Removing Erasures with Explainable Hash Proof Systems

Michel Abdalla[1], Fabrice Benhamouda[2], and David Pointcheval[1]

[1] ENS, CNRS, INRIA, and PSL Research University, Paris, France
{michel.abdalla,david.pointcheval}@ens.fr
www.di.ens.fr/~{abdalla,pointche}
[2] IBM Research, Yorktown Heights, NY, US
fabrice.benhamouda@normalesup.org
www.normalesup.org/~fbenhamo

Abstract. An important problem in secure multi-party computation is the design of protocols that can tolerate adversaries that are capable of corrupting parties dynamically and learning their internal states. In this paper, we make significant progress in this area in the context of password-authenticated key exchange (PAKE) and oblivious transfer (OT) protocols. More precisely, we first revisit the notion of projective hash proofs and introduce a new feature that allows us to *explain* any message sent by the simulator in case of corruption, hence the notion of *Explainable Projective Hashing*. Next, we demonstrate that this new tool generically leads to efficient PAKE and OT protocols that are secure against semi-adaptive adversaries without erasures in the Universal Composability (UC) framework. We then show how to make these protocols secure even against adaptive adversaries, using *non-committing encryption*, in a much more efficient way than generic conversions from semi-adaptive to adaptive security. Finally, we provide concrete instantiations of explainable projective hash functions that lead to the most efficient PAKE and OT protocols known so far, with UC-security against adaptive adversaries, without assuming reliable erasures, in the single global CRS setting.
As an important side contribution, we also propose a new commitment scheme based on DDH, which leads to the construction of the first one-round PAKE adaptively secure under plain DDH without pairing, assuming reliable erasures, and also improves previous constructions of OT and two- or three-round PAKE schemes.

Keywords. Oblivious Transfer, Password Authenticated Key Exchange, Erasures, Universal Composability, Adaptive Adversaries.

1 Introduction

1.1 Motivation

One of the most difficult problems in secure multi-party computation is the design of protocols that can tolerate adaptive adversaries. These are adversaries

which can corrupt parties dynamically and learn their internal states. As stated in the seminal work of Canetti *et al.* [12], this problem is even more difficult when uncorrupted parties may deviate from the protocol by keeping record of past configurations, instead of erasing them, or just because erasures are not reliable. To deal with this problem, they introduced the concept of non-committing encryption (NCE) and showed how to use it to build general multi-party computation protocols that remained secure even in the presence of such adversaries. Unfortunately, the gain in security came at the cost of a significant loss in efficiency. Though these results were later improved (e.g, [6,17,21,27]), NCE still requires a large amount of communication and achieving efficient constructions with adaptive security without assuming reliable erasures remains a difficult task.

To address the efficiency issue with previous solutions, Garay, Wichs, and Zhou [24] (GWZ) introduced two new notions. The first one was the notion of semi-adaptive security in which an adversary is not allowed to corrupt a party if all the parties are honest at the beginning of the protocol. The main advantage of the new notion is that it is only slightly more difficult to achieve than static security but significantly easier than fully-adaptive security. The second new notion was the concept *somewhat non-committing encryption*. Unlike standard NCE schemes, somewhat non-committing encryption only allows the sender of a ciphertext to open it in a limited number of ways, according to an equivocality parameter ℓ.

In addition to being able to build very efficient somewhat non-committing encryption schemes for small values of ℓ, Garay *et al.* [24] also showed how to build a generic compiler with the help of such schemes that converts any semi-adaptively secure cryptographic scheme into a fully-adaptively secure one. Since the equivocality parameter ℓ needed by their compiler is proportional to the input and output domains of the functionality being achieved, they were able to obtain very efficient constructions for functionalities with small domains, such as 1-out-of-2 oblivious transfers (OT). In particular, their results do not require reliable erasures and hold in the universal composability (UC) framework [8,9].

Building on the results of Garay *et al.* [24], Canetti *et al.* [10] showed how to use 1-out-of-2 OT protocols to build reasonably efficient password-based authenticated key exchange (PAKE) protocols in the UC framework against adaptive corruptions without erasures. The number of OT instances used in their protocol is proportional to the number of bits of the password.

Even though both works provide efficient constructions of UC-secure OT and PAKE schemes with adaptive security without erasures, the efficiency gap between these protocols and those which assume reliable erasures (e.g., [1,18]) remains significant. In this work, we aim to reduce this gap.

1.2 Our Approach

In order to build more efficient OT and PAKE schemes with adaptive security without erasures, we start from the constructions of Abdalla *et al.* [1], which were the most efficient OT and PAKE constructions in the UC model with adaptive

corruptions, with a single global common reference string (CRS)[3], and assuming reliable erasures. We then improve them to make them secure against *semi-adaptive* adversaries, without erasures. Finally, we show how to enhance these protocols with *non-committing encryption* (NCE) in order to achieve adaptive security without erasures and without impacting too much their efficiency. All our constructions assume the existence of a single global CRS (notice that even with static corruptions, OT and PAKE in the UC model do not exist in the plain model without CRS [14]).

Hash Proof Systems. At the heart of the OT and PAKE constructions in [1] is the following idea: one party commits to his index (for OT) or his password (for PAKE), and the other party derives from this commitment some hash value which the first party can compute if his commitment was valid and contained some given value (a valid password or a given index), or appears random otherwise. This hash value is then used to mask the values to be transferred in the OT case or is used to derive the session key in the PAKE case.

More precisely, this hash value is computed through a hash proof system or smooth projective hash functions (SPHF) [20]. An SPHF is defined for a language $\mathcal{L} \subseteq \mathcal{X}$. In our case, this language is the language of valid commitments of some value. The first property of an SPHF is that, for a word C in $\mathcal{L}$, the hash value can be computed using either a *secret* hashing key hk (generated by the first party) or a *public* projected key hp (derived from hk and given to the second party) together with a witness w to the fact that C is indeed in $\mathcal{L}$. However, for a word C not in $\mathcal{L}$, the hash value computed with hk is perfectly random, even knowing hp. The latter is known as the *smoothness* property.

Explainable Hash Proof Systems. To make the protocol secure against semi-adaptive adversaries, we face two main problems. The first is the fact the commitment scheme has at the very least to be UC-secure against semi-adaptive adversaries, without relying on erasures. While this is not the case for the original commitment scheme in [1], we show that it is true for a slight variant of it.

The second problem is the main challenge: in case of corruption of an honest player, after this player sent some projection key hp, we need to exhibit a hashing key hk that is compatible with the view of the adversary. In particular, this view may contain a hash value of some commitment under hk. For that purpose, we introduce the notion of explainable hash proof systems (EPHFs) which basically are SPHFs with a trapdoor enabling to generate a projection key hp, and later exhibit a hashing key hk for any hash value.

We propose two constructions of EPHFs. The first one works with any SPHF, as long as there exists a trapdoor which enables to generate, for any hashing key hk, a random hashing key hk' associated to the same projection key as hp. This property is achieved by most known SPHFs. Then to generate a hashing key hk' corresponding to a given projection key hp (associated to some known hk) and a given hash value H, we can draw hk' as above until it corresponds

[3] Here, global CRS just means multiple parties can share the same CRS, as in [18]. Our notion of global CRS is different from that in [11].

to the hash value H. Unfortunately, this can only be done if the set of possible hash values is small. One way to ensure this fact is to truncate the hash value to only ν bits instead of keeping the entire hash value. In this case, the reduction requires $O(2^\nu)$ drawing of hk'.

This reduction gap means that ν has to be logarithmic in the security parameter. If we look carefully at current SPHF constructions over cyclic groups, we remark that hashing keys are usually vectors of scalars, while hash values are typically group elements. Therefore, intuitively, it does not seem possible to recover a hashing key from a hash value, without performing some kind of discrete logarithm computation on the hash value.[4] As a result, it appears that the best we can hope for in this case is to drop the cost from $O(2^\nu)$ down to $O(2^{\nu/2})$, through the use of a baby-step giant-step algorithm, or the Pollard's kangaroo method [30]. A straightforward application of this idea to an SPHF, however, would require computing the discrete logarithm of the hash value, which is impractical. Our second construction consists largely in making this idea work.

From Semi-Adaptive to Adaptive Adversaries. Once we obtain OT and PAKE protocols secure against semi-adaptive adversaries using EPHFs, we still need to transform them into protocols secure against adaptive adversaries.

First, for PAKE, the GWZ transformation cannot directly be used because channels are not authenticated, and some ideas of Canetti *et al.* in [4] need to be combined to deal with this issue. Even then, the GWZ improvement of using somewhat NCE cannot be applied directly because PAKE outputs are session keys, and therefore there is an exponential number of them, which means the equivocality parameter and the communication complexity of the resulting protocol would be exponential in the security parameter. Hence, to transform a semi-adaptively secure PAKE protocol into an adaptively secure one, each bit of each flow of the original protocol needs to be sent through an NCE channel. While the resulting protocol would only be 3-round, its communication complexity would be impractical: even with the most efficient NCE schemes known so far [17], this would multiply the communication complexity of the original protocol by about 320.[5] This is why we propose a new transformation from semi-adaptively secure to adaptively-secure PAKE, in which only $\mathfrak{K} + 8\nu_m$ bits are sent via NCE channels (where $\mathfrak{K}$ is the security parameter and ν_m is the password length).

