

Adaptively UC-Secure Oblivious Transfer from Group Actions

Olivier Blazy^{1,2} , Céline Chevalier^{2,3} , Saqib Kakvi⁴  and
Lola-Baie Mallordy^{1,2} 

¹ LIX, École Polytechnique, Institut Polytechnique de Paris, France

² Inria, France

³ CRED, Université Panthéon-Assas, DIENS, École normale supérieure, PSL University, CNRS, France

⁴ Royal Holloway University of London, United Kingdom

Abstract. Oblivious Transfer (OT) is a fundamental building block for secure computation, in which a sender holds two messages and a receiver learns exactly one of them, without revealing which one was chosen and without learning the other. While generic constructions for Universally Composable (UC) OT exist, instantiating them from post-quantum assumptions remains challenging, particularly under adaptive corruption.

In this work, we present the first adaptively UC-secure OT protocol based on Group Actions, specifically assuming weakly pseudorandom Effective Group Actions (EGA) like CSIDH. We adhere to the framework of [BC15], which derives an adaptively UC-secure OT protocol from SPHF-friendly extractable and equivocal (E^2) commitment schemes.

The central challenge in instantiating this construction from group actions is the lack of SPHF-friendly IND-CCA encryption. Classical CCA transformations such as Fujisaki-Okamoto destroy the algebraic structure required for building SPHFs over ciphertexts, while known group action-based schemes offer no native mechanism to prove ciphertext validity. To overcome this obstacle, we construct from group actions an IND-CCA encryption scheme with enough algebraic structure to efficiently recognize valid ciphertexts.

Our contributions are threefold: (1) a verifiable chameleon hash function with perfect soundness in the standard model, (2) an SPHF-friendly, publicly verifiable IND-CCA encryption scheme in the random oracle model, and (3) we combine these to build an SPHF-friendly E^2 commitment scheme and derive a fully adaptively secure OT protocol in the UC framework. This is the first construction of its kind under group action assumptions, enabling post-quantum secure OT with adaptive security guarantees. These building blocks may also be of independent interest *e.g.*, to obtain digital signatures, non-malleable commitment schemes, or verifiable encryption.

1 Introduction

Oblivious Transfer (OT) was first introduced by Rabin in 1981 [Rab81] as a cryptographic primitive allowing a sender to transmit a message such that the receiver obtains it with probability $1/2$, while the sender remains unaware of the outcome. A more commonly used variant, known as 1-out-of-2 OT, was introduced by Even et al. [EGL85] to build MPC protocols. It was later shown by Crépeau that Rabin’s original protocol is equivalent to

E-mail: olivier.blazy@polytechnique.edu (Olivier Blazy), celine.chevalier@ens.fr (Céline Chevalier), Saqib.Kakvi@rhul.ac.uk (Saqib Kakvi), lola.mallordy@polytechnique.edu (Lola-Baie Mallordy)



the 1-out-of-2 form of OT [Cr 87]. Oblivious transfer has been widely studied, serving as a versatile primitive with many applications beyond MPC [KO97, JZ22, SS05, KK25]. Several generic OT constructions have been proposed, which broadly fall in two categories: those based on public key encryption (PKE) schemes with specific properties [BGJ⁺18, FMV19, DGH⁺20, MR19], and those based on Smooth Projective Hash Functions (SPHF) [HK12, ABB⁺13, BC15]. We focus on the latter approach, since there are no known group action-based PKE schemes achieving the specific properties required for the former.

The generic approach to adaptively Universally Composable (UC)-secure OT protocols described in [ABB⁺13] relies on commitment schemes compatible with an SPHF. Their result was later refined by the work of [BC15], which shows how to build such a commitment scheme from a chameleon hash function paired with an SPHF-friendly CCA-secure encryption scheme.

As shown in [ACP09], an SPHF enables performing implicit decommitments where the committer does not really open its commitment, but rather convinces the receiver that it actually committed to the value it pretended to. These hash functions are defined such that their value can be computed in two different ways if the input belongs to a particular subset (the language). The hash value can be computed either using a private hashing key, or using a public projection key along with a private witness that proves membership to the language. Crucially, these functions exhibit *smoothness* (on inputs not in the language, outputs are perfectly indistinguishable from random) and *pseudorandomness* (even when inputs belong to the language, if the witness is unknown the output remains computationally indistinguishable).

To prove the generality of their framework, the authors of [BC15] provided three different instantiations from several assumptions: the Decisional Diffie-Hellman (DDH) assumption, Learning With Errors (LWE), and the DCR (Decisional Composite Residuosity) assumptions. However, adapting these constructions to the group action setting can be challenging, since it allows for fewer operations. We tackle the challenge of instantiating their construction with the recent new post-quantum construction: cryptographic group actions. Introduced by Alami et al. [ADFMP20], cryptographic group actions offer a generic framework for post quantum cryptography. They can be instantiated with any isogeny-based assumption with the right properties, such as CSIDH [CLM⁺18]. By contrast, its predecessor SIDH [JDF11], not based on group actions, was shown to be insecure [CD23a, MMP⁺23, Rob23]. Once it was broken, the scheme had to be reconstructed from scratch. This highlights the advantage of the group action framework: if a specific instantiation is broken, the group action can be replaced without discarding the entire scheme.

To achieve UC-security, the work of [BC15] relies on both *equivocal* and *extractable* (E^2) SPHF-friendly commitment schemes. A commitment scheme is said to be extractable if there exists a simulated setup (indistinguishable from the honest one) with a trapdoor that allows the simulator to recover the committed value m . Similarly, it is said to be equivocal if there exists a simulated setup with a trapdoor that allows the simulator, on input any commitment c and any message m , to output an opening value δ that opens c to m . These two properties are very useful to achieve UC-security. The work of [BC15] allows building such an E^2 commitment scheme using an SPHF-friendly IND-CCA encryption scheme and a chameleon hash function. However, building an SPHF-friendly IND-CCA encryption is notably difficult: one needs to be able to verify in an efficient way if a word is a valid encryption of some plaintext. Generic transformations from IND-CPA security to IND-CCA security such as the Fujisaki-Okamoto transform [FO99] lose too much algebraic structure to build an SPHF over the language of such ciphertexts (since one would need to verify hash computations, which is cumbersome). The same issue is met if one tries to use generic IND-CCA constructions as the Cramer-Shoup construction [CS02].

Our work thus addresses the following central question:

Can one construct an SPHF-friendly E^2 commitment scheme from group actions?

We answer affirmatively by presenting the first instantiations of all the necessary components from effective group actions (EGA): an SPHF-friendly publicly verifiable IND-CCA encryption scheme, and a chameleon hash. We believe our work goes beyond oblivious transfer: our EGA-based primitives are of independent interest and can be used in other settings. For example, chameleon hash functions are useful primitives that can provide several constructions such as digital signatures [BKGP15], two-tier signature schemes [BS07], on-line/off-line signatures [ST01], chameleon signatures [KR00], sanitizable signatures [ACdMT05], etc. Our publicly verifiable encryption scheme can be employed in group signatures and signature sharing [CD99], and our E^2 commitment scheme enables non-malleable and universally composable commitments schemes [Cre02].

1.1 Contributions

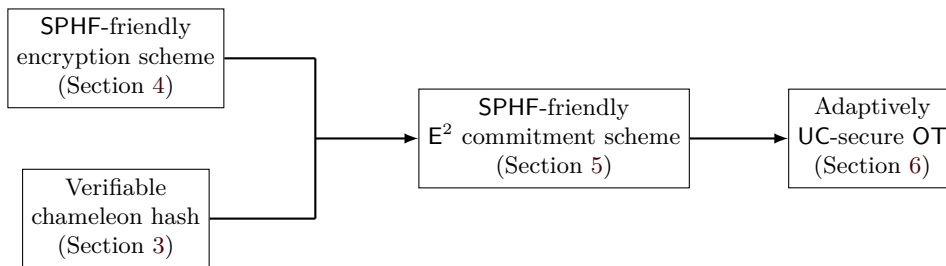


Figure 1: The generic construction for the adaptive OT protocol from [BC15] (and the corresponding sections in which we instantiate each block).

We present three novel constructions from cryptographic group actions:

1. A collision resistant, perfectly sound verifiable chameleon hash function (in the standard model).
2. An SPHF-friendly publicly verifiable IND-CCA encryption scheme $\mathcal{E}$ in the random oracle model (ROM).
3. An SPHF-friendly E^2 commitment scheme.

All of these resulting in an adaptively UC-secure OT protocol solely relying on group action assumptions (as described in Figure 1.1). Our scheme is 3-round, where the first round is a pre-flow in which the server updates the chameleon hash keys. This enables us to make each session independent, and makes the scheme resistant against corruption. Our first contribution is a verifiable chameleon hash function from effective group actions. It provides *perfect soundness*, which means that a hash value that was not honestly computed cannot pass as valid. A chameleon hash function from group actions was recently proposed in [BFP⁺26], but it does not provide verifiability, thus we believe our construction is the first one of its kind.

Our second contribution is a *publicly verifiable* IND-CCA encryption scheme $\mathcal{E}$ from group actions. To follow the generic construction from [BC15], we need an IND-CCA encryption scheme with enough algebraic structure to allow us to build an SPHF over the language of valid ciphertexts. Usual transformations used to achieve IND-CCA security such as the Fujisaki-Okamoto transform [FO99] lose too much of the algebraic structure of the underlying encryption to allow one to verify if a word constitutes a valid ciphertext or not. Generic constructions for IND-CCA encryption schemes such as the Cramer-Shoup construction [CS02] meet the same issue. To circumvent this problem, we propose a publicly verifiable IND-CCA encryption scheme. The ciphertext contains a zero-knowledge proof (ZKP) that it was honestly computed. This allows to lighten the requirement on the

language to be checked by the SPHF as the "valid ciphertext" part is transferred to the ZKP. Since we use a Fiat-Shamir transform to get a non-interactive proof, the IND-CCA security is achieved in the ROM.

Next, we construct our E^2 commitment scheme, and prove its strong simulation indistinguishability (meaning that simulated commitments are indistinguishable from honest ones, even with oracle access to extraction and simulation). We also prove the smoothness of the projective hash function that we define over the languages of valid commitments. Finally, all of these blocks allow us to instantiate the construction from [BC15] and obtain an adaptively UC-secure OT protocol. We provide the complexity of our protocol in terms of the number of group action evaluations and communications.

1.2 Related Work

We review previous generic OT constructions, as well as isogeny-based OT protocols. We summarize the comparisons with our result in Table 1. As pointed out by Canetti et al. [CKL03], it is impossible to achieve a UC-secure OT scheme without any trusted setup, so we compare ourselves with other protocols in the trusted setup model.

Generic OT constructions. Many works have provided generic constructions for OT protocols [ABB⁺13, BC15, BGJ⁺18, FMV19, DGH⁺20, MR19]. However, several of these generic constructions rely on Public Key Encryption (PKE) schemes with specific properties [BGJ⁺18, FMV19, DGH⁺20, MR19], while there are no isogeny-based PKE schemes known to satisfy these additional requirements. For example, the work of [MR19] requires to be able to hash into a curve in the family of supersingular elliptic curves, which is not known so far [MM22]. Similarly, the work of [BGJ⁺18] requires a PKE scheme with a dense public-key space, which is not known to exist from isogeny-based assumptions. The generic construction from [ABB⁺13] builds a UC-secure OT scheme provided that there exists an SPHF-friendly commitment scheme. This result was strengthened in [BC15], which constructs an SPHF-friendly commitment scheme from a chameleon hash function and an SPHF-friendly IND-CCA encryption scheme.

The authors of [DGH⁺20] gave a generic construction for a 2-round UC-secure OT in the Common Reference String (CRS) model from protocols satisfying a very mild form of security, known as elementary OT. The construction of [BLPV18] allows building an adaptively secure two-round OT protocol from any two-round statically secure OT protocol with certain oblivious sampleability properties. However, there are no known isogeny-based OT that satisfy these oblivious sampleability properties.

Isogeny-based OT protocols. Numerous works have realized OT protocols from the SIDH assumption and its variants [BNOB18, Vit18, dSGOPS20, BKW20]. However, many of these schemes [BNOB18, Vit18, dSGOPS20] were broken due to the recent attacks on SIDH [CD23b, MM22]. The work of [ADFMP20] gives a two-message statistically sender-private (SSP) oblivious transfer protocol from any weak pseudorandom (restricted) effective group action. It provides security in the UC framework [Can01]. They achieved semantic security against a computationally bounded sender and an unbounded receiver, under the decisional CSIDH assumption. Two OT protocols based on the reciprocal CSIDH assumption were proposed in [LGDdSG21], however this only considers static corruptions and is not round optimal: the UC-secure version is a 3-round protocol while the simulation secure one requires 4 rounds. Similarly, the author of [BKW20] proposed two constructions: the first one based on SIDH was broken in its original form by an earlier attack proposed in [BKM⁺21]. The second one based on decisional CSIDH was not explicitly described but follows implicitly from their oblivious PRF, and is round optimal but only secure against static corruptions.

The work of [AMPS21] builds on the generic construction of [DGH⁺20], to obtain an adaptively secure OT from isogenies that is round-optimal in the standard model. They do not provide the full construction of an adaptive OT, but only of a *semi-adaptive* (adaptive sender and static receiver, or the other way around) OT scheme. They refer to [BLPV18] for a generic transformation that obtains adaptive security from such a scheme. They start from a weaker version of OT with oblivious sampleability as the one considered in [BLPV18], to be able to instantiate it with isogeny-based cryptography. Then, they use bootstrapping on this weaker OT to get a stronger semi-adaptively secure OT protocol. Their work makes use of several garbled circuits and an equivocal commitment scheme, for which they point to Naor’s construction [Nao91] from one-way functions. Garbling schemes can become heavy in communication, as their sizes grow linearly with the number of the circuit AND gates. Since the authors do not describe directly their adaptive scheme but only refer to [BLPV18] for the generic transformation, they did not provide any cost analysis of their solution. Due to the use of the garbled circuits, we believe that this construction could be quite heavy in communications. In this work, we do not rely on garbling schemes and provide a complexity analysis of our solution, both in communication and computation costs.

Finally, the first round optimal OT protocol from the *computational* CSIDH assumption was designed in [BMM⁺23]. They proposed two constructions, in the setup model and in the plain model, but only tackle static adversaries.

