Semi-Commutative Masking: A Framework for Isogeny-based Protocols, with an Application to Fully Secure Two-Round Isogeny-based OT

Cyprien Delpech de Saint Guilhem[1,2][0000−0002−0147−2566], Emmanuela Orsini[1][0000−0002−1917−1833], Christophe Petit[3,4][0000−0003−3482−6743], and Nigel P. Smart[1,2][0000−0003−3567−3304]

[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
[4] Département d'informatique, Université libre de Bruxelles, Belgium
cyprien.delpechdesaintguilhem@kuleuven.be, emmanuela.orsini@kuleuven.be, christophe.f.petit@gmail.com, nigel.smart@kuleuven.be

Abstract. We define semi-commutative invertible masking structures which aim to capture the methodology of exponentiation-only protocol design (such as discrete logarithm and isogeny-based cryptography). We discuss two instantiations: the first is based on commutative group actions and captures both the action of exponentiation in the discrete logarithm setting and 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 construct two oblivious transfer protocols using this new structure and prove that these UC-securely realise the oblivious transfer functionality in the random-oracle-hybrid model against passive adversaries with static corruptions. Moreover, by starting from one of these two protocols and using the compiler introduced by Döttling et al. (Eurocrypt 2020), we achieve the first fully UC-secure two-round OT protocol based on supersingular isogenies.

1 Introduction

Since its beginnings, isogeny-based cryptography has progressed in several directions. First, that of protocol design, where primitives such as key-exchange and identification protocols [JD11, DFJP14, FTTY19] or signature schemes [GPS17], have already been constructed. Secondly, in the understanding of the concrete security of the computational assumptions [GPST16]. Finally, in the implementation methods for such protocols [CLN16, AJK+16, FLOR18].

Whilst development of discrete-logarithm-based protocols has been rich, in terms of number of primitives, in the context of isogeny-based systems there has been less success. One reason is that the subtleties of isogeny-based primitives can be counter-intuitive (and even dangerous when misunderstood [Gal19]). In particular, as noted in [JD11, DFJP14], isogeny-based systems lack the commutative property which is often exploited in discrete-logarithm-based cryptography. Furthermore, the space of computational problems and their precise formulation is still shifting.

Supersingular isogeny-based protocols have attracted increasing attention mainly for their potential for post-quantum cryptography. In this direction some recent works [Vit19, BDGM19a, BOB18] have proposed oblivious transfer (OT) protocols based on the hardness of supersingular isogeny problems. OT, originally introduced by Rabin in 1982 [Rab81], is a fundamental primitive that has been proved complete for both two-party and multi-party computation, and has been used

as building block in many efficient protocols [NNOB12, KOS16, WRK17]. Due to earlier interest in lattice-based and code-based cryptography, there have already been post-quantum OT protocols [PVW08, BDD+17, BD18] based on the LWE, LPN and McEliece assumptions.

As well as underlying security assumptions, when we consider the state-of-the art in post-quantum OT protocols we also need to take into account different factors, such as the security model and round complexity. Indeed, one of the most desiderable properties, is having OT protocols with high security guarantees and only two rounds of communication. However, this is very hard to achieve and especially in the malicious setting, when one of the parties involved in the computation can arbitrarily deviate from the protocol. Indeed two-round OT with simulation based security is impossible in the plain model [GO94], and we need to rely on setup assumptions such as a common reference string or a random oracle.

Our contribution. We consider a new approach for studying isogeny-based constructions by defining a new general framework for exponentiation-only protocols. We then apply this new structure and describe two simple oblivious transfer protocols with high security guarantees and minimal round complexity, indeed we provide the first fully UC-secure two-round OT protocol based on supersingular isogenies.

Semi-commutative masking. We define new structures called *semi-commutative invertible masking schemes* to capture the exponentiation-only restriction of isogeny-based protocols and help draw out parallels with discrete-logarithm-based protocols. These also capture the absence of full commutativity in supersingular isogenies within a framework that is notationally simpler. In the full version, we show that these structures can also be realised in the discrete logarithm-based setting and in the setting of class group actions on endomorphism rings [CLM+18]. Moreover, we define generic computational problems for our structure and show that these correspond closely to the existing problems in the literature. The combination of our new structure together with instantiation-independent computational problems enables a clearer protocol design methodology. Furthermore, we believe that the hardness assumptions that we present can be extended to ones where more elements are given as a challenge (for example as used in pairing-based crypto). Such extended assumptions may enable the generic construction of schemes and protocols with richer functionalities as they have in the discrete-logarithm setting.

Isogeny-based oblivious transfer. We illustrate the advantage of our framework with two OT protocols constructed from our masking schemes. The first protocol is inspired by the Shamir 3-pass key transport protocol which we modify to satisfy the requirements of OT 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. Notably, our new structure allows us to provide a single proof of security for each protocol which is then valid for different instantiations of the masking scheme.

UC-secure isogeny-based two-round OT. This only provides a two-round passively secure protocol, however we also show how to obtain a two-round maliciously secure protocol. The known methods for maliciously-secure OT are either based on zero-knowledge proofs or on "lossy" encryption schemes [PVW08], which we don't know how to instantiate using isogeny-based constructions and/or without increasing the round complexity. In [DGH+20], Döttling et al. introduced

a general compiler to transform a rather weak and simple two-round *elementary*-OT (eOT), to a fully UC-secure two-round OT, providing also two instantiations: one based on the Computational Diffie-Hellman (CDH) problem and one on the Learning Parity with Noise (LPN) problem. We show (in Appendix 7) that our first protocol satisfies the security requirements of this compiler, establishing the feasibility of two-round UC-secure OT based on semi-commutative masking, and more in particular on supersingular isogenies assumptions. In fact, we achieve the stronger notion of *search*-OT (sOT) security which means that Döttling et al's expensive transformation from eOT to sOT is not required for our protocol. To do so, we introduce a new problem for our masking scheme, called ParallelDouble (Definition 7.3), that is comparable to the one-more static CDH problem (where the adversary has access to both a challenger and a helper oracle and has to solve one more challenge than it was helped on).

Related work. Since De Feo and Jao's work [JD11, DFJP14], others have explored different directions of supersingular isogenies [CLN16, AJK+16, GPST16, GPS17, FLOR18, FTTY19, CLM+18, SGP19, AJJS19, FTY19, LGD20]. However, to the best of our knowledge, our work is the first to present a framework for "exponentiation-based" protocols which unifies supersingular isogenies with previous constructions and also provides a separation between protocol design and analysis of computational assumptions. While we only present an OT protocol is this work, we believe that most of the works stated above can be formulated within our framework.