Second, for OT, while the GWZ transformation is very practical for bit OT (i.e., OT for one-bit messages), it cannot be used for long messages nor for 1-out-of-k OT for large k (e.g., polynomial in the security parameter) for similar

[4] We could alternatively use group elements for the hashing key, but that would require bilinear maps, and the hash value would be in the target group $\mathbb{G}_T$ of the pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. So we would still need to be able to convert a group element from the target group $\mathbb{G}_T$ to the original group $\mathbb{G}$. In any case, the whole comment just highlights our intuition. There might be other ways of avoiding any discrete logarithm computation, using some novel ideas we have not thought about.

[5] We are interested in mimimizing the total communication complexiy of the NCE scheme. With regards to this measure of efficiency, the NCE scheme of Hemenway, Ostrovsky, and Rosen in [27] is less efficient than the scheme of Choi *et al.* [17].

reasons as in the PAKE case. Garay *et al.* [24] proposed a solution for long messages consisting in running ν_m-bit string OT together with zero-knowledge proofs to make sure the same index is used in all protocols. Here, we show how to directly construct ν_m-bit string OT from our specific semi-adaptive protocol at a much lower cost, by avoiding zero-knowledge proofs and reducing the number of bits sent via NCE channels. Contrary to a solution obtained by the GWZ transformation, the communication complexity of this new protocol is polynomial in k (instead of being exponential in k).

Relying only on DDH. As an important side contribution, we propose a new SPHF-friendly commitment scheme based on the plain Decisional Diffie-Hellman assumption (DDH). In addition to being more efficient than the one of Abdalla *et al.* [1], the new commitment scheme also does not require pairings. As a result, the new scheme can be used to significantly improve previous OT and PAKE schemes in the UC model with adaptive adversaries, assuming reliable erasures. Moreover, it also yields to the *first one-round PAKE scheme under plain DDH*, using [1]. All the previously known one-round PAKE schemes (even only secure against statistical corruptions) use pairings, including the recent extremely efficient scheme of Jutla and Roy in [28], where each user only sends four group elements.

For our protocols to be secure, the underlying commitment scheme has to possess strong properties, which makes its design quite challenging. First, we need to be able to extract the inputs of the parties and, in particular, the commitments produced by the adversary. Second, we also need to be able to simulate a party without knowing its input and, in particular, his commitments; but we still need to be able to later open these commitments to the correct input, in case of corruption. In other words, the commitment has to be both equivocable and extractable. Third, to be compatible with SPHF, an additional twist is required: the language $\mathcal{L}$ of commitments of a given value need to be non-trivial. More precisely, it should not be possible for a (polynomial-time) adversary to generate a commitment which may be opened in multiple ways (even if a polynomial-time adversary may not be able to find it), or in other words, a commitment generated by a polynomial-time adversary has to be perfectly binding. This last property is called robustness. Roughly speaking, a commitment satisfying all these three properties is said to be SPHF-friendly.

Efficient constructions of equivocable and extractable commitments fall in two categories: the one following the ideas of Canetti and Fischlin [13] (including [1,3]), and the ones using non-interactive zero-knowledge proofs as decommitment information as the Fischlin-Libert-Manulis schemes [23]. The latter ones are not robust and cannot be used for our purpose. The first basically consists, when the committed value is just one bit b, to commit in an equivocable way to b, and provide two ciphertexts C_0 and C_1, where C_b contains the decommitment information for b and C_{1-b} is random. Extracting such a commitment can be done by decrypting C_0 and C_1 and finding which of them contains a valid decommitment information, while simulating such a commitment just consists

of encryptions of valid decommitment information in C_0 and C_1 (for 0 and 1, respectively).

The difficulty is to find an equivocable commitment and an encryption scheme compatible with an SPHF, which essentially means that they have to be structure-preserving. In [3], the Pedersen [31] commitment scheme is used. But then the decommitment information has to be done bit by bit as it is a scalar, which is very inefficient[6]. To solve this issue, in [1], one of the Haralambiev structure-preserving commitment schemes [26] is used, at the expense of relying on SXDH and pairings. Unfortunately, there does not seem to exist structure-preserving commitment schemes under plain DDH. This is why we developed a new way of constructing SPHF-friendly commitment schemes.

1.3 Organization of the Paper

Due to space restrictions, we focus on OT in the core of the paper. PAKE constructions are detailed in the full version [2].

After recalling some definitions in Section 2, we introduce our new notion of explainable hash proof systems (EPHFs) in Section 3 and present our two constructions. This is our first main contribution. Then, we show how to use EPHFs and SPHF-friendly commitments to construct OT UC-secure against semi-adaptive adversaries, in Section 4. Next, we introduce our new SPHF-friendly commitment scheme under plain DDH, which is our second main contribution. Using the latter, we also provide substantial improvements for OT and PAKE schemes in the UC model, assuming reliable erasures. Finally, in Section 6, we show how to efficiently enhance our OT semi-adaptive protocols with *non-committing encryption* (NCE) in order to achieve adaptive security. In particular, we propose several adaptive versions of our semi-adaptive OT protocols, yielding different trade-offs in terms of communication complexity and number of rounds. In each case, at least one of our new protocols outperforms existing ones. A detailed related work coverage can be found in the full version [2].

To better focus on the core ideas, standard definitions and notations are recalled in the full version [2]. Additional details and proofs for EPHFs, all the proofs of our semi-adaptively and adaptively secure protocols, and proofs and some technical parts of our new SPHF-friendly commitment are in the full version [2].

2 Definitions

Notations. As usual, all the players and algorithms will be possibly probabilistic and stateful. Namely, adversaries can keep a state st during the different phases, and we denote $\xleftarrow{\$}$ the outcome of a probabilistic algorithm or the sampling from a uniform distribution. For example, $\mathcal{A}(x;r)$ will denote the execution of $\mathcal{A}$ with

[6] In addition, the SPHF we can build is a weak form of SPHF, and cannot be used in one-round PAKE protocol for example.

input x and random tape r. For the sake of clarity, sometimes, the latter random tape will be dropped, with the notation $\mathcal{A}(x)$.

Smooth Projective Hash Functions. Projective hashing was first introduced by Cramer and Shoup [20]. Here we use the formalization of SPHF from [7].

Let $(\mathcal{X}_{\mathsf{crs}})_{\mathsf{crs}}$ be a family of domains for the hash functions indexed by crs, and let $(\mathcal{L}_{\mathsf{crs,par}})_{\mathsf{crs,par}}$ be a family of languages, i.e., $\mathcal{L}_{\mathsf{crs,par}}$ is a subset of $\mathcal{X}_{\mathsf{crs}}$. For the sake of simplicity, we write $\mathsf{crs\text{-}par} = (\mathsf{crs}, \mathsf{par})$. In this paper, we focus on languages of commitments, whose corresponding plaintexts satisfy some relations, and even more specifically here equal to some value par. The value crs will be the common reference string for these commitments. The value par is a parameter which is not necessarily public. In case of PAKE for example, it is the expected password.

A key property of an SPHF is that, for a word C in $\mathcal{L}_{\mathsf{crs\text{-}par}}$, the hash value 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 $\mathcal{L}$. More precisely, an SPHF is defined by four algorithms:

- $\mathsf{HashKG}(\mathsf{crs})$ generates a hashing key hk for crs;
- $\mathsf{ProjKG}(\mathsf{hk}, \mathsf{crs}, C)$ derives the projection key hp;
- $\mathsf{Hash}(\mathsf{hk}, \mathsf{crs\text{-}par}, C)$ outputs the hash value (in a set Π, called the *range* of the SPHF) from the hashing key hk, for any word $C \in \mathcal{X}$;
- $\mathsf{ProjHash}(\mathsf{hp}, \mathsf{crs\text{-}par}, C, w)$ outputs the hash value from the projection key hp, and the witness w, for a word $C \in \mathcal{L}$.

On the one hand, the *correctness* of the SPHF assures that if $C \in \mathcal{L}_{\mathsf{crs\text{-}par}}$ with w a witness of this fact, then $\mathsf{Hash}(\mathsf{hk}, \mathsf{crs\text{-}par}, C) = \mathsf{ProjHash}(\mathsf{hp}, \mathsf{crs\text{-}par}, C, w)$. On the other hand, the security is defined through the *smoothness*, which guarantees that, if $C \notin \mathcal{L}_{\mathsf{crs\text{-}par}}$, $\mathsf{Hash}(\mathsf{hk}, \mathsf{crs\text{-}par}, C)$ is *statistically* indistinguishable from a random element, even knowing hp. More formally, an SPHF is smooth if, for any crs, any par, and any $C \notin \mathcal{L}_{\mathsf{crs\text{-}par}}$, the following two distributions are statistically indistinguishable:

$$\{(\mathsf{hp}, H) \mid \mathsf{hk} \xleftarrow{\$} \mathsf{HashKG}(\mathsf{crs}); \mathsf{hp} \leftarrow \mathsf{ProjKG}(\mathsf{hk}, \mathsf{crs}, C); H \leftarrow \mathsf{Hash}(\mathsf{hk}, \mathsf{crs\text{-}par}, C)\}$$

$$\{(\mathsf{hp}, H) \mid \mathsf{hk} \xleftarrow{\$} \mathsf{HashKG}(\mathsf{crs}); \mathsf{hp} \leftarrow \mathsf{ProjKG}(\mathsf{hk}, \mathsf{crs}, C); H \xleftarrow{\$} \Pi\}.$$

We chose to restrict HashKG and ProjKG not to use the parameter par, but just crs (instead of crs-par), as for some applications, such as PAKE, hk and hp have to be independent of par, since par is a secret (the password in case of PAKE). We know that this is a stronger restriction than required for our purpose, since one can use par without leaking any information about it; and some of our applications such as OT do not require par to be private at all. But, this is not an issue, since none of our SPHFs uses par.