Protocols	Assumption	Rounds	Adv. Model
[BMM ⁺ 23]	computational CSIDH	2	static
[LGDdSG21]-1	reciprocal CSIDH	3	static
[LGDdSG21]-2	reciprocal CSIDH	4	static
[AMPS21]	wPR EGA	2	adaptive
[ADFMP20]	wPR EGA	2	static
[BKW20]	decisional CSIDH	2	static
This work	wPR EGA	3	adaptive

Table 1: Comparison with other OT protocols in the trusted setup model. We write wPR for weakly pseudorandom and EGA for effective group action. All of these works are UC-secure except [LGDdSG21]-1 which is simulation-secure.

Outline. In Section 2, we provide some necessary cryptographic background. In Sections 3 and 4, we instantiate the chameleon hash function and IND-CCA encryption scheme from group actions. In Section 5, we define our E^2 commitment scheme and describe the SPHF over the language of valid commitments. Finally in Section 6.3, we recall the generic OT construction from [BC15], and then deduce the cost analysis in terms of group action evaluations for our instantiation of the adaptive OT protocol.

2 Preliminaries

Notations. Sampling an element x uniformly from a set S is denoted by $x \xleftarrow{\$} S$, and concatenation by $||$. By $\mathcal{A}$, we denote the adversary of a security experiment, which is a probabilistic polynomial time (PPT) algorithm. By $[n]$, we denote the set $\{1, \dots, n\}$. For any probabilistic algorithm $P(x)$, we denote by $P(x; r)$ the deterministic version of P with fixed randomness r .

2.1 Zero-knowledge proof of knowledge

A sigma protocol Σ is a three-move proof system for a relation R between a prover P and a verifier V , which are two PPT algorithms with access to some oracles and a shared state. The protocol Σ proceeds as follows:

- The prover, on input a state and a witness $(\text{st}, w) \in R$, sends a commitment com to the verifier V .
- The verifier sends a challenge chall to the prover.
- Given chall , the prover responds with resp to V .
- Finally, the verifier either accepts or rejects the proof transcript $(\text{st}, \text{com}, \text{chall}, \text{resp})$.

This can be made non-interactive using the Fiat-Shamir transform.

Definition 1 (Honest-Verifier Zero-Knowledge (HVZK)). We say Σ is honest-verifier zero-knowledge (HVZK) for the relation R if there exists a simulator $\mathcal{S}$ with access to a random oracle $\mathcal{O}$ such that, for any $(\text{st}, w) \in R$, any adversary $\mathcal{A}$ with polynomial access to $\mathcal{O}$ and any negligible ε , we have

$$\text{Adv}_{\text{ZK}}(\mathcal{A}) := |\Pr[1 \leftarrow \mathcal{A}^{\mathcal{O}}(P^{\mathcal{O}}(\text{st}, w, \text{chall}))] - \Pr[1 \leftarrow \mathcal{A}^{\mathcal{O}}(\mathcal{S}^{\mathcal{O}}(\text{st}, \text{chall}))]| \leq \varepsilon.$$

It is perfectly HVZK if the advantage is 0.

Definition 2 (Special Soundness). We say Σ has special soundness if there exists a polynomial-time extractor such that, given st and any two valid transcripts $(\text{st}, \text{com}, \text{chall}, \text{resp})$ and $(\text{st}, \text{com}, \text{chall}', \text{resp}')$ such that $\text{chall} \neq \text{chall}'$, outputs a witness w satisfying $(\text{st}, w) \in R$.

For a non-interactive zero-knowledge proof (NIZKP) system, we define the following property. The online extractability is a stronger property than the classical soundness, which requires the extractor to output a witness from such a non-interactive proof instantaneously, without having to rewind, fork or program the random oracle.

Definition 3 (Online Extractability [Fis05]). We say Σ has online extractability if for any adversary $\mathcal{A}$ that can produce a valid proof π for a state st with access to a random oracle $\mathcal{O}$, there exists a PPT online extractor that on input (st, π) and the list of $\mathcal{A}$'s queries to $\mathcal{O}$, outputs a witness w satisfying $(\text{st}, w) \in R$ with probability $1 - \varepsilon$ for a negligible ε .

2.2 Smooth Projective Hash Function

We recall the definition of a projective hash function (also known as hash proof systems), first introduced in [CS02]. A projective hashing family is a family of hash functions that can be evaluated in two ways: one can compute a hash for any point in $\mathcal{X}$ with the hashing key, whereas one can only compute a hash on the language $\mathcal{L} \subseteq \mathcal{X}$ using the projected key, which requires a witness that the hashed point belongs to $\mathcal{L}$. We consider a language $\mathcal{L}$ that is a *hard-partitioned* subset of $\mathcal{X}$ i.e., it is computationally hard to distinguish a random element drawn in $\mathcal{L}$ from a randomly chosen element in $\mathcal{X}$.

Definition 4 (Projective Hash Function). A Projective Hash Function $\Pi = (K, P, \mathcal{X}, \Gamma, \mathcal{L})$ over a language $\mathcal{L} \subset \mathcal{X}$ onto a set Γ is defined by the following algorithms:

- $\text{SPHF.Setup}(\lambda) \rightarrow \text{pp}$: takes as input the security parameter and outputs the global parameters. From now on, we assume that pp and $\mathcal{L}$ are implicit inputs.
- $\text{SPHF.KeyGen}(\text{pp}) \rightarrow \text{hk} \in K$: outputs a hashing key hk for the language $\mathcal{L}$.
- $\text{SPHF.ProjKeyGen}(\text{hk}) \rightarrow \text{hp} \in P$: derives the projection key hp from the hashing key hk .

- $\text{SPHF.Hash}(\text{hk}, m \in \mathcal{X}) \rightarrow h \in \Gamma$: outputs the hash h of m from the hashing key hk .
- $\text{SPHF.ProjHash}(\text{hp}, m \in \mathcal{L}, w) \rightarrow h' \in \Gamma$: outputs the hash value h' from the projection key hp , and the witness w that $m \in \mathcal{L}$, such that

$$\text{SPHF.Hash}(\text{hk}, m) = \text{SPHF.ProjHash}(\text{hp}, m, w).$$

Definition 5 (Smoothness [CS02]). A smooth projective hash function is smooth if for all $m \in \mathcal{X} \setminus \mathcal{L}$, the following distributions are indistinguishable

$$\Delta = \left\{ (\mathcal{L}, \text{pp}, m, \text{hp}, h) \left| \begin{array}{l} \text{pp} \leftarrow \text{SPHF.Setup}(\lambda) \\ \text{hk} \leftarrow \text{SPHF.KeyGen}(\text{pp}), \\ \text{hp} \leftarrow \text{SPHF.ProjKeyGen}(\text{hk}) \\ h \leftarrow \text{SPHF.Hash}(\text{hk}, m) \end{array} \right. \right\},$$

$$\Delta' = \left\{ (\mathcal{L}, \text{pp}, m, \text{hp}, h) \left| \begin{array}{l} \text{pp} \leftarrow \text{SPHF.Setup}(\lambda) \\ \text{hk} \leftarrow \text{SPHF.KeyGen}(\text{pp}), \\ \text{hp} \leftarrow \text{SPHF.ProjKeyGen}(\text{hk}) \\ h \xleftarrow{\$} \Gamma \end{array} \right. \right\}.$$

2.3 Cryptography from Group Actions

We instantiate all of these primitives from group actions, to obtain a UC-secure OT protocol as in [BC15]. The first definitions of group actions can be found in [BY91]. It was followed by the first study of isogenies as potential group actions in 1997 [Cou06], however this work was only published in 2006. We use the more recent formal definitions from [ADFMP20], which we recall below.

Definition 6 (Group Action [ADFMP20]). Let $\mathbb{G} = (n, e, \circ)$ be a group of n elements, where e is the group identity and $\circ$ the group law. $\mathbb{G}$ is said to be a group action on a set $\mathcal{X} \subseteq \{0, 1\}^\lambda$ if there exists a map $\star : \mathbb{G} \times \mathcal{X} \rightarrow \mathcal{X}$ that satisfies:

1. *Identity*: for all $x \in \mathcal{X}$, we have $e \star x = x$.
2. *Compatibility*: for all $g, h \in \mathbb{G}$ and $x \in \mathcal{X}$, we have

$$(g \circ h) \star x = g \star (h \star x).$$

We note that for any $g \in \mathbb{G}$, the map $\pi_g : x \mapsto g \star x$ defines a permutation over $\mathcal{X}$. We use $(\mathbb{G}, \mathcal{X}, \star)$ to denote a group action.

Definition 7 (Regular Group Action [ADFMP20]). A group action $(\mathbb{G}, \mathcal{X}, \star)$ is said to be regular if it is both free and transitive:

1. *Free*: for each $g \in \mathbb{G}$, g is the identity if and only if $\exists x \in \mathcal{X}$ such that $x = g \star x$.
2. *Transitive*: for every $x_1, x_2 \in \mathcal{X}$, there exists a group element $g \in \mathbb{G}$ such that $x_2 = g \star x_1$.

For example, CSIDH constitutes a regular group action [CLM⁺18].

Definition 8 (One-way Group Action [ADFMP20]). A group action $(\mathbb{G}, \mathcal{X}, \star)$ is $(\mathcal{D}_{\mathcal{X}}, \mathcal{D}_{\mathbb{G}})$ -one-way if the family of efficiently computable functions

$$\{f_x : \mathbb{G} \rightarrow \mathcal{X}, g \mapsto g \star x\}_{x \in \mathcal{X}}$$

is $(\mathcal{D}_{\mathcal{X}}, \mathcal{D}_{\mathbb{G}})$ -one-way, where $\mathcal{D}_{\mathcal{X}}$ is a distribution over $\mathcal{X}$ and $\mathcal{D}_{\mathbb{G}}$ is a distribution over $\mathbb{G}$ i.e., for all adversaries $\mathcal{A}$ and negligible ε , it holds that

$$\text{Adv}_{\text{OW}}(\mathcal{A}) := \Pr \left[\begin{array}{l} x \leftarrow \mathcal{D}_{\mathcal{X}}, g \leftarrow \mathcal{D}_{\mathbb{G}}, \\ h \leftarrow \mathcal{A}(x, f_x(g)), f_x(h) = f_x(g) \end{array} \right] \leq \varepsilon.$$

Definition 9 (Weak Pseudorandom Group Action [ADFMP20]). A group action $(\mathbb{G}, \mathcal{X}, \star)$ is $(\mathcal{D}_{\mathcal{X}}, \mathcal{D}_{\mathbb{G}})$ -weakly pseudorandom if the family of efficiently computable permutations

$$\{\pi_g : \mathcal{X} \rightarrow \mathcal{X}, x \mapsto g \star x\}_{g \in \mathbb{G}}$$

is $(\mathcal{D}_{\mathcal{X}}, \mathcal{D}_{\mathbb{G}})$ -weakly pseudorandom, where $\mathcal{D}_{\mathcal{X}}$ is a distribution over $\mathcal{X}$ and $\mathcal{D}_{\mathbb{G}}$ is a distribution over $\mathbb{G}$ *i.e.*, for all adversaries $\mathcal{A}$ and negligible ε , it holds that

$$\text{Adv}_{\text{wPR}}(\mathcal{A}) := \left| \Pr [\mathcal{A}^{\pi_g(\$)}(1^\lambda) = 1] - \Pr [\mathcal{A}^{F(\$)}(1^\lambda) = 1] \right| \leq \varepsilon.$$

where $\pi_g(\$)$, resp. $F(\$)$ are oracles that when queried sample $x \leftarrow \mathcal{D}_{\mathcal{X}}$ and output $(x, \pi_g(x))$, resp. $(x, F(x))$ where F is a random function. We denote by $\text{Adv}_{\text{wPR}}(\mathcal{A}, q)$ the game where the adversary makes at most q oracle queries.

We note that breaking the one-wayness implies breaking the weak pseudorandomness of the group action. As pointed out in [ADFMP20], the decisional CSIDH assumption is an instantiation of a weak pseudorandom EGA.

Definition 10 (Effective Group Action [ADFMP20]). A group action $(\mathbb{G}, \mathcal{X}, \star)$ is an effective group action if the following properties are satisfied:

- The group $\mathbb{G}$ is finite and there exists efficient PPT algorithms for:
 1. **Membership testing:** deciding if a given bit string g represents a valid group element in $\mathbb{G}$.
 2. **Equality testing:** deciding if two bit strings g, g' represent the same element in $\mathbb{G}$.
 3. **Sampling:** sampling a statistically close to uniformly random element $r \in \mathbb{G}$.
 4. **Group law:** computing $g \circ h$ for any $g, h \in \mathbb{G}$.
 5. **Inversion:** computing g^{-1} for any $g \in \mathbb{G}$.
- The set $\mathcal{X}$ is finite and there exist efficient algorithms for:
 1. **Membership testing:** deciding if a given bit string $x \in \{0, 1\}^*$ represents a valid element in $\mathcal{X}$.
 2. **Unique representation:** finding the bitstring x that canonically represents any arbitrary set element $x \in \mathcal{X}$.
- There exists a distinguished element $x_0 \in \mathcal{X}$, called the origin, such that its bit string representation is known.
- There exists an efficient algorithm that computes $g \star x$ given any bitstring representation of $g \in \mathbb{G}, x \in \mathcal{X}$.

3 Chameleon Hash from Group Actions

We describe our first building block, a verifiable chameleon hash function. We start by recalling the definition of a chameleon hash function and its security properties, and then provide our construction from group actions.

3.1 Chameleon Hash

We use collision resistant chameleon hash functions. We need the hash values to be verifiable, so we add the VKeyGen and Valid algorithms to the traditional definition. We expect the verification to be perfectly sound.

Definition 11 (Chameleon Hash Function [KR97]). A Chameleon Hash Function is defined by the following algorithms:

- $\text{CH.KeyGen}(\lambda) \rightarrow (\text{ck}, \text{tk})$: outputs the chameleon hash key ck and the trapdoor tk .

