Secure Oblivious Transfer from Semi-Commutative Masking

Cyprien Delpech de Saint Guilhem[1,2], Emmanuela Orsini[1], Christophe Petit[3], and Nigel P. Smart[1,2]

[1] imec-COSIC, KU Leuven, Belgium
[2] Dept Computer Science, University of Bristol, United Kingdom
[3] School of Computer Science, University of Birmingham, United Kingdom
cyprien.delpechdesaintguilhem@kuleuven.be, emmanuela.orsini@kuleuven.be, christophe.f.petit@gmail.com, nigel.smart@kuleuven.be

Abstract. In this work we first define semi-commutative (invertible) masking structures which present a simple abstraction to capture the various examples of protocol design that are based on exponentiation-only style operations (such as discrete logarithm and isogeny based cryptography). We discuss two possible instantiations of our structure: The first is based on commutative group actions and captures both the action of exponentiation in the discrete logarithm setting and also the action of the class group of commutative endomorphism rings of elliptic curves, in the style of the CSIDH key-exchange protocol; the second is based on the semi-commutative action of isogenies of supersingular elliptic curves, in the style of the SIDH key-exchange protocol. We then design two oblivious transfer protocols using this structure and prove that they securely UC-realise the standard OT-functionality in the Random-Oracle-hybrid model against passive adversaries with static corruptions. This paper thus introduces the first oblivious transfer protocol based on supersingular isogenies that is proven secure in the UC framework.

1 Introduction

Oblivious Transfer (OT) is a fundamental cryptographic building block, originally proposed by Rabin in 1981, [Rab81]. OT protocols can be used to create various other high level cryptographic protocols, and have even been shown to be complete for general multi-party computation [Kil88]. In its most basic form, OT is often presented in terms of so-called 1-out-of-2 OT, as first introduced by Even, Goldreich and Lempel [EGL82]. Here, a sender has two messages m_0 and m_1, whilst a receiver has a choice bit $\mathsf{c} \in \{0, 1\}$ which allows him to obtain m_{c}. At the end of the protocol the sender learns nothing about the bit c, and the receiver learns nothing about $m_{1-\mathsf{c}}$.

In considering the state-of-the-art in post-quantum OT protocols we need to understand the various models that are used and the types of OT protocols which are enabled.

There are three *types* of OT: In the first the two messages m_0 and m_1 can only be selected from the set $\{0, 1\}$, this is so-called *bit*-OT. In the second type the two messages can be arbitrary messages selected by the sender, but of the same length, leading to so-called *string*-OT. Finally, we have a variant of string-OT, but where the messages are purely random strings of the same length over which the sender has no control, so called *random*-OT. It is easy to build string-OT from random-OT using an encryption algorithm, and in addition random-OT can be used as a building block in many multi-party computation protocols [NNOB12].

Secondly, one has to consider the overall assumptions; whether the construction assumes the Random Oracle Model (ROM), or a Common Reference String (CRS), or has no such assumption. As well as the underlying hard problem on which security is based, for example CDH or DDH in abelian groups, or Quadratic Residuosity (QR) for RSA-style groups, Learning Parity with Noise (LPN) or Learning with Errors (LWE) for lattice based constructions, or McEliece style problems

for coding theory constructions, or supersingular isogeny (SSI) problems for isogeny based systems.

The third consideration is about the security model. Here protocols can be proved secure in a game-based setting, or a simulation based setting with the latter being divided into those proofs which only provide stand-alone security, and those which provide full Universal Composability (UC) security. The adversaries can be assumed to be fully adaptive (they can decide to corrupt the sender or receiver as the protocol is running), or static (they must choose which party to corrupt at the start). In addition models can capture passive adversaries (which follow the protocol), or malicious ones (which can deviate from the protocol execution). Another efficiency consideration is how many rounds of communication each protocol requires.

Many efficient OT protocols have been built out of discrete logarithm based protocols; see for example the early influential work of Naor and Pinkas [NP01]. The current most efficient and most secure protocols are the DDH and DCR-based constructions of [PVW08] and the CDH-based construction of [BDD+17] which achieves roughly the same computational complexity. Also, the construction of [BDD+17] supersedes the well-known protocol of Chou and Orlandi [CO15a] whose proof has been showed to be flawed [Orl18]. A discussion from the original authors regarding these shortcomings has been added in [CO15b, Section 1.1].

However, any future quantum computer would enable efficient breaking of security of these discrete logarithm based variants. With that threat in mind, both the works of [PVW08] and [BDD+17] were shown to benefit from generalisations to post-quantum variants based on LWE, LPN and McEliece). Also, further lattice based constructions have recently been given in [BD18,Zen18]. However, one would clearly not want to rely on a single assumption to ensure security and there has also been much work into other hard computational problems. One such alternative problem is that of isogeny computation on elliptic curves. Research on protocols based on isogenies has already led to the realisation of key exchange protocols, zero-knowledge proofs and identification protocols [DFJP14,CLM+18].

1.1 Our Contribution

In the traditional discrete logarithm setting, for a given group G, not only can elements be exponentiated – as in the Diffie-Hellman (DH) key exchange with g^a, g^b and g^{ab} – but they can also be multiplied together – with $g^a \cdot g^b = g^{a+b}$. This ability to compose group elements grants more flexibility to protocol designers and enables the realisation of simple and efficient protocols such as Schnorr's zero-knowledge proof of knowledge (ZKPoK) [Sch90]. However, the mathematical structures that are attractive for post-quantum cryptography do not possess as much structure, which is in part why they are conjectured to be more resistant to quantum algorithms.

We first of all focus on a "exponentiation-only" framework to remove the additional freedom of the second operation (i.e. the group multiplication in the discrete logarithm setting). An additional challenge of the setting of the Supersingular Isogeny Diffie-Hellman (SIDH) key exchange, first proposed by De Feo et al. [DFJP14], is that the commutativity of the "exponentiation" operation no longer holds in general. The protocol proposed by De Feo, Jao and Plût works around this at the cost of heavy notation and mathematical mechanisms.

Our first contribution is therefore the definition of a new structure called a *semi-commutative invertible masking scheme* which captures the absence of full commutativity in this new setting, within a framework that is notationally simple and intuitive. We also proceed to show that these schemes can be realised both from the traditional discrete logarithm setting and also from the more recent elliptic curve isogeny setting, both ordinary and supersingular. Furthermore, we discuss the computational problems that we introduce and show that they are in fact very close to the existing problems in the litterature for the respective settings.

Our second contribution is two OT protocols constructed from our new semi-commutative invertible masking scheme. Both protocols achieve UC security against *passive* adversaries with *static* corruptions in the ROM. The first protocol is inspired by the Shamir 3-pass key transport protocol which we modify to satisfy the requirements of oblivious transfer using only two passes. The second protocol is an adaptation of the key-exchange based protocol of Chou and Orlandi [CO15a] to the "exponentiation-only" setting.

While it would be fairly simple to modify our protocols to achieve game-based active security, it is not clear how to achieve active security in the UC-model. Following the blueprint of previous works [BPRS17,BDD+17] leads to new difficulties due to the lack of a rich algebraic structure of our semi-commutative construction, resulting in (potentially) extra rounds of communications. We leave investigation of actively secure variants to future work.

Related work. In addition to the works in other settings mentioned above, there has been only one other very recent work in building post-quantum secure OT protocols from isogenies on elliptic curves [BOB18] that emerged concurrently to ours. This work only proves security in the stand-alone model and it provides no framework with which to derive new protocols. In Table 1 we summarize the current state-of-the-art in both the pre- and post-quantum settings.

Reference	Type	Set-up	Comm.	Security	Assumptions
[CO15a]	string	ROM	2 rds	game-based	gap-DH
[BDD+17]	string	ROM	3 rds	UC, malicious, adaptive	CDH, LPN, McEliece
[BPRS17]	string	ROM	2 rds	GUC, malicious, adaptive	DDH
[PVW08]	bit	CRS	2 rds	UC, malicious, static	DDH, DCR, QR, Regev (LWE)
[BD18]	string		2 rds	game-based	LWE
[Zen18]	string		6 rds	stand-alone, malicious, static	LWE
[BOB18]	string		3 rds	stand-alone, passive, static	SSI-DDH, SSI-CDH
This work	string	ROM	3 rds	UC, passive, static	CDH, CSIDH
This work	string	ROM	2 rds	UC, passive, static	CDH, CSIDH

Table 1: State-of-the-art OT protocols in the pre- and post-quantum settings.

1.2 Paper Overview

After a short preliminaries in Section 2, we present in Section 3 our abstraction of semi-commutative invertible masking scheme. As we do so, we illustrate the construction using the standard discrete logarithm setting, so as to fix ideas for the reader. In Section 4 we show how our abstraction can be instantiated from group actions. This includes not only the traditional discrete logarithm construction, but also the construction from *hard homogeneous spaces*, [CLM+18,Cou06]. In Section 5 we show how to instantiate our abstraction using isogenies between supersingular elliptic curves and discuss how its security relates to the computational problems that arise from the SIDH key-exchange setting.

Then in Section 6 we present two protocols which utilizes this abstraction to generalise from two discrete logarithm based OT protocols. In the first one we present a protocol derived from the Shamir-3-Pass key transport scheme, and one protocol derived from the Chou and Orlandi methodology for constructing OT from the Diffie-Hellmen key agreement protocol.

2 Preliminaries

We denote by λ the computational security parameter. We say that a function $f : \mathbb{N} \to \mathbb{N}$ is *negligible* if for every positive polynomial $p(\cdot)$ and all sufficiently large λ it holds that $f(\lambda) < \frac{1}{p(\lambda)}$. The function f is *noticeable* (or non-negligible) if there exists a positive polynomial $p(\cdot)$ such that for all sufficiently large λ it holds that $f(\lambda) \geq \frac{1}{p(\lambda)}$. We denote by $a \xleftarrow{\$} A$ the uniform sampling of a from a set A, and $\overset{\text{c}}{\approx}$ and $\overset{\text{s}}{\approx}$ computational and statistical indistinguishability, respectively.

We denote by $\mathcal{E} = \{(\mathsf{KGen}_{\mathcal{E}}, \mathsf{Enc}, \mathsf{Dec}), (\mathcal{K}_{\mathcal{E}}, \mathcal{M}_{\mathcal{E}}, \mathcal{C}_{\mathcal{E}})\}$ a symmetric encryption scheme, where $\mathcal{K}_{\mathcal{E}}, \mathcal{M}_{\mathcal{E}}, \mathcal{C}_{\mathcal{E}}$ are the key-space, message-space and ciphertext-space, respectively. We refer the reader to Appendix A for the precise definitions that we require.

Universal Composability We prove security of our protocols in the universal composability (UC) framework [Can01], with static passive corruptions. In particular, we will prove that our protocols UC-realize the OT functionality $\mathcal{F}_{\mathsf{OT}}$ in the $\mathcal{F}_{\mathsf{RO}}$-hybrid model, where the OT functionality is presented in Figure 1 and the random oracle (RO) functionality is described in Figure 2 We refer to Appendix B for a general overview of the UC framework.

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

PARAMETER: n length of the bit-strings

- Upon receiving $(P_S, \mathsf{sid}, \mathsf{m}_0, \mathsf{m}_1)$ from P_S, check if a $(\mathsf{sid}, \mathsf{c})$ was previously stored. If yes, send m_c to P_R; if not, store $(\mathsf{sid}, \mathsf{m}_0, \mathsf{m}_1)$ and continue to run.
- Upon receiving $(P_R, \mathsf{sid}, \mathsf{c})$ from P_R, check if a $(\mathsf{sid}, \mathsf{m}_0, \mathsf{m}_1)$ was previously stored. If yes, send m_c to P_R; if not, store $(\mathsf{sid}, \mathsf{c})$ and continue to run.

Fig. 1: Oblivious transfer functionality

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

The functionality is parametrized by a domain $\mathcal{D}$ and range $\mathcal{R}$. It keeps a list L of pairs of values, which is initially empty and proceeds as follows:

- Upon receiving a value $(\mathsf{sid}, m), m \in \mathcal{D}$, if there is a pair $(m, \hat{h})$, $\hat{h} \in \mathcal{R}$, in the list L, set $h = \hat{h}$. Otherwise choose $h \xleftarrow{\$} \mathcal{R}$ and store the pair (m, h) in L.
- Reply to the activating machine with (sid, h).

Fig. 2: Random oracle functionality

3 Semi-Commutative Masking

In this section we define our abstraction of *semi-commutative invertible masking structure*. This is a structure which enables us to discuss a number of protocols in three different settings; a traditional discrete logarithm style setting, the setting of hard homogeneous spaces related to class groups of the endomorphism ring of elliptic curves (as used in [CLM+18]), and the case of supersingular isogenies (as used in [DFJP14]). This abstraction will allow us to define succintly our protocols, without needing to worry about implementation details. It may also allow other authors to find new applications of supersingular isogenies by abstracting existing discrete logarithm based protocols. To help fix ideas in the reader's mind we will illustrate the abstraction by looking at the specialisation to discrete logarithms in a finite field $\mathbb{F}_p$ such that $q = (p-1)/2$ is prime and g is an element of order q.

A masking structure $\mathcal{M}$ is defined over a set X. Each element $x \in X$ may have multiple *representations*, and we define R_x to be the set of representations of an element $x \in X$. We denote the set of all such sets by $R_X = \{R_x\}_{x \in X}$. The sets of representatives are assumed to be disjoint, i.e.

$$\forall x, x' \in X \text{ s.t. } x \neq x' \quad : \quad R_x \cap R_{x'} = \emptyset,$$

and we define $R = \cup_{x \in X} R_x$ to be the total set of representatives. For example, if we take $X = \langle g \rangle \subset \mathbb{F}_p^*$, then the usual choice for R is, for every $x \in X$, to let $R_x = \{x\}$, but one could also take a redundant representation with two elements per $x \in X$ by letting $R_x = \{x, x+p\}$ as has been done for side-channel protection [Wal99].

A *mask* is a function $\mu : R \longrightarrow R$, and a masking set M is a set of such functions. In our discrete logarithm analogue we can think of M as a set indexed by elements in $\mathbb{Z}_q^*$ which gives an explicit exponentiation algorithm on the set of representatives of the group elements X.

A masking function $\mu \in M$ is said to be *invertible* if

$$\forall \mu \in M, \quad \forall x \in X, \quad \forall r \in R_x, \quad \exists \mu^{-1} \in M \quad : \quad \mu^{-1}(\mu(r)) \in R_x.$$

Note, we do not require that the inverse gives the same representative back, only that it gives a representative in the same set. If all elements $\mu \in M$ are invertible, then we say that the masking set M is *invertible*. In our discrete logarithm example if μ corresponds to the map $g \mapsto g^a$ on some set of group representatives, then μ^{-1} corresponds to the map $g \mapsto g^{1/a}$.

An *invertible masking structure* $\mathcal{M}$ for a set X is then a collection of sets of representative R_X, along with a collection of invertible masking sets $[M_i]_{i=1}^n$, and we write $\mathcal{M} = \{X, R_X, [M_i]_{i=1}^n\}$.

Such an invertible masking structure is said to be *semi-commutative* if

$$\forall i \neq j,\ \ \forall \mu \in M_i,\ \ \forall \mu' \in M_j,\ \ \forall r \in R:\ \ \mu(\mu'(r)) \in R_x \iff \mu'(\mu(r)) \in R_x,$$

for the same set R_x. In our discrete logarithm setting, letting M being the set of exponentiation maps considered earlier, it is easy to see that $\mathcal{M} = \{X, R_X, [M, M]\}$ is a semi-commutative invertible masking structure.

3.1 Problems and Properties

We now present a distinguishing experiment and several computational problems for such masking structures. The security of our protocols will be proven to hold under the assumption that some of these are hard, and the precise security level will then be established when this generic structure is instantiated from concrete constructions in Sections 4 and 5.