If ProjKG does not depend on C and satisfies a slightly stronger smoothness property (called adaptive smoothness, which holds even if C is chosen after hp), we say the SPHF is a KV-SPHF, as such an SPHF was introduced by Katz and Vaikuntanathan in [29]. Otherwise, it is said to be a GL-SPHF, as such an SPHF

was introduced by Gennaro and Lindell in [25]. More formally, a KV-SPHF is said to be smooth if for any crs, any par, and any function f from the set of projection keys to $\mathcal{X}_{\mathsf{crs\text{-}par}} \setminus \mathcal{L}_{\mathsf{crs\text{-}par}}$, the following two distributions are statistically indistinguishable:

$$\{(\mathsf{hp}, H) \mid \mathsf{hk} \xleftarrow{\$} \mathsf{HashKG}(\mathsf{crs}); \mathsf{hp} \leftarrow \mathsf{ProjKG}(\mathsf{hk}, \mathsf{crs}); H \leftarrow \mathsf{Hash}(\mathsf{hk}, \mathsf{crs\text{-}par}, f(\mathsf{hp}))\}$$

$$\{(\mathsf{hp}, H) \mid \mathsf{hk} \xleftarrow{\$} \mathsf{HashKG}(\mathsf{crs}); \mathsf{hp} \leftarrow \mathsf{ProjKG}(\mathsf{hk}, \mathsf{crs}); H \xleftarrow{\$} \Pi\}.$$

See [7] for details on GL-SPHF and KV-SPHF and language definitions.

We would like to remark that one can easily extend the range of an existing SPHF by concatenating several hash values with independent hashing keys on the same word. In this case, the global projection key would be the concatenation of the respective projection keys. It is straightforward to see that the smoothness property of the global SPHF follows directly from a classic hybrid argument over the smoothness property of the underlying SPHF.

SPHF-Friendly Commitment Schemes. In this section, we briefly sketch the definition of SPHF-friendly commitment schemes we will use in this paper (more details are given in the full version [2]). This is a slightly stronger variant of the one in [1], since it requires an additional polynomial-time algorithm C.IsBinding. But the construction in [1] still satisfies it. This is a commitment scheme that is both equivocable and extractable. It is defined by the following algorithms: $\mathsf{C.Setup}(1^{\mathfrak{K}})$ generates the global parameters, passed through the global CRS crs to all other algorithms, while $\mathsf{C.SetupT}(1^{\mathfrak{K}})$ is an alternative that additionally outputs a trapdoor τ; $\mathsf{C.Com}^{\ell}(\boldsymbol{M})$ outputs a pair (C, δ), where C is the commitment of the message $\boldsymbol{M}$ for the label ℓ, and δ is the corresponding opening data, used by $\mathsf{C.Ver}^{\ell}(C, \boldsymbol{M}, \delta)$ to check the correct opening for C, $\boldsymbol{M}$ and ℓ. It always outputs 0 (false) on $\boldsymbol{M} = \perp$. The trapdoor τ can be used by $\mathsf{C.Sim}^{\ell}(\tau)$ to output a pair (C, eqk), where C is a commitment and eqk an equivocation key that is later used by $\mathsf{C.Open}^{\ell}(\mathsf{eqk}, C, \boldsymbol{M})$ to open C on any message $\boldsymbol{M}$ with an appropriate opening data δ. The trapdoor τ can also be used by $\mathsf{C.Ext}^{\ell}(\tau, C)$ to output the committed message $\boldsymbol{M}$ in C, or $\perp$ if the commitment is invalid. Eventually, the trapdoor τ also allows $\mathsf{C.IsBinding}^{\ell}(\tau, C, \boldsymbol{M})$ to check whether the commitment C is binding to the message $\boldsymbol{M}$ or not: if there exists $\boldsymbol{M}' \neq \boldsymbol{M}$ and δ', such that $\mathsf{C.Ver}^{\ell}(C, \boldsymbol{M}', \delta') = 1$, then it outputs 0.

All these algorithms should satisfy some correctness properties: all honestly generated commitments open and verify correctly, can be extracted and are binding to the committed value, while the simulated commitments can be opened on any message.

Then, some security guarantees should be satisfied as well, when one denotes the generation of fake commitments $(C, \delta) \xleftarrow{\$} \mathsf{C.SCom}^{\ell}(\tau, \boldsymbol{M})$, computed as $(C, \mathsf{eqk}) \xleftarrow{\$} \mathsf{C.Sim}^{\ell}(\tau)$ and then $\delta \leftarrow \mathsf{C.Open}^{\ell}(\mathsf{eqk}, C, \boldsymbol{M})$:

- *Setup Indistinguishability*: one cannot distinguish the CRS generated by C.Setup from the one generated by C.SetupT;
- *Strong Simulation Indistinguishability*: one cannot distinguish a real commitment (which is generated by C.Com) from a fake commitment (generated

by $\mathsf{C.SCom}$), even with oracle access to the extraction oracle ($\mathsf{C.Ext}$), the binding test oracle ($\mathsf{C.IsBinding}$), and to fake commitments (using $\mathsf{C.SCom}$);
– *Robustness*: one cannot produce a commitment and a label that extracts to $\boldsymbol{M}$ (possibly $\boldsymbol{M} = \perp$) such that $\mathsf{C.IsBinding}^{\ell}(\tau, C, \boldsymbol{M}) = 0$, even with oracle access to the extraction oracle ($\mathsf{C.Ext}$), the binding test oracle ($\mathsf{C.IsBinding}$), and to fake commitments (using $\mathsf{C.SCom}$).

Note that, for excluding trivial attacks, on fake commitments, the extraction oracle outputs the $\mathsf{C.SCom}$-input message and the binding test oracle accepts for the $\mathsf{C.SCom}$-input message too. Finally, an SPHF-friendly commitment scheme has to admit an SPHF for the following language:

$$\mathcal{L}_{\mathsf{crs\text{-}par}} = \{(\ell, C) \mid \exists \delta,\ \mathsf{C.Ver}^{\ell}(C, \boldsymbol{M}, \delta) = 1\},$$

where $\mathsf{crs\text{-}par} = (\mathsf{crs}, \mathsf{par})$ and $\boldsymbol{M} = \mathsf{par}$.

Basically, compared to the original definition in [1], the main difference is that it is possible to check in polynomial time (using $\mathsf{C.IsBinding}$) whether a commitment is perfectly binding or not, i.e., does not belong to any $\mathcal{L}_{(\mathsf{crs}, \boldsymbol{M}')}$ for $\boldsymbol{M}' \neq \boldsymbol{M}$, where $\boldsymbol{M}$ is the value extracted from the commitment via $\mathsf{C.Ext}$. In addition, in the games for the strong simulation indistinguishability and the robustness, the adversary has access to this oracle $\mathsf{C.IsBinding}$.

Finally, for our PAKE protocols, as in [1], we need another property called strong pseudo-randomness. This property is a strong version of the pseudo-randomness property. However, while the latter is automatically satisfied by any SPHF-friendly commitment scheme, the former may not, because of an additional information provided to the adversary. But, it is satisfied by the SPHF-friendly commitment scheme in [1] and by our new commitment scheme introduced in Section 5, which is the most efficient known so far, based on the plain DDH.

SPHF-Friendly Commitment Schemes without Erasures. We will say that an SPHF-friendly commitment scheme is *without erasures* if this is an SPHF-friendly commitment scheme where δ (and thus the witness) just consists of the random coins used by the algorithm $\mathsf{C.Com}$. Then, an SPHF-friendly commitment scheme without erasures yields directly a commitment scheme that achieves UC-security without erasures.

We remark that slight variants of the constructions in [1, 3] are actually *without erasures*, as long as it is possible to sample obliviously an element from a cyclic group. To make these schemes without erasures, it is indeed sufficient to change the commitment algorithm $\mathsf{C.Com}$ to generate random ciphertexts (with elements obliviously sampled from the corresponding cyclic groups) instead of ciphertexts of 0, for the unused ciphertexts (i.e., the ciphertexts $b_{i,\overline{M_i}}$, for [1], using the notations in that paper). This does not change anything else, since these ciphertexts are not used in the verification algorithm $\mathsf{C.Ver}$.

In the sequel, all SPHF-friendly commitment schemes are assumed to be *without erasures*. Variants of [1, 3] are possible instantiations, but also our quite efficient constructions presented in Section 5 and the full version [2].

3 Explainable Projective Hashing

In this section, we define the notion of explainable projective hash function (EPHF) and then give two generic constructions of EPHF from SPHF. Both constructions work with any SPHF built using the generic framework of [7], basically as long as there is a way to generate the CRS so that the discrete logarithms of all elements are known. This encompasses most SPHFs over cyclic groups. The second construction is more efficient, but only enable building GL-EPHF, while the first construction enables building both GL-EPHF and KV-EPHF and is slightly more generic (it may work with SPHFs which are not built using the generic framework).

3.1 Definition

Let us first suppose there exists an algorithm Setup which takes as input the security parameter $\mathfrak{K}$ and outputs a CRS crs together with a trapdoor τ. In our case Setup will be C.SetupT, and the trapdoor τ will be the commitment trapdoor, which may need to be slightly modified, as we will see in our constructions. This modification generally roughly consists in adding the discrete logarithms of all used elements in the trapdoor C.SetupT and is possible with most concrete commitment schemes.