- $\text{VKeyGen}(\text{ck}) \rightarrow (\text{vk}, \text{vtk})$: outputs a verification key vk and its trapdoor vtk . The trapdoor can be empty or public, if the chameleon hash is publicly verifiable.
- $\text{CHash}(\text{ck}, \text{vk}, m; r) \rightarrow h, d$: picks a random r if it is not given, and outputs a chameleon hash h and a witness d needed to verify h .
- $\text{Coll}(\text{ck}, m, r, m', \text{tk}) \rightarrow r'$: takes as input the public keys, the trapdoor tk , a start message m and randomness r , a target message m' and outputs a randomness r' such that if $\text{CHash}(\text{ck}, \text{vk}, m, r) = h, d$ then $\text{CHash}(\text{ck}, \text{vk}, m', r') = h, d$.
- $\text{Valid}(\text{ck}, \text{vk}, m, h, d, \text{vtk}) \rightarrow b \in \{0, 1\}$: output 1 if h opens to m for the witness d , otherwise 0. The verification can be public if vtk is empty.

Definition 12 (Collision Resistance). A chameleon hash function has ε -collision resistance if for any adversary $\mathcal{A}$, we have

$$\Pr \left[\begin{array}{l} (\text{ck}, \text{tk}) \leftarrow \text{KeyGen}, (\text{vk}, \text{vtk}) \leftarrow \text{VKeyGen} \\ (m_1, r_1, m_2, r_2) \leftarrow \mathcal{A}(\text{ck}, \text{vk}) \\ \text{CHash}(\text{ck}, \text{vk}, m_1; r_1) = \text{CHash}(\text{ck}, \text{vk}, m_2; r_2) \wedge m_1 \neq m_2 \end{array} \right] \leq \varepsilon.$$

Definition 13 (Perfect soundness). A verifiable chameleon hash function is perfectly sound if for any adversary $\mathcal{A}$, we have

$$\Pr \left[\begin{array}{l} (\text{ck}, \text{tk}) \leftarrow \text{KeyGen}, (\text{vk}, \text{vtk}) \leftarrow \text{VKeyGen}(\text{ck}) \\ (m, h, d) \leftarrow \mathcal{A}(\text{ck}, \text{vk}) \\ \text{Valid}(\text{ck}, \text{vk}, m, h, d, \text{vtk}) = 1 \\ \wedge \nexists r \text{ s.t. } (h, d) \leftarrow \text{CHash}(\text{ck}, \text{vk}, m; r) \end{array} \right] = 0.$$

3.2 Our Chameleon Hash Function

For compactness, we write from now on $\forall g = (g_1, \dots, g_\ell) \in \mathbb{G}^\ell, \forall x = (x_1, \dots, x_\ell) \in \mathcal{X}^\ell$,

$$g \star x := (g_1 \star x_1, \dots, g_\ell \star x_\ell).$$

$\text{CH.KeyGen}(\lambda)$:	$\text{CHash}(\text{ck}, \text{vk}, m; r \in \mathbb{G}^\ell)$:
1 : $x_0 = (x_{0,1}, \dots, x_{0,\ell}) \xleftarrow{\$} \mathcal{X}^\ell$	1 : for $i \in [\ell]$:
2 : $a = (a_1, \dots, a_\ell) \xleftarrow{\$} \mathbb{G}^\ell$	2 : $h_i \leftarrow r_i \star x_{m_i,i}$
3 : $x_1 \leftarrow a \star x_0$	3 : $d_i \leftarrow r_i \star f_{m_i,i}$
4 : $(\text{ck}, \text{tk}) \leftarrow ((x_0, x_1), a)$	4 : return (h, d)
5 : return (ck, tk)	$\text{Coll}(\text{ck}, m, r, m', \text{tk})$:
$\text{VKeyGen}(\text{ck})$:	1 : for $i \in [\ell]$:
1 : $(b_0, b_1) \xleftarrow{\$} \mathbb{G}^2$	2 : if $m_i = m'_i$:
2 : $f_0 \leftarrow b_0 \star x_0$	3 : $r'_i \leftarrow r_i$
3 : $f_1 \leftarrow b_1 \star x_1$	4 : else if $m_i = 1$:
4 : $\text{vk} \leftarrow (f_0, f_1)$	5 : $r'_i \leftarrow a_i \cdot r_i$
5 : $\text{vtk} \leftarrow (b_0, b_1)$	6 : else $r'_i \leftarrow a_i^{-1} \cdot r_i$
6 : return (vk, vtk)	7 : return $r' = (r'_1, \dots, r'_\ell)$
	$\text{Valid}(\text{ck}, \text{vk}, m, h, d, \text{vtk} = (b_0, b_1))$:
	1 : if $\forall i \in [\ell], h_i = b_{m_i}^{-1} \star d_i$
	2 : return 1
	3 : return 0

Figure 2: Our chameleon hash algorithms, for messages of ℓ bits $m \in \{0, 1\}^\ell$.

We detail our verifiable chameleon hash function from group actions in Figure 2. This construction is reminiscent of constructions based on the discrete logarithm such as [KR97], with an added verifiability component which relies on the same principle. It only requires a one-way regular group action. We show in the standard model that our scheme is collision resistant, and the verifiability is perfectly sound.

Theorem 1. *The chameleon hash function from Figure 2 is collision resistant assuming that the group action $(\mathbb{G}, \mathcal{X}, \star)$ is one-way and regular. More precisely, for any adversary $\mathcal{A}$ there exists an adversary $\mathcal{B}$ of same complexity such that we have*

$$\text{Adv}_{\text{Coll}}(\mathcal{A}) \leq \text{Adv}_{\text{OW}}(\mathcal{B}).$$

Proof. Assume that there is a PPT adversary $\mathcal{A}$ against the collision resistance of the chameleon hash with advantage ε . We build a reduction $\mathcal{R}$ against the one-wayness of the group action from the adversary $\mathcal{A}$.

The reduction takes as input a challenge $(x, y) \in \mathcal{X}^2$, and must output $h \in \mathbb{G}$ such that $y = h \star x$. The reduction embeds (x, y) in the chameleon hash key $\text{ck} = (x_0, x_1)$, by sampling $a, b \xleftarrow{\$} \mathbb{G}^\ell$ and setting

$$\forall i \in [\ell], (x_{0,i}, x_{1,i}) = (a_i \star x, b_i \star y).$$

It runs VKeyGen honestly with ck , and sends (ck, vk) to the adversary. With probability at least ε , $\mathcal{A}$ outputs a valid collision *i.e.*, a couple $(m_1, r_1), (m_2, r_2)$ such that $\text{CHash}(\text{ck}, \text{vk}, m_1; r_1) = \text{CHash}(\text{ck}, \text{vk}, m_2; r_2)$ and $m_1 \neq m_2$. Since $m_1 \neq m_2$, there is at least an index i such that $m_{1,i} \neq m_{2,i}$. Without loss of generality, assume that $m_{1,i} = 1, m_{2,i} = 0$. We get $r_{1,i} \star (b_i \star y) = r_{2,i} \star (a_i \star x)$ hence the reduction obtains a preimage

$$y = ((r_{1,i} \star b_i)^{-1} \cdot (r_{2,i} \star a_i)) \star x,$$

and the theorem follows. $\square$

Theorem 2. *Let $(\mathbb{G}, \mathcal{X}, \star)$ be a regular group action. Then, our verifiable chameleon hash is perfectly sound.*

Proof. Let (m, h, d) be a tuple such that $\text{Valid}(\text{ck}, \text{vk}, m, h, d, \text{vtk}) = 1$ for a given set of keys $\text{ck} = (x_0, x_1)$ and $(\text{vk} = (f_0, f_1), \text{vtk} = (b_0, b_1))$. We prove that if the hash (h, d) is valid for m , then it must be well-formed (*i.e.*, there exists some $r \in \mathbb{G}^\ell$ such that (h, d) is the output of $\text{CHash}(\text{ck}, \text{vk}, m; r)$). Let $r_i \in \mathbb{G}$ such that $h_i = r_i \star x_{m_i,i}$ and let $r'_i \in \mathbb{G}$ such that

$$d_i = r'_i \star f_{m_i,i} = r'_i \star (b_{m_i} \star x_{m_i,i}).$$

Since the tuple (m, h, d) is valid, we have $h_i = b_{m_i}^{-1} \star d_i$, hence

$$r_i \star x_{m_i,i} = b_{m_i}^{-1} \star ((r'_i \cdot b_{m_i}) \star x_{m_i,i}) = r'_i \star x_{m_i,i}.$$

Since the group action is regular, it results that $r_i = r'_i$ for all $i \in [\ell]$ and we successfully showed that $r = r'$, such that $(h, d) \leftarrow \text{CHash}(\text{ck}, \text{vk}, m; r)$. $\square$

4 IND-CCA encryption from group actions

We now build an SPHF-friendly CCA encryption scheme, which, when combined with the verifiable chameleon hash function from the previous section, yields an SPHF-friendly commitment scheme [BC15]. The instantiations previously proposed in [BC15] were either following the Cramer-Shoup framework for CCA encryption [CS02], or using a

specific lattice-based CCA encryption scheme. The Cramer-Shoup construction requires a universal₂ projective hash function, a primitive that we do not have from group actions yet. To avoid this issue, we instead construct an IND-CCA publicly verifiable encryption scheme from an SPHF and a ZKP system.

4.1 A Smooth Projective Hash Function

We use the hash proof system Π from [ADFMP20] based on weakly pseudorandom EGAs, which we recall below. It was only shown to be universal, and we show here that it is actually also smooth. Their projective hash function is defined over the language $\mathcal{L} \subseteq \mathcal{X}^2$:

$$\mathcal{L} = \{(a_0, a_1) \in \mathcal{X}^2 \mid \exists g \in \mathbb{G}, a_0 = g \star x_0, a_1 = g \star x_1\}.$$

We recall their construction [ADFMP20].

- $\Pi.\text{Setup}(\lambda) \rightarrow (x_0, x_1)$: set public values $\text{pp} = (x_0, x_1) \in \mathcal{X}^2$.
- $\Pi.\text{KeyGen}(\text{pp}) \rightarrow \text{hk} = (h, b)$: output the hashing key $(h, b) \xleftarrow{\$} \mathbb{G}^\ell \times \{0, 1\}^\ell$.
- $\Pi.\text{ProjKeyGen}(\text{hk} = (h, b)) \rightarrow \text{hp} \in \mathcal{X}^\ell$: compute the projective hashing key

$$\text{hp} = y = (h_1 \star x_{b_1}, \dots, h_\ell \star x_{b_\ell}).$$

- $\Pi.\text{Hash}((h, b), (a_0, a_1)) \rightarrow h' \in \mathcal{X}^\ell$: compute the hash $h' = (h_1 \star a_{b_1}, \dots, h_\ell \star a_{b_\ell})$.
- $\Pi.\text{ProjHash}(\text{hp} = y, (a_0, a_1) \in \mathcal{L}, w = g) \rightarrow h' \in \mathcal{X}^\ell$: compute the projective hash $h' = (g \star y_1, \dots, g \star y_\ell)$.

Theorem 3. *The projective hash function Π is ε -smooth, where ε is bounded by*

$$\varepsilon \leq 2\text{Adv}_{\text{wPR}}(\mathcal{B}, \ell + 1)$$

for any PPT adversary $\mathcal{B}$ against the weak pseudorandomness of the group action.

Proof. The smoothness property considers hash evaluations on messages that are not in the language, hence in our case on messages $a = (a_0, a_1) = (g_0 \star x_0, g_1 \star x_1) \notin \mathcal{L}$ for $g_0 \neq g_1$. Thus, we need two different pseudorandomness oracles, to simulate both the action of g_0 and g_1 . We build two successive reductions $\mathcal{R}_0$ and $\mathcal{R}_1$ against the weak pseudorandomness of the group action, where we first tackle the first part of the message a_0 and then the second part a_1 . Let $\mathcal{A}$ be an adversary against the smoothness of Π with advantage ε .

First reduction. In the first reduction, $\mathcal{R}_0$ has access to $\mathcal{A}$ and to an oracle $\mathcal{O}_0$ outputting either random evaluations of a random function F_0 or random evaluations of π_{g_0} . First, the reduction asks $\mathcal{O}_0$ for an evaluation $(x_0, F_0(x_0))$ and uses it in the public parameters $\text{pp} = (x_0, x_1)$, and sets $a_0 = F_0(x_0)$ in the message $a = (a_0, a_1)$. The secret key $b \xleftarrow{\$} \{0, 1\}^\ell$ is honestly sampled. Then, $\mathcal{R}_0$ asks for at most ℓ evaluations $(y_i, F_0(y_i))$ for the indexes i such that $b_i = 0$ and sets $\text{hp} = (y_i)_{i \in [\ell]}$. The hash h' is honestly computed for all i such that $b_i \neq 0$, and $h'_i = F_0(y_i)$ when $b_i = 0$. Lastly, $\mathcal{R}_0$ sends the smoothness instance $(\text{pp}, a, \text{hp}, h')$ to $\mathcal{A}$, and answers the same guess as $\mathcal{A}$.

Let G_0 be the original game realized when F_0 is π_{g_0} , and G_1 be the game realized when F_0 is a random function. Under the pseudorandomness of the group action, the adversary cannot distinguish between these two games. More precisely, we have

$$|\text{Adv}_1(\mathcal{A}) - \text{Adv}_0(\mathcal{A})| = \text{Adv}_{\text{wPR}}(\mathcal{B}, \ell + 1).$$

Second reduction. Let $\mathcal{O}_1$ be another weak pseudorandomness oracle. The second reduction $\mathcal{R}_1$ starting from $\mathbf{G}_1$ is similar to the first one, except that $\mathcal{R}_1$ asks $\mathcal{O}_1$ for an evaluation $(x_1, F_1(x_1))$ and replaces the second element in $\mathbf{pp} = (x_0, x_1)$ and in the message $a = (a_0, a_1 = F_1(x_1))$. Then, $\mathcal{R}_1$ queries $\mathcal{O}_1$ for $(y_i, F_1(y_i))$ for the indexes i such that $b_i = 1$ and sets $\mathbf{hp} = (y_i)_{i \in [\ell]}$. The hash h' is computed as above for i such that $b_i = 0$, and $h'_i = F_1(y_i)$ when $b_i = 1$.