Recent works, concurrent and posterior to ours, have also proposed OT protocols based on supersingular isogenies [Vit19, BOB18, BDGM19b]. The first describes an instantiation which is comparable to ours, especially regarding the computation of inverses and the question of the Weil pairing. It also proposes two protocols inspired by the same exponentiation-based approach and constructed from the same key-exchange and key-transport mechanisms. However, thanks to our new structure, our protocols better refine and separate the required computations. Our first protocol fixes the two elements it requires for all instances, thus reducing the exchange to two flows – the best that can be hoped for, and the maximum allowed for Döttling et al.'s transformation to achieve UC security – instead of three, and it shifts the burden of computing the inverse to the Receiver. This reduces communication further and allows for only one inverse computation to be required. Our second protocol separates the transmission of key material and choice material from the Sender to the Receiver. This permits the Sender to contribute to the final encryption key which is closer in spirit to the original key-exchange protocol. Vitse [Vit19] also proposes an instantiation of her protocols from Kummer varieties; we leave it to further work to establish whether this could yield a new instantiation of our masking structure. Note, the works [BOB18, Vit19] only prove security in the stand-alone and game-based models respectively, as opposed to our proofs in the UC model and there is no extension to malicious security.

Following the blueprint of previous works [BPRS17, BDD+17], Branco et al. [BDGM19b] achieve active security for OT at the cost of three additional rounds of communication. However, this requires the addition of a new mechanism which diverges from the "exponentiation-only" methodology. Furthermore, the security of their isogeny-based mechanism relies on assumptions that were only recently proposed [BOB18] and have not yet been studied at length.

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

2 Preliminaries

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

2.1 Symmetric Encryption

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

Definition 2.1 (Symmetric encryption scheme). *A* symmetric encryption scheme *is a triple of probabilistic polynomial -time 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 2.2 (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 3. We then say that $\mathcal{E}$ is* IND-CPA-secure *if for all PPT adversaries $\mathcal{A}$, it holds that*

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

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

Fig. 3: The $\mathsf{IND\text{-}CPA}_{\mathcal{A},\mathcal{E}}$ security experiment

2.2 Universally Composable Security

We present here 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 internal 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.

Security statement. We then 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 $\overset{c}{\approx}$ denotes 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 $\mathcal{F}$.

3 Semi-Commutative Invertible Masking Structures

We first formally define our new masking structures and discuss some computational problems that arise in this setting. To help fix ideas we illustrate our masking structures with the case of discrete logarithms in a finite field $\mathbb{F}_p$, where $q = (p-1)/2$ is prime and $g \in \mathbb{F}_p$ is an element of order q.

3.1 Masking Structure

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 require that it be efficient to recover x from any representation in R_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. $\forall x, x' \in X$ s.t. $x \neq x',\ R_x \cap R_{x'} = \emptyset$, and we define $R = \cup_{x \in X} R_x$ to be the set of all representatives. For example, if we take $X = \langle g \rangle \subset \mathbb{F}_p^*$, then the usual choice for R is to let $R_x = \{x\}$ for every $x \in X$; but one could also take a redundant representation with two elements letting $R_x = \{x, x+p\}$.

A *mask* is a function $\mu : R \longrightarrow R$, and a masking set M is a set of such functions. In the discrete logarithm case, a natural candidate for M is a set indexed by elements in $\mathbb{Z}_q^*$ which each give 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

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

Note, we only require that μ^{-1} outputs a representative in the same set R_x. If all elements $\mu \in M$ are invertible, then we say that the masking set M is *invertible*. In the discrete logarithm case, if μ corresponds to the map $g \mapsto g^a$, 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

$$\forall i \neq j,\ \forall \mu \in M_i,\ \forall \mu' \in M_j,\ \forall r \in R,\ \mu(\mu'(r)) \in R_x \iff \mu'(\mu(r)) \in R_x. \tag{2}$$

In the discrete logarithm case, with M a set of exponentiation functions, $\mathcal{M} = \{X, R_X, [M, M]\}$ is straightforwardly semi-commutative.

3.2 Problems and Properties

We now present a distinguishing experiment and computational problems for masking structures. The precise security level of these depends from concrete instantiations and reductions to specific computational problems.

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

Fig. 4: The $\mathsf{IND\text{-}Mask}_{\mathcal{A},\mathcal{M}}$ security experiment

Definition 3.1 (IND-Mask security). *We define the $\mathsf{IND\text{-}Mask}_{\mathcal{A},\mathcal{M}}$ experiment in Figure 4 for a masking structure $\mathcal{M} = \{X, R_X, [M_i]_{i=1}^n\}$, and an arbitrary adversary $\mathcal{A}$. We say that $\mathcal{M}$ is* IND-Mask-secure *if for all PPT adversaries $\mathcal{A}$, it holds that*

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

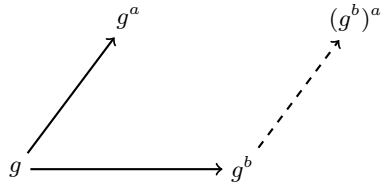
In the discrete logarithm setting, when $R_x = \{x\}$, the map $g \mapsto g^a$ for random $a \in \mathbb{Z}_q^*$ induces a random permutation of the group elements. Therefore for a secret a and two group elements g_0, g_1, the distribution of g_b^a is perfectly uniform, independently of b. This shows 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 the randomly sampled 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.

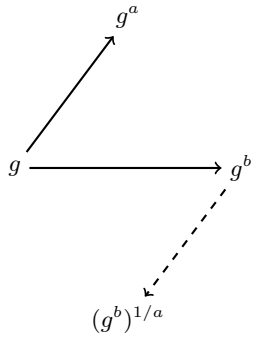
We also define to the following hard problems for semi-commutative invertible masking structures:

Definition 3.2. *Given a masking structure $\mathcal{M} = \{X, R_X, [M_i]_{i=1}^n\}$, we define the following computational problems:*

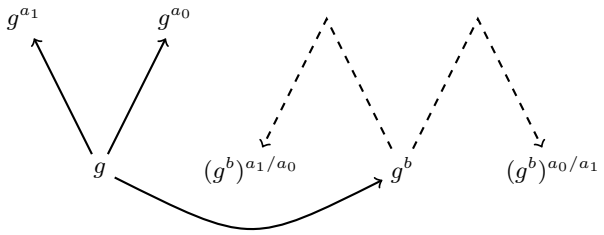
1. Demask*: Given (i, r, r_x) with the promise that $r_x = \mu_x(r)$ for a uniformly random $\mu_x \xleftarrow{\$} M_i$, return μ_x.*
2. Parallel*: Given (i, j, r, r_x, r_y) with the promise that $i \neq j$ and that $r_x = \mu_x(r)$ and $r_y = \mu_y(r)$ for uniformly random $\mu_x \xleftarrow{\$} M_i, \mu_y \xleftarrow{\$} M_j$, 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 $\mu_x^{-1}(r_y) \in R_z$.*



(a) The Parallel problem.



(b) The ParallelInv problem.



(c) The ParallelBoth problem.

Fig. 5: Representations of computational problems.