Definition 3.1 (IND-Mask security). *Given a semi-commutative invertible masking structure $\mathcal{M} = \{X, R_X, [M_i]\}$, we define the IND-Mask$_{\mathcal{A},\mathcal{M}}$ experiment in Figure 3 for an arbitrary adversary $\mathcal{A}$.*

Data: $\mathcal{M} = \{X, R_X, [M_i]\}$, $\lambda \in \mathbb{N}$
Result: $\mathsf{win} \in \{0, 1\}$
$r, \mu_0, \mu_1, i, \mathsf{st} \leftarrow \mathcal{A}(1^\lambda)$ such that $r \in R$, $i \in [n]$, $\mu_0, \mu_1 \in M_j$, $j \neq i$;
$r_0 \leftarrow \mu_0(r), \quad r_1 \leftarrow \mu_1(r)$;
$\mathsf{b} \xleftarrow{\$} \{0, 1\}$;
$\mu \xleftarrow{\$} M_i$;
$r \leftarrow \mu(r_\mathsf{b})$;
$\tilde{\mathsf{b}} \leftarrow \mathcal{A}(1^\lambda, \mathsf{st}, r)$;
if $\tilde{\mathsf{b}} = \mathsf{b}$, **then return** $\mathsf{win} = 1$ **else return** $\mathsf{win} \xleftarrow{\$} \{0, 1\}$;

Fig. 3: The IND-Mask$_{\mathcal{A},\mathcal{M}}$ security experiment

We then say that $\mathcal{M}$ is IND-Mask-secure *if for all PPT adversaries $\mathcal{A}$, it holds that*

$$\left| \Pr\left[\textit{IND-Mask}_{\mathcal{A},\mathcal{M}}(\lambda) = 1\right] - \frac{1}{2} \right| \leq \mathsf{negl}(1^\lambda).$$

We see that in the discrete logarithm setting, when $R_x = \{x\}$, the exponentiation map $(\cdot)^a$ for a random a induces a random premutation of the group elements. Therefore for a secret a and given two group elements g_0, g_1, the distribution of g_b^a is perfectly uniform, independently of b. This shows that such an $\mathcal{M}$ is perfectly IND-Mask-secure.

Note 3.1. In some settings (but not in the discrete logarithm one), it may be possible to distinguish the action of two masks that belong to separate masking sets. It is also possible that this difference is preserved under the action of a mask from a third masking set. Therefore, if an adversary was able to submit arbitrary r_0 and r_1 to the IND-Mask experiment, it could ensure that the difference between them is preserved by the action of μ and hence win the experiment with certainty. By forcing $\mathcal{A}$ to submit a single $r \in R$ and two maps μ_0, μ_1 belonging to the same masking set M_j, the experiment prevents that strategy.

As well as the above hiding property of random masks, we will also make reference to the following hard problems for semi-commutative invertible masking structures:

Definition 3.2. *Given a semi-commutative invertible masking structure $\mathcal{M} = \{X, R_X, [M_i]\}$, we define the following computational problems:*

1. Demask*: Given (i, r, r_x) with the promise that $r_x = \mu_x(r)$ for a uniformly random $\mu_x \xleftarrow{\$} M_i$, return μ_x.*
2. Parallel*: Given (i, j, r, r_x, r_y) with the promise that $i \neq j$ and that $r_x = \mu_x(r)$ and $r_y = \mu_y(r)$ for uniformly random $\mu_x \xleftarrow{\$} M_i, \mu_y \xleftarrow{\$} M_j$, return $z \in X$ such that $\mu_x(r_y) \in R_z$.*
3. ParallelInv*: Given (i, j, r, r_x, r_y) with the promise that $i \neq j$ and that $r_x = \mu_x(r)$ and $r_y = \mu_y(r)$ for uniformly random $\mu_x \xleftarrow{\$} M_i, \mu_y \xleftarrow{\$} M_j$, return $z \in X$ such that $\mu_x^{-1}(r_y) \in R_z$.*
4. ParallelEither*: Given (i, j, r, r_x, r_y) with the promise that $i \neq j$ and that $r_x = \mu_x(r)$ and $r_y = \mu_y(r)$ for uniformly random $\mu_x \xleftarrow{\$} M_i, \mu_y \xleftarrow{\$} M_j$, return $z \in X$ such that either $\mu_x(r_y) \in R_z$ or $\mu_x^{-1}(r_y) \in R_z$.*
5. ParallelBoth*: Given $(i, j, r, r_{x_0}, r_{x_1}, r_y)$ with the promise that $i \neq j$ and that $r_{x_\mathsf{b}} = \mu_\mathsf{b}(r), \mathsf{b} \in \{0, 1\}$ and $r_y = \mu_y(r)$ for uniformly random $\mu_\mathsf{b} \xleftarrow{\$} M_i, \mu_y \xleftarrow{\$} M_j$, return $z \in X$ such that either $\mu_{1-\mathsf{b}}^{-1}(\mu_\mathsf{b}(r_y)) \in R_z$ or $\mu_\mathsf{b}^{-1}(\mu_{1-\mathsf{b}}(r_y)) \in R_z$.*

If we want to make explicit the given masking structure $\mathcal{M}$ to which the (say) Demask *problem refers, then we write* $\mathsf{Demask}^{\mathcal{M}}$.

To motivate these problems we consider them in the context of the discrete logarithm setting, where we take our masking structure as before to have $R_x = \{x\}$ and to have each M_i to be identical to the set of exponentiation maps indexed by $\mathbb{Z}_q^*$.

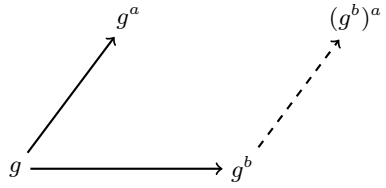
- It is easy to see that the Demask problem is, given (g, h) with the promise that $h = g^a$ for a random a, to return a. This is exactly the discrete logarithm problem (DLP).
- Similarly, the Parallel problem is, given (g, g^a, g^b) for random a, b, to return $g^{a \cdot b}$ which is exactly the computational Diffie-Hellman (CDH) problem. The name "Parallel" is derived from the representation shown in Figure 4a of the problem where the challenge is to compute the parallel operation.
- In the discrete logarithm setting, the ParallelInv problem is to compute $g^{b/a}$ given (g, g^a, g^b). We show here that in this setting it is equivalent to the Parallel problem. Given a challenge (g, g^a, g^b) for Parallel, we let (g^a, g, g^b) be a challenge for ParallelInv. We rewrite this as $(h, h^{a'}, h^{b'})$ with $h = g^a$, $a' = 1/a$ and $b' = b/a$. As a and b are uniformly random, so are a' and b' and hence our ParallelInv solver returns

$$h^{b'/a'} = h^{a \cdot b/a} = (g^a)^b = g^{a \cdot b}$$

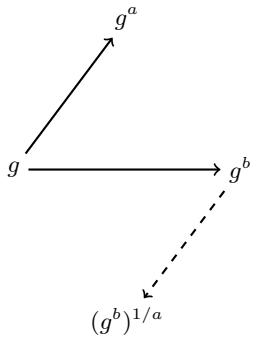
  which is exactly the solution to the Parallel challenge that was given. A similar reduction shows that the ParallelInv problem can be solved using an oracle for the Parallel problem.
  We note that this reduction does not immediately hold in the abstract case, due to the unspecified relation between r and $\mu^{-1}(\mu(r))$, but it can nonetheless be shown to hold for different instantiations.

– The ParallelEither problem is an instance where both the solutions to the Parallel and to the ParallelInv problems, for the same challenge, are accepted. Whilst it is immediate that the ParallelEither problem is at most as hard as any of the other two, a formal reduction to show the reverse implication does not appear to be as trivial. We conjecture that in most settings, and in the discrete logarithm setting in particular, allowing for two possible answers which are both hard to compute on their own does not significantly decrease the hardness of the ParallelEither problem.

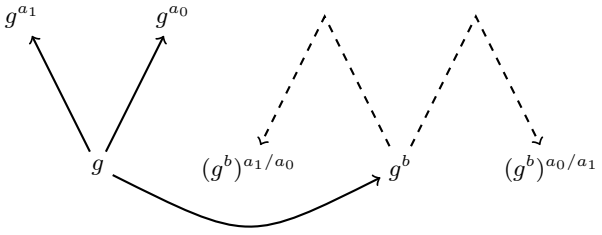
– The solution of the ParallelBoth problem can be seen as a combination of both Parallel and ParallelInv solutions together with the choice of the ParallelEither problem as is shown Figure 4c. Indeed, one can first use a Parallel oracle to compute $\mu_{\mathsf{b}}(r_y)$ for either $\mathsf{b} \in \{0, 1\}$ and then use a ParallelInv oracle to compute $\mu^{-1}_{1-\mathsf{b}}(\mu_{\mathsf{b}}(r_y))$ which shows that ParallelBoth is at most as hard as those two problems. Similarly to the ParallelEither problem, we conjecture that in most settings the ParallelBoth will not be significantly easier as it requires solutions which are both hard to compute.



(a) The Parallel problem.



(b) The ParallelInv problem.



(c) The ParallelBoth problem.

Fig. 4: Representations of computational problems.

4 Instantiation from one-way group actions

We now present a generalisation of the discrete logarithm setting instantiation of our new semi-commutative masking structure. Specifically, we show that *hard homogeneous spaces*, as given in [CLM+18], and which are based on Couveignes's original definition [Cou06], are an example of semi-commutative invertible masking schemes. This forms a direct generalisation of the discrete logarithm setting, which has already been discussed in Section 3. However, it also includes the action, via isogenies, of the class group of the ring of $\mathbb{F}_p$-rational endomorphisms of supersingular isogenies over $\mathbb{F}_p$ on the isomorphism classes of such curves.

4.1 One-way group actions

Before succinctly presenting the two realisations discussed above, we give a summarized definition of hard homogeneous spaces and formally instantiate a semi-commutative masking structure from such spaces. Throughout this section, we let G be a finite commutative group with identity element e and we denote the *group action* of G on a set X with the operator $*$ as follows:

$$\begin{aligned} * : G \times X &\to X \\ g * x &\mapsto x. \end{aligned}$$

Definition 4.1 (Hard (efficient) homogeneous space). *A* homogeneous space X *for* G *is a finite set* X *on which* G *acts freely and transitively. This implies that for any* $g \in G$ *different from* e*, the permutation of* X *induced by the action of* g *has no fixed points; i.e. for given* $x, y \in X$*, there exists a unique* $g \in G$ *such that* $y = g * x$*.*

The space X *is* efficient *if the following tasks are computationally easy (i.e. polynomial-time):*

- *Evaluation of the group operation, inversion and equality testing of elements of* G*,*
- *Sampling a random element from* G *with (close to) uniform distribution,*
- *Deciding membership and equality of a representation of elements of* X*,*
- *Evaluation of the action of a group element* $g \in G$ *on a set element* $x \in X$*;*

The space X *is* hard *if the following tasks are computationally hard (i.e. not polynomial-time):*

- *Given* $x, y \in X$*, return* $g \in G$ *such that* $y = g * x$*; this is the analogue of the DLP for the group action.*
- *Given* $x, y, z \in X$ *such that* $y = g * x$*, return* $g * z$*; this is the analogue of the CDH for the group action.*

We now instantiate a masking structure and show that it realises our definition of a semi-commutative invertible masking structure.

Definition 4.2 (Masking structure from homogeneous space). *Given a space* X *for* G *we define a masking structure* $\mathcal{M}_{X,G} = \{X, R_X, [G, G]\}$ *for* X *as follows:*

- *The* X *in the structure is exactly the set* X *from of space.*

– *We let* $R_x = \{x\}$ *for each* $x \in X$ *and therefore have* $R = X$.
– *The masking tuple* $[G, G]$ *consists of two identical copies of the group* G *that acts on* X.

Lemma 4.1. *Let* X *be an efficient homogeneous space for a commutative group* G*, then the masking structure* $\mathcal{M}_{X,G} = \{X, R_X, [G, G]\}$ *of Definition 4.2 is a semi-commutative masking structure.*

Proof. First we see that all the elements of $\mathcal{M}_{X,G}$ are well-defined and that so is the masking action of $\mu \in G : R \longrightarrow R$ where

$$\mu : r \mapsto \mu * r$$

Next, we have that by definition of a group action, the masking of any $r \in R$ by any $\mu \in M_i$ for all i is indeed invertible. Also, since every M_i is a copy of the group G, the commutativity of G induces the semi-commutativity of $\mathcal{M}_{X,G}$. Finally, the properties of an efficient homogeneous space imply the efficiency of the operations required for a semi-commmutative masking structure. □

We see here that *this* notion of a group action is stronger than our semi-commutative structure since any mask is in fact able to commute with any other. However the advantage of our weaker structure will become apparent in Section 5 with the next instantiation from supersingular isogenies over $\mathbb{F}_{p^2}$.

Note 4.1. Before we discuss the instantiation of the computational problems, we briefly note that the two requirements for the hardness of a homogeneous space correspond exactly to the Demask and Parallel problems for a semi- commutative masking structure.

Also, we have that the Parallel and ParallelInv problems are equivalent as it suffices to swap the first two elements of a challenge (x, y, z) for one problem to obtain a challenge (y, x, z) for the other which yields the same solution. Finally we have that ParallelEither is at most as hard as Parallel or ParallelInv. Hence we have

$$\mathsf{ParallelEither}^{\mathcal{M}_{X,G}} <_P \mathsf{Parallel}^{\mathcal{M}_{X,G}} \cong_P \mathsf{ParallelInv}^{\mathcal{M}_{X,G}}.$$

We also note that $\mathcal{M}_{X,G}$ is perfectly IND-Mask-secure since the action by a uniformly random element in G induces a perfect randomization of any element in X.

4.2 Discrete logarithm setting

The traditional Diffie-Hellman (DH) setting presented in Section 3 is a straightforward realisation of the hard homogeneous space presented in the previous section. Indeed, for any finite abelian group $\langle g \rangle$ of prime order in which the computational Diffie–Hellman problem is hard, we can let X be the set $\langle g \rangle$ and G be the set of exponentiation maps.

4.3 Class group of the endomorphism ring of supersingular elliptic curves over $\mathbb{F}_p$

The second realisation of hard homogeneous spaces we present is a summary of the recent work by Castryck et al. [CLM+18], we refer the reader to the full paper for a precise discussion. The work of Castryck et al. builds upon the Couveignes-Rostovstev-Stolbunov scheme of [Cou06,RS06]

where the public key space is the set of $\mathbb{F}_q$-isomorphism classes of *ordinary* elliptic curves over $\mathbb{F}_q$ whose endomorphism ring is a given order $\mathcal{O}$ in an imaginary quadratic field and whose trace of Frobenius has prescribed sign. The key ideas of the scheme of Courveignes et al. is that the ideal class group $\mathrm{cl}(\mathcal{O})$ acts freely and transitively on that set, and that this class group is commutative which allows for a natural key exchange protocol.

However, and despite recent improvements [FKS18,Kie17], the scheme of Couveignes et al. is inefficient for the following reason. In order to decompose the action of an element of $\mathrm{cl}(\mathcal{O})$ into several smaller actions that are quicker to compute, De Feo-Kieffer-Smith [FKS18] had the idea to chose $p \equiv -1 \mod \ell$ for several small odd primes ℓ. They then searched for an ordinary elliptic curve $E/\mathbb{F}_p$ such that $\#E(\mathbb{F}_p) \equiv 0$ modulo as many ℓ's as possible. This would ensure that $\ell\mathcal{O}$ decomposes as the product of two prime ideals $\mathfrak{l}$ and $\bar{\mathfrak{l}}$ for which the action of the ideal classes $[\mathfrak{l}]$ and $[\bar{\mathfrak{l}}]$ can be computed efficiently. If this works for sufficiently many ℓ's, then a generic element of $\mathrm{cl}(\mathcal{O})$ can be written as a product of small integral powers of such $[\mathfrak{l}]$ and the class group action can be computed efficiently. However, finding a curve $E/\mathbb{F}_p$ such that $\#E(\mathbb{F}_p) \equiv 0$ is hard and they only manage to obtain practical solutions for 7 different values of ℓ.

In order to increase the efficiency of this methodology, Castryck et al. adapt it to make use of *supersingular* elliptic curves defined over a *prime* field $\mathbb{F}_p$. Instead of the full ring of endomorphisms of such curves, which is not commutative, they consider the subring of $\mathbb{F}_p$-rational endomorphisms which is again an order $\mathcal{O}$ in an imaginary quadratic field. As before, the ideal class group $\mathrm{cl}(\mathcal{O})$ acts via isogenies on the set of $\mathbb{F}_p$-isomorphism classes of elliptic curves with $\mathbb{F}_p$-rational endomorphism ring equal to $\mathcal{O}$, we denote this set by $\mathcal{E}_p(\mathcal{O})$. Furthermore, contrary to the ordinary case, this action only has a single orbit.

The reason why this yields an increase in efficiency is that, in the supersingular case, $\#E(\mathbb{F}_p) = p+1$ and hence $\#E(\mathbb{F}_p) \equiv 0$ modulo *all* primes $\ell \mid p+1$ used in building p. This allows for many more values of ℓ to be used which in turn reduces the integral powers of each $[\mathfrak{l}]$ that appear in the decomposition of generic elements in $\mathrm{cl}(\mathcal{O})$. Concretely, Castryck et al. use 74 small odd primes in their implementation for which they heuristically expect that each element in $\mathrm{cl}(\mathcal{O})$ can be written as $[\mathfrak{l}_1]^{e_1}[\mathfrak{l}_2]^{e_2}\cdots[\mathfrak{l}_{74}]^{e_{74}}$ with each $e_i \in \{-5,\ldots,5\}$. In contrast, for a class group of equivalent 256-bit size, using 7 small primes for the same approach would require exponents in the range of 2^{36} which leads to much slower computations.