Now $\mathbf{G}_1$ is exactly the game realized when F_1 is π_{g_1} , and let $\mathbf{G}_2$ be the game realized when F_1 is a random function. Under the pseudorandomness of the group action, the adversary cannot distinguish between these two games and we also have

$$|\text{Adv}_2(\mathcal{A}) - \text{Adv}_1(\mathcal{A})| = \text{Adv}_{\text{wPR}}(\mathcal{B}, \ell + 1).$$

In $\mathbf{G}_2$ where $\mathcal{O}_0$ and $\mathcal{O}_1$ are oracles to random functions, $h \xleftarrow{\$} \mathcal{X}^\ell$ is an independent random value, while in $\mathbf{G}_0$ it is an honestly computed hash $h = g_{b_i} \star y_i = h_i \star a_{b_i}$. Hence the adversary's advantage ε to distinguish between a random value and a hash *i.e.*, between $\mathbf{G}_0$ and $\mathbf{G}_2$ is bounded by

$$\varepsilon \leq 2\text{Adv}_{\text{wPR}}(\mathcal{B}, \ell + 1)$$

by chaining the inequalities. □

4.2 Proof of language membership

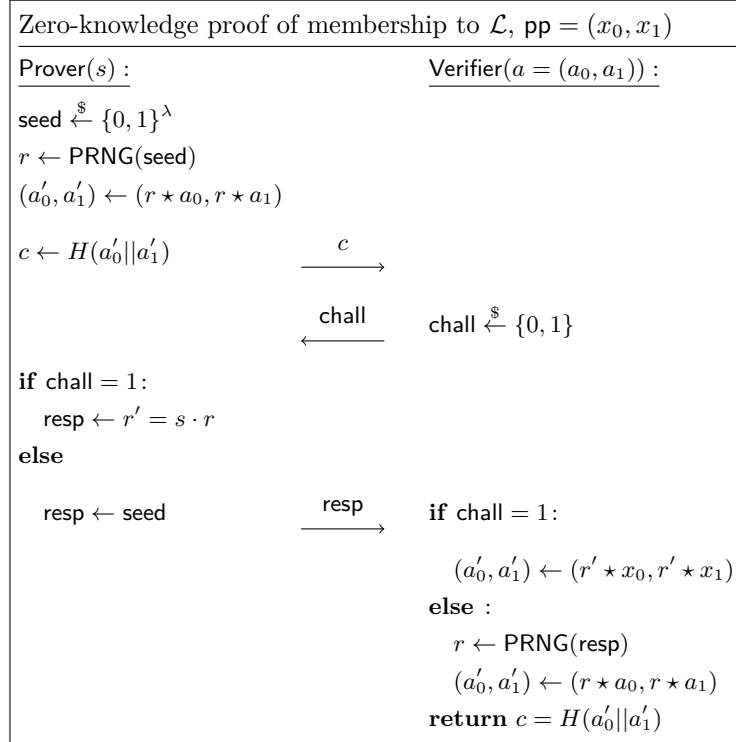


Figure 3: The zero-knowledge proof of membership for the language $\mathcal{L}$, with witness s that $a \in \mathcal{L}$. We call the algorithm of the prover before (resp. after) the challenge is sent P_0 (resp. P_1), similarly V_0 (resp. V_1) for the verifier.

We start our construction with an SPHF as in the Cramer-Shoup encryption scheme, and replace the universal₂ projective hash function by a ZKP of well-formedness. The role of the universal₂ projective hash function is essentially to make the decryption fail as soon

as the ciphertext is not well-formed. We can replace this step by a ZKP proving whether a ciphertext was honestly computed.

We use a ZKP of membership to $\mathcal{L}$ *i.e.*, a ZKP of knowledge of the witness w used to compute the SPHF. This allows us to prove the integrity of a ciphertext, and obtain a publicly verifiable encryption. We give the zero-knowledge protocol in Figure 3, and use its non-interactive form with the Fiat-Shamir transform. The proof is repeated λ times to get appropriate soundness. This is a classical sigma protocol that can be found in very similar forms in other settings, such as the identification protocol presented in [EKKP20]. We prove the security of this zero-knowledge proof of language membership.

Theorem 4 (Zero-Knowledge). *The protocol presented in Figure 3 is perfectly HVZK.*

Proof. We build the following simulator, given a state $(a_0, a_1) \in \mathcal{X}^2$ and a challenge $\text{chall} \xleftarrow{\$} \{0, 1\}$. If $\text{chall} = 0$, the simulator samples $\text{seed} \xleftarrow{\$} \{0, 1\}^\lambda$ and computes $r \leftarrow \text{PRNG}(\text{seed})$ and $c \leftarrow H(r \star a_0 || r \star a_1)$. Then $(c, \text{chall}, \text{seed})$ is a valid transcript.

If $\text{chall} = 1$, the simulator samples $r' \xleftarrow{\$} \mathbb{G}$ and computes $c \leftarrow H(r' \star x_0 || r' \star x_1)$. Then, (c, chall, r') is also a valid transcript. All of the elements of the transcript follow exactly the same distribution as in an honest execution of the protocol, so the adversary has no advantage to distinguish. $\square$

We derive perfect zero-knowledge for the non-interactive protocol through the Fiat-Shamir heuristic.

Theorem 5 (Special Soundness). *The protocol in Figure 3 has special soundness, assuming that H is a collision resistant hash function.*

Proof. Let $(c, 0, \text{resp}_0)$ and $(c, 1, \text{resp}_1)$ be two valid proofs. We build the following extractor. Let $r' = \text{resp}_1$ and $r \leftarrow \text{PRNG}(\text{resp}_0)$. From resp_1 , we have both $a'_0 = r' \star x_0$ and $a'_1 = r' \star x_1$, hence $(a'_0, a'_1) \in \mathcal{L}$. From resp_0 , we have $a''_0 = r \star a_0$ and $a''_1 = r \star a_1$. Since both proofs are valid, we have $c = H(a'_0 || a'_1) = H(a''_0 || a''_1)$. Hence, either we have $a''_0 = a'_0$ and $a''_1 = a'_1$ thus

$$(a_0, a_1) = ((r^{-1}r') \star x_0, (r^{-1}r') \star x_1) \in \mathcal{L},$$

and we found a valid witness $r^{-1}r'$ for $(a_0, a_1) \in \mathcal{L}$, or we found a collision for the hash function H . $\square$

Now, we prove the online-extractability for the non-interactive protocol using the Fiat-Shamir transform, repeated λ times.

Theorem 6 (Online Extractability). *Assume that $|\mathbb{G}| \geq 2^\lambda$. We model the PRNG and the Fiat-Shamir transform as random oracles $\mathcal{O}_{\text{PRNG}}$ and $\mathcal{O}_{\text{FS}}$. The NIZKP protocol derived from Figure 3 is online extractable. More precisely, for q_{PRNG} and q_{FS} the maximum number of queries to the oracles $\mathcal{O}_{\text{PRNG}}$ and $\mathcal{O}_{\text{FS}}$, we have*

$$\text{Adv}_{\text{OE}}(\mathcal{A}) \leq \frac{q_{\text{FS}} + 1}{2^\lambda} + \frac{q_{\text{PRNG}} \cdot q_{\text{FS}}}{|\mathbb{G}|}.$$

Proof. We build the following extractor. We record all of the queries of the adversary to the random oracles $\mathcal{O}_{\text{PRNG}}$ and $\mathcal{O}_{\text{FS}}$.

If the adversary never queried $\mathcal{O}_{\text{FS}}$ on com , then its probability of guessing correctly the challenge $\text{chall} = (\text{chall}_1, \dots, \text{chall}_\lambda)$ is of $1/2^\lambda$. Hence, the probability for $\mathcal{A}$ to win with such a com is no greater than $1/2^\lambda$.

Assume now that the adversary made a query to $\mathcal{O}_{\text{FS}}$ on com . The extractor goes through the queries made to $\mathcal{O}_{\text{PRNG}}$ and looks for a valid 0 response *i.e.*, a tuple (x, y) such that

$$y \leftarrow \mathcal{O}_{\text{PRNG}}(x), \quad \text{com}_i = H(y \star a_0 || y \star a_1)$$

for some $i \in [\lambda]$. If the corresponding chall_i is equal to 1, we can invoke the extractor from Theorem 5 on input $(\text{com}_i, 0, x)$ and $(\text{com}_i, 1, \text{resp}_i)$ and retrieve the witness. Otherwise if $\text{chall}_i = 0$, then $\mathcal{A}$ had a probability of at most $q_{\text{PRNG}}/|\mathbb{G}|$ to make such a query to $\mathcal{O}_{\text{PRNG}}$ since we assume the query to $\mathcal{O}_{\text{FS}}$ was already made *i.e.*, com_i is a fixed value.

If $\mathcal{A}$ never made such a query to $\mathcal{O}_{\text{PRNG}}$, its probability of guessing a correct response resp_i is either $1/2^\lambda$ if $\text{chall}_i = 0$ or $1/|\mathbb{G}|$ if $\text{chall}_i = 1$, thus no greater than $1/2^\lambda$. Hence, its probability of success after a query to $\mathcal{O}_{\text{FS}}$ on com is less than $1/2^\lambda + q_{\text{PRNG}}/|\mathbb{G}|$. Since $\mathcal{A}$ made at most q_{FS} queries to the Fiat-Shamir oracle, it results that

$$\text{Adv}_{\text{OE}}(\mathcal{A}) \leq \frac{1}{2^\lambda} + q_{\text{FS}} \left(\frac{1}{2^\lambda} + \frac{q_{\text{PRNG}}}{|\mathbb{G}|} \right)$$

and the theorem follows. $\square$

4.3 The IND-CCA encryption

We detail our IND-CCA encryption scheme $\mathcal{E}$ in Figure 4. This scheme is inspired by the Cramer-Shoup generic construction, but in which we replace the second component of the ciphertext (the universal_2 projective hash function) by a zero-knowledge proof of honest computation of the first component. We prove our scheme is secure against adaptive chosen ciphertext attacks, under the security of the zero-knowledge proof and the smoothness of the projective hash function.

$\mathcal{E}.\text{KeyGen}(\lambda)$:
1 : $\text{sk} = (h, b) \leftarrow \Pi.\text{KeyGen}(\text{pp})$
2 : $\text{pk} = y \leftarrow \Pi.\text{ProjKeyGen}((h, b))$
3 : return (pk, sk)
$\mathcal{E}.\text{Enc}(\text{pk}, m \in \mathcal{X}^\ell)$:
1 : $s \xleftarrow{\$} \mathbb{G}$
2 : $a \leftarrow (s \star x_0, s \star x_1)$
3 : $c_0 \leftarrow \Pi.\text{ProjHash}(\text{pk}, a, s)$
4 : $e \leftarrow m \oplus c_0$
5 : $\pi \leftarrow \text{NIZKP}$ that $(a, s) \in \mathcal{L}$, using $H(a e)$ in the transcript of the Fiat Shamir transform
6 : return $c = (a, e, \pi)$
$\mathcal{E}.\text{Dec}(\text{sk}, c = (a, e, \pi))$:
1 : if $\text{Verif}(\pi, a, e) = \perp$: return $\perp$
2 : $c_0 \leftarrow \Pi.\text{Hash}((h, b), a)$
3 : return $m \leftarrow e \oplus c_0$

Figure 4: Our IND-CCA encryption scheme $\mathcal{E}$ algorithms. Let H be a collision resistant hash function, and $(x_0, x_1) \in \mathcal{X}^2$ be public values. We write $\mathcal{E}.\text{Enc}(\text{pk}, m; s)$ when s is the randomness used in the encryption.

Theorem 7. *The encryption scheme $\mathcal{E}$ has adaptive IND-CCA security, assuming Π is ε -smooth, and the NIZKP from Figure 3 is both online extractable and zero-knowledge. More precisely, for any adversary $\mathcal{A}$ we have*

$$\text{Adv}_{\text{IND-CCA}}(\mathcal{A}, q) \leq \text{Adv}_{\text{ZK}}(\mathcal{A}) + \left(1 - \frac{1}{|\mathbb{G}|}\right) (q \cdot \text{Adv}_{\text{OE}}(\mathcal{A}) + \varepsilon).$$

Proof. We prove this theorem by a sequence of games.

Let G_0 be the original IND-CCA game:

- The challenger generates keys $(pk, sk) \leftarrow \text{KeyGen}(\lambda)$ and sends pk to $\mathcal{A}$.
- $\mathcal{A}$ sends adaptively decryption queries to the oracle $\mathcal{O}_{\text{Dec}}$.
- $\mathcal{A}$ sends its target m_0, m_1 to the challenger.
- The challenger answers with the challenge $c^* \leftarrow \text{Enc}(m_b, pk)$ for $b \xleftarrow{\$} \{0, 1\}$.
- $\mathcal{A}$ makes a new round of adaptive decryption queries to $\mathcal{O}_{\text{Dec}}$ for any $c \neq c^*$, before giving its guess b' .

Game G_1 : we modify G_0 such that the challenger uses the NIZKP simulator to compute the proof π^* in the challenge without using a witness. It results that

$$|\text{Adv}_1(\mathcal{A}) - \text{Adv}_0(\mathcal{A})| = \text{Adv}_{\text{ZK}}(\mathcal{A}).$$

Game G_2 : we modify G_1 such that the challenger does not use the projective key in the challenge phase. Let (h, b) be the secret hashing key. To encrypt m_b , the challenger samples $a^* \xleftarrow{\$} \mathcal{X}^2$, and computes

$$c_0^* \leftarrow \Pi.\text{Hash}((h, b), a^*), \quad e^* \leftarrow m_b \oplus c_0^*,$$

while π^* is still computed using the simulator as in G_1 . The fact $a^* \in \mathcal{X}^2$ is indistinguishable from $a^* \in \mathcal{L}$ for the adversary comes from the fact that the subset membership problem is hard for $\mathcal{L}$, under the weak pseudorandomness of the EGA.

Claim. Let F be the event that $a^* \in \mathcal{L}$. We have

$$|\text{Adv}_2(\mathcal{A}) - \text{Adv}_1(\mathcal{A})| = \Pr[\bar{F}] |\Pr[G_1 = 1] - \Pr[G_2 = 1 | \bar{F}]|.$$

From the law of total probability, we have

$$\begin{aligned} |\text{Adv}_2(\mathcal{A}) - \text{Adv}_1(\mathcal{A})| &= |\Pr[G_2 = 1] - \Pr[G_1 = 1]| \\ &= |(\Pr[F] \Pr[G_2 = 1 | F] + \Pr[\bar{F}] \Pr[G_2 = 1 | \bar{F}]) - \Pr[G_1 = 1]| \\ &= |(\Pr[F] \Pr[G_2 = 1 | F] + \Pr[\bar{F}] \Pr[G_2 = 1 | \bar{F}]) - \Pr[G_1 = 1]|. \end{aligned}$$

Since when $a^* \in \mathcal{L}$, G_2 perfectly simulates G_1 due to the projective property of the SPHF, thus $\Pr[G_2 = 1 | F] = \Pr[G_1 = 1]$ and the claim follows.