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* $\mu_\mathsf{b}^{-1}(\mu_{1-\mathsf{b}}(r_y)) \in R_z$*.*

To make explicit the given structure $\mathcal{M}$ *to which the (say)* Demask *problem refers, we write* $\mathsf{Demask}^{\mathcal{M}}$*. The name "*Parallel*" is inspired by a similar problem defined by Couveignes [Cou06].*

We motivate these problems 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 $\mathbb{Z}_q^*$. We give a graphical intuition of these problems in Figure 5.

- The Demask problem is, given (g, h) with the promise that $h = g^a$ for a random a, to return a. This is 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 the computational Diffie-Hellman (CDH) problem.
- In the discrete logarithm setting, the ParallelInv problem is to compute $g^{b/a}$ given (g, g^a, g^b). In the full version we show that this is equivalent to the Parallel problem. We note that this

does not immediately hold in the abstract case, due to the absence of 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 in Figure 5c.
  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.

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 such structures. This is also the case for 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

We first 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 $*$ defined as: $* : G \times X \to X$, with $g * x \mapsto y$.

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 then 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 homogeneous space X for G we define a masking structure $\mathcal{M}_{X,G} = \{X, R_X, [G, G]\}$ for X as follows:*

- *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-commutative 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 Prime Fields

The second realisation of hard homogeneous spaces we present is a summary of the recent work by Castryck et al. [CLM+18]. This work 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 Couveignes 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

To avoid a sub-exponential quantum attack vector [CJS14], De Feo, Jao and Plût [DFJP14] consider the use of supersingular elliptic curves over the extension field $\mathbb{F}_{p^2}$ 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 the Extension Field

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 $\mathbb{F}_q$; this 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 addition, $\phi \oplus \varphi : P \mapsto \phi(P) + \varphi(P)$, and multiplication, $\phi \otimes \varphi : P \mapsto \phi(\varphi(P))$, operations. 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 can be 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 K of E there is, up to isomorphism, a unique isogeny $\phi : E \to E'$ having kernel K and we therefore identify E' with the notation E/ϕ. Particularly, we will work with subgroups of the torsion group $E[m]$ for $m \in \mathbb{N}$ which is the group of points of E whose order divides m. When we also have that m^2 divides $\#E(\mathbb{F}_{p^2})$, we can always represent cyclic kernels by generators defined over $\mathbb{F}_{p^2}$.

Semi-commutativity. We introduce the notion of *semi-commutativity* present in this setting; the same notion is behind the SIDH key-exchange protocol [DFJP14] and we generalise it here. We 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. By construction, in each isomorphism class there is a curve $E/\mathbb{F}_{p^2}$ such that the torsion group $E[\ell_i^{e_i}]$ contains $\ell_i^{e_i-1}(\ell_i+1)$ cyclic subgroups of order $\ell_i^{e_i}$ (which each define a different isogeny).

To compute and publish a curve resulting from a secret isogeny, a party generates a secret key by selecting a random point K_i of order $\ell_i^{e_i}$ on a curve E and computes a public curve by computing the unique isogeny with kernel $\langle K_i\rangle$ and publishing the domain curve $E/\langle K_i\rangle$. The issue here is that the structure of $\mathrm{End}(E)$ no longer allows for 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 remains.

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 emerges since applying an isogeny of degree $\ell_i^{e_i}$ preserves the torsion groups $E[\ell_j^{e_j}]$ for $j \neq i$. Therefore, alongside publishing $E/\langle K_i\rangle$ 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 as $K_j = [\alpha_j]P_j + [\beta_j]Q_j$ and applying α_j, β_j to $\{\phi_i(P_j), \phi_i(Q_j)\}$, the other party can then compute an isogeny $\varphi_j : E/\langle K_i\rangle \to E/\langle K_i, K_j\rangle$ which is "parallel" to the isogeny $\phi_j : E \to E/\langle K_j\rangle$ in the sense of Figure 5a.

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, and the parties can then take the j-invariants of their respective curves as an identical shared value.

The Weil pairing. We recall here the notion of the *Weil pairing*. For any integer $m \in \mathbb{N}$, we let $\zeta_m = \{u \mid u^m = 1\} \subset \mathbb{F}_{p^2}^*$. For any curve $E/\mathbb{F}_{p^2}$, the Weil pairing is a map $e_m : E[m] \times E[m] \longrightarrow \zeta_m$, that satisfies $e_m(\phi(P), \phi(Q)) = e_m(P, Q)^{\deg(\phi)}$, where $\phi : E \to E'$ is any isogeny.

5.2 Masking Structure

To define 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 one of the two classes with curves $E/\mathbb{F}_{p^2}$ with trace $t = p^2 + 1 - \#E(\mathbb{F}_{p^2}) \in \{-2p, 2p\}$; these two classes are the largest of the five [AAM18].

Representatives. For each j-invariant $x \in X$, there is a canonical choice of curve E_x [Sil86]. For each E_x we take the appropriate twist of the curve such that they belong to the same isogeny class. We define the set R_x of representatives as the set of tuples $(E_x, \{\{P_i, Q_i\}_{i\in[n]}\})$ where $\{P_i, Q_i\}$ is a basis of the torsion group $E_x[\ell_i^{e_i}]$ as above.

For a given curve and torsion order, there exists a deterministic and efficient algorithm $\mathsf{Basis}(E, i)$ which outputs a basis $\{P_i, Q_i\} \subset E_x[\ell_i^{e_i}]$ [AJK+16, Section 3.2]; for each torsion order, we fix a generator $q_i \in \zeta_{\ell_i^{e_i}}$ such that for any curve E, $e_m(P_i, Q_i) = q_i$ for $\{P_i, Q_i\} \leftarrow \mathsf{Basis}(E, i)$. 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.

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

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

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

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 $\mathbb{P}^1(\mathbb{Z}/\ell_i^{e_i}\mathbb{Z})$. We therefore define n masking sets $[M_i]_{i\in[n]}$ where each M_i is the projective line $\mathbb{P}_i := \mathbb{P}^1(\mathbb{Z}/\ell_i^{e_i}\mathbb{Z})$.

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 $\mathsf{Basis}(E_y, i)$ algorithm.

The output of the computation of the mask $\mu(r)$ is therefore the curve $E_y \overset{\chi}{\simeq} E_x/\langle K_i \rangle$ together with the basis points $\{\{\chi \circ \phi_i(P_j), \chi \circ \phi_i(Q_j)\}\}$ for $j \neq i$ and the output of $\mathsf{Basis}(E_y, i)$.

Inverting the mask. Since our masking sets M_i do not derive from a group structure, 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 $\phi_i(E[\ell_i^{e_i}]) \subset E'[\ell_i^{e_i}]$ of the $\ell_i^{e_i}$-torsion group under the isogeny $\tilde{\phi}_i$ defined by K_i and 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 $\mu \in M_i = \mathbb{P}_i$ and elements r and $r' = \mu(r)$ with $r' = (E', \{\{P_j, Q_j\}_{j\in[n]}\})$, 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 $[\hat{\alpha}_i : \hat{\beta}_i] \in \mathbb{P}_i$, given μ and r. We note that the dependency of μ^{-1} on μ and r is consistent with the definition of the inverse of a mask as stated in Section 3.