An *explainable projective hashing* (EPH) is an SPHF with the following additional property: it is possible to generate a random-looking projection key hp, and then receive some hash value H, some value par and some word $C \notin \mathcal{L}_{\mathsf{crs\text{-}par}}$, and eventually generate a valid hashing key hk which corresponds to hp and H, as long as we know τ. In other words, it is possible to generate hp and then "explain" any hash H for a word outside the language $\mathcal{L}_{\mathsf{crs\text{-}par}}$, by giving the appropriate hk.

While dual projective hashing [33] implies a weak version of smoothness, our notion of EPH implies the usual notion of smoothness, and is thus stronger than SPHF. Then, an EPHF can be either a GL-EPHF or a KV-EPHF, depending on whether the word C is known when hp is generated.

GL-EPHF. Formally, a GL-EPHF is defined by the following algorithms:

- $\mathsf{Setup}(1^{\mathfrak{K}})$ takes as input the security parameter $\mathfrak{K}$ and outputs the global parameters, passed through the global CRS crs or crs-par to all the other algorithms, plus a trapdoor τ;
- HashKG, ProjKG, Hash, and ProjHash behave as for a classical SPHF;
- $\mathsf{SimKG}(\mathsf{crs}, \tau, C)$ outputs a projection key hp together with an explainability key expk (C is not given as input for KV-EPHF);
- $\mathsf{Explain}(\mathsf{hp}, \mathsf{crs\text{-}par}, C, H, \mathsf{expk})$ outputs an hashing key hk corresponding to hp, crs-par, C, and H.

It must satisfy the same properties as an SPHF together with the following properties, for any $(\mathsf{crs}, \tau) \xleftarrow{\$} \mathsf{Setup}(1^{\mathfrak{K}})$:

– *Explainability Correctness.* For any par, any $C \notin \mathcal{L}_{\mathsf{crs\text{-}par}}$ and any hash value H, if $(\mathsf{hp}, \mathsf{expk}) \xleftarrow{\$} \mathsf{SimKG}(\mathsf{crs}, \tau, C)$ and $\mathsf{hk} \xleftarrow{\$} \mathsf{Explain}(\mathsf{hp}, \mathsf{crs\text{-}par}, C, H, \mathsf{expk})$, then $\mathsf{hp} = \mathsf{ProjKG}(\mathsf{hk}, \mathsf{crs}, C)$ and $H = \mathsf{Hash}(\mathsf{hk}, \mathsf{crs\text{-}par}, C)$, with overwhelming probability (over the random tape of $\mathsf{Explain}$);
– *Indistinguishability.* As for smoothness, we consider two types of indistinguishability: a GL-EPHF is indistinguishable, if for any par and any $C \notin \mathcal{L}_{\mathsf{crs\text{-}par}}$, the two following distributions are statistically indistinguishable:

$$\left\{ (\mathsf{hk}, \mathsf{hp}) \;\middle|\; \begin{array}{c} H \xleftarrow{\$} \Pi; (\mathsf{hp}, \mathsf{expk}) \xleftarrow{\$} \mathsf{SimKG}(\mathsf{crs}, \tau, C); \\ \mathsf{hk} \xleftarrow{\$} \mathsf{Explain}(\mathsf{hp}, \mathsf{crs\text{-}par}, C, H, \mathsf{expk}) \end{array} \right\}$$
$$\left\{ (\mathsf{hk}, \mathsf{hp}) \;\middle|\; \mathsf{hk} \xleftarrow{\$} \mathsf{HashKG}(\mathsf{crs}); \mathsf{hp} \leftarrow \mathsf{ProjKG}(\mathsf{hk}, \mathsf{crs}, C) \right\}.$$

KV-EPHF. A KV-EPHF is a GL-EPHF, for which ProjKG and SimKG does not take as input the word C, and which satisfies the same smoothness as a KV-SPHF, and a stronger indistinguishability property. A KV-EPHF is ε-indistinguishable, if for any par and any function f from the set of projection keys to $\mathcal{X} \setminus \mathcal{L}_{\mathsf{crs\text{-}par}}$, the two following distributions are statistically indistinguishable:

$$\left\{ (\mathsf{hk}, \mathsf{hp}) \;\middle|\; \begin{array}{c} H \xleftarrow{\$} \Pi; (\mathsf{hp}, \mathsf{expk}) \xleftarrow{\$} \mathsf{SimKG}(\mathsf{crs}, \tau, \perp); \\ \mathsf{hk} \xleftarrow{\$} \mathsf{Explain}(\mathsf{hp}, \mathsf{crs\text{-}par}, f(\mathsf{hp}), H, \mathsf{expk}) \end{array} \right\}$$
$$\left\{ (\mathsf{hk}, \mathsf{hp}) \;\middle|\; \mathsf{hk} \xleftarrow{\$} \mathsf{HashKG}(\mathsf{crs}); \mathsf{hp} \leftarrow \mathsf{ProjKG}(\mathsf{hk}, \mathsf{crs}, \perp) \right\}.$$

3.2 First Construction

This first construction enables to transform any GL-SPHF (or KV-SPHF) satisfying some properties of re-randomization of the hashing key into a GL-EPHF (respectively, a KV-SPHF). These properties are satisfied by any GL-SPHF (or KV-SPHF) built from the generic framework [7], when τ contains the discrete logarithms of all elements defining the language, as shown in the full version [2]. We first present the construction for GL-EPHF.

GL-EPHF. Here are the properties we require:

(a) For any hashing key hk and associated projection key hp, it is possible to draw a random hk' corresponding to hp, such that hk' looks like a fresh hashing key (conditioned on the fact that its projection key is hp). More precisely, we suppose there exists a randomized algorithm $\mathsf{InvProjKG}$, which takes as input τ, a hashing key hk, $\mathsf{crs\text{-}par}$, and a word $C \notin \mathcal{L}_{\mathsf{crs\text{-}par}}$, and outputs a random hashing key hk', satisfying $\mathsf{ProjKG}(\mathsf{hk}', \mathsf{crs}, C) = \mathsf{hp}$. For any $\mathsf{crs\text{-}par}$, for any $C \notin \mathcal{L}_{\mathsf{crs\text{-}par}}$, for any hashing key $\mathsf{hk} \xleftarrow{\$} \mathsf{HashKG}(\mathsf{crs})$, the two following distributions are supposed to be statistically indistinguishable:

$$\{\mathsf{hk}' \mid \mathsf{hk}' \xleftarrow{\$} \mathsf{HashKG}(\mathsf{crs}) \text{ such that } \mathsf{ProjKG}(\mathsf{hk}, \mathsf{crs}, C) = \mathsf{ProjKG}(\mathsf{hk}', \mathsf{crs}, C)\}$$
$$\{\mathsf{hk}' \mid \mathsf{hk}' \xleftarrow{\$} \mathsf{InvProjKG}(\tau, \mathsf{hk}, \mathsf{crs}, C)\}.$$

For GL-SPHFs built from the generic framework [7], if we look at the discrete logarithms of all the group elements defining the language and all the ones in the projection key, hashing keys corresponding to a given projection key hp essentially are the solutions of a linear system (the right-hand side of the system corresponds to hp, while coefficients of the system depend on the language). $\mathsf{InvProjKG}$ can then output a uniform solution of this linear system.

(b) A stronger property than smoothness, called strong smoothness, is required. Informally, it ensures that smoothness holds even when the hashing key is conditioned on any projection key. Formally, a GL-SPHF is strongly smooth if for any $\mathsf{crs\text{-}par}$, for any $C \notin \mathcal{L}_{\mathsf{crs\text{-}par}}$, for any projection key hp (generated by $\mathsf{hk} \xleftarrow{\$} \mathsf{HashKG}(\mathsf{crs})$ and $\mathsf{hp} \leftarrow \mathsf{ProjKG}(\mathsf{hk}, \mathsf{crs}, C)$), the two following distributions are statistically indistinguishable:

$$\left\{ \mathsf{Hash}(\mathsf{hk}', \mathsf{crs\text{-}par}, C) \;\middle|\; \begin{array}{c} \mathsf{hk}' \xleftarrow{\$} \mathsf{HashKG}(\mathsf{crs}) \text{ such that} \\ \mathsf{ProjKG}(\mathsf{hk}', \mathsf{crs}, C) = \mathsf{hp} \end{array} \right\}$$
$$\left\{ H \;\middle|\; H \xleftarrow{\$} \Pi \right\};$$

(c) There exists a parameter ν linear in $\log \mathfrak{K}$ and a randomness extractor $\mathsf{Extract}$ with range $\{0,1\}^\nu$, such that the two following distributions are statistically indistinguishable:

$$\{\mathsf{Extract}(H) \mid H \xleftarrow{\$} \Pi\} \qquad\qquad \{H \mid H \xleftarrow{\$} \{0,1\}^\nu\}.$$

Details on the randomness extractor can be found in the full version [2]. But we can use either a deterministic extractor exists for Π, which is possible for many cyclic groups [16], or a probabilistic extractor with an independent random string in the CRS.

Then, if the hash values H computed by Hash or $\mathsf{ProjHash}$ are replaced by $\mathsf{Extract}(H)$, the resulting SPHF is a GL-EPHF. Indeed, if $\mathsf{SimKG}(\mathsf{crs}, \tau, C)$ just generates $\mathsf{hk} \xleftarrow{\$} \mathsf{HashKG}(\mathsf{crs})$ and $\mathsf{hp} \leftarrow \mathsf{ProjKG}(\mathsf{hk}, \mathsf{crs}, C)$, and outputs hp and $\mathsf{expk} = (\tau, \mathsf{hk})$. Then, $\mathsf{Explain}(\mathsf{hp}, \mathsf{crs\text{-}par}, C, H, \mathsf{expk})$ just runs $\mathsf{hk}' \xleftarrow{\$} \mathsf{InvProjKG}(\tau, \mathsf{hk}, \mathsf{crs}, C)$ many times until it finds hk' such that $\mathsf{Hash}(\mathsf{hk}', \mathsf{crs\text{-}par}, C) = H$. It aborts if does not find a valid hk' after $2^\nu \mathfrak{K}$ times. Thanks to the smoothness and the above properties, its abort probability is negligible in the security parameter $\mathfrak{K}$.[7] Since ν is linear in $\log \mathfrak{K}$, the resulting algorithm $\mathsf{Explain}$ runs in polynomial time in $\mathfrak{K}$. A formal proof can be found in the full version [2].