Lemma 4.2. *For a fixed prime field $\mathbb{F}_p$ and appropriate order $\mathcal{O}$ of an imaginary quadratic field, let $X = \mathcal{E}_p(\mathcal{O})$, and let $G = cl(\mathcal{O})$. Then X is an efficient homogeneous space for G.*

Proof. As stated in the discussion above, we have that G acts freely and transitively on X and furthermore it inherits the commutative structure of $\mathcal{O}$ and therefore this is a well-defined homogeneous space.

Also, due to the decomposition into classes of small prime ideals with small integral exponents the evaluation of the group operation, inversion, equality and sampling, as well as the action of a group element on a set element x are all efficient. Furthermore, since X can be represented as the set of Montgomery coefficients of the $\mathbb{F}_p$-isomorphism classes, equality of elements of X is efficient as well. □

As in the previous setting, the Demask and Parallel problems for the semi- commutative masking structure $\mathcal{M}_{X,G}$ induced by the homogeneous space of Lemma 4.2 immediately translate to

analogues of the DLP and CDH in the class group action setting; and so does our prior discussion on the equivalence of ParallelInv and Parallel and on the hardness of ParallelEither. The classical and post-quantum security of the DLP analogue in this setting was already succinctly discussed in [CLM+18, Section 7] and was addressed in greater detail in the very recent work of [BS18] which provides a finer estimation of the required security parameters. We leave the analysis of the security of the CDH analogue for further work.

5 Instantiation from supersingular isogenies over $\mathbb{F}_{p^2}$

In Section 4 above, the commutative property of G gives stronger algebraic properties to the induced masking structure than the weaker *semi-commutativity* which we require. Furthermore, the first realisation presented in Section 4.2 also possesses a group structure on the set X which is compatible with the action of G. This additional structure on X plays a key role in the design of several protocols as it enables increased flexibility, such as the OT protocol of Chou and Orlandi [CO15a], but it also leads to increased attack vectors such as Pohlig-Hellman-style attacks. The second realisation of Section 4.3 does not possess such a structure on X compatible with the action G which eliminates that attack vector. However the commutative property of G itself still enables the Demask problem for $\mathcal{M}_{X,G}$ to be presented as an instance of the abelian shift problem for which a sub-exponential quantum algorithm with time complexity of $L_p[1/2]$ is known to exist [CJS14].

In an effort to avoid this sub-exponential quantum attack vector, De Feo, Jao and Plût [DFJP14] consider the use of supersingular elliptic curves over an *extension* of $\mathbb{F}_p$ whose *full* endomorphism ring is an order in a quaternion algebra and therefore non-commutative. In this section we summarize this approach succinctly, construct a semi- commutative masking structure from this setting and discuss the hardness of the induced problems.

5.1 Supersingular isogenies over $\mathbb{F}_{p^2}$

Preliminaries. Let E_1 and E_2 be elliptic curves defined over a finite field $\mathbb{F}_q$. An *isogeny* $\phi : E_1 \to E_2$ over $\mathbb{F}_q$ is a non-constant rational map over $\mathbb{F}_q$ which is also a group homomorphism from $E_1(\mathbb{F}_q)$ to $E_2(\mathbb{F}_q)$. For the isogenies that we consider, we identify their degrees with the size of their kernels. Two curves E_1, E_2 are said to be *isogenous* over $\mathbb{F}_q$ if there exists an isogeny $\phi : E_1 \to E_2$ over $\mathbb{F}_q$ which holds if and only if $\#E_1(\mathbb{F}_q) = \#E_2(\mathbb{F}_q)$. A set of elliptic curves over $\mathbb{F}_q$ that are all isogenous to one another is called an *isogeny class*.

An *endomorphism* over $\mathbb{F}_q$ of an elliptic curve E is a particular isogeny $E \to E$ over $\mathbb{F}_{q^m}$ for some m. The set of endomorphisms of E together with the zero map, denoted $\mathrm{End}(E)$, forms a ring under the following operations

$$\phi \oplus \varphi : P \mapsto \phi(P) + \varphi(P) \qquad \text{and} \qquad \phi \otimes \varphi : P \mapsto \phi(\varphi(P)).$$

The full ring $\mathrm{End}(E)$ is isomorphic to either an order in a quaternion algebra, in which case we say that E is supersingular, or to an order in an imaginary quadratic field, in which case we say that E is ordinary. Curves that are in the same isogeny class are either all supersingular or all ordinary.

Here we focus on the supersingular case. All supersingular curves are defined over the field $\mathbb{F}_{p^2}$ for a prime p and for every prime $\ell \nmid p$ there exist $\ell + 1$ isogenies, up to isomorphism, of degree ℓ originating from any given supersingular curve.

Given a curve E and a subgroup Φ of $E(\mathbb{F}_{p^2})$ there is, up to isomorphism, a unique isogeny $E \to E'$ having kernel Φ and we therefore identify E' with the notation E/Φ:

$$\forall \Phi < E(\mathbb{F}_{p^2}), \qquad \exists! \phi : E \longrightarrow E/\Phi.$$

Particularly, we will work with subgroups of the torsion group $E[m]$ for $m \in \mathbb{N}$ which is the group of $\mathbb{F}_{p^2}$-points of E whose order divides m. Whilst it is inefficient to specify a whole subgroup to specify an isogeny, in the special case of kernels generated by $\mathbb{F}_{p^2}$-rational points, one can specify a *generator* of the kernel to allow for a short representation and efficient computation of an isogeny.

Semi-commutativity. To fix ideas, and introduce some *semi-commutativity* to this setting despite the non-commutativity of $\mathrm{End}(E)$, we recap on how isogenies can be used to construct the SIDH key-exchange protocol. We generalise slightly the presentation of [DFJP14] and discuss the case where $\mathbb{F}_q$ is fixed to be $\mathbb{F}_{p^2}$ where p is a prime of the form $\ell_1^{e_1}\ell_2^{e_2}\cdots\ell_n^{e_n} \cdot f \pm 1$ for n small primes $\ell_1, \ldots, \ell_n$ and a small cofactor f.

Since we have that $\ell_i \nmid p$ for all i, there is a curve $E/\mathbb{F}_{p^2}$ in each isomorphism class such that the torsion group $E[\ell_i^{e_i}]$ is isomorphic to $(\mathbb{Z}/\ell_i^{e_i}\mathbb{Z}) \times (\mathbb{Z}/\ell_i^{e_i}\mathbb{Z})$. This implies that $E[\ell_i^{e_i}]$ contains $\ell_i^{e_i-1}(\ell_i + 1)$ cyclic subgroups of order $\ell_i^{e_i}$ (which each define a different isogeny).

In the setting of key exchange, a party generates a secret key by selecting a random point K_i of order $\ell_i^{e_i}$ on a common curve E and computes a public key by computing the unique isogeny with kernel $\langle K_i \rangle$ and publishing the domain curve $E/\langle K_i \rangle$. The computation of this isogeny is efficient due to its very smooth degree. The issue here is that the structure of $\mathrm{End}(E)$ no longer allows for the composition of arbitrary isogenies to commute and an analogue of the $(g^a)^b = (g^b)^a$ equality is not immediate. However, with isogenies of co-prime degrees some commutative structure can still be achieved.

To solve this, in addition to the curve E, the parties agree on bases $\{P_i, Q_i\}$ for each of the torsion groups $E[\ell_i^{e_i}]$. The semi-commutative structure then comes from the fact that applying an isogeny of degree $\ell_i^{e_i}$ preserves the torsion groups $E[\ell_j^{e_j}]$ for $j \neq i$ since every point in $E[\ell_j^{e_j}]$ has order co-prime to $\ell_i^{e_i}$. Therefore, alongside publishing the domain curve $E/\langle K_i \rangle$ for their secret isogeny ϕ_i, parties also publish $\{\{\phi_i(P_j), \phi_i(Q_j)\}_{j \neq i}\}$, the images under ϕ_i of the bases for the other torsion groups. By expressing their secret kernel $K_j = [\alpha_j]P_j + [\beta_j]Q_j$ in the bases of the initial torsion groups and applying these coefficients to the images $\{\phi_i(P_j), \phi_i(Q_j)\}$, the other party is then able to compute an isogeny $\varphi_j : E/\langle K_i \rangle \to E/\langle K_i, K_j \rangle$ which can be considered equivalent to the isogeny $\phi_j : E \to E/\langle K_j \rangle$.

Whilst the two resulting curves $E/\langle K_i, K_j \rangle$ and $E/\langle K_j, K_i \rangle$ may not be identical, they will be isomorphic, as the kernel $\langle K_i, K_j \rangle$ defines a unique isogeny up to isomorphism, and the parties can then take the j-invariants of their respective curves as an identical shared value.

The Weil pairing. As we now include points and their images under secret isogenies in our protocols, we recall here the notion of the *Weil pairing* which can be a vector for distinguishing attacks.

For any integer $m \in \mathbb{N}$, we let $\zeta_m = \{u \mid u^m = 1\} \subset \mathbb{F}_{p^2}^*$. For any curve E defined over $\mathbb{F}_{p^2}$, the Weil pairing is a map e_m evaluated on pairs of points,

$$e_m : E[m] \times E[m] \longrightarrow \zeta_m,$$

that satisfies the following relation:

$$e_m(\phi(P), \phi(Q)) = e_m(P, Q)^{\deg\phi}$$

where $\phi : E \to E'$ is any isogeny.

5.2 Masking structure.

To start defining a semi-commutative masking structure, we fix $p = \ell_1^{e_1}\ell_2^{e_2}\cdots\ell_n^{e_n} \cdot f \pm 1$ as above. In this setting, there are five supersingular isogeny classes and we let X denote the set of all j-invariants that belong to the biggest of the five classes.

Representatives. For each j-invariant $x \in X$, there is a canonical choice of curve E_x as described in [AJK+16, Section 3.1] and [GPS17, Section 2.4]. For each E_x we take the appropriate twist of the curve to ensure that they all belong to the same isogeny class. We then define the set R_x of representatives to be the set of tuples $(E_x, \{\{P_i, Q_i\}_{i\in[n]}\})$ where E_x is the canonical curve for x or an appropriate twist and $\{P_i, Q_i\}$ is a basis of the torsion group $E_x[\ell_i^{e_i}]$ as in Section 5.1. For a given curve and torsion order, there exist a deterministic and efficient algorithm $\mathsf{Basis}(E, i)$ as is shown in [AJK+16, Section 3.2], and for each torsion order, we fix a global value $q_i \in \zeta_{\ell_i^{e_i}}$ such that for any curve E, the value of the Weil pairing evaluated on the basis points output by $\mathsf{Basis}(E, i)$ is equal to q_i. This will be used to derive new torsion points when required, but these are still free to be modified under the action of isogenies. Hence for each x, there will be a unique choice of E_x but many choices of bases of torsion groups that originate from the deterministic one.

In our protocols, when a shared common element $x \in X$ is required together with a representative $r \in R_x$, then the parties should agree on a j-invariant and then derive the curve and all the basis points in the deterministic way described above.

Masking sets. We first observe that for any $K_i = [\alpha_i]P_i + [\beta_i]Q_i$ on a curve E, the point $[\lambda]K_i$, for $\lambda \in (\mathbb{Z}/\ell_i^{e_i}\mathbb{Z})^*$, generates the same subgroup of $E[\ell_i^{e_i}]$. By defining the equivalence relation $\sim_R$ by

$$(\alpha, \beta) \sim_R (\alpha', \beta') \qquad \Longleftrightarrow \qquad \exists\lambda \in (\mathbb{Z}/\ell_i^{e_i}\mathbb{Z})^* \text{ s.t. } (\alpha', \beta') = (\lambda\alpha, \lambda\beta),$$

we can then identify any such kernel K_i with the equivalence class of (α_i, β_i) which we denote $[\alpha_i : \beta_i]$. We recall that the projective line $\mathrm{P}^1(\mathbb{Z}/\ell_i^{e_i}\mathbb{Z})$ is the set of equivalence classes $[\alpha_i : \beta_i]$ such that the ideal $\langle\alpha_i, \beta_i\rangle$ is the whole of $\mathbb{Z}/\ell_i^{e_i}\mathbb{Z}$.

Since K_i has exact order $\ell_i^{e_i}$, at least one of α_i and β_i must not be divisible by ℓ_i and hence the ideal of the ring $\mathbb{Z}/\ell_i^{e_i}\mathbb{Z}$ generated by α_i, β_i is always the unit ideal, i.e. the whole of $\mathbb{Z}/\ell_i^{e_i}\mathbb{Z}$. This implies that all the possible choices for K_i can be exactly identified with the points on the projective line $\mathrm{P}^1(\mathbb{Z}/\ell_i^{e_i}\mathbb{Z})$. We therefore define n masking sets $[M_i]_{i\in[n]}$ where each M_i is the projective line $\mathrm{P}_i := \mathrm{P}^1(\mathbb{Z}/\ell_i^{e_i}\mathbb{Z})$.

Masking action. Computing the result of a mask $\mu(r) \in R_y$ on a representative $r \in R_x$ then consists in computing one of its representatives K_i in $E_x[\ell_i^{e_i}]$ and the isogeny $\phi_i : E_x \to E_x/\langle K_i \rangle$. Note that the curve $E_x/\langle K_i \rangle$ with j-invariant $y \in X$ may not be the same curve as the canonical choice E_y. However they will be isomorphic over $\mathbb{F}_{p^2}$ due to the appropriate choice of twist in the definition of our set R_y and the isomorphism $\chi : E_x/\langle K_i \rangle \longrightarrow E_y$ will be easy to compute.

To be able to compose isogenies in a semi-commutative way, computing $\mu(r)$ also requires computing the images of $\{\{P_j, Q_j\}\}$ for $j \neq i$ first under ϕ_i and then under the isomorphism χ to obtain bases of the torsion groups of E_y. It also requires generating a new basis for $E_y[\ell_i^{e_i}]$ using the $\mathsf{Basis}(E, i)$ algorithm.

Inverting the mask. Since our masking sets M_i are no longer derived from a group, we do not have an immediate instantiation of an inverse operation. However, for every isogeny $\phi : E \to E'$ of degree ℓ, there is a unique dual isogeny $\hat{\phi} : E' \to E$ also of degree ℓ such that the composition is the multiplication-by-ℓ map: $\hat{\phi} \circ \phi = [\ell] : E \to E$. Whilst not a perfect inverse operation, in this setting the multiplication-by- $\ell_i^{e_i}$ map preserves the structure of the $\ell_j^{e_j}$-torsion groups for all $j \neq i$ and that is all we require for semi-commutativity to hold.

Hence, given a kernel generator $K_i \in E[\ell_i^{e_i}]$ for some curve E, one can compute a generator of the image $\phi_i(E[\ell_i^{e_i}]) < E'[\ell_i^{e_i}]$ of the torsion group, under the isogeny ϕ_i defined by K_i composed with an appropriate isomorphism, to obtain $\hat{K}_i \in E/\langle K_i \rangle$ which is a generator of the kernel of the unique dual isogeny $\hat{\phi}_i$.

Given a mask $\mu \in M_i = \mathrm{P}_i$ and elements r and $r' = \mu(r)$ with $r' = (E', \{\{P_i, Q_i\}_{i\in[n]}\})$, computing the inverse μ^{-1} amounts to computing a point $\hat{K}_i$ as above and expressing it as $(\hat{\alpha}_i, \hat{\beta}_i)$ in the deterministically generated basis for $E'[\ell_i^{e_i}]$ which can be done efficiently as is shown in [AJK+16]. This then allows us to define μ^{-1} uniquely as $[\hat{\alpha}_i : \hat{\beta}_i] \in \mathrm{P}_i$, given μ and r.

We note that this dependency of μ^{-1} on μ and r is consistent with the definition of the inverse of a mask as stated in Section 3 and discussed in the presentation of the construction of the protocol Π^1_{OT}.

Masking structure. We are then able to formally define a masking structure in this setting.