Then from now on, we only need to study the case $a^* \notin \mathcal{L}$.

Game G_3 : the same as G_2 except that the oracle $\mathcal{O}_{\text{Dec}}(\cdot, sk)$ outputs $\perp$ for all ciphertexts $c = (a, e, \pi)$ such that $a \notin \mathcal{L}$.

Claim. Assuming that the adversary makes at most q queries to $\mathcal{O}_{\text{Dec}}$, we have

$$|\text{Adv}_3(\mathcal{A}) - \text{Adv}_2(\mathcal{A})| \leq q \cdot \text{Adv}_{\text{OE}}(\mathcal{A}).$$

Since $\mathcal{O}_{\text{Dec}}$ outputs $\perp$ whenever the proof of membership π is false, the probability that the decryption algorithm does not output $\perp$ on a ciphertext (a, e, π) where $a \notin \mathcal{L}$ is the probability for the adversary to forge a valid proof without any witness. Hence, the claim follows directly by using successively q times the online extractability of the zero-knowledge proof.

Game G_4 : the same as G_3 except that the challenger samples c_0^* uniformly when encrypting m_b .

Claim. We have $|\text{Adv}_4(\mathcal{A}) - \text{Adv}_3(\mathcal{A})| \leq \varepsilon$.

The inputs of the challenger in G_3 and G_4 are identical, except that $c_0^* = \Pi.\text{Hash}((h, b), a^*)$ in G_3 while we now choose $c_0^* \xleftarrow{\$} \mathcal{X}^\ell$ at random in G_4 . Since Π is ε -smooth, the probability for $\mathcal{A}$ to distinguish between these distributions is bounded by ε .

Claim. We have $\text{Adv}_4(\mathcal{A}) = 0$.

The inputs of the adversary in G_4 are all independent of the message m_b . Since c_0^* is drawn uniformly, it acts as a one-time pad and the distribution of $e^* = m_b \oplus c_0^*$ is also uniform.

Finally, by chaining the inequalities and noting that $\Pr[\bar{F}] = (1 - 1/|\mathbb{G}|)$, we get

$$\text{Adv}_{\text{IND-CCA}}(\mathcal{A}, q) \leq \text{Adv}_{\text{ZK}}(\mathcal{A}) + (1 - 1/|\mathbb{G}|)(q \cdot \text{Adv}_{\text{OE}}(\mathcal{A}) + \varepsilon).$$

□

5 SPHF-friendly E^2 commitment scheme

We now have all of the blocks required to obtain an E^2 commitment scheme CS based on group actions [BC15]. We start by recalling the definition of an SPHF-friendly E^2 commitment scheme. Then, we propose a construction and prove that it achieves *strong simulation indistinguishability* under the weak pseudorandomness of the group action. Finally, we provide the corresponding SPHF over the language of valid commitments.

5.1 E^2 commitment scheme

To achieve UC-security, we rely on *equivocable* and *extractable* (E^2) commitment schemes. Essentially, there exists a simulated setup (indistinguishable from an honest setup) with a trapdoor that allows the simulator to recover the committed value m (extractable), and another simulated setup with a trapdoor that allows the simulator, on input any commitment c and any value m , to output an opening value δ that opens c to m (equivocable). For the sake of simplicity, we merge these two simulated setups into a single simulated SetupT , that outputs the global parameters and a trapdoor for both simulation and equivocation.

Definition 14 (Commitment scheme [BCC88]). An E^2 commitment scheme is defined by the following algorithms:

- $\text{Setup}(\lambda) \rightarrow \text{pp}$: takes as input the security parameter and outputs the global parameters, passed through the other algorithms.
- $\text{SetupT}(\lambda) \rightarrow \text{pp}, \tau$: the simulated setup, outputs the global parameters along with a trapdoor τ .
- $\text{Commit}^\ell(m) \rightarrow (\text{com}, r)$: takes as input a message m and a label ℓ and outputs a pair (com, r) where com is the commitment to m for the label ℓ .
- $\text{Open}^\ell(\text{com}, m, r) \rightarrow b \in \{0, 1\}$: outputs 1 if r is a valid opening for the commitment com to the message m and label ℓ , otherwise outputs 0.
- $\text{SCom}^\ell(\tau) \rightarrow (\text{com}, \text{eqk})$: takes as input the trapdoor τ and a label ℓ , and outputs a pair (com, eqk) where com is a commitment and eqk an equivocation key.
- $\text{OpenGen}^\ell(\text{eqk}, \text{com}, m) \rightarrow r$: takes as input a commitment com , a label ℓ , a message m and an equivocation key eqk and outputs an opening value r for com and ℓ on m using eqk .
- $\text{Extract}^\ell(\tau, \text{com}) \rightarrow m/\perp$: takes as input the trapdoor τ , a commitment com and a label ℓ and outputs the committed message m , or $\perp$ if the commitment is invalid.

A real commitment should be indistinguishable from a simulated commitment, even with oracle access to extracted values and simulated commitments. This is captured by the following property defined in [ABB⁺13].

Definition 15 (Strong Simulation Indistinguishability). A commitment scheme has strong simulation indistinguishability (as in Figure 5) if for any adversary $\mathcal{A}$ there exists a negligible ε such that

$$|\Pr[\text{Exp}_{\text{SSIND}}(\mathcal{A}) = 1] - 1/2| \leq \varepsilon.$$

$\text{Exp}_{\text{SSIND}}(\mathcal{A})$:	$\mathcal{O}_{\text{Extract}}(\tau, \text{com})$:
1 : $L_{\text{SCom}} \leftarrow \emptyset$	1 : if $\exists m, (m, \text{com}) \in L_{\text{SCom}}$:
2 : $(\text{pp}, \tau) \leftarrow \text{SetupT}(\lambda)$	2 : return m
3 : $m \leftarrow \mathcal{A}^{\mathcal{O}_{\text{Extract}}, \mathcal{O}_{\text{SCom}}}(\text{pp})$	3 : return $\text{Extract}(\tau, \text{com})$
4 : $b \xleftarrow{\$} \{0, 1\}$	$\mathcal{O}_{\text{SCom}}(\tau, m)$:
5 : if $b = 0$:	1 : $(\text{com}, \text{eqk}) \leftarrow \text{SCom}(\tau)$
6 : $(\text{com}, \text{eqk}) \leftarrow \text{SCom}(\tau)$	2 : $r \leftarrow \text{OpenGen}(\text{eqk}, \text{com}, m)$
7 : $r \leftarrow \text{OpenGen}(\text{eqk}, \text{com}, m)$	3 : Add (m, com) to L_{SCom}
8 : Add (m, com) to L_{SCom}	4 : return (com, r)
9 : if $b = 1$:	
10 : $(\text{com}, r) \leftarrow \text{Commit}(m)$	
11 : $b' \leftarrow \mathcal{A}^{\mathcal{O}_{\text{Extract}}, \mathcal{O}_{\text{SCom}}}(\text{com}, r)$	
12 : return $b = b'$	

Figure 5: The Strong Simulation Indistinguishability game and its oracles $\mathcal{O}_{\text{Extract}}(\tau, \cdot)$ and $\mathcal{O}_{\text{SCom}}(\tau, \cdot)$. The list L_{SCom} keeps track of the simulated commitments (m, com) , such that $\mathcal{O}_{\text{Extract}}$ can answer consistently when input with a simulated commitment.

SPHF-friendly commitments. We define the following pseudorandomness property for simulated commitments for an SPHF over the language of valid commitments. Let CS be an E^2 commitment scheme and let $\mathcal{L}_m$ be the language of valid commitments for CS to a message m such that

$$\mathcal{L}_m^{\text{CS}} := \{\text{com} \mid \exists s, \text{CS.Open}(s, \text{vtk}, \text{com}, m) = 1\}.$$

Essentially, this property requires the evaluations of the SPHF to look uniformly distributed over *simulated* commitments.

$\text{Exp}_{\text{PR}}(\mathcal{A})$:
1 : $(\text{pp}, \tau) \leftarrow \text{CS.SetupT}(\lambda)$
2 : $b \xleftarrow{\$} \{0, 1\}, L_{\text{SCom}} \leftarrow \emptyset$
3 : $m, m' \leftarrow \mathcal{A}^{\mathcal{O}_{\text{Extract}}(\tau, \cdot), \mathcal{O}_{\text{SCom}}(\tau, \cdot)}(\text{pp})$
4 : $\text{hk} \leftarrow \text{SPHF.KeyGen}(\mathcal{L}_m)$
5 : $\text{hp} \leftarrow \text{SPHF.ProjKeyGen}(\text{hk})$
6 : if $b = 0$:
7 : $(\text{com}, \text{eqk}) \leftarrow \text{CS.SCom}(\tau)$
8 : $h \leftarrow \text{SPHF.Hash}(\text{hk}, \text{com})$
9 : if $b = 1$: $h \xleftarrow{\$} \Gamma$
10 : $b' \leftarrow \mathcal{A}^{\mathcal{O}_{\text{Extract}}(\tau, \cdot), \mathcal{O}_{\text{SCom}}(\tau, \cdot)}(\text{com}, \text{hp}, h)$
11 : return $b = b'$

Figure 6: The pseudorandomness game for an SPHF defined over the language $\mathcal{L}_m^{\text{CS}}$, the oracles $\mathcal{O}_{\text{Extract}}$ and $\mathcal{O}_{\text{SCom}}$ are defined as in Figure 5, and the list L_{SCom} keeps track of the oracle queries such that $\mathcal{O}_{\text{Extract}}$ can answer consistently.

Definition 16 (Pseudorandomness [ABB⁺13]). An SPHF defined over the language $\mathcal{L}_m$ for a commitment scheme CS is pseudorandom (as in Figure 6) if for any adversary $\mathcal{A}$ there exists a negligible ε such that

$$|\Pr[\text{Exp}_{\text{PR}}(\mathcal{A}) = 1] - 1/2| \leq \varepsilon.$$

5.2 Our E^2 commitment scheme from group actions

We now describe the E^2 commitment scheme resulting from our encryption scheme and chameleon hash function. The verification chameleon hash keys (vk, vtk) are generated in a pre-flow by the server, which keeps the trapdoor vtk and sends vk to the sender.

- $\text{SetupT}(\lambda) \rightarrow \text{pp} = (\text{pk}, \text{ck}), \tau = (\text{sk}, \text{tk})$: generate the encryption keys (pk, sk) , the chameleon hash keys (ck, tk) . The non-simulated **Setup** algorithm is exactly the same, except that the trapdoor $\tau = (\text{sk}, \text{tk})$ is erased at the end.
- $\text{Commit}(\text{pp}, \text{vk}, m \in \{0, 1\}^\ell) \rightarrow (\text{com}, s)$: sample $r = (r_0^1, r_1^1, \dots, r_0^\ell, r_1^\ell) \xleftarrow{\$} \mathbb{G}^{2 \times \ell}$ and compute

$$((h_i, d_{m_i}^i)_{i \in [\ell]}) \leftarrow \text{CHash}(\text{ck}, \text{vk}, m; (r_{m_i}^i)_{i \in [\ell]}).$$

Then, sample $d_{1-m_i}^i \xleftarrow{\$} \mathcal{X}$ for all $i \in [\ell]$ and encrypt the value $d = (d_0^i, d_1^i)_{i \in [\ell]} \in \mathcal{X}^{2\ell}$ as $b \leftarrow \mathcal{E}.\text{Enc}(\text{pk}, d; s)$. Return $\text{com} = (h, b)$ with opening the randomness s .

- $\text{Open}(s, \text{vtk}, \text{com} = (h, b), m) \rightarrow b' \in \{0, 1\}$: extract d_{i, m_i} from the ciphertext b using the randomness s for all $i \in [\ell]$, and return

$$\text{Valid}(\text{ck}, \text{vk}, m, h, \{d_{i, m_i}\}_{i \in [\ell]}, \text{vtk}).$$

- $\text{SCom}^\ell(\tau = (\text{sk}, \text{tk})) \rightarrow (\text{com}, \text{eqk})$: compute for $i \in [\ell]$:

$$(h_i, d_0^i) \leftarrow \text{CHash}(\text{ck}, \text{vk}, 0; r_0^i).$$

Then use tk to compute $r_1^i \leftarrow \text{Coll}(\text{ck}, 0, r_0^i, 1, \text{tk})$ for each $i \in [\ell]$. Let

$$(h_i, d_1^i) \leftarrow \text{CHash}(\text{ck}, \text{vk}, 1; r_1^i).$$

Encrypt $d = (d_0^i, d_1^i)_{i \in [\ell]}$ as $b \leftarrow \mathcal{E}.\text{Enc}(\text{pk}, d; s)$, and return the commitment $\text{com} = (h, b)$ with $\text{eqk} = s$.

- $\text{OpenGen}^\ell(\text{eqk} = s, \text{com} = (h, b), m) \rightarrow d$: uses the equivocation key s to extract the opening values $\{d_{i, m_i}\}_{i \in [\ell]}$ from b in order to open to m .
- $\text{Extract}^\ell(\tau = (\text{tk}, \text{sk}), \text{com} = (h, b), \text{vtk}) \rightarrow m/\perp$: compute

$$(d_j^i)_{j \in \{0, 1\}, i \in [\ell]} \leftarrow \mathcal{E}.\text{Dec}(\text{sk}, b).$$

Then, for each bit $j \in \{0, 1\}$ and $i \in [\ell]$, run the algorithm $\text{Valid}(\text{ck}, \text{vk}, j, h_i, d_j^i, \text{vtk})$.

If only one opening value d_j^i passes the verification, then $m_i = j$. If this is the case for all $i \in [\ell]$, output $m = (m_1, \dots, m_\ell)$. If there is an ambiguity, output $\perp$.

Theorem 8. *The commitment scheme CS has strong simulation indistinguishability under the assumption that the group action is weakly pseudorandom, more precisely, we have*

$$\text{Adv}_{\text{SSIND}}(\mathcal{A}) \leq \frac{q_{\text{Extract}} \cdot \ell}{|\mathcal{X}|} + 4\text{Adv}_{\text{wPR}}(\mathcal{A})$$

for q_{Extract} the number of queries to the extraction oracle.

