i,0}$ and $r_{i,1}$ as respective randomness: $a_i = g_2^{r_{i,0}} = g_2^{r_{i,1}} T$, and $d_{i,j} = g_1^{r_{i,j}}$ for $j = 0, 1$, which makes $d_{i,j}$ the opening value for a_i to the value $j \in \{0,1\}$. This leads to $\boldsymbol{a}$;
  - $\boldsymbol{b}$ is built as above: $\boldsymbol{b} = (b_{i,j})_{i,j} = 2m\text{-}\mathsf{MCS}_{\mathsf{pk}}^{\ell'}(\boldsymbol{d}; \boldsymbol{s})$, with random scalars $(s_{i,j})_{i,j}$.
- $\mathsf{OpenCom}^{\ell}(\mathsf{eqk}, C, \boldsymbol{M})$ simply extracts the useful values from $\mathsf{eqk} = \boldsymbol{s}$ to make the opening value $\delta = (s_{1,M_1}, \ldots, s_{m,M_m})$ in order to open to $\boldsymbol{M} = (M_i)_i$.
- $\mathsf{ExtCom}^{\ell}(\tau, C)$ takes as input the extraction trapdoor, namely the Cramer-Shoup decryption key. Given $\boldsymbol{b}$, it can decrypt all the $b_{i,j}$ into $d_{i,j}$ and check whether $e(g_1, a_i/T^j) = e(d_{i,j}, g_2)$ or not. If, for each i, exactly one $j = M_i$ satisfies the equality, then the extraction algorithm outputs $(M_i)_i$, otherwise (no correct decryption or ambiguity with several possibilities) it outputs $\perp$.

4.3 Security Properties

The above commitment scheme $\mathcal{E}^2\mathcal{C}$ is a labeled $\mathtt{E}^2$-commitment, with both strong-simulation-indistinguishability and strong-binding-extractability, under the DDH assumptions in both $\mathbb{G}_1$ and $\mathbb{G}_2$. It is thus a UC-secure commitment scheme. The stronger `VIND-PO-CCA` security notion for the encryption scheme is required because the $\mathsf{SCom}/\mathsf{Com}$ oracle does not only output the commitment (and thus the ciphertexts) but also the opening values which include the random coins of the encryption, but just for the plaintext components that are the same in the two vectors, since the two vectors only differ for unnecessary data (namely the $d_{i,1-M_i}$'s) in the security proof. More details can be found in the full version [1].

5 SPHF-Friendly Commitments

5.1 Smooth Projective Hash Functions

Projective hash function families were first introduced by Cramer and Shoup [17], but we here use the definitions of Gennaro and Lindell [20], provided to build secure password-based authenticated key exchange protocols, together with non-malleable commitments.

Let X be the domain of these functions and let L be a certain subset of this domain (a language). A key property of these functions is that, for words C in L, their values can be computed by using either a *secret* hashing key hk or a *public* projection key hp but with a witness w of the fact that C is indeed in L:

- $\mathsf{HashKG}(L)$ generates a hashing key hk for the language L;
- $\mathsf{ProjKG}(\mathsf{hk}, L, C)$ derives the projection key hp, possibly depending on the word C;
- $\mathsf{Hash}(\mathsf{hk}, L, C)$ outputs the hash value from the hashing key, on any word $C \in X$;
- $\mathsf{ProjHash}(\mathsf{hp}, L, C, w)$ outputs the hash value from the projection key hp, and the witness w, for $C \in L$.

The *correctness* of the SPHF assures that if $C \in L$ with w a witness of this fact, then $\mathsf{Hash}(\mathsf{hk}, L, C) = \mathsf{ProjHash}(\mathsf{hp}, L, C, w)$. On the other hand, the security is defined through the *smoothness*, which guarantees that, if $C \notin L$, $\mathsf{Hash}(\mathsf{hk}, L, C)$ is *statistically* indistinguishable from a random element, even knowing hp.

Note that HashKG and ProjKG can just depend partially on L (a superset L') and not at all on C: we then note $\mathsf{HashKG}(L')$ and $\mathsf{ProjKG}(\mathsf{hk}, L', \perp)$ (see [6] for more details on GL-SPHF and KV-SPHF and language definitions).