Definition 5.1 (Masking structure from supersingular isogenies). *Given a prime p defining the finite field $\mathbb{F}_{p^2}$ as above, we define the masking structure $\mathcal{M}_p = \{X, R_X, [M_i]_{i\in[n]}\}$ where the individual components are defined as above.*

Lemma 5.1. *The masking structure $\mathcal{M}_p$ of Definition 5.1 is a semi- commutative masking structure.*

Proof. First we see that the elements of $\mathcal{M}_p$ together with the action of any $\mu \in M_i$ on any r are well-defined. Since the composition of any isogeny with its dual results in an endomorphism of the starting curve, our method of inverting a given mask yields the same j-invariant regardless of the starting r or masking index i.

The semi-commutative property of our structure follows from the semi- commutative property of isogenies of co-prime degrees when the appropriate images of the bases of the torsion groups are revealed which they are in our computation of an arbitrary mask action.

The required efficiency of the computations for $\mathcal{M}_p$ follows from the comments above regarding the computation of isogenies of smooth degrees and expression of points in arbitrary torsion bases. Equality in X and M_i and membership in X are immediate to check. □

5.3 Computational problems

Due to its recent introduction and the relatively complec mathematics involved, the problem landscape of the SIDH setting is still currently undergoing intense study from the community. Urbanik and Jao [UJ18] have proposed a detailed presentation and study of the analogues of the dLog and CDH problems that arise from the SIDH key-exchange of De Feo, Jao and Plût [DFJP14]. Galbraith and Vercauteren also have written a survey of these problems [GV17], with a stronger focus on the mathematics of isogenies of elliptic curves.

Here we frame Urbanik and Jao's discussion of these problems in [UJ18, Section 4] in our setting that uses n distinct small primes ℓ_i. Whilst we give a very general presentation, in practice our OT schemes only require $n = 3$ (contrasted with $n = 2$ in the case of the SIDH key-exchange) which constitutes only a small extension of the original setting.

The isogeny problem. In its simplest form, the intuition behind the security of isogeny-based cryptography is that it is hard to compute a hidden isogeny, up to isomorphism, when given only the initial and final j-invariants. The *general isogeny problem* can be stated as follows.

Definition 5.2 (General isogeny problem [GV17, Definition 1]). *Given j-invariants $j, j' \in \mathbb{F}_{p^2}$, return an isogeny $\phi : E \longrightarrow E'$ (if it exists), where $j(E) = j$ and $j(E') = j'$.*

Given that the elements of X in the masking structure $\mathcal{M}_p$ are the supersingular j-invariants of $\mathbb{F}_{p^2}$ and that the elements of the masking sets M_i can be uniquely identified with isogenies between isomorphism classes, it would first seem that the Demask problem for $\mathcal{M}_p$ can be instantiated as the general isogeny problem of Definition 5.2. To recover some commutative structure, however, we have to reveal the images of the bases of the torsion points. This constitutes significantly more information and therefore is conjectured to be an easier problem to solve.

Additional information. This has led to the definition in the literature of a specific SIDH problem. Here we merge the definitions of [GV17] and [UJ18] for the case of $n = 2$ small primes in the composition of p.

Definition 5.3 (2-i-isogeny problem [GV17, Def. 2][UJ18, Prob. 4.1]). *Let (E, P_1, Q_1, P_2, Q_2) be such that $E/\mathbb{F}_{p^2}$ is a supersingular curve and P_j, Q_j is a basis for $E[\ell_j^{e_j}]$ for $j \in \{1, 2\}$. Let E' be such that there is an isogeny $\phi : E \longrightarrow E'$ of degree $\ell_i^{e_i}$. Let P'_j, Q'_j be the images under ϕ of P_j, Q_j for $j \neq i$. The* 2-i-isogeny problem*, given $(E, P_1, Q_1, P_2, Q_2, E', P'_j, Q'_j)$, is to determine an isogeny $\tilde{\phi} : E \longrightarrow E'$ of degree $\ell_i^{e_i}$ such that $P'_j = \tilde{\phi}(P_j)$ and $Q'_j = \tilde{\phi}(Q_j)$.*

This definition leads to the following natural generalisation which we show corresponds exactly to the computational problem that we need.

Definition 5.4 (n-i**-isogeny problem).** *Let $(E, \{P_j, Q_j\}_{j=1}^n)$ be such that $E/\mathbb{F}_{p^2}$ is a supersingular curve and P_j, Q_j is a basis for $E[\ell_j^{e_j}]$ for $j \in [n]$. Let E' be such that there is an isogeny $\phi : E \longrightarrow E'$ of degree $\ell_i^{e_i}$. Let $\{P'_j, Q'_j\}$ be the images under ϕ of $\{P_j, Q_j\}$ for $j \neq i$. The* n-i-isogeny problem*, given $(E, \{P_j, Q_j\}_{j=1}^n, E', \{P'_j, Q'_j\}_{j\neq i})$, is to determine an isogeny $\tilde{\phi} : E \longrightarrow E'$ of degree $\ell_i^{e_i}$ such that $P'_j = \tilde{\phi}(P_j)$ and $Q'_j = \tilde{\phi}(Q_j)$ for all $j \neq i$.*

Lemma 5.2. *Let $p = \ell_1^{e_1}\ell_2^{e_2}\cdots\ell_n^{e_n} \cdot f \pm 1$ be a prime and let $\mathcal{M}_p$ be a masking structure as defined in Definition 5.1. Then the* Demask *problem for $\mathcal{M}_p$ is an instance of the n-i-isogeny problem.*

Proof. The specification of i in (i, r, r_x) together with the random mask μ_x satisfies the promise of existence of an isogeny ϕ of degree $\ell_i^{e_i}$. Also, By definition of R_x for each $x \in X$ for $\mathcal{M}_p$, the representative r_x contains exactly the information of the curve E' together with the images of the appropriate torsion points.

We note that r_x does not contain additional information as the basis points of $E'[\ell_i^{e_i}]$ are derived deterministically from E'. □

Computational SIDH. The isogeny problems defined above can be viewed as the analogues of the discrete logarithm problem of computing an unknown exponent in the general case and in the specific SIDH setting. This naturally leads to an analogue of the CDH problem which is defined as follows in the case of $n = 2$.

Definition 5.5 (2**-computational SIDH problem [UJ18, Problem 4.3]).** *Let E, E_A, E_B be supersingular curves such that there exist isogenies $\phi_A : E \longrightarrow E_A$ and $\phi_B : E \longrightarrow E_B$ with kernels K_A and K_B and degrees $\ell_1^{e_1}$ and $\ell_2^{e_2}$ respectively. Let P_1, Q_1 and P_2, Q_2 be bases of $E[\ell_1^{e_1}]$ and $E[\ell_2^{e_2}]$ respectively, and let $P'_1 = \phi_B(P_1)$, $Q'_1 = \phi_B(Q_1)$ and $P'_2 = \phi_A(P_2)$, $Q'_2 = \phi_A(Q_2)$ be the images of the basis under the isogeny of coprime degree. The* 2-computational SIDH problem *is, given $(E, P_1, Q_1, P_2, Q_2, E_A, P'_2, Q'_2, E_B, P'_1, Q'_1)$, to identify the isomorphism class of the curve $E/\langle K_A, K_B\rangle$.*

Note 5.1. We abbreviate the previous problem as 2-CSIDH and stress that it has no relation to the CSIDH scheme of [CLM+18].

This problem can also be generalised in a natural way to the following which then yields the appropriate instantiation for our structure.

Definition 5.6 (n-i, j**-computational SIDH problem).** *Let E, E_A, E_B be supersingular curves such that there exist isogenies $\phi_A : E \longrightarrow E_A$ and $\phi_B : E \longrightarrow E_B$ with kernels K_A and K_B and degrees $\ell_i^{e_i}$ and $\ell_j^{e_j}$ respectively with $i \neq j$. Let $\{P_k, Q_k\}$ be bases of $E[\ell_k^{e_k}]$, for $k \in [n]$, and let $P_k^A = \phi_A(P_k)$, $Q_k^A = \phi_A(Q_k)$, for $k \neq i$, and $P_k^B = \phi_B(P_k)$, $Q_k^B = \phi_B(Q_k)$, for $k \neq j$ be the images of the bases under the isogeny of coprime degree. The* n-i, j-computational SIDH problem *is, given $(E, \{P_k, Q_k\}_{k\in[n]}, E_A, \{P_k^A, Q_k^A\}_{k\neq i}, E_B, \{P_k^B, Q_k^B\}_{k\neq j})$, to identify the isomorphism class of the curve $E/\langle K_A, K_B\rangle$.*

Lemma 5.3. *Let $p = \ell_1^{e_1}\ell_2^{e_2}\cdots\ell_n^{e_n} \cdot f \pm 1$ be a prime and let $\mathcal{M}_p$ be a masking structure as defined in Definition 5.1. Then the* Parallel *problem for $\mathcal{M}_p$ is an instance of the n-i, j-CSIDH problem.*

Proof. As for Lemma 5.2, the specification (i, j, r, r_x, r_y) of the Parallel problem for $\mathcal{M}_p$ satisfies the promise of existence of the two isogenies of coprime degrees and contains all the required information on the images of the torsion bases. Also, the goals of the problems agree since the solution to the Parallel problem for $\mathcal{M}_p$ requires $z \in X$ which is exactly the j-invariant which identifies the isomorphism class uniquely. Again, r_x and r_y do not contain additional information since the bases for the ith and jth torsion groups are computed deterministically. □

Inverse CSIDH problem. Regarding the ParallelInv problem for $\mathcal{M}_p$, we do not have an immediate reduction to the Parallel problem as we had for the previous instantiation. Nonetheless we are still able to prove an equivalence between the two.

Lemma 5.4. *Let $p = \ell_1^{e_1}\ell_2^{e_2}\cdots\ell_n^{e_n} \cdot f \pm 1$ be a prime and let $\mathcal{M}_p$ be a masking structure as defined in Definition 5.1. Then the* ParallelInv *problem for $\mathcal{M}_p$ is equivalent to the* Parallel *problem for $\mathcal{M}_p$.*

Proof. We show that the ParallelInv problem for $\mathcal{M}_p$ can be solved using an oracle for the Parallel problem for $\mathcal{M}_p$. The converse implication holds by a symmetric argument.

With (i, j, r, r_x, r_y) we are given the promise that there exists a mask $\mu_x \in M_i$ such that $r_x = \mu_x(r)$ and therefore that there exists a mask $\mu_x^{-1} \in M_i$, by definition of $\mathcal{M}_p$. However, it is not guaranteed by the definition of a semi-commutative masking structure that $\mu_x^{-1}(\mu_x(r)) = r$, and therefore we cannot simply submit (i, j, r_x, r, r_y) to the Parallel oracle as we might not be satisfying the promise regarding the images of the points.

Instead we will compute an alternative r' that does satisfy this promise. Leaving aside the detail of the additional computation of isomorphisms to ensure that all points belong to the canonical choice of curve for the moment, and letting $\phi_x : E \longrightarrow E_x$ denote the isogeny specified by μ_x, we can identify μ_x^{-1} with the dual isogeny $\hat{\phi}_x$ as described in Section 5.2. Since the composition of ϕ_x with its dual results in the multiplication-by-$\ell_i^{e_i}$ map on E, we are able to compute the images under μ_x^{-1} of the basis points P_k^x, Q_k^x as $[\ell_i^{e_i}]P_k, [\ell_i^{e_i}]Q_k$, for $k \neq i$, without knowledge of μ_x.

As we know that the choice of E is canonical for each j-invariant, we use the same E for r' as is given in r. Also, the basis points of $E[\ell_i^{e_i}]$ in r' are expected to be those deterministically fixed for E so we can compute them without knowledge of μ_x as well.

This shows that we can compute all the elements of r' such that they satisfy the appropriate promise for the Parallel problem with the additional relation that the hidden isogeny of degree $\ell_i^{e_i}$ between r_x and r' is exactly μ_x^{-1}, up to isomorphism, for μ_x as uniquely fixed by r and r_x.

Since we are only interested in a correct result of j-invariant from the Parallel solver, we do not need to address further the isomorphims of curves used to ensure the canonical choice of curve. By submitting the tuple (i, j, r_x, r', r_y) to the Parallel oracle, we can obtain the desired solution to the given ParallelInv challenge. □

In this setting, we similarly conjecture that the hardness of the ParallelEither and ParallelBoth problems is comparable to that of the Parallel and ParallelInv problems as no additional information is revealed and only similarly hard-to-compute solutions are required.

Decisional SIDH. Galbraith and Vercauteren also fomalise a decisional variant of the SIDH problem in the case of $n = 2$.

Definition 5.7 (2-i-decisional SIDH problem [GV17, Definition 3]).
Let (E, P_1, Q_1, P_2, Q_2) be such that $E/\mathbb{F}_{p^2}$ is a supersingular curve and P_j, Q_j is a basis for $E[\ell_j^{e_j}]$ for $j \in \{1, 2\}$. Let E' be an elliptic curve and let $P'_j, Q'_j \in E'[\ell_j^{e_j}]$ for $j \neq i$. Let $0 < d < e_i$. The 2-i-decisional SIDH problem *is, given $(E, P_1, Q_1, P_2, Q_2, E', P'_j, Q'_j, d)$ for $j \neq i$, to determine if there exists an isogeny $\phi : E \to E'$ of degree ℓ_i^d such that $\phi(P_j) = P'_j$ and $\phi(Q_j) = Q'_j$.*

As for the computational problems, we can generalise the above problem to our setting.

Definition 5.8 (n-i-decisional SIDH problem).
Let $(E, \{P_j, Q_j\}_{j\in[n]})$ be such that $E/\mathbb{F}_{p^2}$ is a supersingular curve and P_j, Q_j is a basis for $E[\ell_j^{e_j}]$ for $j \in [n]$. Let E' be an elliptic curve and let $P'_j, Q'_j \in E'[\ell_j^{e_j}]$ for $j \neq i$. Let $0 < d < e_i$. The n-i-decisional SIDH problem *is, given $(E, \{P_j, Q_j\}_{j\in[n]}, E', \{P'_j, Q'_j\}_{j\neq i}, d)$, to determine if there exists an isogeny $\phi : E \to E'$ of degree ℓ_i^d such that $\phi(P_j) = P'_j$ and $\phi(Q_j) = Q'_j$ for $j \neq i$.*

Whilst we do not have an equivalence between the IND-Mask experiment and the n-i-DSIDH as presented above, we see that an oracle for the latter with $d = e_i$ is sufficient to obtain a noticeable advantage against the former. Also, it would seem that our IND-Mask experiment corresponds to a worst case of the n-i-DSIDH as it uses a maximal degree of $d = e_i$. Given the state of the art in cryptanalysis for these problems, we conjecture that the IND-Mask problem for $\mathcal{M}_p$ is not significantly easier than the n-i-DISDH for the same parameters.

As hinted at in Note 3.1, the Weil pairing is in fact a useful tool against the IND-Mask experiment. Indeed, if the adversary had free control over the values r_0 and r_1 of the experiment, it could give two representatives whose basis points of the same torsion group evaluated to different values under the Weil pairing. This difference would be preserved under the secret masking action of the experiment and this would enable it to win trivially. Restricting the adversary's input to be a single representative r and two masks that determine r_0 and r_1 and preserve the values of Weil pairing on the points of r thus prevents this strategy.

5.4 Security analysis

As mentioned above, one of the main advantage of the SIDH approach as opposed to the hard homoegneous space approach (including CSIDH) is that no subexponential attack is known on the SIDH protocol, even using a quantum computer. On the other hand in SIDH protocol, the action of the secret isogeny on a large torsion subgroup is made public. A recent paper [Pet17] has shown how to exploit this additional information to break "overstretched" variants of the SIDH protocol.

More precisely, let $N_1 \approx p^\alpha$ be the degree of the isogeny to compute, and let $N_2 \approx p^\beta$ be the order of torsion points images revealed in the protocol. The original SIDH protocol uses $\alpha \approx \beta \approx \frac{1}{2}$, but [Pet17] describes a generalization to any coprime, powersmooth values N_1, N_2. Under some parameter restrictions and heuristic assumptions, the best attack in [Pet17] computes the isogeny in polynomial time assuming $\frac{1}{4}\beta > \alpha > 1$. Another attack removes the restriction that $\alpha > 1$ but it requires $\beta = O(\alpha^2)$.

In our instantiation above, for any i one can fix $N_1 = \ell_i^{e_i}$ and $N_2 = \prod_{j \neq i} \ell_j^{e_j}$. We also have $N_1 N_2 | (p \pm 1)$ so the first attack in [Pet17] does not apply. The second attack, however, applies whenever n is larger than $O(e_i \log \ell_i)$. At the moment this is the only constraint on the parameters that is implied by the techniques of [Pet17].