We observe that ν impacts on the running time of SimKG which will only be used in the proofs of our PAKE and OT protocols (and not in their constructions), so that ν only impacts on the tightness of the proofs of the resulting protocols. In all comparisons in this article, we will use $\nu = 1$, which hinders performances of

[7] Notice that the strong smoothness is necessary to prove that as, otherwise, it would have been possible that for some projection key hp, no such hk' exist, and $\mathsf{Explain}$ would not run in expected polynomial time. See details in the full version [2].

our scheme; but our schemes are still very efficient. In practice, to gain constant factors, it would be advisable to use a greater ν, and thus larger blocks. Finally, the range of the EPHF can be easily extended just by using multiple copies of the EPHF: for a range of ν', hk becomes a tuple of $\lceil \nu'/\nu \rceil$ original hashing keys, the same for hp and H.

KV-EPHF. In the first generic construction for GL-SPHF, we get a KV-EPHF, if Property *(a)* and Property *(b)* hold even if C can depend on hp. In other words, instead of quantifying on any $C \notin \mathcal{L}_{\mathsf{crs\text{-}par}}$, we quantify on any function f from the set of projection keys to $\mathcal{X} \setminus \mathcal{L}_{\mathsf{crs\text{-}par}}$, and replace C by $f(\mathsf{hp})$ in the definition (similarly to what is done for the smoothness of KV-SPHF or the indistinguishability of KV-EPHF).

As for GL-EPHF, any KV-SPHF built using the generic framework satisfies these properties and so can be transformed into KV-EPHF, as long as discrete logarithms of all elements in the matrix Γ can be known from τ

3.3 Second Construction

We show a more efficient construction for GL-EPHF from any GL-SPHF built using the generic framework in the full version [2]. The idea is to use the algebraic properties of this framework to replace the costly search for hk' in Explain (which requires $O(2^\nu)$ guesses) by the computation of a small (less than 2^ν) discrete logarithm in ProjHash. This can be done in $O(2^{\nu/2})$ group operations by ProjHash, using Pollard's kangaroo method in [30]. The parameter ν can therefore be twice larger in our second construction, which makes it approximately twice more efficient.

4 Semi-Adaptive OT without Erasures

In this section, we propose a new OT protocol that is UC-secure against semi-adaptive adversaries, without requiring reliable erasures. The new protocol is very similar to the UC-secure OT construction in [1], except that the underlying SPHF-friendly commitment scheme has to be *without erasures* and the underlying SPHF has to be *explainable*. The security proof, which can be found in the full version [2], is however more complex.

4.1 Semi Adaptivity

The semi-adaptive setting has been introduced in [24], for two-party protocols when channels are authenticated: the adversary is not allowed to corrupt any player if the two players were honest at the beginning of the protocol. When channels are not authenticated, as for PAKE, we restrict the adversary not to corrupt a player P_i if an honest flow has been sent on its behalf, and it has been received by P_j, without being altered.

In addition to those restrictions on the adversary, there are also some restrictions on the simulator and the protocol. First, the simulator has to be

The functionality $\mathcal{F}_{(1,k)\text{-OT}}$ is parameterized by a security parameter $\mathfrak{K}$. It interacts with an adversary $\mathcal{S}$ and a set of parties $P_1, \ldots, P_n$ via the following queries:

- **Upon receiving an input $(\mathtt{Send}, \mathsf{sid}, \mathsf{ssid}, P_i, P_j, (m_1, \ldots, m_k))$ from party P_i**, with $m_i \in \{0,1\}^{\mathfrak{K}}$: record the tuple $(\mathsf{sid}, \mathsf{ssid}, P_i, P_j, (m_1, \ldots, m_k))$ and reveal $(\mathtt{Send}, \mathsf{sid}, \mathsf{ssid}, P_i, P_j)$ to the adversary $\mathcal{S}$. Ignore further `Send`-message with the same ssid from P_i.
- **Upon receiving an input $(\mathtt{Receive}, \mathsf{sid}, \mathsf{ssid}, P_i, P_j, s)$ from party P_j,** with $s \in \{1, \ldots, k\}$: record the tuple $(\mathsf{sid}, \mathsf{ssid}, P_i, P_j, s)$, and reveal $(\mathtt{Receive}, \mathsf{sid}, \mathsf{ssid}, P_i, P_j)$ to the adversary $\mathcal{S}$. Ignore further `Receive`-message with the same ssid from P_j.
- **Upon receiving a message $(\mathtt{Sent}, \mathsf{sid}, \mathsf{ssid}, P_i, P_j)$ from the adversary $\mathcal{S}$**: ignore the message if $(\mathsf{sid}, \mathsf{ssid}, P_i, P_j, (m_1, \ldots, m_k))$ or $(\mathsf{sid}, \mathsf{ssid}, P_i, P_j, s)$ is not recorded; otherwise send $(\mathtt{Sent}, \mathsf{sid}, \mathsf{ssid}, P_i, P_j)$ to P_i and ignore further `Sent`-message with the same ssid from the adversary.
- **Upon receiving a message $(\mathtt{Received}, \mathsf{sid}, \mathsf{ssid}, P_i, P_j)$ from the adversary $\mathcal{S}$**: ignore the message if $(\mathsf{sid}, \mathsf{ssid}, P_i, P_j, (m_1, \ldots, m_k))$ or $(\mathsf{sid}, \mathsf{ssid}, P_i, P_j, s)$ is not recorded; otherwise send $(\mathtt{Received}, \mathsf{sid}, \mathsf{ssid}, P_i, P_j, m_s)$ to P_j and ignore further `Received`-message with the same ssid from the adversary.

Fig. 1. Ideal Functionality for 1-out-of-k Oblivious Transfer $\mathcal{F}_{(1,k)\text{-OT}}$

setup-preserving, which means, in our case, that it first has to generate the CRS, before simulating the protocol execution. Second, the simulator has to be *input-preserving*, which means that if the adversary corrupts some user and honestly runs the protocol for some input x, the simulator submits the same input to the functionality. Third, the protocol has to be *well-formed*, which means that the number of flows and the size of each flow is independent of the input and the random tapes of the users. All these restrictions are clearly satisfied by our simulators and protocols. Formal definitions can be found in [24].

4.2 Oblivious Transfer

The ideal functionality of an Oblivious Transfer (OT) protocol is depicted in Fig. 1. It is inspired from [18]. In Fig. 2, we describe a 2-round 1-out-of-k OT for ν_m-bit messages, that is UC-secure against semi-adaptive adversaries. It can be built from any SPHF-friendly commitment scheme, admitting a $\mathsf{GL\text{-}EPHF}$, with range $\Pi = \{0,1\}^{\nu_m}$, for the language: $\mathcal{L}_{\mathsf{crs\text{-}par}} = \{(\ell, C) \mid \exists \delta, \mathsf{C.Ver}^{\ell}(C, \boldsymbol{M}, \delta) = 1\}$, where $\mathsf{crs\text{-}par} = (\mathsf{crs}, \mathsf{par})$ and $\boldsymbol{M} = \mathsf{par}$.

In case of corruption of the database (sender) after it has sent its flow, since we are in the semi-adaptive setting, the receiver was already corrupted and thus the index s was known to the simulator. The latter can thus generate "explainable" hp_t for all $t \neq s$, so that when the simulator later learns the messages m_t, it can explain hp_t with appropriate hk_t. Erasures are no longer required, contrarily to [1].

The restriction that Π has to be of the form $\{0,1\}^{\nu_m}$ is implicit in [1]. Any SPHF can be transformed to an SPHF with range Π of the form $\{0,1\}^{\nu_m}$, using a

CRS: $\mathsf{crs} \xleftarrow{\$} \mathsf{C.Setup}(1^{\mathfrak{K}})$.

Index query on s:

1. P_j computes $(C, \delta) \xleftarrow{\$} \mathsf{C.Com}^{\ell}(s)$ with $\ell = (\mathsf{sid}, \mathsf{ssid}, P_i, P_j)$
2. P_j sends C to P_i

Database input $(m_1, \ldots, m_k)$:

1. P_i computes $\mathsf{hk}_t \xleftarrow{\$} \mathsf{HashKG}(\mathsf{crs})$, $\mathsf{hp}_t \leftarrow \mathsf{ProjKG}(\mathsf{hk}_t, \mathsf{crs}, (\ell, C))$, $K_t \leftarrow \mathsf{Hash}(\mathsf{hk}_t, (\mathsf{crs}, t), (\ell, C))$, and $M_t \leftarrow K_t \mathsf{\ xor\ } m_t$, for $t = 1, \ldots, k$
2. P_i sends $(\mathsf{hp}_t, M_t)_{t=1,\ldots,k}$

Data recovery:
Upon receiving $(\mathsf{hp}_t, M_t)_{t=1,\ldots,k}$, P_j computes $K_s \leftarrow \mathsf{ProjHash}(\mathsf{hp}_s, (\mathsf{crs}, s), (\ell, C), \delta)$ and gets $m_s \leftarrow K_s \mathsf{\ xor\ } M_s$.