Masking structure. We formally define a masking structure in this setting.

Definition 5.1 (Masking structure from supersingular isogenies). *Let p be a prime 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 semi-commutative.*

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. Then, 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. Also, the semi-commutative property of our structure follows from the semi-commutative property of isogenies of co-prime degrees. Finally, 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

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 discrete logarithm 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 [GV18], 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 the OT scheme presented in this paper will only require $n = 2$, as in the case of the SIDH key-exchange. Our second OT protocol (described in the full version) will require $n = 3$, 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 [GV18, 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 [GPS17, Pet17, GV18, KMP+20].

Additional information. This has led to the definition in the literature of a specific SIDH problem. Here we merge the definitions of [GV18] and [UJ18] for the case of $n = 2$ small primes in the composition of p.

Definition 5.3 (2-i-**isogeny problem [GV18, Def. 2][UJ18, Prob. 4.1]).** *Let $i \in \{1, 2\}$ and 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*, is, given $(E, P_1, Q_1, P_2, Q_2, E', P'_j, Q'_j)$, 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 n be an integer, $i \in \{1, \ldots, n\}$ and let $(E, \{P_j, Q_j\}_{j=1}^n)$ be a tuple 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*, for $i \in [n]$, is, given $(E, \{P_j, Q_j\}_{j=1}^n, E', \{P'_j, Q'_j\}_{j\neq i})$, 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 bases 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 $E/\langle K_A, K_B\rangle$.*

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*, for $i, j \in [n]$, 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. □

Regarding the ParallelInv problem for $\mathcal{M}_p$, we do not have an immediate reduction to the Parallel problemas we had for the previous instantiation. To follow the same proof strategy as for masking structures from homogeneous spaces one would have to swap r and r_x to submit a challenge to the oracle for the ParallelInv problem. The map from r_x to r would then be the inverse of the one from r to r_x but the map from r_x to r_y would no longer satisfy the promise of the ParallelInv problem. We discuss this interesting subtlety in the definitions of the CDH problem in the full version of this work. We nonetheless conjecture that, as they are very similar, the hardness of the ParallelInv problem is close to that of the Parallel problem. 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 formalise a decisional variant of the SIDH problem in the case of $n = 2$.

Definition 5.7 (**2-i-decisional SIDH problem [GV18, 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.

Security analysis. As mentioned above, one of the main advantage of the SIDH approach as opposed to the hard homogeneous space approach (including CSIDH) is that no sub-exponential attack is known on the SIDH protocol, even using a quantum computer. On the other hand in the SIDH protocol, the action of the secret isogeny on a large torsion subgroup is revealed. A paper by Petit [Pet17] and a recent follow-up work by Kutas et al. [KMP+20] show how to exploit this additional information to break variants of the SIDH protocol with unbalanced parameters or weak starting curves.

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] and [KMP+20] describe a generalization to any coprime, power-smooth values N_1, N_2. Under some parameter restrictions and heuristic assumptions, the best attack in [KMP+20] computes the isogeny in classical polynomial time assuming $\beta > 2\alpha > 2$ or $\beta > 3\alpha > 3/2$. Furthermore, Kutas et al. show an attack requiring only $\beta > 2\alpha$ (with no lower bound on α) when the protocol uses a weak starting curve.

In our instantiation above, for any i one can fix $\alpha = e_i \log \ell_i$ and $\beta = \sum_{j\neq i} e_j \log \ell_j$. We also have $\alpha + \beta \leq 1$ so the first attack in [Pet17] and its improvement in [KMP+20] does not apply if the starting curve is not weak. The second attack of [Pet17], however, applies whenever the number n of factors ℓ_i is larger than $O(e_i \log \ell_i)$ for some i. The second one from [KMP+20] applies if any starting curve is weak. The notion of weak however depends on p, α, β and the chosen curve so choosing correct parameters (as those chosen in SIDH are) prevents this from happening.

One may fear that these attacks 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 also that $n = 2$ in our construction

$\mathsf{CDH}_1(G = \langle g \rangle)$

1 : $a \xleftarrow{\$} \mathbb{Z}_q^*$

2 : $b \xleftarrow{\$} \mathbb{Z}_q^*$

3 : **output** (g, g^a, g^b)

(a) Traditional CDH.

$\mathsf{CDH}_2(G = \langle g \rangle)$

1 : $h \xleftarrow{\$} G$

2 : $a \xleftarrow{\$} \mathbb{Z}_q^*$

3 : $b \xleftarrow{\$} \mathbb{Z}_q^*$

4 : **output** (h, h^a, h^b)

(b) Random base CDH.

$\mathsf{CDH}_3(G = \langle g \rangle)$

1 : $h \xleftarrow{\$} G$

2 : $a \xleftarrow{\$} \mathbb{Z}_q^*$

3 : $i \xleftarrow{\$} G$

4 : **output** (h, h^a, i)

(c) Single promise CDH.

Fig. 6: Three versions of the CDH challenge creation.

corresponds to the SIDH protocol parameters, so our semi-commutative masking construction with $n = 2$ will remain secure as long as SIDH remains secure.

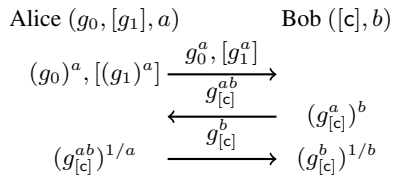
5.4 Different Formulations of the CDH Problem

In Figure 6, we present three subtly different versions of the CDH problem, simplified to their challenge creation and written using group exponentiation notation. The first formulation, in Figure 6a, reflects the original definition of the CDH problem where the first element of the tuple (g, g^a, g^b) is always the pre-defined generator g. This formulation differs from our definition of the Parallel problem as the r element of our challenge tuple (i, r, r_x, r_y) does not have to be any pre-defined value. Instead our formulation is aligned on the second version, presented in Figure 6b. This then allows for the equivalence between the Parallel and ParallelInv problems to be proven formally in the setting of homogeneous spaces. Indeed we can construct a tuple (i, j, r', r'_x, r'_y), with $r' = r_x, r'_x = r$ and $r'_y = r_y$, where the promise of a map in M_i between r' and r'_x is satisfied because of the inverse, and the promise of a map in M_j taking r' to r'_y holds because, in this setting, there necessarily exists a map between any two elements.

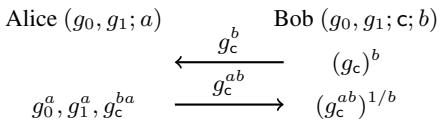
However, the second implication does not hold in the setting of supersingular isogenies. Indeed, swapping r and r_x results in a isogeny of degree $\ell_i^{e_i} \cdot \ell_j^{e_j}$ between the curves in r'_x and r'_y as opposed to an isogeny of degree $\ell_j^{e_j}$ as promised by the problem.