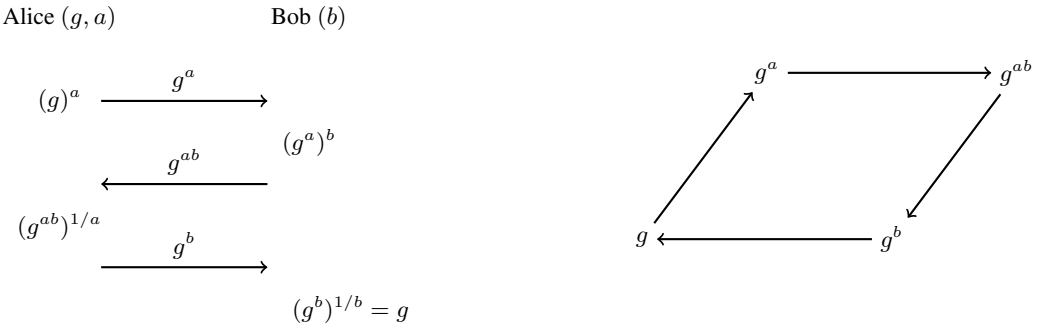
One may fear that the attacks in [Pet17] will get improved over time, leading to further restrictions on n. We note that $n = 3$ is sufficient to instantiate Protocols Π^1_{OT} and Π^2_{OT}. Moreover the first protocol could even be instantiated with $n = 2$ (see Note 6.1). We note that $n = 2$ in our construction corresponds to the SIDH protocol parameters, so our semi-commutative masking construction with $n = 2$ will remain secure as long as SIDH remains secure.

6 Oblivious Transfer Protocols

In this section, we present several OT protocols constructed from a semi-commutative masking structure $\mathcal{M}$ as presented in Section 3. We formally prove their UC security for passive adversaries with static corruptions in the $\mathcal{F}_{\mathsf{RO}}$-hybrid model under the assumption that $\mathcal{M}$ is IND-Mask-secure and that the $\mathsf{ParallelEither}^{\mathcal{M}}$ and $\mathsf{ParallelBoth}^{\mathcal{M}}$ problems are hard.

6.1 First Construction

Motivation. For our first OT protocol based on general semi-commutative invertible masking schemes we take as inspiration the two-party Shamir three-pass protocol for secure message transmission as shown in Figure 5a, sometimes called the Massey-Omura encryption scheme. This scheme corresponds to the "masking diagram" given in Figure 5b. In this protocol, Alice's input is a message g together with a secret mask a whilst Bob's input is another secret mask b. To transmit g, Alice first sends a masked g^a to Bob who replies with his own masked g^{ab}. Now Alice undoes her mask, as the inverse exponentiation commutes with Bob's, and replies with $g^{ab/a} = g^b$. Since Bob knows his own mask b, he inverts it and recovers g. When viewed as a key transport protocol the element g is seen as Alice's input key which is transmitted to Bob.

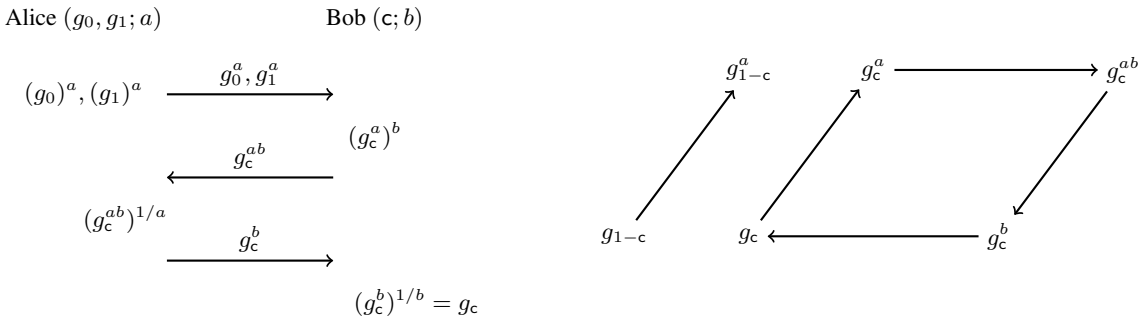


(a) Sketch of protocol flows

(b) Masking diagram

Fig. 5: The Shamir three-pass protocol

This protocol can be modified to yield an OT protocol as shown in Figure 6. This protocol can be seen as being based on a key transport protocol which is expanded to achieve the requirements of oblivious transfer. In contrast, our second construction will later take a key *agreement* protocol and turn it into an OT protocol.



(a) Sketch of protocol flows

(b) Masking diagram

Fig. 6: Sketch of the OT protocol derived from the Shamir three-pass protocol

In the OT protocol Alice, acting like the Sender, now has two inputs g_0 and g_1 and masks both using her secret mask a to send g_0^a, g_1^a to Bob, the Receiver. In addition to his mask b, Bob now also has a choice bit $\mathsf{c} \in \{0, 1\}$ and he uses both to reply to Alice with $(g_\mathsf{c}^a)^b$. They then continue as before until Bob recovers g_c. Whilst the security of the Shamir three-pass protocol holds against external adversaries due to the masking of the message, the OT protocol also needs to provide security guarantees against internal adversaries, namely Alice or Bob. The intuition for security in this case is that the mask a cannot be deduced from either g_0^a or g_1^a and therefore the first message hides both of Alice's inputs from Bob. Also when Bob applies his own mask to one of the two messages, this hides his input bit c from Alice who doesn't know b.

The problem with this protocol for *general* semi-commutative masking schemes is that Alice needs to be able to invert the mask $1/a$ on g_c^{ab} without knowing g_c. Whilst this is easy in the discrete logarithm case, it is not in general possible for semi-commutative masking schemes. This is due to the subtle fact that in the definition of the inverse of a mask,

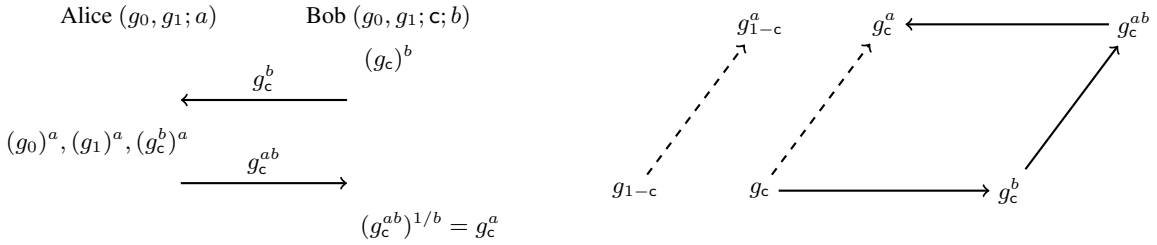
$$\forall \mu \in M, \quad \forall x \in X, \quad \forall r \in R_x, \quad \exists \mu^{-1} \in M \quad : \quad \mu^{-1}(\mu(r)) \in R_x,$$

the μ^{-1} is specified *after* μ and r and may therefore depend on $\mu(r)$. Whilst such an μ^{-1} is required to exist *for all* $r \in R_x$ for a given μ and x, if may be different for each value of r or x. Therefore in a general semi-commutative masking scheme, the $1/a$ mask may be different depending on whether it needs to be applied to g_0^a or g_1^a. As the aim of the protocol is to hide which of these two values was chosen by Bob, Alice lacks some information to compute her un-masking properly.

We thus modify the OT protocol so as to remove the need to apply the inverse mask on an unknown base. In our new (discrete logarithm based) variant, the elements g_0 and g_1 are common to both parties. Rather using her mask a to send g_0^a, g_1^a to Bob (the Receiver), Alice (the Sender)

does not go first. Instead, Bob first communicates his masked choice g_{c}^b, and then Alice applies her mask a and replies with g_{c}^{ab}. At that moment, she also computes g_0^a, g_1^a internally. She then uses these internal values as the inputs to a hash function to derive two symmetric keys k_0 and k_1. Those are used to encrypt Alice's actual OT inputs m_0 and m_1 as two ciphertexts e_0 and e_1 which she sends alongside g_{c}^{ab}. This allows Bob to recover g_{c}^a and hence decrypt e_{c} to recover m_{c}.

As she no longer communicates one of g_0 or g_1 to Bob since these are now common to both of them, this is no longer exactly a *message transport* protocol. Instead, it can be seen as a *randomness transport* protocol where Alice communicates her random mask a applied to Bob's choice g_{c}. As g_0 and g_1 are now established once and re-used for every instance of the protocol, this allows the flows to have only *two* pass rather than three. Figure 7 abstracts the symmetric encryption and only shows the flows and the masking diagram that leads to Bob receiving the value g_{c}^a.



(a) Sketch of protocol flows

(b) Masking diagram.

Fig. 7: Sketch of the final Shamir three-pass OT protocol

Construction. We can now formally define our first OT protocol from semi-commutative invertible masking schemes, as the new variant of Figure 7 can be instantiated in such a setting. A formal description of the protocol Π_{OT}^1 is given in Figure 8.

Let $\mathcal{M} = \{X, R_X, [M_A, M_B, M_C]\}$ be a semi-commutative masking structure with three masking sets; let $\mathcal{E} = \{(\mathsf{KGen}_{\mathcal{E}}, \mathsf{Enc}, \mathsf{Dec}), (\mathcal{K}_{\mathcal{E}}, \mathcal{M}_{\mathcal{E}}, \mathcal{C}_{\mathcal{E}})\}$ be a symmetric encryption scheme and let $\mathcal{F}_{\mathsf{RO}}$ be an instance of the RO ideal functionality with domain $\mathcal{D} = X$ and range $\mathcal{R} = \mathcal{K}_{\mathcal{E}}$. We assume that random sampling from masking sets $M_i, i \in \{A, B, C\}$, evaluation of masks, evaluation of Enc, Dec, and inversion in M_i are all efficient operations for the masking structure $\mathcal{M}$ and for the symmetric encryption scheme $\mathcal{E}$.

As we said before, the idea of the protocol is that both the sender, P_S, and receiver, P_R, have as common input arbitrary elements $x_0 \neq x_1 \in X$ along with representations $r_0 \in R_{x_0}, r_1 \in R_{x_1}$. In the first pass, P_R takes a random mask $\beta \in M_B$ and sends $r_{\mathsf{c}}^\beta = \beta(r_{\mathsf{c}})$ to P_S, where c is its choice bit. In the second pass, P_S samples a random mask $\alpha \in M_A$ and computes $r_0^\alpha = \alpha(r_0)$ and $r_1^\alpha = \alpha(r_1)$. These elements uniquely determine $x_{\mathsf{b}}^\alpha \in X, \mathsf{b} \in \{0, 1\}$. Thus the sender can compute two private keys $\mathsf{k}_{\mathsf{b}}, \mathsf{b} \in \{0, 1\}$, by invoking twice the random oracle functionality $\mathcal{F}_{\mathsf{RO}}$ on input x_{b}^α, and encrypt its input messages $\mathsf{m}_0, \mathsf{m}_1$ accordingly. P_S then sends the ciphertexts $\mathsf{e}_{\mathsf{b}} \leftarrow \mathsf{Enc}(\mathsf{k}_{\mathsf{b}}, \mathsf{m}_{\mathsf{b}}), \mathsf{b} \in \{0, 1\}$, and $r_{\mathsf{c}}^{\alpha\beta} = \alpha(r_{\mathsf{c}}^\beta)$ to P_R. The receiver has now all the information

Protocol Π^1_{OT}

PARAMETER: length n of the P_S's input strings.
SENDER'S INPUT: $\mathsf{m}_0, \mathsf{m}_1 \in \mathcal{M}_{\mathcal{E}}$.
RECEIVER'S INPUT: $\mathsf{c} \in \{0,1\}$.
COMMON INPUTS: Arbitrary $x_0 \neq x_1 \in X$ together with $r_0 \in R_{x_0}, r_1 \in R_{x_1}$ are shared and re-used for every instance of the protocol; an instance of the random oracle ideal functionality $\mathcal{F}_{\mathsf{RO}} : \{0,1\}^\lambda \to \mathcal{K}_{\mathcal{E}}$.

Receiver 1
- Sample $\beta \xleftarrow{\$} M_B$ uniformly at random.
- Compute $r_{\mathsf{c}}^{\beta} := \beta(r_{\mathsf{c}})$ and $\beta^{-1} \in M_B$.
- Send r_{c}^{β} to P_S.

Sender 1
- Sample $\alpha \xleftarrow{\$} M_A$ and compute $r_{\mathsf{b}}^{\alpha} := \alpha(r_{\mathsf{b}}) \in R_{x_{\mathsf{b}}^{\alpha}}$, $\mathsf{b} \in \{0,1\}$
- For $\mathsf{b} \in \{0,1\}$, call $\mathcal{F}_{\mathsf{RO}}$ twice on input x_{b}^{α} obtaining k_{b}, and compute $\mathsf{e}_{\mathsf{b}} \leftarrow \mathsf{Enc}(\mathsf{k}_{\mathsf{b}}, \mathsf{m}_{\mathsf{b}})$
- Compute $r_{\mathsf{c}}^{\alpha\beta} := \alpha(r_{\mathsf{c}}^{\beta})$
- Send $(r_{\mathsf{c}}^{\alpha\beta}, \mathsf{e}_0, \mathsf{e}_1)$ to P_R.

Receiver 2
- Compute $r_{\mathsf{c}}^{\alpha} := \beta^{-1}(r_{\mathsf{c}}^{\alpha\beta})$ and $\mathsf{k}_R := \mathcal{F}_{\mathsf{RO}}(x_\alpha)$ where $r_{\mathsf{c}}^{\alpha} \in R_{x_{\mathsf{c}}^{\alpha}}$.
- Return $\mathsf{m}_{\mathsf{c}} := \mathsf{Dec}(\mathsf{k}_R, \mathsf{e}_{\mathsf{c}})$.

Fig. 8: The protocol Π^1_{OT} for realizing $\mathcal{F}_{\mathsf{OT}}$ from semi-commutative masking.

needed to recover the message m_{c} corresponding to its choice bit: it can apply the inverse β^{-1} to $r_{\mathsf{c}}^{\alpha\beta}$ using the semi-commutativity of $\mathcal{M}$, so that

$$\beta^{-1}(r_{\mathsf{c}}^{\alpha\beta}) = \beta^{-1}(\alpha(r_{\mathsf{c}}^{\beta})) = \beta^{-1}(\alpha(\beta(r_{\mathsf{c}}))) \in R_{x_{\mathsf{c}}^{\alpha}},$$

and recover $\mathsf{k}_{\mathsf{c}} = \mathcal{F}_{\mathsf{RO}}(x_{\mathsf{c}}^{\alpha})$. This easily implies correctness of the scheme. Security is given by the following theorem.

Theorem 6.1. *The protocol Π^1_{OT} of Figure 8 securely UC-realizes the functionality $\mathcal{F}_{\mathsf{OT}}$ of Figure 1 in the $\mathcal{F}_{\mathsf{RO}}$-hybrid model for semi-honest adversaries and static corruptions, under the assumption that $\mathcal{E}$ is IND-CPA-secure, that $\mathcal{M}$ is IND-Mask-secure and that the* $\mathsf{ParallelEither}^{\mathcal{M}}$ *problem is hard.*

Proof. We prove that there exists a PPT simulator $\mathcal{S}$, with access to an ideal functionality $\mathcal{F}_{\mathsf{OT}}$, which simulates the adversary's view. We divide the proof according to the selection of the corrupt parties.

Corrupt receiver and corrupt sender. As both parties are corrupt, the simulator $\mathcal{S}$ may read their inputs from their internal state and use those to create a perfect simulation of the transcript and of the parties' internal states. It presents this simulation to its internal copy of $\mathcal{A}$, together with an perfect simulation of $\mathcal{F}_{\mathsf{RO}}$, with which it is then able to perfectly answer $\mathcal{Z}$'s queries by forwarding them to $\mathcal{A}$ and returning the responses. Since it knows all of the inputs, it forwards them to $\mathcal{F}_{\mathsf{OT}}$ at the right moment to ensure that the dummy corrupt parties return the correct output to $\mathcal{Z}$.