5.2 Robust Commitments

For a long time, SPHFs have been used to implicitly check some statements, on language membership, such as "C indeed encrypts x". This easily extends to perfectly binding commitments with labels: $L_x = \{(\ell, C) | \exists\delta, \mathsf{VerCom}^{\ell}(C, x, \delta) = 1\}$. But when commitments are equivocable, this intuitively means that a commitment C with the label ℓ contains any x and is thus in all the languages L_x. In order to be able to use SPHFs with E^2-commitments, we want the commitments generated by polynomially-bounded adversaries to be perfectly binding, and thus to belong to at most one language L_x. We thus need a *robust verification* property for such *E^2-commitments.*

Definition 5 (Robustness). *One cannot produce a commitment and a label that extracts to x' (possibly $x' = \perp$) such that there exists a valid opening data to a different input x, even with oracle access to the extraction oracle (*ExtCom*) and to fake commitments (using* SCom*). $\mathcal{C}$ is said (t, ε)-robust if $\mathsf{Succ}_{\mathcal{C}}^{robust}(t) \leq \varepsilon$, according to the experiment $\mathsf{Exp}_{\mathcal{A}}^{robust}(\mathfrak{K})$ in Figure 5.*

It is important to note that the latter experiment $\mathsf{Exp}_{\mathcal{A}}^{\mathtt{robust}}(\mathfrak{K})$ may not be run in polynomial time. Robustness implies strong-binding-extractability.

```
$\mathsf{Exp}^{\texttt{robust}}_{\mathcal{A}}(\mathfrak{K})$
    $(\rho, \tau) \overset{\$}{\leftarrow} \mathsf{SetupComT}(1^{\mathfrak{K}})$
    $(C, \ell) \overset{\$}{\leftarrow} \mathcal{A}^{\mathsf{SCom}^{\cdot}(\tau,\cdot), \mathsf{ExtCom}^{\cdot}(\tau,\cdot)}(\rho)$
    $x' \leftarrow \mathsf{ExtCom}^{\ell}(\tau, C)$
    if $(\ell, x', C) \in \Lambda$ then return $0$
    if $\exists x \neq x', \exists \delta, \mathsf{VerCom}^{\ell}(C, x, \delta)$ then return $1$
    else return $0$
```

Fig. 5. Robustness

5.3 Properties of SPHF-Friendly Commitments

We are now ready to define SPHF-friendly commitments, which admit an SPHF on the languages $L_x = \{(\ell, C) | \, \exists \delta, \, \mathsf{VerCom}^{\ell}(C, x, \delta) = 1\}$, and to discuss about them:

Definition 6 (SPHF-Friendly Commitments). *An SPHF-friendly commitment is an* E^2*-commitment that admits an SPHF on the languages* L_x*, and that is both strongly-simulation-indistinguishable and robust.*

Let us consider such a family $\mathcal{F}$ of SPHFs on languages L_x for $x \in X$, with X a non trivial set (with at least two elements), with hash values in the set G. From the smoothness of the SPHF on L_x, one can derive the two following properties on SPHF-friendly commitments, modeled by the experiments in Figure 6. The first notion of *smoothness* deals with adversary-generated commitments, that are likely perfectly binding from the robustness, while the second notion of *pseudo-randomness* deals with simulated commitments, that are perfectly hiding. They are inspired by the security games from [20].

In both security games, note that when hk and hp do not depend on x nor on C, and when the smoothness holds even if the adversary can choose C after having seen hp (*i.e.*, the SPHF is actually a KV-SPHF [6]), they can be generated from the beginning of the games, with hp given to the adversary much earlier.

Smoothness of SPHF-Friendly Commitments. If the adversary $\mathcal{A}$, with access to the oracles SCom and ExtCom, outputs a fresh commitment (ℓ, C) that extracts to $x' \leftarrow \mathsf{ExtCom}^{\ell}(\tau, C)$, then the robustness guarantees that for any $x \neq x'$, $(\ell, C) \notin L_x$ (excepted with small probability), and thus the distribution of the hash value is statistically indistinguishable from the random distribution, even when knowing hp. In the experiment $\mathsf{Exp}^{\texttt{c-smooth}}_{\mathcal{A}}(\mathfrak{K})$, we let the adversary choose x, and we have: $\mathsf{Adv}^{\texttt{c-smooth}}_{\mathcal{C},\mathcal{F}}(t) \leq \mathsf{Succ}^{\texttt{robust}}_{\mathcal{C}}(t) + \mathsf{Adv}^{\texttt{smooth}}_{\mathcal{F}}$.