Proof. We first start with the real game in the case $b = 0$ i.e., the challenger sends a simulated commitment.

Game G_1 : instead of computing first $(h_i, d_0^i)_i$ a hash of a 0^ℓ and then $(h_i, d_1^i)_i$ a hash of 1^ℓ , the challenger starts by computing $(h, d_m) \leftarrow \text{CHash}(\text{ck}, \text{vk}, m; r_m)$ and then

$$(h, d_{1-m}) \leftarrow \text{CHash}(\text{ck}, \text{vk}, \{1 - m_i\}_{i \in [\ell]}; r_{1-m})$$

where r_{1-m} is output by $\text{Coll}(\text{ck}, m, r_m, \{1 - m_i\}_{i \in [\ell]}, \text{tk})$.

Claim. We have $|\text{Adv}_1(\mathcal{A}) - \text{Adv}_0(\mathcal{A})| = 0$.

Since this is just reordering, G_1 is perfectly indistinguishable from the original game.

Game G_2 : instead of computing d_{1-m} with the key $\text{vk} = (f_0, f_1)$, the challenger uses a different verification key

$$\text{vk}' = (f'_0, f'_1) = (\text{vtk}'_0 \star x_0, \text{vtk}'_1 \star x_1)$$

in the challenge phase and computes $d_{1-m,i} = r_i \star f'_{1-m_i}$ for $i \in [\ell]$.

Additionally, we denote by $m^*, \text{com}^* = (h^*, b^*)$ the challenge. We modify the oracle $\mathcal{O}_{\text{Extract}}$ in the following way. On input a query (h, b) , for all indexes i such that $(h_i, d_i) = (h_i^*, d_i^*)$, the oracle answers consistently with m_i^* . On the remaining indexes, the oracle uses its key τ to extract m . If there is any index i such that $h_i = h_i^*$ extracts to m_i^* with $d_i \neq d_i^*$, the oracle aborts.

If $\mathcal{A}$ makes a query to $\mathcal{O}_{\text{SCom}}$ on a value $m_i = m_i^*$, the oracle re-samples the randomness until each coefficient $h_i \neq h_i^*$ is different from the challenge.

Claim. We have

$$|\text{Adv}_2(\mathcal{A}) - \text{Adv}_1(\mathcal{A})| \leq \frac{q_{\text{Extract}} \cdot \ell}{|\mathcal{X}|} + 2\text{Adv}_{\text{wPR}}(\mathcal{A}).$$

Let q_{Extract} be the number of queries made to the extraction oracle, the oracle aborts only whenever the adversary guesses the correct value that should have been $d^* = r^* \star \text{vk}_{m_i^*}$. Under the weak pseudorandomness of the EGA, its probability to do so is bounded by $q_{\text{Extract}} \cdot \ell / |\mathcal{X}|$, which is negligible.

Under the assumption that the subset membership problem is hard for $\mathcal{L}$ (which is hard under the weak pseudorandomness of the EGA [ADFMP20]), the distributions of $r_i \star f'_{1-m_i}$ and $r_i \star f_{1-m_i}$ are both indistinguishable, hence the adversary cannot distinguish between those games. The claim follows, by using twice the weak pseudorandomness of the group action for f'_0 and f'_1 .

Game G_3 : finally, the challenger replaces all the values computed with vtk' by outputs of weak pseudorandomness oracles $\mathcal{O}_{F_0}$ and $\mathcal{O}_{F_1}$. If F_0 and F_1 are random oracles, this game emulates perfectly the real game for $b = 1$ where the challenger computes an honest commitment. If $F_i = \pi_{\text{vtk}'_i}$, we are in the previous game. Under the weak pseudorandomness of the group action, G_2 and G_3 are indistinguishable

$$|\text{Adv}_3(\mathcal{A}) - \text{Adv}_2(\mathcal{A})| \leq 2\text{Adv}_{\text{wPR}}(\mathcal{A}).$$

Finally, we have $\text{Adv}_3(\mathcal{A}) = 0$ as all the values are now random in G_3 , and the theorem follows by chaining the inequalities. $\square$

5.3 SPHF over the language of valid ciphertexts

To finalize our construction, we build a smooth projective hash function over the language of valid commitments. For this, it suffices to have an SPHF over the language of valid ciphertexts for $\mathcal{E}$, from which one can derive an SPHF over valid commitments by conjunction [ACP09, BHBC⁺13]. We begin by constructing an SPHF over the language of

valid ciphertexts, and then extend it to the language of commitments. We construct a projective hash function $\Pi_{\mathcal{E}}$ over the language $\mathcal{L}_m^{\text{pk}}$ defined as

$$\mathcal{L}_m^{\text{pk}} = \{c \mid \exists r, c = \mathcal{E}.\text{Enc}(\text{pk}, m; r)\}.$$

More precisely, we have the following expression

$$\mathcal{L}_m^{\text{pk}} = \left\{ (a, e, \pi) \mid \begin{array}{l} \text{Verify}(a, \pi) = \top, \\ \exists s \in \mathbb{G}, a = (s \star x_0, s \star x_1) \\ e \oplus m = \Pi.\text{ProjHash}(\text{pk}, a, s) \end{array} \right\}.$$

We define the projective hash function $\Pi_{\mathcal{E}}$ over $\mathcal{L}_m^{\text{pk}}$ as follow.

- $\Pi_{\mathcal{E}}.\text{Setup}(\lambda) \rightarrow \text{pp} : \text{return } \text{pk} = y \in \mathcal{X}^{\ell}$ the encryption key from the language $\mathcal{L}_m^{\text{pk}}$.
- $\Pi_{\mathcal{E}}.\text{KeyGen}(\text{pp}) \rightarrow \text{hk} : \text{output a random value } g \xleftarrow{\$} \mathbb{G}^{\ell}$.
- $\Pi_{\mathcal{E}}.\text{ProjKeyGen}(\text{hk} = g) \rightarrow \text{hp} \in \mathcal{X}^{\ell} : \text{set } \text{hp} = g \star y = y'$.
- $\Pi_{\mathcal{E}}.\text{Hash}(\text{hk} = g, c = (a, e, \pi)) \rightarrow h : \text{if } \text{Verify}(a, \pi) = \perp, \text{return a random element } h \xleftarrow{\$} \mathcal{X}^{\ell}. \text{ Else return } h = g \star (e \oplus m).$
- $\Pi_{\mathcal{E}}.\text{ProjHash}(\text{hp} = y', c = (a, e, \pi), w = s) \rightarrow h : \text{return } h = s \star y'.$

Theorem 9. *The projective hash function $\Pi_{\mathcal{E}}$ is ε -smooth, with ε bounded by*

$$\varepsilon \leq \text{Adv}_{\text{wPR}}(\mathcal{B}, 2\ell)$$

for any adversary $\mathcal{B}$ against the weak pseudorandomness of the group action.

Proof. We build the following reduction $\mathcal{R}$ to the weak pseudorandomness of the group action. Let $\mathcal{A}$ be an adversary against the smoothness of $\Pi_{\mathcal{E}}$ with advantage ε . If the proof π is not valid, $\Pi_{\mathcal{E}}.\text{Hash}$ already outputs a random value $h \xleftarrow{\$} \mathcal{X}^{\ell}$, thus we assume π to be a valid proof.

Setup. The reduction has access to an oracle $\mathcal{O}$ outputting random evaluations for a function F , which is either a random function or a function π_g for $g \in \mathbb{G}$. First, $\mathcal{R}$ asks the oracle $\mathcal{O}$ for couples $(y_i, F(y_i))$ for $i \in [\ell]$. It sets $\text{pk} = (y_i)_{i \in [\ell]}$ as the encryption key and $\text{hp} = (F(y_i))_{i \in [\ell]}$ as the projective key.

Main reduction. Then, $\mathcal{R}$ asks for new couples $(x_i, F(x_i))$ for $i \in [\ell]$ to the oracle $\mathcal{O}$. It derives a ciphertext $c = (a, e, \pi)$ by computing honestly a, π and c_0 using pk . Then, it sets m such that $e \oplus m = (x_i)_{i \in [\ell]}$ and the hash value is $h = (F(x_i))_{i \in [\ell]}$ where $F(x_i) = F(e_i \oplus m_i)$. Finally, the reduction sends the smoothness instance

$$(\mathcal{L}_{m'}^{\text{pk}}, \text{pp} = \text{pk}, c, \text{hp}, h)$$

for any $m \neq m'$ to the adversary, and answers the same guess as the adversary.

If F is a random function, then h follows a random distribution. Otherwise, if $F = \pi_g$, then h is an honestly computed hash of $c \notin \mathcal{L}_{m'}$ for the hashing key $\text{hk} = g$. Thus, the advantage ε of $\mathcal{A}$ against the smoothness of $\Pi_{\mathcal{E}}$ is reduced to solving the weak pseudorandomness of the group action in at most 2ℓ oracle queries, hence we obtain

$$\varepsilon \leq \text{Adv}_{\text{wPR}}(\mathcal{B}, 2\ell),$$

and the theorem follows. $\square$

5.4 SPHF over the language of valid commitments

Let $\mathbf{pk} = (y_{0,1}, y_{1,1}, \dots, y_{0,\ell}, y_{1,\ell}) \in \mathcal{X}^{2\ell}$ and $\mathbf{vtk} = (\mathbf{vtk}_0, \mathbf{vtk}_1)$ be respectively the encryption key of $\mathcal{E}$ and the chameleon hash verification key used in CS. For any message $m \in \{0, 1\}^\ell$, we define

$$\mathbf{pk}_m = (y_{m_i,i})_{i \in [\ell]} \in \mathcal{X}^\ell.$$

We extend the SPHF over the language $\mathcal{L}_m^{\text{CS}}$ of valid commitments defined as

$$\begin{aligned} \mathcal{L}_m^{\text{CS}} &= \{\text{com} \mid \exists s, \text{Open}(s, \mathbf{vtk}, \text{com}, m) = 1\} \\ &= \left\{ (h, b) \mid \begin{array}{l} \exists (s, d), b = \mathcal{E}.\text{Enc}(\mathbf{pk}, d; s) \\ \text{CH}.\text{Valid}(\text{ck}, \text{vk}, m, h, (d_{m_i,i})_{i \in [\ell]}, \mathbf{vtk}) = 1 \end{array} \right\}. \end{aligned}$$

There is no constraint on $d_{1-m_i,i}$ except in the encryption

$$d_{1-m_i,i} = (e \oplus c_0)_{1-m_i,i},$$

which always admits a solution. Hence, we can discard it and only focus on the constraints on $d_{m_i,i}$. Then, we can express the language $\mathcal{L}_m^{\text{CS}}$ in terms of $\mathcal{L}_d^{\mathbf{pk}_m}$ as

$$\begin{aligned} \mathcal{L}_m^{\text{CS}} &= \left\{ (h, b) \mid \begin{array}{l} \exists d \in \mathcal{X}^\ell, d = (\mathbf{vtk}_{m_i} \star h_i)_{i \in [\ell]} \\ \exists s, (b_{m_i})_{i \in [\ell]} = \mathcal{E}.\text{Enc}(\mathbf{pk}_m, d; s) \end{array} \right\} \\ &= \left\{ (h, b) \mid \begin{array}{l} \exists d \in \mathcal{X}^\ell, d = (\mathbf{vtk}_{m_i} \star h_i)_{i \in [\ell]} \\ (b_{m_i})_{i \in [\ell]} \in \mathcal{L}_d^{\mathbf{pk}_m} \end{array} \right\}, \end{aligned}$$

Thus, we derive a projective hash function Π_{CS} over $\mathcal{L}_m^{\text{CS}}$ in the following way.

- $\Pi_{\text{CS}}.\text{Setup}(\lambda) \rightarrow \mathbf{pp}$: return $\mathbf{pp} = \mathbf{pk}$ the encryption key from the language.
- $\Pi_{\text{CS}}.\text{KeyGen}(\mathbf{pp}) \rightarrow \tilde{\mathbf{hk}}$: let $\mathbf{hk} \leftarrow \Pi_{\mathcal{E}}.\text{KeyGen}(\lambda)$. Return $\tilde{\mathbf{hk}} = (\mathbf{hk}, \mathbf{vtk})$ with $\mathbf{vtk}$ the chameleon hash verification key used in CS.
- $\Pi_{\text{CS}}.\text{ProjKeyGen}(\tilde{\mathbf{hk}} = (\mathbf{hk}, \mathbf{vtk})) \rightarrow \mathbf{hp} \in \mathcal{X}^\ell$: return $\mathbf{hp} \leftarrow \Pi_{\mathcal{E}}.\text{ProjKeyGen}(\mathbf{hk})$.
- $\Pi_{\text{CS}}.\text{Hash}(\tilde{\mathbf{hk}}, \text{com} = (h, b)) \rightarrow \tilde{h}$: Let $d = (\mathbf{vtk}_{m_i} \star h_i)_{i \in [\ell]} \in \mathcal{X}^\ell$ and $b_m = (b_{m_i,i})_{i \in [\ell]}$. Return for the language $\mathcal{L}_d^{\mathbf{pk}_m}$

$$\tilde{h} \leftarrow \Pi_{\mathcal{E}}.\text{Hash}(\mathbf{hk}, b_m).$$

- $\Pi_{\text{CS}}.\text{ProjHash}(\mathbf{hp} = \mathbf{pk}_m, \text{com} = (h, (a, e, \pi)), w = s) \rightarrow \tilde{h}$: set $b_m = (b_{m_i,i})_{i \in [\ell]}$ and return

$$\tilde{h} \leftarrow \Pi_{\mathcal{E}}.\text{ProjHash}(\mathbf{hp}, b_m, s).$$

We prove the smoothness of Π_{CS} directly relies on the weak pseudorandomness of the group action.

Theorem 10. *The projective hash function Π_{CS} is ε_{CS} -smooth, where ε_{CS} is bounded by*

$$\varepsilon_{\text{CS}} \leq 3\text{Adv}_{\text{wPR}}(\mathcal{A}, 2\ell) + \text{Adv}_{\text{OE}}(\mathcal{A}).$$

Proof. We build the following reduction $\mathcal{R}$ to the smoothness of $\Pi_{\mathcal{E}}$. Let $\mathcal{A}$ be an adversary against the smoothness of Π_{CS} with advantage ε_{CS} .