Corrupt receiver and honest sender. We formally describe the simulator $\mathcal{S}_{R^*}$ in Figure 9.
We show that for every semi-honest adversary $\mathcal{A}$ who corrupts P_R and any environment $\mathcal{Z}$,

$$\mathsf{HYBRID}^{\mathcal{F}_{\mathsf{RO}}}_{\Pi^1_{\mathsf{OT}},\mathcal{A},\mathcal{Z}} \overset{\mathrm{c}}{\approx} \mathsf{IDEAL}_{\mathcal{F}_{\mathsf{OT}},\mathcal{S}_{R^*},\mathcal{Z}},$$

Simulator $\mathcal{S}_{R^*}$

- Throughout the execution, $\mathcal{S}_{R^*}$ simulates the $\mathcal{F}_{\mathsf{RO}}$ by answering every new query with a random value from $\mathcal{K}_{\mathcal{E}}$ and maintaining a list of past queries to answer repeated queries consistenly. As in the previous case, it presents the simulated transcript and corrupt receiver state as computed below to $\mathcal{A}$ and uses it to answer queries from $\mathcal{Z}$.
- When $\mathcal{Z}$ activates the corrupt Receiver, its private input c is visible by $\mathcal{S}_{R^*}$ which can then compute r_{c}^{β} to perfectly simulate Receiver 1.
- To simulate Sender 1, $\mathcal{S}_{R^*}$ samples $\alpha \xleftarrow{\$} M_A$ and computes $r_{\mathsf{c}}^{\alpha\beta}$ honestly.
  Since m_{c} appears on the corrupt Receiver's output tape, the simulator computes k_{c} and e_{c} as prescribed by the protocol.
  However, since $\mathcal{S}_{R^*}$ does not learn the honest input $\mathsf{m}_{1-\mathsf{c}}$, it samples $\mathsf{k}_{1-\mathsf{c}} \xleftarrow{\$} \mathcal{K}_{\mathcal{E}}$ at random and sets $\mathsf{e}_{1-\mathsf{c}} \leftarrow \mathsf{Enc}(\mathsf{k}_{1-\mathsf{c}}, \mathsf{m})$ for an arbitrary $\mathsf{m} \in \mathcal{M}_{\mathcal{E}}$.
- If $\mathcal{Z}$ queries either $\mathcal{F}_{\mathsf{RO}}(x_{\mathsf{c}}^{\alpha})$ before activating Sender 2, then $\mathcal{S}_{R^*}$ aborts the simulation by returning $\perp$ to $\mathcal{Z}$.
- Finally, $\mathcal{S}_{R^*}$ finishes the protocol as prescribed.

Fig. 9: The simulator $\mathcal{S}_{R^*}$ of Theorem 6.1

by proceeding via a sequence of hybrid simulators.

We begin with a hybrid $\mathcal{H}_0$ which knows the inputs of the honest sender. As it learns the input c of the corrupt receiver as soon as it is activated by $\mathcal{Z}$, it is able to present a perfect simulation of the protocol.

The second hybrid $\mathcal{H}_1$ samples $\mathsf{k}_{1-\mathsf{c}} \xleftarrow{\$} \mathcal{K}_{\mathcal{E}}$ at random. Instead, $\mathcal{F}_{\mathsf{RO}}(x_{1-\mathsf{c}}^{\alpha})$ will be set to a random value if it is queried during the execution.

Claim. Any environment $\mathcal{Z}$ that can distinguish the simulations of $\mathcal{H}_1$ and $\mathcal{H}_0$ can be used to solve the ParallelEither problem for $\mathcal{M}$. Such an environment is capable of distinguishing if and only if it queries $\mathcal{F}_{\mathsf{RO}}(x_{1-\sigma}^{\alpha})$. Let $\mathcal{A}$ be an adversary for which $\mathcal{Z}$ distinguishes between $\mathcal{H}_0$ and $\mathcal{H}_1$ with some advantage ϵ, we use this to build a reduction $\mathcal{B}$ against the ParallelEither problem for $\mathcal{M}$ which proceeds as follows.
Upon receiving a challenge (C, A, r, r_x, r_y), $C \neq A$, $r_x = (r)$ and $r_y = \alpha(r)$, $\mathcal{B}$ simulates an execution of the protocol with $\mathcal{Z}$ as follows:

- First set $r_0 \coloneqq r$ and $r_1 \coloneqq r_x$, and set $r_{\mathsf{c}}^{\alpha} \coloneqq r_y$.
- Set the keys and ciphertexts as $\mathcal{H}_1$ does and simulate Receiver 1 honestly.
- Since $\mathcal{B}$ does not know the $\alpha \in M_A$ such that $r_y = \alpha(r)$, it cannot compute $r_{\mathsf{c}}^{\alpha\beta} = \alpha(r_{\mathsf{c}}^{\beta})$ honestly. Instead, it sets $r_{\mathsf{c}}^{\alpha\beta} = \beta(r_y)$. This can be done do since it is simulating the internal value β. This remains consistent with the protocol as we still have that $\beta^{-1}(r_{\mathsf{c}}^{\alpha\beta}) \in R_y$ and $r_y = (r_{\mathsf{c}}^{\alpha}) \in R_y$, as set at the beginning of $\mathcal{B}$.
- If $\mathsf{c} = 0$, then $r_{1-\mathsf{c}} = \gamma(r_{\mathsf{c}})$ and therefore $\gamma(r_{\mathsf{c}}^{\alpha}) = \gamma(r_y) \in R_{x_{1-\mathsf{c}}^{\alpha}}$. If instead $\mathsf{c} = 1$, then $r_{1-\mathsf{c}}^{\alpha} = \gamma^{-1}(r_{\mathsf{c}}^{\alpha}) = \gamma^{-1}(r_y)$.
  Therefore we see that, independently of c, if $\mathcal{Z}$ queries $\mathcal{F}_{\mathsf{RO}}(x_{1-\mathsf{c}}^{\alpha})$, then one of the solutions to the ParallelEither problem is present on the list of past queries.

When $\mathcal{Z}$ terminates, $\mathcal{B}$ therefore returns a random entry on the list of random oracle queries. If $\mathcal{Z}$ has advantage ϵ in distinguishing between $\mathcal{H}_1$ and $\mathcal{H}_0$, $\mathcal{B}$ then has an advantage ϵ/q_H in solving the ParallelEither problem, where q_H denotes the number of queries to $\mathcal{F}_{\mathsf{RO}}$ made during the execution.

The final hybrid $\mathcal{H}_2$ replaces $\mathsf{m}_{1-\mathsf{c}}$ by an arbitrary $\mathsf{m} \in \mathcal{M}_{\mathcal{E}}$ in the computation of $\mathsf{e}_{1-\mathsf{c}}$. This removes the last occurence of $\mathsf{m}_{1-\mathsf{c}}$ in the simulator and we have that $\mathcal{H}_2$ is identical to the original $\mathcal{S}_{R^*}$.

Claim. Any environment $\mathcal{Z}$ that can distinguish between a simulation of $\mathcal{H}_2$ and of $\mathcal{H}_1$ with advantage ϵ can be used to break the IND-CPA property of $\mathcal{E}$ with advantage at least ϵ.
We can build an adversary against the IND-CPA property of $\mathcal{E}$ by querying the challenger for a ciphertext of either m or $\mathsf{m}_{1-\mathsf{c}}$. This reduction emulates either $\mathcal{H}_2$ or $\mathcal{H}_1$ perfectly as $\mathsf{k}_{1-\mathsf{c}}$ is not accessible to $\mathcal{Z}$ and therefore not required by $\mathcal{H}_2$ or $\mathcal{H}_1$ at any point.

Under the assumption that $\mathcal{E}$ is IND-CPA-secure and that the ParallelEither problem is hard for $\mathcal{M}$, we have that the simulation generated by $\mathcal{S}_{R^*}$ is indistinguishable from a real world execution, for any environment $\mathcal{Z}$. This concludes the proof that $\mathsf{HYBRID}^{\mathcal{F}_{\mathsf{RO}}}_{\Pi^1_{\mathsf{OT}},\mathcal{A},\mathcal{Z}} \overset{\mathrm{c}}{\approx} \mathsf{IDEAL}_{\mathcal{F}_{\mathsf{OT}},\mathcal{S}_{R^*},\mathcal{Z}}$.

Honest receiver and corrupt sender. We formally describe the simulator $\mathcal{S}_{S^*}$ in Figure 10

Simulator $\mathcal{S}_{S^*}$

- $\mathcal{S}_{S^*}$ simulates $\mathcal{F}_{\mathsf{RO}}$ consistently and presents the state and transcript computed as follows to an internal copy of $\mathcal{A}$ to reply to the queries from $\mathcal{Z}$.
- As it does not know the c of the honest receiver, $\mathcal{S}_{S^*}$ proceeds by setting $\mathsf{c} = 0$ internally which remains out of the view of $\mathcal{A}$. It then samples $\beta \xleftarrow{\$} M_B$ and sets $r^\beta_{\mathsf{c}} = \beta(r_0)$ consistently.
- As it knows the inputs m_0, m_1 of the corrupt Sender, $\mathcal{S}_{S^*}$ computes Sender 2 consistently with z_β using the correct plaintexts.
- Finally, $\mathcal{S}_{S^*}$ finishes the protocol as prescribed.

Fig. 10: The simulator $\mathcal{S}_{S^*}$ of Theorem 6.1

We show that for every semi-honest adversary $\mathcal{A}$ who corrupts P_S and any environment $\mathcal{Z}$, it holds that

$$\mathsf{HYBRID}^{\mathcal{F}_{\mathsf{RO}}}_{\Pi^1_{\mathsf{OT}},\mathcal{A},\mathcal{Z}} \overset{\mathrm{c}}{\approx} \mathsf{IDEAL}_{\mathcal{F}_{\mathsf{OT}},\mathcal{S}_{S^*},\mathcal{Z}}.$$

The simulation of $\mathcal{S}_{S^*}$ is not a perfect simulation of a real world execution only if the honest receiver had actually received input $\mathsf{c} = 1$ from $\mathcal{Z}$. In that case, any environment that can distinguish between a simulation of $\mathcal{S}_{S^*}$ and the real world with advantage ϵ can be used to break the IND-Mask security of $\mathcal{M}$ with advantage at least ϵ. We build a reduction $\mathcal{B}$ against the IND-Mask experiment as follows.
It first selects an arbitrary r as well as two masks $\gamma_0, \gamma_1 \in M_C$ and sends $(r, \gamma_0, \gamma_1, B)$ to the IND-Mask experiment. Upon receiving $\tilde{r}$, $\mathcal{B}$ then begins the distinguishing experiment with $\mathcal{Z}$ by setting $r_0 = \gamma_0(r), r_1 = \gamma_1(r)$ and returning $r^\beta_{\mathsf{c}} = \tilde{r}$ to the adversary when $\mathcal{Z}$ activates Receiver 1. Not knowing β is not a problem for the simulation as the receiver is honest and therefore $\mathcal{B}$ does not need to simulate its state to $\mathcal{A}$.
This is a perfect simulation of either the real world or of $\mathcal{S}_{S^*}$ as either r_1 or r_0 is used by the IND-Mask experiment in the computation of r^β_{c}. Thus if $\mathcal{Z}$ distinguishes between the two, then $\mathcal{B}$ can distinguish the hidden bit of the IND-Mask experiment.

Honest receiver and honest sender. In this final case, the simulator $\mathcal{S}$ chooses arbitrary inputs $\mathsf{m}_0 = \mathsf{m}_1 = \mathsf{m} \in \mathcal{M}_{\mathcal{E}}$ and $\mathsf{c} = 1$ and simulates a transcript to $\mathcal{A}$ using those. If an environment $\mathcal{Z}$ is capable of distinguishing this simulation from a real execution of the protocol then this implies that is is able to extract information regarding the arbitrary inputs used by $\mathcal{S}$. However the previous

two cases show that, even with the additional information of the corrupted party's internal state, any environment is not able to identify a simulation that does not have any information the honest party's inputs. By combining techniques from both cases above, we can therefore show that the simulation of $\mathcal{S}$ is indistinguishable from a real world execution under the assumption that $\mathcal{S}$ is IND-CPA-secure, that $\mathcal{M}$ is IND-Mask-secure and that the $\mathsf{ParallelEither}^{\mathcal{M}}$ problem is hard.

This completes the proof that for any $\mathcal{A}$ there exists a $\mathcal{S}$ such that, for any $\mathcal{Z}$,

$$\mathsf{HYBRID}^{\mathcal{F}_{\mathsf{RO}}}_{\Pi^1_{\mathsf{OT}},\mathcal{A},\mathcal{Z}} \overset{\mathrm{c}}{\approx} \mathsf{IDEAL}_{\mathcal{F}_{\mathsf{OT}},\mathcal{S},\mathcal{Z}}.$$

□

Note 6.1. We remark that protocol Π^1_{OT} only requires the third masking set M_C as a proof artefact and that only two set would be sufficient to execute the protocol.

6.2 Second Construction

Motivation. We now describe our second OT protocol based on semi-commutative masking. This construction is inspired by the OT protocol of Chou and Orlandi [CO15a] in the sense that is uses an underlying key exchange mechanism and transforms it to acheive the requirements of oblivious transfer, which differs from the key transport-based approach of our first protocol in Section 6.1.

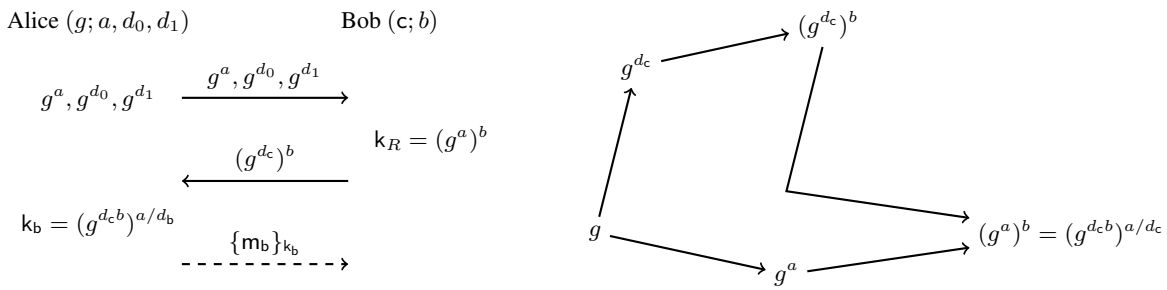
We note that the several problems that have emerged with the protocol of Chou and Orlandi [CO15b, Section 1.1] only arise when considering active adversaries. As we only prove security against passive adversaries, we do not address these problems here.

Again we motivate the proposed OT protocol in the case of semi-commutative invertible masking schemes by looking at the discrete logarithm variant first. Here Alice's inputs are two messages $\mathsf{m}_0, \mathsf{m}_1$ and an ephemeral mask a whilst Bob's is another mask b together with his choice c. To agree on the key under which the selected message will be encrypted, Alice sends a masked g^a to Bob who derives the decryption key g^{ab}. However Bob cannot simply reply with his masked g^b since Alice would then not know which of m_0 or m_1 to encrypt under the shared key (and Bob telling her would void any privacy guarantees for himself). Instead, Alice also communicates two random masks g^{d_0} and g^{d_1} to allow Bob to make a selection.

By masking $(g^{d_\mathsf{c}})^b$ with the same b as he uses to derive the key, Bob obliviously communicates his choice and his mask to Alice which is then able to derive two keys (by unmasking d_b and then adding her mask a) of which only one will be equal to the exchanged key g^{ab}. We sketch the protocol flows in Figure 11.

The protocol is intuitively secure as Alice cannot deduce the mask b from Bob's message and Bob cannot deduce the key $\mathsf{k}_{1-\mathsf{c}}$ as it is not able to recover d_b^{-1} from Alice's first message.

Construction. Formally, let $\mathcal{M} = \{X, R_X, [M_A, M_B, M_C]\}$ be a semi-commutative masking structure; let $\mathcal{E} = \{(\mathsf{KGen}_\mathcal{E}, \mathsf{Enc}, \mathsf{Dec}), (\mathcal{K}_\mathcal{E}, \mathcal{M}_\mathcal{E}, \mathcal{C}_\mathcal{E})\}$ be a symmetric encryption scheme and let $\mathcal{F}_{\mathsf{RO}}$ be an instance of the RO ideal functionality with domain $\mathcal{D} = X$ and range $\mathcal{R} = \mathcal{K}_\mathcal{E}$. We formally describe the protocol Π^2_{OT} in Figure 12.



(a) Sketch of protocol flows

(b) Masking diagram.

Fig. 11: Sketch of the OT protocol derived from the key agreement protocol.

Protocol Π^2_{OT}

PARAMETER: length n of the Sender's input strings.
SENDER'S INPUT: $\mathsf{m}_0, \mathsf{m}_1 \in \mathcal{M}_\mathcal{E}$.
RECEIVER'S INPUT: $\mathsf{c} \in \{0,1\}$.
COMMON INPUTS: An arbitrary $x \in X$ together with $r \in R_x$ is shared and re-used for every instance of the protocol; an instance of the random oracle ideal functionality $\mathcal{F}_{\mathsf{RO}} : \{0,1\}^\lambda \to \mathcal{K}_\mathcal{E}$.