Pseudo-Randomness of SPHF on Robust Commitments. If the adversary $\mathcal{A}$ is given a commitment C by SCom on x' with label ℓ, both adversary-chosen, even with access to the oracles SCom and ExtCom, then for any x, it cannot distinguish the hash value of (ℓ, C) on language L_x from a random value, even being given hp, since C could have been generated as $\mathsf{Com}^{\ell}(x'')$ for some $x'' \neq x$,

```
$\mathsf{Exp}_{\mathcal{A}}^{\texttt{c-smooth-}b}(\mathfrak{K})$
    $(\rho,\tau) \overset{\$}{\leftarrow} \mathsf{SetupComT}(1^{\mathfrak{K}})$
    $(C,\ell,x,\mathsf{state}) \overset{\$}{\leftarrow} \mathcal{A}^{\mathsf{SCom}^{\cdot}(\tau,\cdot),\mathsf{ExtCom}^{\cdot}(\tau,\cdot)}(\rho);\ x' \leftarrow \mathsf{ExtCom}^{\ell}(\tau,C)$
    $\textbf{if } (\ell,x',C) \in \Lambda \textbf{ then return } 0$
    $\mathsf{hk} \overset{\$}{\leftarrow} \mathsf{HashKG}(L_x);\ \mathsf{hp} \leftarrow \mathsf{ProjKG}(\mathsf{hk},L_x,(\ell,C))$
    $\textbf{if } b=0 \vee x'=x \textbf{ then } H \leftarrow \mathsf{Hash}(\mathsf{hk},L_x,(\ell,C)) \textbf{ else } H \overset{\$}{\leftarrow} G$
    $\textbf{return } \mathcal{A}^{\mathsf{SCom}^{\cdot}(\tau,\cdot),\mathsf{ExtCom}^{\cdot}(\tau,\cdot)}(\mathsf{state},\mathsf{hp},H)$
```

```
$\mathsf{Exp}_{\mathcal{A}}^{\texttt{c-ps-rand-}b}(\mathfrak{K})$
    $(\rho,\tau) \overset{\$}{\leftarrow} \mathsf{SetupComT}(1^{\mathfrak{K}})$
    $(\ell,x,x',\mathsf{state}) \overset{\$}{\leftarrow} \mathcal{A}^{\mathsf{SCom}^{\cdot}(\tau,\cdot),\mathsf{ExtCom}^{\cdot}(\tau,\cdot)}(\rho);\ (C,\delta) \overset{\$}{\leftarrow} \mathsf{SCom}^{\ell}(\tau,x')$
    $\mathsf{hk} \overset{\$}{\leftarrow} \mathsf{HashKG}(L_x); \mathsf{hp} \leftarrow \mathsf{ProjKG}(\mathsf{hk},L_x,(\ell,C))$
    $\textbf{if } b=0 \textbf{ then } H \leftarrow \mathsf{Hash}(\mathsf{hk},L_x,(\ell,C)) \textbf{ else } H \overset{\$}{\leftarrow} G$
    $\textbf{return } \mathcal{A}^{\mathsf{SCom}^{\cdot}(\tau,\cdot),\mathsf{ExtCom}^{\cdot}(\tau,\cdot)}(\mathsf{state},C,\mathsf{hp},H)$
```

Fig. 6. Smoothness and Pseudo-Randomness

which excludes it to belong to L_x, under the robustness. In the experiment $\mathsf{Exp}_{\mathcal{A}}^{\texttt{c-ps-rand}}(\mathfrak{K})$, we let the adversary choose (ℓ,x), and we have: $\mathsf{Adv}_{\mathcal{C},\mathcal{F}}^{\texttt{c-ps-rand}}(t) \le \mathsf{Adv}_{\mathcal{C}}^{\texttt{s-sim-ind}}(t) + \mathsf{Succ}_{\mathcal{C}}^{\texttt{robust}}(t) + \mathsf{Adv}_{\mathcal{F}}^{\texttt{smooth}}$.