Setup. Let $\mathcal{O}_{F_0}$ and $\mathcal{O}_{F_1}$ be two oracles for the weak pseudorandomness of the group actions *i.e.*, oracles to either random functions F_0 and F_1 or to functions $F_0 = \pi_g$ and $F_1 = \pi_{g'}$. To generate ck and vk , the reduction calls $\mathcal{O}_{F_0}$ and $\mathcal{O}_{F_1}$ to get couples $(x_i, F_0(x_i))$ and $(x'_i, F_1(x'_i))$ for $i \in [\ell]$, and sets

$$\text{ck} = (x = (x_i)_i, x' = (x'_i)_i), \text{vk} = ((F_0(x_i))_i, (F_1(x'_i))_i).$$

To compute the hash part (h, d) of the commitment for a message $m \in \{0, 1\}^\ell$, the reduction uses the oracles and asks for each $i \in [\ell]$ for a couple $(h_i, F_{m_i}(h_i))$ to $\mathcal{O}_{F_{m_i}}$ and sets $d_i = F_{m_i}(h_i)$.

Then, the reduction takes as input a smoothness instance for $\Pi_{\mathcal{E}}$ of the form

$$(\mathcal{L} = \mathcal{L}_d^{\text{pk}_m}, \text{pp} = \text{pk}_m, c = (a, e, \pi), \text{hp}, \tilde{h}),$$

for the language $\mathcal{L}_d^{\text{pk}_m}$. As before, we assume that π is valid, since otherwise we already know that h is sampled randomly.

Main reduction. We extend the ciphertext to be an encryption of a message of length 2ℓ as required in the commitment. Let $\text{pk} = (y_{i,j})_{j \in \{0,1\}, i \in [\ell]}$ such that $(y_{i,m_i})_{i \in [\ell]} = \text{pk}_m$ completed by random values. We use the online extractability of the ZKP to obtain a witness s such that $a = (s \star x_0, s \star x_1)$. Let $b = (a, e', \pi')$ be the extended ciphertext of c such that $(e'_{m_i,i})_{i \in [\ell]} = e$ and

$$(e'_{1-m_i,i})_{i \in [\ell]} \leftarrow \text{Enc}(\text{pk}_{1-m}, (d_{1-m_i,i})_{i \in [\ell]}; s)$$

for $(d_{1-m_i,i})_{i \in [\ell]} \xleftarrow{\$} \mathcal{X}^\ell$. The proof π' is computed honestly with witness s and challenge $H(a||e')$. The reduction sends to $\mathcal{A}$ the instance

$$(\mathcal{L}_m^{\text{CS}}, \text{pp} = (\text{pk}, \text{ck}, \text{vk}), \text{com} = (h, b), \text{hp}, \tilde{h}).$$

Then, the reduction answers with the same guess as $\mathcal{A}$.

If $F_0 = \pi_g$ and $F_1 = \pi_{g'}$, we have

$$\tilde{h} \leftarrow \Pi_{\text{CS}}.\text{Hash}(\text{hk} = (\text{hk}, \text{vtk} = (g, g')), \text{com})$$

if and only if $\tilde{h} \leftarrow \Pi_{\mathcal{E}}.\text{Hash}(\text{hk}, d)$. In this case, we can bound the adversary's advantage by $\varepsilon_{\mathcal{E}}$ the probability to break the smoothness of $\Pi_{\mathcal{E}}$.

Otherwise, if F_0 and F_1 are random functions, $\tilde{h}$ follows a uniform distribution. By applying successively the weak pseudorandomness of the group action on both $\mathcal{O}_{F_0}$ and $\mathcal{O}_{F_1}$, the advantage of the adversary to distinguish between outputs of random functions F_0 and F_1 and outputs of $F_0 = \pi_g, F_1 = \pi_{g'}$ is bounded by $2 \cdot \text{Adv}_{\text{wPR}}(\mathcal{A}, 2\ell)$. Hence, the advantage ε_{CS} of $\mathcal{A}$ against the smoothness of Π_{CS} is bounded by

$$\varepsilon_{\text{CS}} \leq \varepsilon_{\mathcal{E}} + 2 \cdot \text{Adv}_{\text{wPR}}(\mathcal{A}, 2\ell) + \text{Adv}_{\text{OE}}(\mathcal{A}),$$

since we assume that the extraction of the witness s succeeded. As we know from the previous Theorem 9 that $\varepsilon_{\mathcal{E}} \leq \text{Adv}_{\text{wPR}}(\mathcal{A}, 2\ell)$, the theorem follows. $\square$

Theorem 11. *The projective hash function Π_{CS} is pseudorandom under the assumption that the group action is weakly pseudorandom, more precisely, we have*

$$\text{Adv}_{\text{PR}}(\mathcal{A}) \leq \text{Adv}_{\text{wPR}}(\mathcal{A}, 2\ell + 2) + \text{Adv}_{\text{ZK}}(\mathcal{A}).$$

Proof. We build the following reduction $\mathcal{R}$ to the weak pseudorandomness of the EGA. Let $\mathcal{A}$ be an adversary against the pseudorandomness of Π_{CS} with advantage ε .

The adversary sends the reduction a message m and a language $\mathcal{L}_{m'}$. Let $\mathcal{O}_F$ be the oracle for the weak pseudorandomness of the group actions. The reduction starts by computing a commitment honestly, but asks the oracle $\mathcal{O}_F$ to compute $a = (F(x_0), F(x_1))$ and $c_0 = (F(y_{i,j}))_{i \in [\ell], j \in \{0,1\}}$ for $\mathbf{pk} = (y_{i,j})$ without the randomness s . It uses the simulator of the ZKP to compute π . Then for a hashing key $\mathbf{hk} = g \in \mathbb{G}^\ell$, the reduction computes $\tilde{h} = g \star (F(y_{i,m_i}))_{i \in [\ell]}$ for the language $\mathcal{L}_{m'}$. Finally, it sends $(\text{com}, \text{hp}, \tilde{h})$ to $\mathcal{A}$.

The reduction answers with the same guess as $\mathcal{A}$. If $F = \pi_s$, then $\tilde{h}$ is the projective hash on a commitment with witness s . If F is a random function, then $\tilde{h}$ follows a uniform distribution. Hence, the advantage $p_{\mathcal{R}}$ for the reduction to break the weak pseudorandomness of the group action in $2\ell + 2$ queries is at least

$$p_{\mathcal{R}} \geq \varepsilon - \text{Adv}_{\text{ZK}}(\mathcal{A}).$$

□

6 UC-secure Oblivious Transfer

We finally instantiated all of the blocks needed to build an adaptively secure OT protocol. We start by recalling the ideal functionality for an OT protocol, and then we recall the generic construction from [BC15]. From that, we can finally estimate the resulting cost that would yield our instantiation from group actions, in computation as well as in communications.

6.1 Ideal Functionality

The functionality $\mathcal{F}$ is parametrized by a security parameter λ . It interacts with a simulator $\mathcal{S}$ and a set of parties $(P_i)_{i \in [n]}$.

- **Upon receiving from P_i an input** $(\text{Send}, \text{sid}, \text{ssid}, P_i, P_j, (m_1, \dots, m_k))$, **with** $m_i \in \{0,1\}^\lambda$: record the tuple $(\text{sid}, \text{ssid}, P_i, P_j, (m_1, \dots, m_k))$ and reveal $(\text{Send}, \text{sid}, \text{ssid}, P_i, P_j)$ to $\mathcal{S}$. Ignore further Send message with the same ssid from P_i .
- **Upon receiving an input** $(\text{Receive}, \text{sid}, \text{ssid}, P_i, P_j, s)$ **from P_j , with $s \in [k]$** : record the tuple $(\text{sid}, \text{ssid}, P_i, P_j, s)$ and reveal $(\text{Receive}, \text{sid}, \text{ssid}, P_i, P_j)$ to $\mathcal{S}$. Ignore further Receive message with the same ssid from P_j .
- **Upon receiving an input** $(\text{Sent}, \text{sid}, \text{ssid}, P_i, P_j)$ **from $\mathcal{S}$** : ignore if the message $(\text{sid}, \text{ssid}, P_i, P_j, (m_1, \dots, m_k))$ or $(\text{sid}, \text{ssid}, P_i, P_j, s)$ is not recorded, otherwise send $(\text{Sent}, \text{sid}, \text{ssid}, P_i, P_j)$ to P_i . Ignore further Sent message with the same ssid from $\mathcal{S}$.
- **Upon receiving an input** $(\text{Received}, \text{sid}, \text{ssid}, P_i, P_j)$ **from $\mathcal{S}$** : ignore if the message $(\text{sid}, \text{ssid}, P_i, P_j, (m_1, \dots, m_k))$ or $(\text{sid}, \text{ssid}, P_i, P_j, s)$ is not recorded, otherwise send $(\text{Received}, \text{sid}, \text{ssid}, P_i, P_j, m_s)$ to P_j . Ignore further Received message with the same ssid from $\mathcal{S}$.

Figure 7: The ideal functionality $\mathcal{F}$ for 1-out-of- k Oblivious Transfer. We denote by P_i the sender, and P_j the receiver. The session id is sid , and ssid the sub session id to allow for clones or forks of a session.

The goal of the UC framework is to ensure protocols remain secure, even if arbitrarily composed with other protocols. The framework relies on the indistinguishability between the real world and an ideal world. An ideal functionality for a protocol is a trusted third party that computes the desired outcome of the protocol perfectly and securely. We provide the ideal functionality $\mathcal{F}$ for oblivious transfer in Figure 6.1. To prove that our protocol perfectly emulates $\mathcal{F}$, we construct a simulator $\mathcal{S}$ for any polynomial adversary such that no polynomial environment can distinguish between the real world (real players interacting with themselves and $\mathcal{A}$) and the ideal world (dummy players interacting with $\mathcal{S}$ and $\mathcal{F}$) with non-negligible advantage. We consider *adaptive* adversaries *i.e.*, that can corrupt users whenever it wants to. After corrupting a player, $\mathcal{A}$ has access to all its internal state and private values, and takes its control.

6.2 Generic Adaptive OT protocol

We follow the generic construction from [BC15] to obtain a 1-out-of- k oblivious transfer protocol. It builds onto an SPHF-friendly E^2 commitment scheme [ABB⁺13] denoted by CS. They construct such an E^2 commitment scheme from an SPHF-friendly IND-CCA encryption $\mathcal{E}$, and a verifiable chameleon hash function CH, as we saw in the previous sections. This results in a 3-round UC-secure OT protocol, for which we recall the construction in Figure 8.

Note that this construction also requires an encryption scheme E , that is any IND-CPA encryption scheme. This scheme is only used by the user to encrypt the nonce J , and the keys are refreshed at each session to preserve both forward and backward secrecy.

Theorem 12 ([BC15]). *The OT scheme described in Figure 8 is UC-secure against adaptive adversaries, assuming reliable erasures and authenticated channels, and assuming the chameleon hash is verifiable and collision resistant, the encryption $\mathcal{E}$ is IND-CCA and admits an SPHF on the language of valid ciphertexts.*

6.3 Complexity analysis

We provide a complexity analysis of our protocol. We summarize the communication cost, as well as the computation cost in terms of the number of group action evaluations in Table 2. We denote by $|\mathcal{X}|$ the size of an element of the set $\mathcal{X}$ and by $|\mathbb{G}|$ the size of an element from the group $\mathbb{G}$.

For the user, computing a commitment requires computing a chameleon hash with 2ℓ group action evaluation and an encryption for $2 + 2\ell + 2\lambda$ evaluations. As for the server, computing a projective key requires ℓ evaluations, and a hash $\Pi_{CS}.\text{Hash}$ requires $2\lambda + 2\ell$ evaluations.

Round	Communication	Complexity (# evals)
1 (Server)	$ \text{pk} + 2 \cdot \mathcal{X} $	2
2 (User)	$(3\ell + 2) \cdot \mathcal{X} + \lambda \cdot \mathbb{G} $	$2 + 4\ell + 2\lambda$
3 (Server)	$2\ell k \cdot \mathcal{X} $	$k(2\lambda + 3\ell)$

Table 2: Complexity analysis for our 1-out-of- k OT protocol to request an index $j \in \{0, 1\}^\ell$. Let λ be the security parameter. We denote by $|\text{pk}|$ the public key for any IND-CPA encryption scheme.