Fig. 2. UC-Secure 1-out-of-k OT from an SPHF-Friendly Commitment for Semi-Adaptive Adversaries

randomness extractor, as long as the initial range is large enough. However, this is not necessarily the case for EPHF, since the extractor might not be efficiently invertible. That is why we prefer to make this assumption on Π explicit.[8]

5 A New SPHF-Friendly Commitment Scheme

In this section, we present our new efficient SPHF-friendly commitment scheme under the plain DDH. Due to lack of space, we only give an overview of the scheme and a comparison with previous SPHF-friendly commitment schemes. Details are left to the full version [2].

5.1 Scheme

High-Level Intuition. The basic idea of our scheme is a generalization of the schemes in [1, 3, 13, 15]. In these schemes, the commitment of a bit b consists of an equivocable commitment[9] (also known as trapdoor commitment [22]) a of b together with two ciphertexts C_0 and C_1 (with an IND-CCA encryption scheme),

[8] As pointed out by an anonymous reviewer, if ν_m is linear in $\log \mathfrak{K}$, this assumption is not necessary, as any extractor can be inversed by evaluating it on $2^{\nu_m}\mathfrak{K}$ randomly chosen inputs, similarly to what Explain does in the construction of Section 3.2.

[9] For the resulting commitment scheme to not require erasures, we suppose that it is not only possible to generate the opening data of a simulated commitment for any message, but also the corresponding random coins used by C.Com. Please note that we do not require the opening data to be the random coins, to provide more efficient construction, as the one in [1] using the Haralambiev commitment scheme TC4 [26] (see details in the sequel).

such that C_b contains a valid opening d_b of the commitment a for b, while C_{1-b} is sampled obliviously.

To extract some commitment C, it is sufficient to know the decryption key of the underlying IND-CCA encryption scheme and check whether C_0 or C_1 contains a valid opening d_0 or d_1 of a for 0 or 1. To simulate a commitment C, it is sufficient to know a trapdoor enabling to construct a commitment a and two valid openings d_0 and d_1 for both 0 and 1.

The robustness property basically comes from the fact the adversary cannot generate a commitment a and two valid openings d_0 and d_1, without breaking the binding property of the commitment a. Therefore, any commitment C generated by a polynomial-time adversary is perfectly binding.

However, for the resulting commitment to be compatible with SPHF, the underlying primitives (equivocable commitment and IND-CCA encryption scheme) have to be algebraic. In [3], Abdalla et al. propose to use the Pedersen commitment [31], as the equivocable commitment, together with the Cramer-Shoup [19] encryption scheme. Unfortunately, as the openings of the Pedersen commitments are scalars, they have to be encrypted bit-by-bit for the resulting commitment to be SPHF-friendly. This makes the commitment size of one bit to be quadratic in the security parameter (or the commitment to contain a linear number of group elements). This issue was solved in [1] by replacing the Pedersen commitment, by the Haralambiev commitment TC4 [26], for which the opening is a group element. However, this was at the expense on relying on bilinear groups (and SXDH) instead of plain DDH.

More precisely, the Haralambiev commitment of a bit b consists in a group element $a = g^{r_b} T^b$, with r_b a random scalar, and g, T two public generators of a cyclic group $\mathbb{G}$ of prime order p. The opening of a is $d_b = \hat{h}^{r_b}$ with $\hat{h}$ another generator of $\mathbb{G}$. This can be check using a pairing as follows: $e(a/T^b, \hat{h}) \stackrel{?}{=} e(g, d_b)$.

Pairings are only used to check the validity of an opening, and are only required in the security proof, as the committer needs to reveal r_b anyway (as it is part of his random tape), and r_b is sufficient to check the validity of the opening information d_b of a without pairing.

In our new scheme, we replace the need of a pairing by adding a 2-universal hash [20]. A 2-universal hash proof system can be seen as a designated-verifier one-time-simulation-sound zero-knowledge proof, which basically means that *i*) it can only be checked by the simulator which generated the CRS, *ii*) the simulator can generate fake or simulated proof for false statement, *iii*) and the adversary cannot generate proof for false statement even if it sees one fake proof. Finally, the Cramer-Shoup (IND-CCA) encryption scheme can be replaced by the ElGamal encryption scheme, as the 2-universal hash provides a form of non-malleability which is sufficient for our purpose[10]. As the construction is no longer black-box, new ideas are required in the proof of security of the scheme.

Our New Scheme. Our new scheme is formally described and proven in the full version [2].

[10] Actually, a Cramer-Shoup ciphertext basically consists in an ElGamal ciphertext plus a Diffie-Hellman element and a proof that everything is well-formed.

Basically, the setup $\mathsf{C.SetupT}(1^{\mathfrak{K}})$ generates a cyclic group $\mathbb{G}$ of order p, together with four generators $g, h = g^x, \hat{h} = g^{\hat{x}}, T = g^t$, a tuple $(\alpha, \beta, \gamma, \alpha', \beta', \gamma') \leftarrow \mathbb{Z}_p^6$, and H is a random collision-resistant hash function from some family $\mathcal{H}$. It then computes the tuple $(c = g^{\alpha}\hat{h}^{\gamma}, d = g^{\beta}h^{\gamma}, c' = g^{\alpha'}\hat{h}^{\gamma'}, d' = g^{\beta'}h^{\gamma'})$. The CRS crs is set as $(g, h, \hat{h}, H, c, d, c', d', T)$ and the trapdoor τ is the tuple $(\alpha, \alpha', \beta, \beta', \gamma, \gamma')$ (a.k.a., extraction trapdoor) together with t (a.k.a., equivocation trapdoor) and $(x, \hat{x})$ (only used in the EPHF).

To commit a vector of bits $\boldsymbol{M} = (M_i)_i \in \{0,1\}^m$ under a label ℓ, for $i = 1, \ldots, m$, we choose two random scalars $r_{i,M_i}, s_{i,M_i} \xleftarrow{\$} \mathbb{Z}_p$ and set

$$
\begin{array}{lll}
u_{i,M_i} = g^{s_{i,M_i}} & v_{i,M_i} = h^{s_{i,M_i}}\hat{h}^{r_{i,M_i}} & w_{i,M_i} = (c^{r_{i,M_i}} \cdot d^{s_{i,M_i}}) \cdot (c'^{r_{i,M_i}} d'^{s_{i,M_i}})^{\xi} \\
u_{i,\overline{M_i}} \xleftarrow{\$} \mathbb{G} & v_{i,\overline{M_i}} \xleftarrow{\$} \mathbb{G} & w_{i,\overline{M_i}} \xleftarrow{\$} \mathbb{G},
\end{array}
$$

together with $a_i \leftarrow g^{r_{i,M_i}} T^{M_i}$, where $\xi = H(\ell, (a_i, (u_{i,b}, v_{i,b})_b)_i)$. The commitment is then $C = (a_i, (u_{i,b}, v_{i,b}, w_{i,b})_b)_i \in \mathbb{G}^{8m}$, while the opening information is the $2m$-tuple $\delta = (r_{i,M_i}, s_{i,M_i})_i \in \mathbb{Z}_p^{2m}$.

The pair (u_{i,M_i}, v_{i,M_i}) is the ElGamal encryption of the opening $d_{i,M_i} = \hat{h}^{r_{i,M_i}}$ of the equivocable commitment a_i, while w_{i,M_i} is the 2-universal hash proving that $\log_g a_i/T^{M_i}$, the discrete logarithm in base g of a_i (i.e., r_{i,M_i} when generated honestly), is equal to the discrete logarithm in base $\hat{h}$ of the plaintext d_{i,M_i}.

The equivocation trapdoor t enables to open a_i to both 0 and 1, and so enables simulating commitments, while the equivocation trapdoor $(\alpha, \alpha', \beta, \beta', \gamma, \gamma')$ is the hashing key for the 2-universal hash proof system, i.e., enables to check the validity of the proof w_{i,M_i} as follows: $w_{i,b} \stackrel{?}{=} (a_i/T^b)^{\alpha+\xi\alpha'} \cdot u_{i,b}^{\beta+\xi\beta'} \cdot v_{i,b}^{\gamma+\xi\gamma'}$.

5.2 Complexity and Comparison

Table 1 compares our new schemes with existing non-interactive UC-secure commitments with a single global CRS. Since in most cryptographic schemes relying on SPHF-friendly commitments, such as the OT and PAKE schemes in [1], the most important metrics tend to be the size of the commitments and the size of the projection keys, Table 1 focuses on these parameters. In this context, as Table 1 shows, our new construction is the most efficient SPHF-friendly commitment scheme (even for $\mathsf{KV\text{-}SPHF}$, since group elements in $\mathbb{G}_2$ are larger than elements in $\mathbb{G}_1$) resulting in the most efficient OT and PAKE schemes so far (adaptively secure, assuming reliable erasures, under any assumption, with a single global CRS). In addition, since the new commitment scheme is secure under plain DDH, it allows for the construction of the first one-round PAKE (adaptively secure, assuming reliable erasures) under plain DDH, since the scheme of Abdalla, Chevalier, and Pointcheval [3] does not support $\mathsf{KV\text{-}SPHF}$ (which is required for one-round PAKE construction [1]).