5.4 Our Commitment Scheme $\mathcal{E}^2\mathcal{C}$ is SPHF-Friendly

In order to be *SPHF-friendly*, the commitment first needs to be *strongly-simulation-indistinguishable* and *robust*. We have already shown the former property, and the latter is also proven in the full version [1]. One additionally needs an SPHF able to check the verification equation: using the notations from Section 4.2, $C = (\boldsymbol{a},\boldsymbol{b})$ is a commitment of $\boldsymbol{M} = (M_i)_i$, if there exist $\delta = (s_{1,M_1},\ldots,s_{m,M_m})$ and $(d_{1,M_1},\ldots,d_{m,M_m})$ such that $b_{i,M_i} = (u_{i,M_i}, v_{i,M_i}, e_{i,M_i}, w_{i,M_i}) = \mathsf{CS}_{\mathsf{pk}}^{\ell'}(d_{i,M_i},\theta;s_{i,M_i})$ (with a particular θ) and $e(g_1, a_i/T^{M_i}) = e(d_{i,M_i}, g_2)$, for $i = 1,\ldots,m$. Since e is non-degenerated, we can eliminate the need of d_{i,M_i}, by lifting everything in $\mathbb{G}_T$, and checking that, first, the ciphertexts are all valid:

$$e(u_{i,M_i},g_2) = e(g_1^{s_{i,M_i}},g_2) \qquad e(v_{i,M_i},g_2) = e(h_1^{s_{i,M_i}},g_2)$$
$$e(w_{i,M_i},g_2) = e((cd^{\theta})^{s_{i,M_i}},g_2)$$

and, second, the plaintexts satisfy the appropriate relations:

$$e(e_{i,M_i},g_2) = e(f_1^{s_{i,M_i}},g_2)\cdot e(g_1,a_i/T^{M_i}).$$

From these expressions we derive several constructions of such SPHFs in the full version [1], and focus here on the most interesting ones for the following applications:

- First, when C is sent in advance (known when generating hp), as in the OT protocol described in Section 7, for $\mathsf{hk} = (\eta, \alpha, \beta, \mu, \varepsilon) \stackrel{\$}{\leftarrow} \mathbb{Z}_p^5$, and $\mathsf{hp} = (\varepsilon, \mathsf{hp}_1 = g_1^\eta h_1^\alpha f_1^\beta (cd^\theta)^\mu) \in \mathbb{Z}_p \times \mathbb{G}_1$:

$$\begin{aligned} H &= \mathsf{Hash}(\mathsf{hk}, \boldsymbol{M}, C) \\ &\stackrel{\mathsf{def}}{=} \textstyle\prod_i \left(e(u_{i,M_i}^\eta \cdot v_{i,M_i}^\alpha, g_2) \cdot (e(e_{i,M_i}, g_2)/e(g_1, a_i/T^{M_i}))^\beta \cdot e(w_{i,M_i}^\mu, g_2)\right)^{\varepsilon^{i-1}} \\ &= e(\textstyle\prod_i \mathsf{hp}_1^{s_{i,M_i}\varepsilon^{i-1}}, g_2) \stackrel{\mathsf{def}}{=} \mathsf{ProjHash}(\mathsf{hp}, \boldsymbol{M}, C, \delta) = H'. \end{aligned}$$

- Then, when C is not necessarily known for computing hp, as in the one-round PAKE, described in Section 6, for $\mathsf{hk} = (\eta_{i,1}, \eta_{i,2}, \alpha_i, \beta_i, \mu_i)_i \stackrel{\$}{\leftarrow} \mathbb{Z}_p^{5m}$, and $\mathsf{hp} = (\mathsf{hp}_{i,1} = g_1^{\eta_{i,1}} h_1^{\alpha_i} f_1^{\beta_i} c^{\mu_i}, \mathsf{hp}_{i,2} = g_1^{\eta_{i,2}} d^{\mu_i})_i \in \mathbb{G}_1^{2m}$:

$$\begin{aligned} H &= \mathsf{Hash}(\mathsf{hk}, \boldsymbol{M}, C) \\ &\stackrel{\mathsf{def}}{=} \textstyle\prod_i \left(e(u_{i,M_i}^{(\eta_{i,1}+\theta\eta_{i,2})} \cdot v_{i,M_i}^{\alpha_i}, g_2) \cdot (e(e_{i,M_i}, g_2)/e(g_1, a_i/T^{M_i}))^{\beta_i} \cdot e(w_{i,M_i}^{\mu_i}, g_2)\right) \\ &= e(\textstyle\prod_i (\mathsf{hp}_{i,1}\mathsf{hp}_{i,2}^\theta)^{s_{i,M_i}}, g_2) \stackrel{\mathsf{def}}{=} \mathsf{ProjHash}(\mathsf{hp}, \boldsymbol{M}, C, \delta) = H'. \end{aligned}$$

5.5 Complexity and Comparisons

As summarized in Table 1, the communication complexity is linear in $m \cdot \mathfrak{K}$ (where m is the bit-length of the committed value and $\mathfrak{K}$ is the security parameter), which is much better than [2] that was linear in $m \cdot \mathfrak{K}^2$, but asymptotically worse than the two proposals in [19] that are linear in $\mathfrak{K}$, and thus independent of m (as long as $m = \mathcal{O}(\mathfrak{K})$).