Sender 1
- Sample $\alpha \xleftarrow{\$} M_A, \gamma_\mathsf{b} \xleftarrow{\$} M_C$, $\mathsf{b} \in \{0,1\}$, at random, and compute $r^\alpha := \alpha(r)$, $r_\mathsf{b} := \gamma_\mathsf{b}(r)$ and $\gamma_\mathsf{b}^{-1} \in M_C$.
- Send (r^α, r_0, r_1) to P_R

Receiver 1
- Sample $\beta \xleftarrow{\$} M_B$ uniformly at random.
- Compute $r^\beta_\mathsf{c} := \beta(r_\mathsf{c})$.
- Compute $\mathsf{k}_\mathsf{c} := \mathcal{F}_{\mathsf{RO}}(x^{\alpha\beta})$ where $\beta(r^\alpha) \in R_{x^{\alpha\beta}}$.
- Send r^β_c to P_S.

Sender 2
- For $\mathsf{b} \in \{0,1\}$, compute $\mathsf{k}_\mathsf{b} := \mathcal{F}_{\mathsf{RO}}(x_\mathsf{b})$ where $\alpha(\gamma_\mathsf{b}^{-1}(r^\beta_\mathsf{c})) \in R_{x_\mathsf{b}}$.
- For $\mathsf{b} \in \{0,1\}$, compute $\mathsf{e}_\mathsf{b} \leftarrow \mathsf{Enc}(\mathsf{k}_\mathsf{b}, \mathsf{m}_\mathsf{b})$.
- Send $(\mathsf{e}_0, \mathsf{e}_1)$ to P_R.

Receiver 2 Return $\mathsf{m}_\mathsf{c} := \mathsf{Dec}(\mathsf{k}_\mathsf{c}, \mathsf{e}_\mathsf{c})$.

Fig. 12: The protocol Π^2_{OT} for realizing $\mathcal{F}_{\mathsf{OT}}$ from semi-commutative masking.

Protocol Π^2_{OT} makes use of random sampling from M_i, evaluation of masks, evaluation of H, evaluation of Enc, Dec, as well as membership and equality testing in X and $\mathcal{C}_\mathcal{E}$ and inversion in M_i. All these operations are assumed to be efficient for the masking structure $\mathcal{M}$ and for the symmetric scheme $\mathcal{E}$. Because $\mathcal{M} = \{X, R_X[M_A, M_B, M_C]\}$ is semi-commutative, we see that

$$\alpha(\gamma_\mathsf{b}^{-1}(r^\beta_\mathsf{c})) = \alpha(\gamma_\mathsf{b}^{-1}(\beta(\gamma_\mathsf{c}(r)))) \in R_{x^{\alpha\beta}} \iff \mathsf{b} = \mathsf{c}$$

which shows that, if both parties execute the protocol honestly, $\mathsf{k}_R = \mathsf{k}_\mathsf{c}$ and hence P_R recovers the correct message m_c.

Theorem 6.2. *The protocol Π^2_{OT} of Figure 12 securely UC-realizes the functionality $\mathcal{F}_{\mathsf{OT}}$ of Figure 1 in the $\mathcal{F}_{\mathsf{RO}}$-hybrid model for semi-honest adversaries and static corruptions, under the assump-*

tion that $\mathcal{E}$ is IND-CPA-secure, that $\mathcal{M}$ is IND-Mask-secure and that the $\mathsf{ParallelBoth}^{\mathcal{M}}$ *problem is hard.*

Proof. We prove that there exists a PPT simulator $\mathcal{S}$, with access to an ideal functionality $\mathcal{F}_{\mathsf{OT}}$, which simulates the adversary's view. We divide the proof according to the selection of the corrupt parties.

Corrupt receiver and corrupt sender. As both parties are corrupt, the simulator $\mathcal{S}$ may read their inputs from their internal state and use those to create a perfect simulation of the transcript and of the parties' internal states. It presents this simulation to its internal copy of $\mathcal{A}$, together with an perfect simulation of $\mathcal{F}_{\mathsf{RO}}$, with which it is then able to perfectly answer $\mathcal{Z}$'s queries by forwarding them to $\mathcal{A}$ and returning the responses. Since it knows all of the inputs, it forwards them to $\mathcal{F}_{\mathsf{OT}}$ at the right moment to ensure that the dummy corrupt parties return the correct output to $\mathcal{Z}$.

Corrupt receiver and honest sender. We formally describe the simulator $\mathcal{S}_{R^*}$ in Figure 13.

Simulator $\mathcal{S}_{R^*}$

- $\mathcal{S}_{R^*}$ simulates $\mathcal{F}_{\mathsf{RO}}$ consistently and presents the state and transcript computed as follows to an internal copy of $\mathcal{A}$ to reply to the queries from $\mathcal{Z}$.
- When $\mathcal{Z}$ activates the honest Sender, $\mathcal{S}_{R^*}$ computes Sender 1 honestly to send (r^α, r_0, r_1) to $\mathcal{A}$.
- When $\mathcal{Z}$ activates the corrupt Receiver, its private input c is visible by $\mathcal{S}_{R^*}$ which can then compute r_{c}^β to perfectly simulate Receiver 1.
- To simulate Sender 2, since m_{c} appears on the corrupt Receiver's output tape, the simulator computes k_{c} and e_{c} as prescribed by the protocol.
  However, since $\mathcal{S}_{R^*}$ does not learn the honest input $\mathsf{m}_{1-\mathsf{c}}$, it samples $\mathsf{k}_{1-\mathsf{c}} \xleftarrow{\$} \mathcal{K}_{\mathcal{E}}$ at random and sets $\mathsf{e}_{1-\mathsf{c}} \leftarrow \mathsf{Enc}(\mathsf{k}_{1-\mathsf{c}}, \mathsf{m})$ for an arbitrary $\mathsf{m} \in \mathcal{M}_{\mathcal{E}}$.
- If $\mathcal{Z}$ queries either $\mathcal{F}_{\mathsf{RO}}(x_{\mathsf{b}}^\alpha)$ before activating Sender 2, then $\mathcal{S}_{R^*}$ aborts the simulation by returning $\perp$ to $\mathcal{Z}$.
- Finally, $\mathcal{S}_{R^*}$ finishes the protocol as prescribed.

Fig. 13: The simulator $\mathcal{S}_{R^*}$ of Theorem 6.2

We show that for every semi-honest adversary $\mathcal{A}$ who corrupts P_R and any environment $\mathcal{Z}$, it holds that

$$\mathsf{HYBRID}^{\mathcal{F}_{\mathsf{RO}}}_{\Pi^2_{\mathsf{OT}},\mathcal{A},\mathcal{Z}} \overset{\mathrm{c}}{\approx} \mathsf{IDEAL}_{\mathcal{F}_{\mathsf{OT}},\mathcal{S}_{R^*},\mathcal{Z}},$$

by proceeding via a sequence of hybrid simulators, going from the real execution to the ideal execution, defined as follows.

The first hybrid $\mathcal{H}_0$ knows the inputs of the honest sender and is therefore able to compute $\mathsf{e}_{1-\mathsf{c}}$ honestly using the correct random oracle query to obtain the key. This is then a perfect simulation of a real-world execution.

The second hybrid $\mathcal{H}_1$ samples $\mathsf{k}_{1-\mathsf{c}} \xleftarrow{\$} \mathcal{K}_{\mathcal{E}}$ at random and does not query the random oracle on $x_{1-\mathsf{c}}$ where $\alpha(\gamma^{-1}_{1-\mathsf{c}}(r^\beta_{\mathsf{c}})) \in R_{x_{1-\mathsf{c}}}$.

Claim. Any environment $\mathcal{Z}$ that distinguishes an interaction with $\mathcal{H}_1$ from one with $\mathcal{H}_0$ with advantage ϵ can be used to solve the $\mathsf{ParallelBoth}$ problem for $\mathcal{M}$ with advantage at least ϵ/q_H where q_H denotes the number of queries made by $\mathcal{Z}$ to the random oracle. Such an environment is capable

of distinguishing if and only if it submits the query for $\mathsf{k}_{1-\mathsf{c}}$ to the random oracle. We use this to build a reduction $\mathcal{B}$ agains the ParallelBoth problem for $\mathcal{M}$ which proceeds as follows.

Upon receiving a challenge $(C, A, r, r_{x_0}, r_{x_1}, r_y)$, $\mathcal{B}$ first sets $z^\alpha := r_y$ and $z_i := r_{x_i}$ to simulate Sender 1 and then samples $\beta \xleftarrow{\$} M_B$ to compute Receiver 1 perfectly upon activation of P_R^* which reveals c.

Since it now does not know the $\alpha \in M_A$ such that $r^\alpha = \alpha(r)$, $\mathcal{B}$ computes k_c from $\beta(r^\alpha)$ which it can do as it knows β and which yields the correct $x^{\alpha\beta}$ as the masks commute. For the other key, it sets $\mathsf{k}_{1-\mathsf{c}} \xleftarrow{\$} \mathcal{K}_\mathcal{E}$ as $\mathcal{S}_1$ would. It then returns the ciphertexts encrypting $\mathsf{m}_0, \mathsf{m}_1$ under these keys.

When $\mathcal{Z}$ terminates, $\mathcal{B}$ selects a random entry on the list of random oracle queries and applies β^{-1}. The un-selected key $\mathsf{k}_{1-\mathsf{c}}$ is the hash of the element of X represented by $\alpha(\gamma_{1-\mathsf{c}}^{-1}(\beta(\gamma_\mathsf{c}(r))))$ where $\gamma_i \in M_C$ is such that $r_{x_i} = \gamma_i(r)$. So by applying β^{-1}, $\mathcal{B}$ obtains exactly a representative one of the solutions to the ParallelBoth problem as long as it selected the correct entry on the hash list. If $\mathcal{Z}$ has advantage ϵ in distinguishing between $\mathcal{H}_1$ and $\mathcal{H}_0$, $\mathcal{B}$ then has an advantage ϵ/q_H in solving the ParallelBoth problem.

The final hybrid $\mathcal{H}_2$ replaces $\mathsf{m}_{1-\mathsf{c}}$ by an arbitrary $\mathsf{m} \in \mathcal{M}_\mathcal{E}$ in the computation of $\mathsf{e}_{1-\mathsf{c}}$. This removes the last occurrence of $\mathsf{m}_{1-\mathsf{c}}$ in the simulator and we have that $\mathcal{H}_2$ is identical to $\mathcal{S}_{R^*}$.

Claim. Any environment $\mathcal{Z}$ that can distinguish between a simulation of $\mathcal{H}_2$ and of $\mathcal{H}_1$ with advantage ϵ can be used to break the IND-CPA property of $\mathcal{E}$ with advantage at least ϵ.
We can build an adversary against the IND-CPA property of $\mathcal{E}$ by querying the challenger for a ciphertext of either m or $\mathsf{m}_{1-\mathsf{c}}$. This reduction emulates either $\mathcal{H}_2$ or $\mathcal{H}_1$ perfectly as $\mathsf{k}_{1-\mathsf{c}}$ is not accessible to $\mathcal{Z}$ and therefore not required by $\mathcal{S}_2$ or $\mathcal{S}_1$ at any point.

Under the assumption that $\mathcal{E}$ is IND-CPA-secure and that the ParallelBoth problem is hard for $\mathcal{M}$, we have that the simulation generated by $\mathcal{S}_{R^*}$ is indistinguishable from a real world execution, for any environment $\mathcal{Z}$. This concludes the proof that $\mathsf{HYBRID}^{\mathcal{F}_\mathsf{RO}}_{\Pi^2_\mathsf{OT},\mathcal{A},\mathcal{Z}} \overset{\mathrm{c}}{\approx} \mathsf{IDEAL}_{\mathcal{F}_\mathsf{OT},\mathcal{S}_{R^*},\mathcal{Z}}$.

Honest receiver and corrupt sender. We formally describe the simulator $\mathcal{S}_{S^*}$ in Figure 14.

Simulator $\mathcal{S}_{S^*}$

- $\mathcal{S}_{S^*}$ simulates $\mathcal{F}_\mathsf{RO}$ consistently and presents the state and transcript computed as follows to an internal copy of $\mathcal{A}$ to reply to the queries from $\mathcal{Z}$.
- When $\mathcal{Z}$ activates the corrupt sender, $\mathcal{S}_{S^*}$ computes Sender 1 honestly to send (r^α, r_0, r_1) to $\mathcal{A}$.
- As it does not know the c of the honest receiver, $\mathcal{S}_{S^*}$ proceeds by setting $\mathsf{c} = 0$ internally which remains out of the view of $\mathcal{A}$. It then samples $\beta \xleftarrow{\$} M_B$ and sets $r_0^\beta = \beta(r_0)$ consistently. It also computes k_0 accordingly.
- As it knows the inputs m_0, m_1 of the corrupt Sender, $\mathcal{S}_{S^*}$ computes Sender 2 consistently with r_c^β using the correct plaintexts.
- Finally, $\mathcal{S}_{S^*}$ finishes the protocol as prescribed.

Fig. 14: The simulator $\mathcal{S}_{S^*}$ of Theorem 6.2

We show that for every semi-honest adversary $\mathcal{A}$ who corrupts P_S and any environment $\mathcal{Z}$, it holds that

$$\mathsf{HYBRID}^{\mathcal{F}_\mathsf{RO}}_{\Pi^2_\mathsf{OT},\mathcal{A},\mathcal{Z}} \overset{\mathrm{c}}{\approx} \mathsf{IDEAL}_{\mathcal{F}_\mathsf{OT},\mathcal{S}_{S^*},\mathcal{Z}}.$$

The simulation of $\mathcal{S}_{S^*}$ is not a perfect simulation of a real world execution only if the honest receiver had actually received input $\mathsf{c} = 1$ from $\mathcal{Z}$. In that case, any environment that can distinguish between a simulation of $\mathcal{S}_{S^*}$ and the real world with advantage ϵ can be used to break the IND-Mask security of $\mathcal{M}$ with advantage at least ϵ. We build a reduction $\mathcal{B}$ against the IND-Mask experiment as follows.
It first simulates Sender 1 as prescribed by the protocol and sends $(r, \gamma_0, \gamma_1, B)$ to the IND-Mask experiment. Upon receiving $\tilde{r}$, $\mathcal{B}$ then returns $r_{\mathsf{c}}^{\beta} = \tilde{r}$ to the adversary when $\mathcal{Z}$ activates Receiver 1. Not knowing β is not a problem for the simulation as the receiver is honest and therefore $\mathcal{B}$ does not need to simulate its state to $\mathcal{A}$.
This is a perfect simulation of either the real world or of $\mathcal{S}_{S^*}$ as either r_1 or r_0 is used by the IND-Mask experiment in the computation of r_{c}^{β}. Thus if $\mathcal{Z}$ distinguishes between the two, then $\mathcal{B}$ can distinguish the hidden bit of the IND-Mask experiment.

Honest receiver and honest sender. In this final case, the simulator $\mathcal{S}$ chooses arbitrary inputs $\mathsf{m}_0 = \mathsf{m}_1 = \mathsf{m} \in \mathcal{M}_{\mathcal{E}}$ and $\mathsf{c} = 1$ and simulates a transcript to $\mathcal{A}$ using those. If an environment $\mathcal{Z}$ is capable of distinguishing this simulations from a real execution of the protocol then this implies that is is able to extract information regarding the arbitrary inputs used by $\mathcal{S}$. However the previous two cases show that, even with the additional information of the corrupted party's internal state, any environment is not able to identify a simulation that does not have any information the honest party's inputs. By combining techniques from both cases above, we can therefore show that the simulation of $\mathcal{S}$ is indistinguishable from a real world execution under the assumption that $\mathcal{S}$ is IND-CPA-secure, that $\mathcal{M}$ is IND-Mask-secure and that the $\mathsf{ParallelBoth}^{\mathcal{M}}$ problem is hard.

This completes the proof that for any $\mathcal{A}$ there exists a $\mathcal{S}$ such that, for any $\mathcal{Z}$,

$$\mathsf{HYBRID}^{\mathcal{F}_{\mathsf{RO}}}_{\Pi^2_{\mathsf{OT}},\mathcal{A},\mathcal{Z}} \overset{\mathrm{c}}{\approx} \mathsf{IDEAL}_{\mathcal{F}_{\mathsf{OT}},\mathcal{S},\mathcal{Z}}.$$

□

Acknowledgements

This work has been supported in part by ERC Advanced Grant ERC-2015-AdG-IMPaCT, by the Defense Advanced Research Projects Agency (DARPA) and Space and Naval Warfare Systems Center, Pacific (SSC Pacific) under contract No. N66001-15-C-4070, and by EPSRC via grants EP/M012824 and EP/N021940/1.