Table 1. Comparison with existing non-interactive UC-secure commitments with a single global CRS

	SPHF-Friendly	W/o Erasure	Assumption	C size	hp size	KV / GL SPHF
[13]		✓[a]	DDH	$9m \times \mathbb{G}$	$2m \times \mathbb{Z}_p$	–
[3][b]	✓	✓[a]	DDH	$(m + 16m\mathfrak{K}) \times \mathbb{G}$	$2m\mathfrak{K} \times \mathbb{Z}_p$	– / $(3m + 2) \times \mathbb{G} + (\mathbb{Z}_p)$[a]
[23][c] 1			DLin	$5 \times \mathbb{G}$	$16 \times \mathbb{G}$	–
[23][c] 2			DLin	$37 \times \mathbb{G}$	$3 \times \mathbb{G}$	–
[1]	✓	✓[a]	SXDH	$8m \times \mathbb{G}_1 + m \times \mathbb{G}_2$	$m \times \mathbb{Z}_p$	$2m \times \mathbb{G}_1$ / $\mathbb{G}_1 + (\mathbb{Z}_p)$[a]
§ 5.1	✓	✓	DDH	$7m \times \mathbb{G}$	$2m \times \mathbb{Z}_p$	$4m \times \mathbb{G}$ / $2 \times \mathbb{G} + (\mathbb{Z}_p)$[d]

m = bit-length of the committed value, $\mathfrak{K}$ = security parameter;
we suppose there exists a family of efficient collision-resistant hash functions (for efficiency reason, since DDH implies the existence of such families).

[a] commitments in [1,3,13] were not described as without erasures, but slight variants of them are, as explained in Section 2.

[b] we consider a slight variant without one-time signature but using labels and multi-Cramer-Shoup ciphertexts, as in the scheme in [1] (which makes the scheme more efficient). The size of the projection key is computed using the most efficient methods in [1];

[c] we use a Pedersen commitment as a chameleon hash and multi-Cramer-Shoup ciphertexts to commit to multiple bits in a non-malleable way (see [1] for a description of the multi-Cramer-Shoup encryption scheme). We do not know a SPHF on such commitment, since the opening information of a Pedersen commitment is a scalar;

[d] this $\mathbb{Z}_p$ element may only be $\mathfrak{K}$-bit long and is useless when $m = 1$.

6 Adaptive OT without Erasures

As explained in [24], one can transform any semi-adaptive protocols into adaptive ones by sending all the flows through secure channels. Such secure channels can be constructed using non-committing encryption (NCE) [5,12,17,21]. However, even the most efficient instantiation of NCE [17] requires $8\nu_{\mathsf{NCE}}\mathfrak{K}$ group elements to send ν_{NCE} bits securely, with ElGamal encryption scheme as (trapdoor) simulatable encryption scheme. If ν_{NCE} is $\Omega(\mathfrak{K})$, this can be reduced to about $320\nu_{\mathsf{NCE}}$ group elements.

In this section, we propose several adaptive versions of our semi-adaptive OT and PAKE protocols. Some are optimized for the number of rounds, while others are optimized for the communication complexity. In each case, at least one of our new protocols performs better than existing protocols. Only the high-level intuition is given in this section. Details are given in the full version [2].

First Scheme. A first efficient way to construct a bit (i.e., $\nu_m = 1$) 1-out-of-2 OT secure against adaptive adversary consists in applying the generic transformation of Garay *et al.* [24] to our semi-adaptive OT.

This transformation uses the notion of ℓ-somewhat non-committing encryption scheme. This scheme enables to send securely long messages, but which restricts the non-committing property to the following: it is only possible to produce random coins corresponding to ℓ different messages. Then, to get an adaptive OT from a semi-adaptive OT, it is sufficient to execute the protocol in a 8-somewhat non-committing channel. Indeed, the simulator can send via this channel 8

versions of the transcript of the protocol: depending on which user gets corrupted first and on which were their inputs and outputs. There are two choices of inputs for the sender (the two index queries) and two outputs (the message m_s), hence four choices in total; and there are four choices of inputs for the receiver (the two messages m_0 and m_1). Hence the need for 8 versions.

In [24], the authors also show how to extend their bit OT based on the DDH version of the static OT of Peikert *et al.* [32] to string OT by repeating the protocol in parallel and adding an equivocable commitment to the index and a zero-knowledge proof to ensure that the sender always uses the same index s. Actually, for both of our instantiations and for the one in [24], we can do better, just by using the same commitment C to s (in our case) or the same CRS (the one obtained by coin tossing) and the same public key of the dual encryption system (in their case). This enables us to get rid off the additional zero-knowledge proof and can also be applied to the QR instantiation in [24]. In addition, the commitment C to s (in our case) or the CRS and the public key (in their case) only needs to be sent in the first somewhat non-committing channel.

Furthermore, if the original semi-adaptive OT is a 1-out-of-k OT (with $k = 2^{\nu_k}$), then we just need to use a 2^{k+1}-somewhat NCE instead of a 8-somewhat NCE encrypt (because there are 2^k possible inputs for the sender, and k possible inputs and 2 possible outputs for the receiver, so $2^k + 2k \leq 2^{k+1}$ possible versions for the transcript).

Finally, the combination of all the above remarks yields a ν_m-bit string 1-out-of-k OT scheme requiring only ν_m 2^{k+1}-somewhat NCE channels, and so only $\nu_m(k+1)$ bits sent through NCE.

Second Scheme. Our second scheme can be significantly more efficient than our first one, for several parameter choices. Essentially, it consists in using NCE channels to send $k\nu_m$ random bits to mask the messages (in case the sender is corrupted first) and $2\nu_k$ random bits to enable the simulator to make the commitment binding to the index s (in case the receiver gets corrupted first). Methods used for this second part are specific to our new SPHF-friendly commitment scheme, but can also be applied to the commitment scheme in [1].

The scheme is depicted in Fig. 3. Our 1-out-of-k OT protocol uses a NCE channel of $\nu_{\mathsf{NCE}} = 2\nu_k + k\nu_m$ bits, where $k = 2^{\nu_k}$, for ν_m-bit strings. This channel is used to send a random value R. The last $k\nu_m$ bits of R are k ν_m-bit values $R_1, \ldots, R_k$. These values are used to mask the messages $m_1, \ldots, m_k$ sent by the sender, to be able to reveal the correct messages, in case of corruption of the sender (when both the sender and the receiver were honest at the beginning, and so when $m_1, \ldots, m_k$ were completely unknown to the simulator).

The first $2\nu_k$ bits of R are used to make the commitment C (which is normally simulated when the receiver is honest) perfectly binding to the revealed index s, in case of corruption of the receiver (when both the sender and the receiver were honest at the beginning, and so when s was completely unknown to the simulator). More precisely, they are used to partially hide the last component of commitments: the $w_{i,b}$; the bit R_{2i+b-1} indicates whether $w_{i,b}$ has to be inverted or not before use. The full security proof is given in the full version [2].

CRS: $\mathsf{crs} \xleftarrow{\$} \mathsf{C.Setup}(1^\mathfrak{K})$ and $\mathsf{NCE.param} \xleftarrow{\$} \mathsf{NCE.Setup}(1^\mathfrak{K})$.

Pre-flow:

1. P_i generates $(\mathsf{ek}, \mathsf{dk}) \xleftarrow{\$} \mathsf{NCE.KG}(\mathsf{NCE.param})$
2. P_i sends ek to P_j

Index query on s:

1. P_j chooses a random $R \xleftarrow{\$} \{0,1\}^{\nu_{\mathsf{NCE}}}$ and computes $\chi \xleftarrow{\$} \mathsf{NCE.Enc}(\mathsf{ek}, R)$
2. P_j computes $(C = ((e_{I,b}, u_{I,b}, v_{I,b}, w_{I,b})_{I,b}), \delta) \xleftarrow{\$} \mathsf{C.Com}^\ell(s)$ with $\ell = (\mathsf{sid}, \mathsf{ssid}, P_i, P_j)$
3. P_j sets $w'_{I,b} = w_{I,b}$ if $R_{2I+b-1} = 0$ and $w'_{I,b} = 1/w_{I,b}$ otherwise, for $I = 1, \ldots, \nu_k$ and $b = 0, 1$;
   and sets $C' = ((e_{I,b}, u_{I,b}, v_{I,b}, w'_{I,b})_{I,b})$
4. P_j sends χ and C' to P_i

Database input $(m_1, \ldots, m_k)$:

1. P_i computes $R \xleftarrow{\$} \mathsf{NCE.Dec}(\mathsf{dk}, \chi)$
2. P_i sets $w_{I,b} = w'_{I,b}$ if $R_{2I+b-1} = 0$ and $w_{I,b} = 1/w'_{I,b}$ otherwise, for $I = 1, \ldots, \nu_k$ and $b = 0, 1$;
   and sets $C = ((e_{I,b}, u_{I,b}, v_{I,b}, w_{I,b})_{I,b})$
3. P_i sets $(R_t)_t$ to the last $k\nu_m$ bits of R (R_t being a ν_m-bit variable)
4. P_i computes $\mathsf{hk}_t \xleftarrow{\$} \mathsf{HashKG}(\mathsf{crs})$, $\mathsf{hp}_t \leftarrow \mathsf{ProjKG}(\mathsf{hk}_t, \mathsf{crs}, (\ell, C))$,
   $K_t \leftarrow \mathsf{Hash}(\mathsf{hk}_t, (\mathsf{crs}, t), (\ell, C))$, and $M_t \leftarrow R_t \mathsf{\ xor\ } K_t \mathsf{\ xor\ } m_t$, for $t = 1, \ldots, k$
5. P_i sends $(\mathsf{hp}_t, M_t)_{t=1,\ldots,k}$

Data recovery:
Upon receiving $(\mathsf{hp}_t, M_t)_{t=1,\ldots,k}$, P_j computes $K_s \leftarrow \mathsf{ProjHash}(\mathsf{hp}_s, (\mathsf{crs}, s), (\ell, C), \delta)$ and gets $m_s \leftarrow R_s \mathsf{\ xor\ } K_s \mathsf{\ xor\ } M_s$, with $(R_t)_t$ the last $k\nu_m$ bits of R.