Basically, the first scheme in [19] consists of a Cramer-Shoup-like encryption C of the message x, and a perfectly-sound Groth-Sahai [23] NIZK π that C contains x. The actual commitment is C and the opening value on x is $\delta = \pi$. The trapdoor-setup provides the Cramer-Shoup decryption key and changes the Groth-Sahai setup to the perfectly-hiding setting. The indistinguishable setups of the Groth-Sahai mixed commitments ensure the setup-indistinguishability. The extraction algorithm uses the Cramer-Shoup decryption algorithm, while the equivocation uses the simulator of the NIZK. The `IND-CCA` security notion for C and the computational soundness of π make it strongly-binding-extractable, the `IND-CCA` security notion and the zero-knowledge property of the NIZK provide the strong-simulation-indistinguishability. It is thus UC-secure. However, the verification is not robust: because of the perfectly-hiding setting of Groth-Sahai proofs, for any ciphertext C and for any message x, there exists a proof π that makes the verification of C on x. As a consequence, it is not SPHF-friendly. The second construction is in the same vein: they cannot be used in the following applications.

6 Password-Authenticated Key Exchange

6.1 A Generic Construction

The ideal functionality of a Password-Authenticated Key Exchange (PAKE) has been proposed in [11]. In Figure 7, we describe a one-round PAKE that

Table 1. Comparison with existing non-interactive UC-secure commitments with a single global CRS (m = bit-length of the committed value, $\mathfrak{K}$ = security parameter)

	SPHF-Friendly	Commitment C	Decommitment δ	Assumption
[2][a]	yes	$(m + 16m\mathfrak{K}) \times \mathbb{G}$	$2m\mathfrak{K} \times \mathbb{Z}_p$	DDH
[19], 1	no	$5 \times \mathbb{G}$	$16 \times \mathbb{G}$	DLIN
[19], 2	no	$37 \times \mathbb{G}$	$3 \times \mathbb{G}$	DLIN
this paper	yes	$8m \times \mathbb{G}_1 + m \times \mathbb{G}_2$	$m \times \mathbb{Z}_p$	SXDH

[a] slight variant without one-time signature but using labels for the IND-CCA security of the multi-Cramer-Shoup ciphertexts, as in our new scheme, and supposing that an element in the cyclic group $\mathbb{G}$ has size $2\mathfrak{K}$, to withstand generic attacks.

is UC-secure against adaptive adversaries, assuming erasures. It can be built from any SPHF-friendly commitment scheme (that is $\mathtt{E}^2$, strongly-simulation-indistinguishable, and robust as described in Section 5), if the SPHF is actually a KV-SPHF [6] and the algorithms HashKG and ProjKG do not need to know the committed value π (nor the word (ℓ, C) itself). We thus denote L_π the language of the pairs (ℓ, C), where C is a commitment that opens to π under the label ℓ, and L the union of all the L_π (L does not depend on π).

Theorem 7. *The Password-Authenticated Key-Exchange described on Figure 7 is UC-secure in the presence of adaptive adversaries, assuming erasures, as soon as the commitment scheme is SPHF-friendly with a KV-SPHF.*

6.2 Concrete Instantiation

Using our commitment $\mathcal{E}^2\mathcal{C}$ introduced Section 4 together with the second SPHF described Section 5 (which satisfies the above requirements for HashKG and ProjKG), one gets a quite efficient protocol, described in the full version [1]. More precisely, for m-bit passwords, each player has to send $\mathsf{hp} \in \mathbb{G}_1^{2m}$ and

CRS: $\rho \overset{\$}{\leftarrow} \mathsf{SetupCom}(1^{\mathfrak{K}})$.