Formulating the CDH challenge differently and removing the promise between r and r_y, as presented in Figure 6c, would enable a formal reduction to be built between the Parallel and ParallelInv problems in this less structured setting. We note that the CDH_2 and CDH_3 formulations are in fact equivalent in our first setting of homogeneous spaces.

Changing our definitions to allow for this reduction to be proven would however cause the computational problems to be further removed from their usage in practice. Indeed, the messages exchanged in protocols constructed in this setting typically satisfy the promises of our problems as they currently stand and we therefore chose not to modify our definitions.



(a) The Shamir three-pass protocol and its OT variant



(b) Sketch of final OT protocol flows

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

6 Two Oblivious Transfer Protocols from Semi-Commutative Masking

6.1 First Construction - A 2-round OT Protocol

In this section we construct an OT protocol from a semi-commutative masking structure $\mathcal{M}$. We prove its UC security for passive adversaries with static corruptions in the $\mathcal{F}_{\mathsf{RO}}$-hybrid model assuming that $\mathcal{M}$ is IND-Mask-secure and that the ParallelEither$^{\mathcal{M}}$ problem is hard.

Motivation. Our OT protocol is inspired by the two-party Shamir three-pass protocol for secure message transmission shown in Figure 7a (ignoring the elements in square brackets), also known as the Massey-Omura encryption scheme. Here, Alice's input is a message g together with a secret mask a and Bob's input is another secret mask b. To transmit g, Alice first sends g^a to Bob who replies by masking it as g^{ab}. Now Alice removes her mask and replies with $g^{ab/a} = g^b$ to Bob who then inverts b and recovers g. This protocol can be modified to yield an OT protocol by including the elements in square brackets; this was proposed by Wu et al. [WZW03].

Alice, acting as Sender, now has two inputs g_0 and g_1 and masks both with 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 replies to Alice with $(g_{\mathsf{c}}^a)^b$. They then continue as before until Bob recovers g_{c}. The intuition for security 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 does not know b.

We remove the need to apply the inverse mask $1/a$ to g_{c}^{ab} since Alice's ignorance of c makes this impossible for general semi-commutative masking schemes due to the definition of inverse masks. In our new (discrete logarithm based) variant, the elements g_0 and g_1 are common to both parties. Rather than using 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 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 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* passes rather than three. Figure 7b abstracts the symmetric encryption and only shows the flows that lead to Bob receiving the value g_{c}^a.

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 $\mathcal{F}_{\mathsf{RO}} : \{0,1\}^\lambda \to \mathcal{K}_\mathcal{E}$.

OT_1 **(Receiver 1)**
- Sample $\beta \xleftarrow{\$} M_B$ uniformly at random.
- Compute $r^\beta_\mathsf{c} := \beta(r_\mathsf{c})$ and $\beta^{-1} \in M_B$.
- Send r^β_c to P_S.

OT_2 **(Sender 1)**
- Sample $\alpha \xleftarrow{\$} M_A$ and compute $r^\alpha_\mathsf{b} := \alpha(r_\mathsf{b}) \in R_{x^\alpha_\mathsf{b}},\ \mathsf{b} \in \{0,1\}$
- For $\mathsf{b} \in \{0,1\}$, call $\mathcal{F}_{\mathsf{RO}}$ twice on input x^α_b obtaining k_b, and compute $\mathsf{e}_\mathsf{b} \leftarrow \mathsf{Enc}(\mathsf{k}_\mathsf{b}, \mathsf{m}_\mathsf{b})$
- Compute $r^{\alpha\beta}_\mathsf{c} := \alpha(r^\beta_\mathsf{c})$
- Send $(r^{\alpha\beta}_\mathsf{c}, \mathsf{e}_0, \mathsf{e}_1)$ to P_R.

OT_3 **(Receiver 2)**
- Compute $r^\alpha_\mathsf{c} := \beta^{-1}(r^{\alpha\beta}_\mathsf{c})$ and $\mathsf{k}_R := \mathcal{F}_{\mathsf{RO}}(x^\alpha_\mathsf{c})$ where $r^\alpha_\mathsf{c} \in R_{x^\alpha_\mathsf{c}}$.
- 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.

Construction. We now formally define our OT protocol from semi-commutative invertible masking schemes. Let $\mathcal{M} = \{X, R_X, [M_A, M_B, M_C]\}$ be an SCM 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 $\mathcal{E}$. The protocol Π^1_{OT} is formally defined in Figure 8.

As described above, 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 $r^\beta_\mathsf{c} = \beta(r_\mathsf{c})$ 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 $r^\alpha_0 = \alpha(r_0)$ and $r^\alpha_1 = \alpha(r_1)$. These elements uniquely determine $x^\alpha_\mathsf{b} \in X, \mathsf{b} \in \{0,1\}$. 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^α_b) 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 $r^{\alpha\beta}_\mathsf{c} = \alpha(r^\beta_\mathsf{c})$ to P_R. The receiver has now all the information needed to recover the message m_c corresponding to its choice bit: it can apply the inverse β^{-1} to $r^{\alpha\beta}_\mathsf{c}$ using the semi-commutativity of $\mathcal{M}$, so that

$$\beta^{-1}(r^{\alpha\beta}_\mathsf{c}) = \beta^{-1}(\alpha(r^\beta_\mathsf{c})) = \beta^{-1}(\alpha(\beta(r_\mathsf{c}))) \in R_{x^\alpha_\mathsf{c}},$$

and recover $\mathsf{k}_\mathsf{c} = \mathcal{F}_{\mathsf{RO}}(x^\alpha_\mathsf{c})$. 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 assump-*

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 consistently. 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 1, 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

tion 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 a 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 $\mathcal{Z}$, we have that $\mathsf{HYBRID}^{\mathcal{F}_{\mathsf{RO}}}_{\Pi^1_{\mathsf{OT}},\mathcal{A},\mathcal{Z}} \overset{\mathrm{c}}{\approx} \mathsf{IDEAL}_{\mathcal{F}_{\mathsf{OT}},\mathcal{S}_{R^*},\mathcal{Z}}$, 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 $\mathsf{ParallelEither}$ 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 $\mathsf{ParallelEither}$ 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 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 since it is simulating the internal

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 $r_{\mathsf{c}}^{\beta} = \beta(r_0)$ consistently.
- As it knows the inputs m_0, m_1 of the corrupt Sender, $\mathcal{S}_{S^*}$ computes Sender 1 consistently with r_{c}^{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

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 occurrence 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 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.

The reduction 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 $r_{\mathsf{c}}^{\beta} = \tilde{r}$ 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 $\mathcal{A}$. 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 on 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. Protocol Π^1_{OT} only requires the third masking set M_C as a proof artefact and that only two sets would be sufficient to execute the protocol.