References

AJK+16. Reza Azarderakhsh, David Jao, Kassem Kalach, Brian Koziel, and Christopher Leonardi. Key compression for isogeny-based cryptosystems. In K. Emura, G. Hanaoka, and R. Zhang, editors, *Proceedings of the 3rd ACM International Workshop on ASIA Public-Key Cryptography, APKC*, pages 1–10. ACM, 2016.

BD18. Zvika Brakerski and Nico Döttling. Two-message statistical sender-private ot from lwe. Cryptology ePrint Archive, Report 2018/530, 2018. https://eprint.iacr.org/2018/530.

BDD+17. Paulo S. L. M. Barreto, Bernardo David, Rafael Dowsley, Kirill Morozov, and Anderson C. A. Nascimento. A framework for efficient adaptively secure composable oblivious transfer in the ROM. Cryptology ePrint Archive, Report 2017/993, 2017. http://eprint.iacr.org/2017/993.

BOB18. Paulo Barreto, Glaucio Oliveira, and Waldyr Benits. Supersingular isogeny oblivious transfer. arXiv preprint 1805.06589, 2018. https://arxiv.org/abs/1805.06589.

BPRS17. Megha Byali, Arpita Patra, Divya Ravi, and Pratik Sarkar. Fast and universally-composable oblivious transfer and commitment scheme with adaptive security. Cryptology ePrint Archive, Report 2017/1165, 2017. https://eprint.iacr.org/2017/1165.

BS18. Xavier Bonnetain and Andreé Schrottenloher. Quantum security analysis of csidh and ordinary isogeny-based schemes. Cryptology ePrint Archive, Report 2018/537, 2018. https://eprint.iacr.org/2018/537.

Can01. Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd Annual Symposium on Foundations of Computer Science*, pages 136–145. IEEE Computer Society Press, October 2001.

CJS14. Andrew Childs, David Jao, and Vladimir Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *Journal of Mathematical Cryptology*, 8(1):1–29, 2014. A pre-print version appears at https://arxiv.org/abs/1012.4019.

CLM+18. Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. Csidh: An efficient post-quantum commutative group action. Cryptology ePrint Archive, Report 2018/383, 2018. https://eprint.iacr.org/2018/383.

CO15a. Tung Chou and Claudio Orlandi. The simplest protocol for oblivious transfer. In Kristin E. Lauter and Francisco Rodríguez-Henríquez, editors, *Progress in Cryptology - LATINCRYPT 2015: 4th International Conference on Cryptology and Information Security in Latin America*, volume 9230 of *Lecture Notes in Computer Science*, pages 40–58. Springer, Heidelberg, August 2015.

CO15b. Tung Chou and Claudio Orlandi. The simplest protocol for oblivious transfer. Cryptology ePrint Archive, Report 2015/267, 2015. http://eprint.iacr.org/2015/267.

Cou06. Jean-Marc Couveignes. Hard homogeneous spaces. Cryptology ePrint Archive, Report 2006/291, 2006. http://eprint.iacr.org/2006/291.

DFJP14. Luca De Feo, David Jao, and Jérôme Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *Journal of Mathematical Cryptology*, 8(3):209–247, 2014. A pre-print version appears at https://eprint.iacr.org/2011/506.

EGL82. Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology – CRYPTO'82*, pages 205–210. Plenum Press, New York, USA, 1982.

FKS18. Luca De Feo, Jean Kieffer, and Benjamin Smith. Towards practical key exchange from ordinary isogeny graphs, 2018. Preprint.

GPS17. Steven D. Galbraith, Christophe Petit, and Javier Silva. Identification protocols and signature schemes based on supersingular isogeny problems. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 3–33. Springer, Heidelberg, December 2017.

GV17. Steven D. Galbraith and Frederik Vercauteren. Computational problems in supersingular elliptic curve isogenies. Cryptology ePrint Archive, Report 2017/774, 2017. http://eprint.iacr.org/2017/774.

Kie17. Jean Kieffer. Étude et accélération du protocole d'échange de clés de couveignes-rostovtsev-stolbunov. Master's thesis, Université Paris VI, 2017. Mémoire du Master 2, https://arxiv.org/abs/1804.10128.

Kil88. Joe Kilian. Founding cryptography on oblivious transfer. In *20th Annual ACM Symposium on Theory of Computing*, pages 20–31. ACM Press, May 1988.

NNOB12. Jesper Buus Nielsen, Peter Sebastian Nordholt, Claudio Orlandi, and Sai Sheshank Burra. A new approach to practical active-secure two-party computation. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 681–700. Springer, Heidelberg, August 2012.

NP01. Moni Naor and Benny Pinkas. Efficient oblivious transfer protocols. In S. Rao Kosaraju, editor, *12th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 448–457. ACM-SIAM, January 2001.

Orl18. Claudio Orlandi. Rump session talk, TPMPC 2018, Aarhus, Denmark, 2018.

Pet17. Christophe Petit. Faster algorithms for isogeny problems using torsion point images. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017, Part II*, volume 10625 of *Lecture Notes in Computer Science*, pages 330–353. Springer, Heidelberg, December 2017.

PVW08. Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 554–571. Springer, Heidelberg, August 2008.

Rab81. Michael O. Rabin. How to exchange secrets by oblivious transfer. Technical Report TR-81, Aiken Computation Laboratory, Harvard University, 1981.

RS06. Alexander Rostovtsev and Anton Stolbunov. Public-Key Cryptosystem Based On Isogenies. Cryptology ePrint Archive, Report 2006/145, 2006. http://eprint.iacr.org/2006/145.

Sch90. Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *Advances in Cryptology – CRYPTO'89*, volume 435 of *Lecture Notes in Computer Science*, pages 239–252. Springer, Heidelberg, August 1990.

UJ18. David Urbanick and David Jao. Sok: The problem landscape of sidh. In *APKC'18: Proceedings of the 5th ACM on ASIA Public-Key Cryptography Workshop*, pages 53–60. ACM, 2018.

Wal99. Colin D. Walter. Montgomery exponentiation needs no final subtractions. *Electronics Letters*, 35(21):1831–1832, 1999.

Zen18. Bing Zeng. Founding cryptography on smooth projective hashing. Cryptology ePrint Archive, Report 2018/444, 2018. https://eprint.iacr.org/2018/444.

A Symmetric encryption

We recall the syntax of a symmetric encryption scheme and the definition of IND-CPA security.

Definition A.1 (Symmetric encryption scheme). *A* symmetric encryption scheme *is a triple of probabilistic polynomial -time (PPT) algorithms* $\mathcal{E} := (\mathsf{KGen}_{\mathcal{E}}(\cdot), \mathsf{Enc}(\cdot,\cdot), \mathsf{Dec}(\cdot,\cdot))$ *together with a triple of sets* $(\mathcal{K}_{\mathcal{E}}, \mathcal{M}_{\mathcal{E}}, \mathcal{C}_{\mathcal{E}})$. *The* key generation algorithm $\mathsf{KGen}_{\mathcal{E}}(1^{\lambda})$ *takes as input a security parameter* 1^{λ} *and outputs a uniformly distributed key* $\mathsf{k} \xleftarrow{\$} \mathcal{K}_{\mathcal{E}}$. *The* encryption algorithm $\mathsf{Enc}(\mathsf{k}, m)$ *takes as input a key* $\mathsf{k} \in \mathcal{K}_{\mathcal{E}}$ *and a message* $m \in \mathcal{M}_{\mathcal{E}}$ *and outputs a ciphertext* $c \in \mathcal{C}_{\mathcal{E}}$. *The* decryption algorithm $\mathsf{Dec}(\mathsf{k}, c)$ *takes as input a key* $\mathsf{k} \in \mathcal{K}_{\mathcal{E}}$ *and a ciphertext* $c \in \mathcal{C}_{\mathcal{E}}$ *and outputs a message* $m' \in \mathcal{M}_{\mathcal{E}}$ *or a failure message* $\perp$. *For correctness, we require that* $\forall\, m \in \mathcal{M}_{\mathcal{E}}, \forall\, \mathsf{k} \in \mathcal{K}_{\mathcal{E}}, \mathsf{Dec}(\mathsf{k}, \mathsf{Enc}(\mathsf{k}, m)) = m$.

Definition A.2 (IND-CPA security).

Let $\mathcal{E} = (\mathsf{KGen}_{\mathcal{E}}, \mathsf{Enc}, \mathsf{Dec})$, *together with* $\mathcal{K}_{\mathcal{E}}, \mathcal{M}_{\mathcal{E}}, \mathcal{C}_{\mathcal{E}}$ *be a symmetric encryption scheme. For an arbitrary adversary* $\mathcal{A}$, *we define the* $\mathit{IND\text{-}CPA}_{\mathcal{A},\mathcal{E}}(\lambda)$ *experiment in Figure 15.*

Data: $\mathcal{E}$, $\lambda \in \mathbb{N}$
Result: $\mathsf{win} \in \{0,1\}$
1 $\mathsf{k} \xleftarrow{\$} \mathsf{KGen}_{\mathcal{E}}(1^{\lambda})$;
2 $(\mathsf{m}_0, \mathsf{m}_1), \mathsf{st} \leftarrow \mathcal{A}(1^{\lambda})$ such that $\mathsf{m}_0, \mathsf{m}_1 \in \mathcal{M}_{\mathcal{E}}$;
3 $\mathsf{b} \xleftarrow{\$} \{0,1\}$;
4 $\mathsf{e} \leftarrow \mathsf{Enc}(\mathsf{k}, \mathsf{m}_{\mathsf{b}})$;
5 $\tilde{\mathsf{b}} \leftarrow \mathcal{A}(1^{\lambda}, \mathsf{st}, \mathsf{e})$;
6 **if** $\tilde{\mathsf{b}} = \mathsf{b}$, **then return** $\mathsf{win} = 1$ **else return** $\mathsf{win} \xleftarrow{\$} \{0,1\}$;

Fig. 15: The $\mathsf{IND\text{-}CPA}_{\mathcal{A},\mathcal{E}}$ security experiment

We then say that $\mathcal{E}$ *is* IND-CPA-secure *if for all PPT adversaries* $\mathcal{A}$, *it holds that*

$$\left| \Pr\left[\mathit{IND\text{-}CPA}_{\mathcal{A},\mathcal{E}}(\lambda) = 1 \right] - \frac{1}{2} \right| \leq \mathsf{negl}(1^{\lambda}).$$

B UC security

We present a semi-formal overview of the universally composable (UC) model of security established by Canetti [Can01]. Protocols that aim to achieve security in this model are defined in three steps. First, the protocol and its execution in the presence of an adversary are formalized, this represents the *real-life model* which we also call the *real world*. Next, an ideal process for executing the task is defined; its role is to act as a trusted party by separately receiving the input of each party, honestly computing the result of the protocol internally and returning the output assigned to each party. In this *ideal world*, the parties do not communicate with one another but instead solely rely on the *ideal functionality* to provide them with their output. Finally, we say that the protocol in question *UC-realizes* the ideal functionality if running the protocol is equivalent to emulating the ideal functionality. We provide a brief discussion with additional formal details for the case of *semi-honest* adversaries with *static corruptions*.

In the real world, the parties involved in the execution of a protocol Π perform their own computation and communicate with one another when required to do so. Also present in the execution model is an adversary $\mathcal{A}$ which not only observes the messages exchanged but is also responsible for their delivery. This implies that it can choose to deliver them in the wrong order or to not deliver them at all. We however assume that communication is authenticated and that $\mathcal{A}$ can therefore only deliver messages that were previously sent, without modifying them, and that it cannot deliver the same message more than once.

The final entity present in this execution model is the environment $\mathcal{Z}$ which represents all of the events happening on the network at the time of the protocol execution. This environment is responsible for deciding the inputs and receiving the outputs of all the parties executing the protocol; this communication takes place outside of the view of $\mathcal{A}$ but we note that $\mathcal{A}$ still learns the inputs and outputs of corrupt parties as it is able to read their internal state. Furthermore, $\mathcal{Z}$ interacts with $\mathcal{A}$ *throughout* the execution of the protocol Π.

In the ideal world, the parties instead interact with an ideal functionality $\mathcal{F}$ in a simple way: they pass their private inputs to $\mathcal{F}$ and wait for it to return their assigned output. There is also an adversary $\mathcal{S}$ which is responsible for the delivery of messages. As we assume that the functionality is a trusted third party, this adversary cannot observe the content of the messages. Finally, the same environment $\mathcal{Z}$ is present in the ideal world. $\mathcal{Z}$ also prescribes the inputs and observes the outputs of all parties and may interact with $\mathcal{S}$ throughout the execution of the ideal process.

In the *static corruptions* strategy, the adversary ($\mathcal{A}$ or $\mathcal{S}$) may choose, at the beginning of the execution only, to corrupt one or more parties in the protocol. After the execution begins, it is not allowed to corrupt new parties.

We also formalize *semi-honest* adversarial behaviour, also called *honest-but-curious*, by saying that the adversary may not send messages on behalf of corrupt parties. Instead, it is given read access to all of their internal state which includes their private input and output as well as their internal computations. In the real world, this forces $\mathcal{A}$ to follow the protocol honestly and in the ideal world, it restricts $\mathcal{S}$ to simply forwarding messages between parties and the functionality.

In addition to these two model of computation, the UC-framework also considers the $\mathcal{G}$-hybrid model where the parties in both real and ideal world have access to a copy of the ideal functionality

$\mathcal{G}$. In the real world, this is an independent trusted party that executes the functionality honestly. In the ideal world, $\mathcal{S}$ executes an internal copy of the functionality $\mathcal{G}$ and only interacts with $\mathcal{F}$. Particularly, the random oracle model (ROM) of classical models of cryptography is modelled here using a $\mathcal{F}_{\mathsf{RO}}$ functionality as shown in Figure 2 and by proving the security of protocol in the $\mathcal{F}_{\mathsf{RO}}$-hybrid model.

To then prove that a protocol Π securely UC-realizes an ideal functionality $\mathcal{F}$, one must show that, for every adversary $\mathcal{A}$ interacting with Π in the real world, there exists an adversary $\mathcal{S}$ (often called the *simulator*) interacting with $\mathcal{F}$ in the ideal world such that *no environment* $\mathcal{Z}$ should be able to distinguish if it is interacting with $\mathcal{A}$ or with $\mathcal{S}$.

In other words, for every $\mathcal{A}$, one needs to design an $\mathcal{S}$ which is capable of *simulating* the view of $\mathcal{A}$ (which includes the transcript of the protocol and the interal state of the corrupt parties) such that no $\mathcal{Z}$ can distinguish the simulation from a real execution.

In this work, we restrict all of the entities $\mathcal{A}, \mathcal{S}, \mathcal{Z}$ to PPT algorithms.

Malicious behaviour. A much stronger form of security allows the adversary $\mathcal{A}$ in the real world to *behave arbitrarily*, or *maliciously* and thus not follow the protocol specification. In this setting, the corrupt parties are removed from the execution environment (in both real and ideal world) and the adversary $\mathcal{A}$ is directly responsible for generating their messages in the execution of Π. In the ideal world, this implies that $\mathcal{S}$ has to engage with the functionality $\mathcal{F}$ on behalf of the corrupt parties.

In the proof of simulation, $\mathcal{S}$ is then able to run an internal black-box copy of $\mathcal{A}$ and is required to *detect* if $\mathcal{A}$ deviates from the protocol and *extract* from this the inputs to the functionality that will yield the correct outputs for the honest parties, as otherwise the environment would detect that it is interacting with the ideal adversary $\mathcal{S}$. This notion is significantly harder to achieve as it essentially guarantees that no real world adversary, even if it deviates arbitrarily from the protocol, is capable of extracting more information than is revealed by the ideal functionality.

Summing up, we say that the protocol Π securely realises the functionality $\mathcal{F}$ in the $\mathcal{G}$-hybrid model, if for every adversary $\mathcal{A}$, there exists a simulator $\mathcal{S}$ such that for every environment $\mathcal{Z}$,

$$\mathsf{HYBRID}^{\mathcal{G}}_{\Pi,\mathcal{A},\mathcal{Z}} \overset{\mathrm{c}}{\approx} \mathsf{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}},$$

where $\overset{\mathrm{c}}{\approx}$ is the standard notation of computational indistinguishability, $\mathsf{HYBRID}^{\mathcal{G}}_{\Pi,\mathcal{A},\mathcal{Z}}$ denotes the output of $\mathcal{Z}$ in an execution of the real protocol with the adversary $\mathcal{A}$ controlling the corrupted parties, and $\mathsf{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}$ denotes the output of $\mathcal{Z}$ in the ideal execution, where the simulator $\mathcal{S}$ plays the role of the honest parties in Π against an internal $\mathcal{A}$ and interacts as the corrupt parties with the functionality $\mathcal{F}$.