Protocol execution by P_i with π_i:

1. P_i generates $\mathsf{hk}_i \overset{\$}{\leftarrow} \mathsf{HashKG}(L)$, $\mathsf{hp}_i \leftarrow \mathsf{ProjKG}(\mathsf{hk}_i, L, \perp)$ and erases any random coins used for the generation
2. P_i computes $(C_i, \delta_i) \overset{\$}{\leftarrow} \mathsf{Com}^{\ell_i}(\pi_i)$ with $\ell_i = (\mathrm{sid}, P_i, P_j, \mathsf{hp}_i)$
3. P_i stores δ_i, completely erases random coins used by Com and sends hp_i, C_i to P_j

Key computation: Upon receiving hp_j, C_j from P_j

1. P_i computes $H'_i \leftarrow \mathsf{ProjHash}(\mathsf{hp}_j, L_{\pi_i}, (\ell_i, C_i), \delta_i)$ and $H_j \leftarrow \mathsf{Hash}(\mathsf{hk}_i, L_{\pi_i}, (\ell_j, C_j))$ with $\ell_j = (\mathrm{sid}, P_j, P_i, \mathsf{hp}_j)$
2. P_i computes $\mathsf{sk}_i = H'_i \cdot H_j$.

Fig. 7. UC-Secure PAKE from an SPHF-Friendly Commitment

Table 2. Comparison with existing UC-secure PAKE schemes

	Adaptive	One-round	Communication complexity	Assumption
[2][a]	yes	no	$2 \times (2m + 22m\mathfrak{K}) \times \mathbb{G} + \mathrm{OTS}$[b]	DDH
[30]	no	yes	$\approx 2 \times 70 \times \mathbb{G}$	DLIN
[6]	no	yes	$2 \times 6 \times \mathbb{G}_1 + 2 \times 5 \times \mathbb{G}_2$	SXDH
this paper	yes	yes	$2 \times 10m \times \mathbb{G}_1 + 2 \times m \times \mathbb{G}_2$	SXDH

[a] with the commitment variant of note "a" of Table 1.
[b] OTS: one-time signature (public key size and signature size) to link the flows in the PAKE protocol.

$C \in \mathbb{G}_1^{8m} \times \mathbb{G}_2^m$, which means $10m$ elements from $\mathbb{G}_1$ and m elements from $\mathbb{G}_2$. In Table 2, we compare our new scheme with some previous UC-secure PAKE.

7 Oblivious Transfer

7.1 A Generic Construction

The ideal functionality of an Oblivious Transfer (OT) protocol is depicted in the full version [1]. It is inspired from [15]. In Figure 8, we describe a 3-round OT that is UC-secure against adaptive adversaries, and a 2-round variant which is UC-secure against static adversaries. They can be built from any SPHF-friendly commitment scheme, where L_t is the language of the commitments that open to t under the associated label ℓ, and from any IND-CPA encryption scheme $\mathcal{E} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ with plaintext size at least $\mathfrak{K}$, and from any Pseudo-Random Generator (PRG) F with input size equal to plaintext size, and output size equal to the size of the messages in the database. Details on encryption schemes and PRGs can be found in the full version [1]. Notice the adaptive version can be seen as a variant of the static version where the last flow is sent over a somewhat secure channel, as in [15]; and the preflow and pk and c are used to create this somewhat secure channel.

Theorem 8. *The two Oblivous Transfer schemes described in Figure 8 are UC-secure in the presence of adaptive adversaries and static adversaries respectively, assuming reliable erasures and authenticated channels, as soon as the commitment scheme is SPHF-friendly.*