6.2 Second Construction - OT from Key-Exchange

Motivation Our second OT protocol is inspired by the OT protocol of Chou and Orlandi [CO15a] in that it uses an underlying key exchange mechanism and then transforms it to achieve oblivious transfer. The problems that have emerged in their construction [CO15b, Section 1.1] do not arise when considering passive adversaries so we do not address them here.

Again we motivate our proposed OT protocol by looking at the discrete logarithm variant. Here, Alice's inputs are two messages $\mathsf{m}_0, \mathsf{m}_1$ and an ephemeral mask a and 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 g^a to Bob who derives the decryption key g^{ab}. But Bob cannot simply reply with g^b, since Alice would then not know which of m_0 or m_1 to encrypt. Instead, Alice 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 share with Bob. We sketch the protocol flows in Figure 11. The protocol is intuitively secure as Alice cannot deduce b from Bob's message and Bob cannot deduce the key $\mathsf{k}_{1-\mathsf{c}}$ as he is not able to recover d_{b}^{-1} from Alice's first message.

Construction. We now formally define our second OT protocol from semi-commutative invertible masking schemes. 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. Protocol Π^2_{OT} makes use of random sampling from M_i,

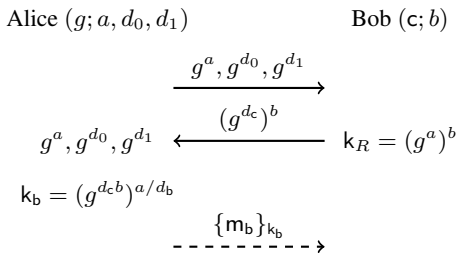


Fig. 11: Sketch of the OT protocol derived from the key agreement protocol.

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.

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 $\alpha(\gamma_\mathsf{b}^{-1}(r^\beta_\mathsf{c})) = \alpha(\gamma_\mathsf{b}^{-1}(\beta(\gamma_\mathsf{c}(r)))) \in R_{x^{\alpha\beta}}$ if and only if $\mathsf{b} = \mathsf{c}$. This 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 assumption 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.

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 $\mathcal{F}_{\mathsf{RO}}(x_{\mathsf{b}})$ before activating Receiver 1, 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

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. 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 $\alpha(\gamma^{-1}_{1-\mathsf{c}}(r_{\mathsf{c}}^\beta)) \in R_{x_{1-\mathsf{c}}}$.

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 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 $\mathcal{B}$ against 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 $z^\alpha \coloneqq r_y$ and $z_i \coloneqq r_{x_i}$ 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.

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

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. 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 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 $\mathsf{ParallelBoth}^{\mathcal{M}}$ problem is hard.
This completes the proof that for any $\mathcal{A}$ there exists a $\mathcal{S}$ such that, for any $\mathcal{Z}$,

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

□

7 Active Secure Two-round OT from Commutative Masking

We now show how to compile our 2-round OT protocol Π^1_{OT}, described in Section 6.1, to a 2-round maliciously UC-secure protocol using the generic transformations introduced by Döttling et al. [DGH+20].

7.1 Additional OT Security Notions

A 2-round OT protocol with public setup consists of four algorithms $(\mathsf{Setup}, \mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$ such that:

- $\mathsf{Setup}(1^\lambda)$ generates a public input pin.
- $\mathsf{OT}_1(\mathsf{pin}, \mathsf{c})$, where $\mathsf{c} \in \{0,1\}$ is the P_R choice bit, outputs $(\mathsf{st}, \mathsf{ot\_}P_R)$
- $\mathsf{OT}_2(\mathsf{pin}, \mathsf{ot\_}P_R, \mathsf{m}_0, \mathsf{m}_1)$, where $\mathsf{m}_0, \mathsf{m}_1$ are the sender's input messages, outputs $\mathsf{ot\_}P_S$
- $\mathsf{OT}_3(\mathsf{st}, \mathsf{ot\_}P_S)$ outputs m_{c}

First we need to recall some security notions [DGH+20] for the receiver P_R and the sender P_S. The first definition states that P_S should not learn anything about P_R's choice bit c.

Definition 7.1 (Receiver's indistinguishability security). *For every PPT adversary $\mathcal{A}$:*

$$|\Pr[\mathcal{A}(\mathsf{pin}, \mathsf{OT}_1(\mathsf{pin}, 0)) = 1] - \Pr[\mathcal{A}(\mathsf{pin}, \mathsf{OT}_1(\mathsf{pin}, 1)) = 1]| = \mathsf{negl}(\lambda),$$

where pin *is the public output of the setup phase.*

The next definition concerns the security of the sender; it states that P_R cannot compute both secret values y_0 and y_1 used by OT_2 to protect m_0 and m_1, but not necessarily in the same experiment.

Definition 7.2 (Sender's search security). *Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an adversary where $\mathcal{A}_2$ outputs a string y^*. Consider the following experiment $\mathsf{Exp}^{\mathsf{pin},\rho,w}_{\mathsf{sOT}}(\mathcal{A})$, indexed by a pin, random coins $\rho \in \{0,1\}^\lambda$ and a bit $w \in \{0,1\}$.*

1. *Run $(\mathsf{ot\_}P_R, \mathsf{st}) \leftarrow \mathcal{A}_1(1^\lambda, \mathsf{pin}; \rho)$.*
2. *Compute $(\mathsf{ot\_}P_S, \mathsf{y}_0, \mathsf{y}_1) \xleftarrow{\$} \mathsf{OT}_2(\mathsf{pin}, \mathsf{ot\_}P_R)$.*
3. *Run $\mathsf{y}^* \leftarrow \mathcal{A}_2(\mathsf{st}, \mathsf{ot\_}P_S, w)$ and output 1 iff $\mathsf{y}^* = \mathsf{y}_w$.*

We say that $\mathcal{A}$ breaks a scheme's Sender's search (sOT) security *if there exists a non-negligible function ϵ such that*

$$\Pr_{\mathsf{pin},\rho}[\Pr[\mathsf{Exp}^{\mathsf{pin},\rho,0}_{\mathsf{sOT}}(\mathcal{A}) = 1] > \epsilon \text{ and } \Pr[\mathsf{Exp}^{\mathsf{pin},\rho,1}_{\mathsf{sOT}}(\mathcal{A}) = 1] > \epsilon] > \epsilon,$$

where $\mathsf{pin} \xleftarrow{\$} \mathsf{Setup}$ and $\rho \xleftarrow{\$} \{0,1\}^\lambda$.