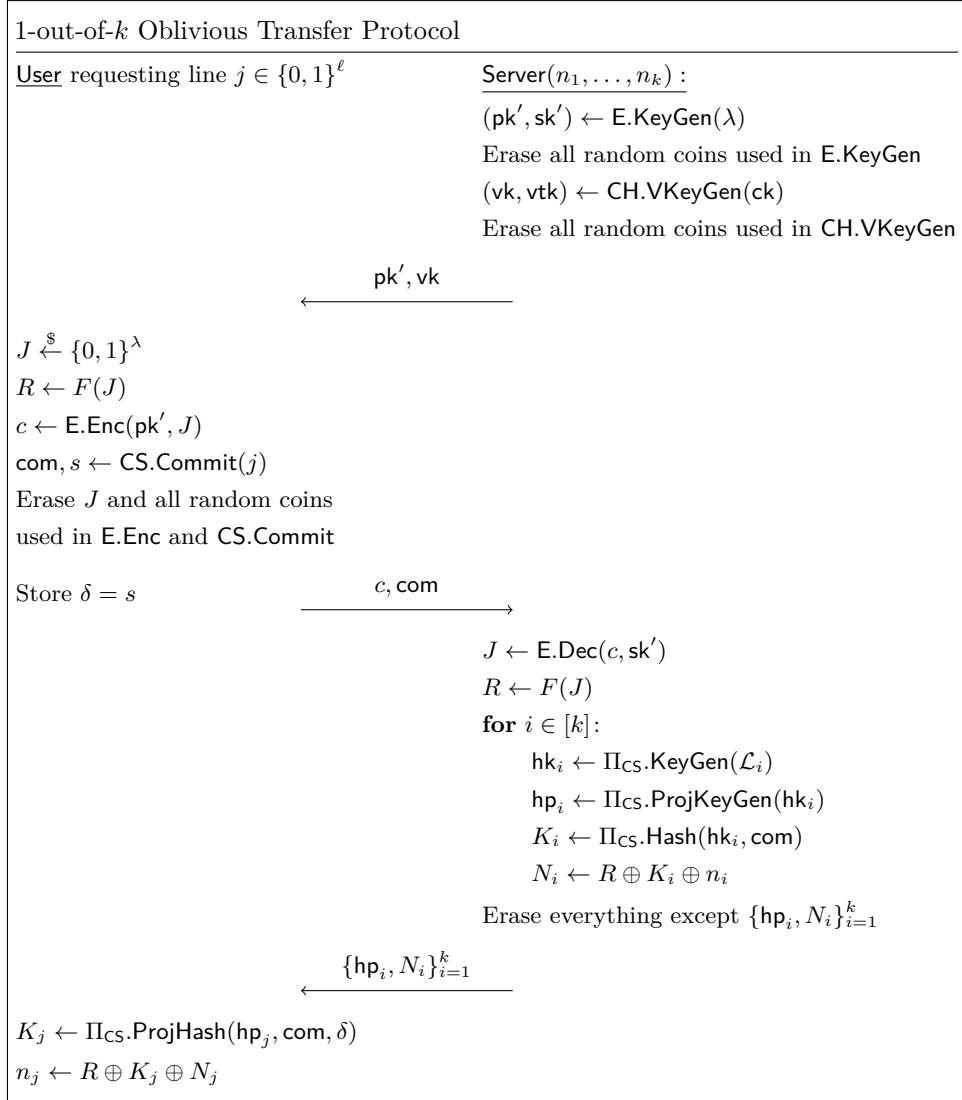


Figure 8: Generic construction of a 1-out-of- k oblivious transfer protocol [BC15]. Let $(ck, tk) \leftarrow CH.KeyGen(\lambda)$ be chameleon hash keys already generated in the setup (the ones used by the commitment scheme CS as detailed in Section 5). Let F be a pseudorandom generator, E any IND-CPA encryption scheme and Π_{CS} the SPHF over the language $\mathcal{L}_m$ of valid commitments to m for CS .

Conclusion

We investigated the feasibility of constructing an SPHF-friendly E^2 commitment scheme from effective group actions, a foundational component for realizing OT protocols in the absence of suitable public-key encryption schemes. Building upon the generic OT construction of Blazy and Chevalier [BC15], which was already instantiated with the DDH, LWE and DCR assumptions, we provide for the first time an instantiation of all the required primitives—namely, an SPHF-friendly publicly verifiable IND-CCA encryption scheme and a verifiable chameleon hash function—within the group action framework. Our construction requires a weakly pseudorandom regular EGA, which can be in practice

instantiated with the decisional CSIDH assumption. Our contribution not only enables an adaptively secure OT protocol in the UC framework from group actions but also lays the groundwork for a broader class of cryptographic primitives. The components we construct have independent utility: from chameleon hash functions used in digital signature [BKKP15] or two-tier signature schemes [BS07] to publicly verifiable encryption scheme in group signatures [CD99], and E^2 commitment scheme to universally composable commitment schemes [Cre02]. As such, our results enrich the toolkit available for post-quantum cryptography.

Acknowledgments. This work is partly supported by ANR Project PRIVA-SIQ (ANR-23-CE39-0008).

References

- [ABB⁺13] Michel Abdalla, Fabrice Benhamouda, Olivier Blazy, Céline Chevalier, and David Pointcheval. Sphf-friendly non-interactive commitments. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 214–234. Springer, 2013.
- [ACdMT05] Giuseppe Ateniese, Daniel H. Chou, Breno de Medeiros, and Gene Tsudik. Sanitizable signatures. In Sabrina De Capitani di Vimercati, Paul F. Syverson, and Dieter Gollmann, editors, *ESORICS 2005*, volume 3679 of *LNCS*, pages 159–177, Milan, Italy, September 12–14, 2005. Springer, Heidelberg, Germany. doi:[10.1007/11555827_10](https://doi.org/10.1007/11555827_10).
- [ACP09] Michel Abdalla, Céline Chevalier, and David Pointcheval. Smooth projective hashing for conditionally extractable commitments. In *Annual International Cryptology Conference*, pages 671–689. Springer, 2009.
- [ADFMP20] Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis. Cryptographic group actions and applications. In *Advances in Cryptology—ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part II 26*, pages 411–439. Springer, 2020.
- [AMPS21] Navid Alamati, Hart Montgomery, Sikhar Patranabis, and Pratik Sarkar. Two-round adaptively secure mpc from isogenies, lpn, or cdh. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 305–334. Springer, 2021.
- [BC15] Olivier Blazy and Céline Chevalier. Generic construction of uc-secure oblivious transfer. In *Applied Cryptography and Network Security: 13th International Conference, ACNS 2015, New York, NY, USA, June 2-5, 2015, Revised Selected Papers 13*, pages 65–86. Springer, 2015.
- [BCC88] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *Journal of computer and system sciences*, 37(2):156–189, 1988.
- [BFP⁺26] Andrea Basso, Luca De Feo, Sikhar Patranabis, Ilinca Radulescu, and Benjamin Wesolowski. Forensic categories: a framework for SQIsign-like primitives. Cryptology ePrint Archive, Paper 2026/1171, 2026. URL: <https://eprint.iacr.org/2026/1171>.

- [BGJ⁺18] Saikrishna Badrinarayanan, Vipul Goyal, Abhishek Jain, Yael Tauman Kalai, Dakshita Khurana, and Amit Sahai. Promise zero knowledge and its applications to round optimal mpc. In *Annual International Cryptology Conference*, pages 459–487. Springer, 2018.
- [BHBC⁺13] Fabrice Ben Hamouda, Olivier Blazy, Céline Chevalier, David Pointcheval, and Damien Vergnaud. Efficient uc-secure authenticated key-exchange for algebraic languages. In *Public-Key Cryptography–PKC 2013: 16th International Conference on Practice and Theory in Public-Key Cryptography, Nara, Japan, February 26–March 1, 2013. Proceedings 16*, pages 272–291. Springer, 2013.
- [BKKP15] Olivier Blazy, Saqib A Kakvi, Eike Kiltz, and Jiaxin Pan. Tightly-secure signatures from chameleon hash functions. In *Public-Key Cryptography–PKC 2015: 18th IACR International Conference on Practice and Theory in Public-Key Cryptography, Gaithersburg, MD, USA, March 30–April 1, 2015, Proceedings 18*, pages 256–279. Springer, 2015.
- [BKM⁺21] Andrea Basso, Péter Kutas, Simon-Philipp Merz, Christophe Petit, and Antonio Sanso. Cryptanalysis of an oblivious prf from supersingular isogenies. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 160–184. Springer, 2021.
- [BKW20] Dan Boneh, Dmitry Kogan, and Katharine Woo. Oblivious pseudorandom functions from isogenies. In *Advances in Cryptology–ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part II 26*, pages 520–550. Springer, 2020.
- [BLPV18] Fabrice Benhamouda, Huijia Lin, Antigoni Polychroniadou, and Muthuramakrishnan Venkitasubramaniam. Two-round adaptively secure multiparty computation from standard assumptions. In *Theory of Cryptography: 16th International Conference, TCC 2018, Panaji, India, November 11–14, 2018, Proceedings, Part I 16*, pages 175–205. Springer, 2018.
- [BMM⁺23] Saikrishna Badrinarayanan, Daniel Masny, Pratyay Mukherjee, Sikhar Patranabis, Srinivasan Raghuraman, and Pratik Sarkar. Round-optimal oblivious transfer and mpc from computational csidh. In *IACR International Conference on Public-Key Cryptography*, pages 376–405. Springer, 2023.
- [BNOB18] Paulo Barreto, Anderson Nascimento, Glaucio Oliveira, and Waldyr Benits. Supersingular isogeny oblivious transfer (siot). *arXiv preprint arXiv:1805.06589*, 2018.
- [BS07] Mihir Bellare and Sarah Shoup. Two-tier signatures, strongly unforgeable signatures, and fiat-shamir without random oracles. In *International Workshop on Public Key Cryptography*, pages 201–216. Springer, 2007.
- [BY91] Gilles Brassard and Moti Yung. One-way group actions. In *Advances in Cryptology-CRYPTO’90: Proceedings 10*, pages 94–107. Springer, 1991.
- [Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 136–145. IEEE, 2001.

- [CD99] Jan Camenisch and Ivan Damgaard. Verifiable encryption and applications to group signatures and signature sharing. Cryptology ePrint Archive, Paper 1999/008, 1999. URL: <https://eprint.iacr.org/1999/008>.
- [CD23a] Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCs*, pages 423–447, Lyon, France, April 23–27, 2023. Springer, Heidelberg, Germany. doi:10.1007/978-3-031-30589-4_15.
- [CD23b] Wouter Castryck and Thomas Decru. An efficient key recovery attack on sidh. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 423–447. Springer, 2023.
- [CKL03] Ran Canetti, Eyal Kushilevitz, and Yehuda Lindell. On the limitations of universally composable two-party computation without set-up assumptions. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 68–86. Springer, 2003.
- [CLM⁺18] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. Csidh: an efficient post-quantum commutative group action. In *Advances in Cryptology—ASIACRYPT 2018: 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2–6, 2018, Proceedings, Part III 24*, pages 395–427. Springer, 2018.
- [Cou06] Jean-Marc Couveignes. Hard homogeneous spaces. 2006.
- [Cré87] Claude Crépeau. Equivalence between two flavours of oblivious transfers. In *Conference on the Theory and Application of Cryptographic Techniques*, pages 350–354. Springer, 1987.
- [Cre02] Giovanni Di Crescenzo. Equivocable and extractable commitment schemes. In *International Conference on Security in Communication Networks*, pages 74–87. Springer, 2002.
- [CS02] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *International conference on the theory and applications of cryptographic techniques*, pages 45–64. Springer, 2002.
- [DGH⁺20] Nico Döttling, Sanjam Garg, Mohammad Hajiabadi, Daniel Masny, and Daniel Wichs. Two-round oblivious transfer from cdh or lpn. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 768–797. Springer, 2020.
- [dSGOPS20] Cyprien Delpech de Saint Guilhem, Emmanuela Orsini, Christophe Petit, and Nigel P Smart. Semi-commutative masking: a framework for isogeny-based protocols, with an application to fully secure two-round isogeny-based ot. In *International Conference on Cryptology and Network Security*, pages 235–258. Springer, 2020.
- [EGL85] Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. *Communications of the ACM*, 28(6):637–647, 1985.

- [EKKP20] Ali El Kaafarani, Shuichi Katsumata, and Federico Pintore. Lossy csi-fish: efficient signature scheme with tight reduction to decisional csidh-512. In *IACR International Conference on Public-Key Cryptography*, pages 157–186. Springer, 2020.
- [Fis05] Marc Fischlin. Communication-efficient non-interactive proofs of knowledge with online extractors. In *Annual International Cryptology Conference*, pages 152–168. Springer, 2005.
- [FMV19] Daniele Friolo, Daniel Masny, and Daniele Venturi. A black-box construction of fully-simulatable, round-optimal oblivious transfer from strongly uniform key agreement. In *Theory of Cryptography Conference*, pages 111–130. Springer, 2019.
- [FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Annual international cryptology conference*, pages 537–554. Springer, 1999.
- [HK12] Shai Halevi and Yael Tauman Kalai. Smooth projective hashing and two-message oblivious transfer. *Journal of Cryptology*, 25(1):158–193, 2012.
- [JDF11] David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *International workshop on post-quantum cryptography*, pages 19–34. Springer, 2011.
- [JZ22] Xi Jia and Meng Zhang. Encrypted packet inspection based on oblivious transfer. *Security and Communication Networks*, 2022(1):4743078, 2022.
- [KK25] Ferhat Karakoç and Alptekin Küpçü. Enabling two-party secure computation on set intersection. *IEEE Transactions on Dependable and Secure Computing*, 2025.
- [KO97] Eyal Kushilevitz and Rafail Ostrovsky. Replication is not needed: Single database, computationally-private information retrieval. In *Proceedings 38th annual symposium on foundations of computer science*, pages 364–373. IEEE, 1997.
- [KR97] Hugo Krawczyk and Tal Rabin. Chameleon hashing and signatures. *Internet-http://wwwresearch.ibm.com/security/projects.html*, 1997.
- [KR00] Hugo Krawczyk and Tal Rabin. Chameleon signatures. In *NDSS 2000*, San Diego, CA, USA, February 2–4, 2000. The Internet Society.
- [LGDdSG21] Yi-Fu Lai, Steven D Galbraith, and Cyprien Delpech de Saint Guilhem. Compact, efficient and uc-secure isogeny-based oblivious transfer. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 213–241. Springer, 2021.
- [MM22] Luciano Maino and Chloe Martindale. An attack on sidh with arbitrary starting curve. *Cryptology ePrint Archive*, 2022.
- [MMP⁺23] Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A direct key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 448–471, Lyon, France, April 23–27, 2023. Springer, Heidelberg, Germany. doi:10.1007/978-3-031-30589-4_16.

- [MR19] Daniel Mansy and Peter Rindal. Endemic oblivious transfer. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 309–326, 2019.
- [Nao91] Moni Naor. Bit commitment using pseudorandomness. *Journal of cryptology*, 4(2):151–158, 1991.
- [Rab81] Michael O Rabin. How to exchange secrets with oblivious transfer. *Cryptology ePrint Archive*, 1981.
- [Rob23] Damien Robert. Breaking SIDH in polynomial time. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 472–503, Lyon, France, April 23–27, 2023. Springer, Heidelberg, Germany. doi:[10.1007/978-3-031-30589-4_17](https://doi.org/10.1007/978-3-031-30589-4_17).
- [SS05] L’ubica Staneková and Martin Stanek. Fast contract signing with batch oblivious transfer. In *IFIP International Conference on Communications and Multimedia Security*, pages 1–10. Springer, 2005.
- [ST01] Adi Shamir and Yael Tauman. Improved online/offline signature schemes. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 355–367, Santa Barbara, CA, USA, August 19–23, 2001. Springer, Heidelberg, Germany. doi:[10.1007/3-540-44647-8_21](https://doi.org/10.1007/3-540-44647-8_21).
- [Vit18] Vanessa Vitse. Simple oblivious transfer protocols compatible with kummer and supersingular isogenies. *Cryptology ePrint Archive*, 2018.