7.2 Concrete Instantiation and Comparison

Using our commitment $\mathcal{E}^2\mathcal{C}$ introduced Section 4 together with the first SPHF described Section 5, one gets the protocol described in the full version [1], where the number of bits of the commited value is $m = \lceil \log k \rceil$. For the statically secure version, the communication cost is, in addition to the database $\boldsymbol{m}$ that is sent in $\boldsymbol{M}$ in a masked way, 1 element of $\mathbb{Z}_p$ and k elements of $\mathbb{G}_1$ (for **hp**, by using the same scalar ε for all hp_t's) for the sender, while the receiver sends $\lceil \log k \rceil$ elements of $\mathbb{G}_2$ (for $\boldsymbol{a}$) and $\lceil 8 \log k \rceil$ elements of $\mathbb{G}_1$ (for $\boldsymbol{b}$), in only two rounds. In

```
CRS: $\rho \xleftarrow{\$} \mathsf{SetupCom}(1^{\mathfrak{K}}), \mathtt{param} \xleftarrow{\$} \mathsf{Setup}(1^{\mathfrak{K}})$.
Pre-flow (for adaptive security only):
  1. $P_i$ generates a key pair $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{KeyGen}(\mathtt{param})$ for $\mathcal{E}$
  2. $P_i$ stores $\mathsf{sk}$, completely erase random coins used by $\mathsf{KeyGen}$, and sends $\mathsf{pk}$ to $P_i$
Index query on $s$:
  1. $P_j$ chooses a random value $S$, computes $R \leftarrow F(S)$ and encrypts $S$ under $\mathsf{pk}$:
     $c \xleftarrow{\$} \mathsf{Encrypt}(\mathsf{pk}, S)$ (for adaptive security only; for static security: $c = \perp, R = 0$)
  2. $P_j$ computes $(C, \delta) \xleftarrow{\$} \mathsf{Com}^{\ell}(s)$ with $\ell = (\mathrm{sid}, \mathrm{ssid}, P_i, P_j)$
  3. $P_j$ stores $\delta$ and completely erase $R$, $S$ and random coins used by $\mathsf{Com}$ and $\mathsf{Encrypt}$
     and sends $C$ and $c$ to $P_i$
Database input $(m_1, \ldots, m_k)$:
  1. $P_i$ decrypts $S \leftarrow \mathsf{Decrypt}(\mathsf{sk}, c)$ and gets $R \leftarrow F(S)$ (for static security: $R = 0$)
  2. $P_i$ computes $\mathsf{hk}_t \xleftarrow{\$} \mathsf{HashKG}(L_t)$, $\mathsf{hp}_t \leftarrow \mathsf{ProjKG}(\mathsf{hk}_t, L_t, (\ell, C))$,
     $K_t \leftarrow \mathsf{Hash}(\mathsf{hk}_t, L_t, (\ell, C))$, and $M_t \leftarrow R \oplus K_t \oplus m_t$, for $t = 1, \ldots, k$
  3. $P_i$ erases everything except $(\mathsf{hp}_t, M_t)_{t=1,\ldots,k}$ and sends them over a secure channel
Data recovery:
Upon receiving $(\mathsf{hp}_t, M_t)_{t=1,\ldots,k}$, $P_j$ computes $K_s \leftarrow \mathsf{ProjHash}(\mathsf{hp}_s, L_s, (\ell, C), \delta)$
and gets $m_s \leftarrow R \oplus K_s \oplus M_s$.
```

Fig. 8. UC-Secure 1-out-of-k OT from an SPHF-Friendly Commitment (for Adaptive and Static Security)

the particular case of $k = 2$, the scalar can be avoided since the message consists of 1 bit, so our construction just requires: 2 elements from $\mathbb{G}_1$ for the sender, and 1 from $\mathbb{G}_2$ and 8 from $\mathbb{G}_1$ for the receiver, in two rounds. For the same security level (static corruptions in the CRS, with erasures), the best known solution from [15] required to send at least 23 group elements and 7 scalars, in 4 rounds. If adaptive security is required, our construction requires 3 additional elements in $\mathbb{G}_1$ and 1 additional round, which gives a total of 13 elements in $\mathbb{G}_1$, in 3 rounds. This is also more efficient then the best known solution from [15], which requires 26 group elements and 7 scalars, in 4 rounds.

Acknowledgments

We thank Ralf Küsters for his comments on a preliminary version. This work was supported in part by the French ANR-12-INSE-0014 SIMPATIC Project and in part by the European Commission through the FP7-ICT-2011-EU-Brazil Program under Contract 288349 SecFuNet. The third author was funded by a Sofja Kovalevskaja Award of the Alexander von Humboldt Foundation and the German Federal Ministry for Education and Research.

References

1. Abdalla, M., Benhamouda, F., Blazy, O., Chevalier, C., Pointcheval, D.: SPHF-friendly non-interactive commitment schemes. In: Advances in Cryptology – ASI-

ACRYPT 2013 (2013), full version available from the authors' web pages.