7.2 Two rounds OT with Active UC-Security

We provide an intermediary result which enables us to use the general compiler from [DGH+20] to get an actively secure 2-round OT protocol starting from Π^1_{OT}. First we introduce and discuss a new security assumption derived from the Parallel problem but more suited to active adversaries. Then we show that our protocol satisfies the security notions of Definitions 7.1 and 7.2. Finally, by applying the general transformations from sOT to UC OT described in [DGH+20], we obtain a fully UC-secure two-round OT protocol. We note that we are able to remove the random oracle from our protocol to achieve sOT security; therefore the resulting OT protocol requires only the CRS. We define our new computational problem as follows.

Definition 7.3 (ParallelDouble)**.** *Given $(i, j, r, r_{x_0}, r_{x_1}, r_y)$ with the promise that $i \neq j$ and that $r_{x_\mathsf{b}} = \mu_{x_\mathsf{b}}(r)$, $\mathsf{b} \in \{0,1\}$ and $r_y = \mu_y(r)$ for random $\mu_{x_\mathsf{b}} \xleftarrow{\$} M_i$ and $\mu_y \xleftarrow{\$} M_j$, and given a* one-time *access to an oracle $\mathcal{O}_y$ which, when given $r \in R$ returns $\mu_y(r)$, compute $z_0, z_1 \in X$ such that* both $\mu_{x_\mathsf{b}}(r_y) \in R_{z_\mathsf{b}}$.

The instantiation of this problem in the discrete logarithm case is, when given (g, g^a, g^b, g^c) and a one-time access to an exponentiation-by-c oracle, to return both g^{ac} and g^{bc}. For practical efficiency, it is also desirable that g^a and g^b remain constant across multiple instances of the ParallelDouble problem, with only g^c being randomly sampled in each instance. This version of the problem is similar to the one-more static CDH problem where an adversary has to successfully compute one more CDH challenge than it was able to ask from a helper oracle [BMV08].

Security of the Π^1_{OT} protocol. We then prove that protocol Π^1_{OT} achieves Receiver's indistinguishability and Sender's search security.

Proposition 7.1. *The protocol Π^1_{OT} in Figure 8 satisfies computational receiver's indistinguishability security and sender's* sOT *security under the assumption that $\mathcal{M}$ is IND-Mask-secure and that the* ParallelDouble$^{\mathcal{M}}$ *problem is hard.*

Proof. Receiver's indistinguishability follows from the IND-Mask-security assumption. By setting the public inputs r_0 and r_1 in Π^1_{OT} as they are computed in the IND-Mask experiment, the random mask μ is distributed in the same way as the mask β in OT_1. Therefore if an adversary breaks the receiver's indistinguishability for Π^1_{OT}, this can be reduced to a solution to the IND-Mask problem.

Sender's search security. To prove sOT security for Π^1_{OT} we assume the existence of an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and a non-negligible ϵ such that

$$\Pr_{\mathsf{pin},\rho}[\Pr[\mathsf{Exp}^{\mathsf{pin},\rho,0}_{\mathsf{sOT}}(\mathcal{A}) = 1] > \epsilon \text{ and } \Pr[\mathsf{Exp}^{\mathsf{pin},\rho,1}_{\mathsf{sOT}}(\mathcal{A}) = 1] > \epsilon] > \epsilon,$$

and we build a reduction $\mathcal{B}$ that is given a ParallelDouble challenge $(i, j, r, r_{x_0}, r_{x_1}, r_y)$ with access to an oracle $\mathcal{O}_y$ (Definition 7.3). Instead of running Setup to generate r_0 and r_1, $\mathcal{B}$ sets $r_0 \leftarrow r_{x_0}$ and $r_1 \leftarrow r_{x_1}$; also $\mathcal{B}$ samples $\rho \xleftarrow{\$} \{0,1\}^\lambda$. As this ensures that pin is distributed identically to the output of Setup, pin and ρ are good for $\mathcal{A}$ with probability at least ϵ.

After $\mathcal{B}$ runs $\mathcal{A}_1$, which outputs $(\mathsf{ot\_}P_R, \mathsf{st})$, it queries the oracle to obtain $\mathsf{ot\_}P_{S,0} \leftarrow \mathcal{O}_y(\mathsf{ot\_}P_R)$. It also computes $\mathsf{ot\_}P_{S,1} \leftarrow \mu(\mathsf{ot\_}P_{S,0})$ for a random $\mu \in M_k$ with $i \neq k \neq j$; it also computes μ^{-1}. Then, for $w \in \{0,1\}$, $\mathcal{B}$ runs $\mathsf{y}^*_w \leftarrow \mathcal{A}_2(\mathsf{st}, \mathsf{ot\_}P_{S,w}, w)$ and updates $\mathsf{y}^*_1 \leftarrow \mu^{-1}(\mathsf{y}^*_1)$. Finally $\mathcal{B}$ returns y^*_0 and the updated y^*_1 as the ParallelDouble answer.

Since $\Pr[\mathsf{Exp}^{\mathsf{pin},\rho,0}_{\mathsf{sOT}}(\mathcal{A}) = 1] > \epsilon$ and $\Pr[\mathsf{Exp}^{\mathsf{pin},\rho,1}_{\mathsf{sOT}}(\mathcal{A}) = 1] > \epsilon$, with probability ϵ^2, $\mathcal{A}_2$ is successful for both inputs $(\mathsf{st}, \mathsf{ot\_}P_{S,0}, 0)$ and $(\mathsf{st}, \mathsf{ot\_}P_{S,1}, 1)$ as the two messages are made independent by $\mathcal{B}$'s addition of μ. If this happens, then y^*_0 is exactly one of the answers, and the update of y^*_1 by $\mathcal{B}$ removes the extra mask μ and means that y^*_1 is then the other answer to the ParallelDouble problem. Hence $\mathcal{B}$ is successful with probability at least ϵ^3.

Theorem 7.1. *Under the assumption that $\mathcal{M}$ is IND-Mask-secure and that the* $\mathsf{ParallelDouble}^{\mathcal{M}}$ *problem is hard, there exists a 2-round UC-secure OT protocol constructed from Π^1_{OT}.*

Proof. This follows from the transformations and results of [DGH+20, Theorems 8, 9, 11, 12, 14, 19 and 21].

Corollary 7.1. *By instantiating the semi-commutative masking scheme, there exists an actively secure 2-round OT protocol based on supersingular isogenies.*

We remark here that the isogeny-based OT protocols proposed by Vitse [Vit19], while being semantically secure against malicious adversaries, require three rounds of communication; this implies that they cannot be transformed to achieve two-round OT with fully UC-security using the work of Döttling et al.

Acknowledgements

This work has been supported in part by ERC Advanced Grant ERC-2015-AdG-IMPaCT, by CyberSecurity Research Flanders with reference number VR20192203, by the Defense Advanced Research Projects Agency (DARPA) and Space and Naval Warfare Systems Center, Pacific (SSC Pacific) under contracts No. N66001-15-C-4070 and No. HR001120C0085, by the FWO under an Odysseus project GOH9718N and by EPSRC grant EP/S01361X/1

References

AAM18. Gora Adj, Omran Ahmadi, and Alfred Menezes. ON ISOGENY GRAPHS OF SUPERSINGULAR ELLIPTIC CURVES OVER FINITE FIELDS. Cryptology ePrint Archive, Report 2018/132, 2018. https://eprint.iacr.org/2018/132.