Fig. 3. UC-Secure 1-out-of-k OT from our SPHF-Friendly Commitment for Adaptive Adversaries

Remark 1. Though the new protocol uses our new commitment scheme, it could alternatively use the commitment scheme in [1], by just replacing $w_{i,b}$ by the last part of the Cramer-Shoup ciphertexts in these schemes. The proof would be very similar. This replacement may yield a more efficient scheme (under SXDH however) when ν_m is large, since the projection key in [1] is shorter than for our scheme and multiple projection keys need to be sent due to the generic transformation of SPHF to EPH.

Comparison. In Table 2, we give a detailed comparison of our OT schemes with the DDH-based OT in [24]. The QR-based one in less efficient anyway. We see that, for every parameters ν_m and k, at least one of our two schemes (if not both) is the most efficient scheme regarding both the number of rounds and the communication complexity.

Table 2. Comparison of 1-out-of-k OT UC-Secure against Adaptive Adversaries, without Erasures, with $k = 2^{\nu_k}$

	Rnd[a]	Communication Complexity
[24]	≥ 8	$(k+1)\cdot\nu_m \times \text{NCE} + 3\cdot(2^k+2k)\cdot\nu_m \times \mathbb{G}$ $+(2^k+2k)\cdot\big(\text{com}(4\times\mathbb{G}) + 2\nu_k \times \mathbb{G} + \nu_k \times \text{ZK} + 4\nu_m\nu_k \times \mathbb{G}\big)$
1st	4	$(k+1)\cdot\nu_m \times \text{NCE} + 3\cdot(2^k+2k)\cdot\nu_m \times \mathbb{G}$ $+(2^k+2k)\cdot\big(7\nu_k \times \mathbb{G} + \nu_m\cdot(2\times\mathbb{G} + (\mathbb{Z}_p)^{\text{b}} + 2)\big)$
2nd	3	$(k\nu_m + 2\nu_k) \times \text{NCE} + 7\nu_k \times \mathbb{G} + \nu_m\cdot\big(2\times\mathbb{G} + (\mathbb{Z}_p)^{\text{b}} + 2\big)$

[a] number of rounds
[b] this element in $\mathbb{Z}_p$ is not required when $\nu_m = \nu_k = 1$
Legend:
- ZK: zero-knowledge proof used in [24].
- com(x): communication complexity of a UC-commitment scheme for x bits. This is used to generate the CRS for the scheme in [32]. If this commitment is interactive, this increases the number of required rounds.
- $x \times$ NCE: x bits sent by non-committing encryption scheme.

The exact communication complexity cost depends on the exact instantiation of NCE. But in all cases, at least one of our schemes outperforms existing schemes both in terms of number of bits sent via a NCE channel, and in terms of auxiliary elements (elements which are not directly used by the NCE scheme). In addition, our second scheme always uses the smallest number of auxiliary elements; and it requires $k\nu_m + 2\nu_k$ bits to be sent via a NCE channel, which is not worse than the $(k+1)\nu_m$ bits required by our first scheme, as long as $\nu_m \geq 2\nu_k$.

Here are some details on the comparison. We suppose we use the NCE scheme proposed in [17] (which is 2-round) and the ElGamal encryption as simulation encryption scheme for the NCE scheme and the somewhat NCE construction (which also requires a simulation encryption scheme). So all our schemes are secure under DDH (plus existence of collision resistant hash functions and symmetric key encryption, but only for efficiency, since DDH implies that also).

In the comparison, we extend the schemes in [24] to 1-out-of-k schemes using the method explained in Section 6 and the 1-out-of-k version of the schemes of Peikert *et al.* [32], which consists in doing ν_k schemes in parallel and secret sharing the messages (where $k = 2^{\nu_k}$).

To understand the costs in the table, recall that a 2^l-somewhat non-committing encryption scheme works as follows: one player sends a l-bit value I using a full NCE scheme (2 rounds) together with 2^l public keys all samples obviously except the I^{th} one, and then the other player sends 2^l ciphertexts samples obliviously except the I^{th} one which contains a symmetric key K. Then to send any message through this 2^l-somewhat NCE channel, a player just sends 8 messages all random except the I^{th} one which is an encryption of the actual message under K. This means that if the original semi-adaptive protocol is x-round, then the protocol resulting from the transformation of Garay *et al.*, is $(x+2)$-round; and this costs a total of $3\cdot 2^l$ group elements, in addition of the group elements for the l-bit non-committing encryption.

Acknowledgments

This work was partially done while the second author was student at ENS, CNRS, INRIA, and PSL Research University, Paris, France. The first author and the third author were supported by the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013 Grant Agreement no. 339563 – CryptoCloud). The second author was supported in part by the CFM Foundation and by the Defense Advanced Research Projects Agency (DARPA) and Army Research Office (ARO) under Contract No. W911NF-15-C-0236.

References

1. Abdalla, M., Benhamouda, F., Blazy, O., Chevalier, C., Pointcheval, D.: SPHF-friendly non-interactive commitments. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part I. LNCS, vol. 8269, pp. 214–234. Springer, Heidelberg (Dec 2013)
2. Abdalla, M., Benhamouda, F., Pointcheval, D.: Removing erasures with explainable hash proof systems. Cryptology ePrint Archive, Report 2014/125 (2014), http://eprint.iacr.org/2014/125
3. 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 (Aug 2009)
4. 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 (Aug 2005)
5. Beaver, D.: Commodity-based cryptography (extended abstract). In: 29th ACM STOC. pp. 446–455. ACM Press (May 1997)
6. Beaver, D.: Plug and play encryption. In: Kaliski Jr., B.S. (ed.) CRYPTO'97. LNCS, vol. 1294, pp. 75–89. Springer, Heidelberg (Aug 1997)
7. 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 2013, Part I. LNCS, vol. 8042, pp. 449–475. Springer, Heidelberg (Aug 2013)
8. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. Cryptology ePrint Archive, Report 2000/067 (2000), http://eprint.iacr.org/2000/067
9. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: 42nd FOCS. pp. 136–145. IEEE Computer Society Press (Oct 2001)
10. Canetti, R., Dachman-Soled, D., Vaikuntanathan, V., Wee, H.: Efficient password authenticated key exchange via oblivious transfer. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 449–466. Springer, Heidelberg (May 2012)
11. Canetti, R., Dodis, Y., Pass, R., Walfish, S.: Universally composable security with global setup. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 61–85. Springer, Heidelberg (Feb 2007)
12. Canetti, R., Feige, U., Goldreich, O., Naor, M.: Adaptively secure multi-party computation. In: 28th ACM STOC. pp. 639–648. ACM Press (May 1996)
13. Canetti, R., Fischlin, M.: Universally composable commitments. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 19–40. Springer, Heidelberg (Aug 2001)

14. 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, Heidelberg (May 2005)
15. 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)
16. Chevalier, C., Fouque, P.A., Pointcheval, D., Zimmer, S.: Optimal randomness extraction from a Diffie-Hellman element. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 572–589. Springer, Heidelberg (Apr 2009)
17. Choi, S.G., Dachman-Soled, D., Malkin, T., Wee, H.: Improved non-committing encryption with applications to adaptively secure protocols. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 287–302. Springer, Heidelberg (Dec 2009)
18. 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.) PKC 2013. LNCS, vol. 7778, pp. 73–88. Springer, Heidelberg (Feb / Mar 2013)
19. 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, Heidelberg (Aug 1998)
20. 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 (Apr / May 2002)
21. Damgård, I., Nielsen, J.B.: Improved non-committing encryption schemes based on a general complexity assumption. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 432–450. Springer, Heidelberg (Aug 2000)
22. Feige, U., Shamir, A.: Witness indistinguishable and witness hiding protocols. In: 22nd ACM STOC. pp. 416–426. ACM Press (May 1990)
23. 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, Heidelberg (Dec 2011)
24. 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 (Aug 2009)
25. Gennaro, R., Lindell, Y.: A framework for password-based authenticated key exchange. ACM Transactions on Information and System Security 9(2), 181–234 (2006)
26. Haralambiev, K.: Efficient Cryptographic Primitives for Non-Interactive Zero-Knowledge Proofs and Applications. Ph.D. thesis, New York University (2011)
27. Hemenway, B., Ostrovsky, R., Rosen, A.: Non-committing encryption from ϕ-hiding. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015, Part I. LNCS, vol. 9014, pp. 591–608. Springer, Heidelberg (Mar 2015)
28. Jutla, C.S., Roy, A.: Dual-system simulation-soundness with applications to UC-PAKE and more. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015, Part I. LNCS, vol. 9452, pp. 630–655. Springer, Heidelberg (Nov / Dec 2015)
29. 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 (Mar 2011)
30. Montenegro, R., Tetali, P.: How long does it take to catch a wild kangaroo? In: Mitzenmacher, M. (ed.) 41st ACM STOC. pp. 553–560. ACM Press (May / Jun 2009)

31. 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, Heidelberg (Aug 1992)
32. 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 (Aug 2008)
33. Wee, H.: Dual projective hashing and its applications - lossy trapdoor functions and more. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 246–262. Springer, Heidelberg (Apr 2012)