2. 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 (Aug 2009)
3. 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 (Aug 2005)
4. Beaver, D.: Adaptive zero knowledge and computational equivocation (extended abstract). In: 28th ACM STOC. pp. 629–638. ACM Press (May 1996)
5. Bellovin, S.M., Merritt, M.: Encrypted key exchange: Password-based protocols secure against dictionary attacks. In: 1992 IEEE Symposium on Security and Privacy. pp. 72–84. IEEE Computer Society Press (May 1992)
6. Benhamouda, F., Blazy, O., Chevalier, C., Pointcheval, D., Vergnaud, D.: New techniques for SPHFs and efficient one-round PAKE protocols. In: Canetti, R., Garay, J.A. (eds.) CRYPTO (1). Lecture Notes in Computer Science, vol. 8042, pp. 449–475. Springer (2013), full version available on the Cryptology ePrint Archive as reports 2013/034 and 2013/341.
7. Blazy, O., Pointcheval, D., Vergnaud, D.: Round-optimal privacy-preserving protocols with smooth projective hash functions. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 94–111. Springer (Mar 2012)
8. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: 42nd FOCS. pp. 136–145. IEEE Computer Society Press (Oct 2001)
9. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. Cryptology ePrint Archive, Report 2000/067 (2005), `http://eprint.iacr.org/`
10. Canetti, R., Fischlin, M.: Universally composable commitments. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 19–40. Springer (Aug 2001)
11. Canetti, R., Halevi, S., Katz, J., Lindell, Y., MacKenzie, P.D.: Universally composable password-based key exchange. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 404–421. Springer (May 2005)
12. Canetti, R., Krawczyk, H.: Universally composable notions of key exchange and secure channels. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 337–351. Springer (Apr / May 2002)
13. Canetti, R., Lindell, Y., Ostrovsky, R., Sahai, A.: Universally composable two-party and multi-party secure computation. In: 34th ACM STOC. pp. 494–503. ACM Press (May 2002)
14. Canetti, R., Rabin, T.: Universal composition with joint state. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 265–281. Springer (Aug 2003)
15. Choi, S.G., Katz, J., Wee, H., Zhou, H.S.: Efficient, adaptively secure, and composable oblivious transfer with a single, global crs. In: Kurosawa, K., Hanaoka, G. (eds.) Public Key Cryptography. Lecture Notes in Computer Science, vol. 7778, pp. 73–88. Springer (2013)
16. Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Krawczyk, H. (ed.) CRYPTO'98. LNCS, vol. 1462, pp. 13–25. Springer (Aug 1998)
17. 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 (Apr / May 2002)
18. Dolev, D., Dwork, C., Naor, M.: Nonmalleable cryptography. SIAM Journal on Computing 30(2), 391–437 (2000)

19. Fischlin, M., Libert, B., Manulis, M.: Non-interactive and re-usable universally composable string commitments with adaptive security. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 468–485. Springer (Dec 2011)
20. Gennaro, R., Lindell, Y.: A framework for password-based authenticated key exchange. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 524–543. Springer (May 2003), http://eprint.iacr.org/2003/032.ps.gz
21. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game, or a completeness theorem for protocols with honest majority. In: Aho, A. (ed.) 19th ACM STOC. pp. 218–229. ACM Press (May 1987)
22. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. Journal of the ACM 38(3), 691–729 (1991)
23. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer (Apr 2008)
24. Haralambiev, K.: Efficient Cryptographic Primitives for Non-Interactive Zero-Knowledge Proofs and Applications. Ph.D. thesis, New York University (2011)
25. Hofheinz, D., Müller-Quade, J.: Universally composable commitments using random oracles. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 58–76. Springer (Feb 2004)
26. Horvitz, O., Katz, J.: Universally-composable two-party computation in two rounds. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 111–129. Springer (Aug 2007)
27. Kalai, Y.T.: Smooth projective hashing and two-message oblivious transfer. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 78–95. Springer (May 2005)
28. Katz, J.: Universally composable multi-party computation using tamper-proof hardware. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 115–128. Springer (May 2007)
29. 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 (May 2001)
30. Katz, J., Vaikuntanathan, V.: Round-optimal password-based authenticated key exchange. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 293–310. Springer (Mar 2011)
31. Naor, M., Pinkas, B.: Efficient oblivious transfer protocols. In: 12th SODA. pp. 448–457. ACM-SIAM (Jan 2001)
32. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Feigenbaum, J. (ed.) CRYPTO'91. LNCS, vol. 576, pp. 129–140. Springer (Aug 1992)
33. 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 (Aug 2008)
34. Rabin, M.O.: How to exchange secrets with oblivious transfer. Technical Report TR81, Harvard University (1981)