AJJS19. Reza Azarderakhsh, Amir Jalali, David Jao, and Vladimir Soukharev. Practical supersingular isogeny group key agreement. Cryptology ePrint Archive, Report 2019/330, 2019. https://eprint.iacr.org/2019/330.

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 statistically sender-private OT from LWE. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part II*, volume 11240 of *LNCS*, pages 370–390. Springer, Heidelberg, November 2018.

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.

BDGM19a. Pedro Branco, Jintai Ding, Manuel Goulão, and Paulo Mateus. A framework for universally composable oblivious transfer from one-round key-exchange. In Martin Albrecht, editor, *17th IMA International Conference on Cryptography and Coding*, volume 11929 of *LNCS*, pages 78–101. Springer, Heidelberg, December 2019.

BDGM19b. Pedro Branco, Jintai Ding, Manuel Goulão, and Paulo Mateus. A framework for universally composable oblivious transfer from one-round key-exchange. Cryptology ePrint Archive, Report 2019/726, 2019. https://eprint.iacr.org/2019/726. To appear at the 17th IMA International Conference on Cryptography and Coding.

BMV08. Emmanuel Bresson, Jean Monnerat, and Damien Vergnaud. Separation results on the "one-more" computational problems. In Tal Malkin, editor, *CT-RSA 2008*, volume 4964 of *LNCS*, pages 71–87. Springer, Heidelberg, April 2008.

BOB18. Paulo Barreto, Glaucio Oliveira, and Waldyr Benits. Supersingular isogeny oblivious transfer. Cryptology ePrint Archive, Report 2018/459, 2018. https://eprint.iacr.org/2018/459.

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 FOCS*, 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. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part III*, volume 11274 of *LNCS*, pages 395–427. Springer, Heidelberg, December 2018.

CLN16. Craig Costello, Patrick Longa, and Michael Naehrig. Efficient algorithms for supersingular isogeny Diffie-Hellman. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 572–601. Springer, Heidelberg, August 2016.

CO15a. Tung Chou and Claudio Orlandi. The simplest protocol for oblivious transfer. In Kristin E. Lauter and Francisco Rodríguez-Henríquez, editors, *LATINCRYPT 2015*, volume 9230 of *LNCS*, 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.

DGH+20. Nico Döttling, Sanjam Garg, Mohammad Hajiabadi, Daniel Masny, and Daniel Wichs. Two-round oblivious transfer from CDH or LPN. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 768–797. Springer, Heidelberg, May 2020.

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

FLOR18. A. Faz-Hernández, J. López, E. Ochoa-Jiménez, and F. Rodríguez-Henríquez. A faster software implementation of the supersingular isogeny diffie-hellman key exchange protocol. *IEEE Transactions on Computers*, 67(11):1622–1636, Nov 2018.

FTTY19. Atsushi Fujioka, Katsuyuki Takashima, Shintaro Terada, and Kazuki Yoneyama. Supersingular isogeny Diffie-Hellman authenticated key exchange. In Kwangsu Lee, editor, *ICISC 18*, volume 11396 of *LNCS*, pages 177–195. Springer, Heidelberg, November 2019.

FTY19. Atsushi Fujioka, Katsuyuki Takashima, and Kazuki Yoneyama. One-round authenticated group key exchange from isogenies. In Ron Steinfeld and Tsz Hon Yuen, editors, *ProvSec 2019*, volume 11821 of *LNCS*, pages 330–338. Springer, Heidelberg, October 2019.

Gal19. Steven Galbraith. Isogeny crypto. Blog post from ellipticnews, 2019. https://ellipticnews.wordpress.com/2019/11/09/isogeny-crypto/, last accessed Apr 15, 2020.

GO94. Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32, December 1994.

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, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 3–33. Springer, Heidelberg, December 2017.

GPST16. Steven D. Galbraith, Christophe Petit, Barak Shani, and Yan Bo Ti. On the security of supersingular isogeny cryptosystems. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 63–91. Springer, Heidelberg, December 2016.

GV18. Steven D. Galbraith and Frederik Vercauteren. Computational problems in supersingular elliptic curve isogenies. *Quantum Information Processing*, 17(10):265, Aug 2018.

JD11. David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In Bo-Yin Yang, editor, *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011*, pages 19–34. Springer, Heidelberg, November / December 2011.

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.

KMP+20. Péter Kutas, Chloe Martindale, Lorenz Panny, Christophe Petit, and Katherine E. Stange. Weak instances of sidh variants under improved torsion-point attacks. Cryptology ePrint Archive, Report 2020/633, 2020. https://eprint.iacr.org/2020/633.

KOS16. Marcel Keller, Emmanuela Orsini, and Peter Scholl. MASCOT: Faster malicious arithmetic secure computation with oblivious transfer. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016*, pages 830–842. ACM Press, October 2016.

LGD20. Yi-Fu Lai, Steven D. Galbraith, and Cyprien Delpech de Saint Guilhem. Compact, efficient and uc-secure isogeny-based oblivious transfer. Cryptology ePrint Archive, Report 2020/1012, 2020. https://eprint.iacr.org/2020/1012.

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, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 681–700. Springer, Heidelberg, August 2012.

Pet17. Christophe Petit. Faster algorithms for isogeny problems using torsion point images. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part II*, volume 10625 of *LNCS*, 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, *CRYPTO 2008*, volume 5157 of *LNCS*, 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.

SGP19. Rajeev Anand Sahu, Agnese Gini, and Ankan Pal. Supersingular isogeny-based designated verifier blind signature. Cryptology ePrint Archive, Report 2019/1498, 2019. https://eprint.iacr.org/2019/1498.

Sil86. Joseph H Silverman. *The arithmetic of elliptic curves*, volume 106 of *Graduate Texts in Mathematics*. Springer Science & Business Media, 1986.

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.

Vit19. Vanessa Vitse. Simple oblivious transfer protocols compatible with supersingular isogenies. In Johannes Buchmann, Abderrahmane Nitaj, and Tajje eddine Rachidi, editors, *AFRICACRYPT 19*, volume 11627 of *LNCS*, pages 56–78. Springer, Heidelberg, July 2019.

WRK17. Xiao Wang, Samuel Ranellucci, and Jonathan Katz. Global-scale secure multiparty computation. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 39–56. ACM Press, October / November 2017.

WZW03. Qian-Hong Wu, Jian-Hong Zhang, and Yu-Min Wang. Practical t-out-n oblivious transfer and its applications. In Sihan Qing, Dieter Gollmann, and Jianying Zhou, editors, *ICICS 03*, volume 2836 of *LNCS*, pages 226–237. Springer, Heidelberg, October 2003.