Efficient and Tight Oblivious Transfer from PKE with Tight Multi-User Security

Saikrishna Badrinarayanan Daniel Masny Pratyay Mukherjee

August 26, 2021

Abstract

We propose an efficient oblivious transfer in the random oracle model based on public key encryption with pseudorandom public keys. The construction is as efficient as the state of art though it has a significant advantage. It has a tight security reduction to the multi-user security of the underlying public key encryption. In previous constructions, the security reduction has a multiplicative loss that amounts in at least the amount of adversarial random oracle queries. When considering this loss for a secure parameter choice, the underlying public key encryption or elliptic curve would require a significantly higher security level which would decrease the overall efficiency.

Our OT construction can be instantiated from a wide range of assumptions such as DDH, LWE, or Codes as well as many public key encryption schemes such as the NIST PQC finalists. Since tight multi-user security is a very natural requirement which many public key encryption schemes suffice, many public key encryption schemes can be straightforwardly plugged in our construction without the need of reevaluating or adapting the parameter choice.

1 Introduction

An oblivious transfer (OT) [Rab81, EGL82] is an interactive protocol between two parties called a sender and a receiver. At the end of the protocol, the sender outputs two messages m_0, m_1 while the receiver outputs b, m_b for a choice bit b. Security requires that the sender does not learn b and the receiver does not learn m_{1-b}. OT is a fundamental building block in cryptography [Kil88], particularly in secure multi-party computation (MPC) [Yao82, Yao86, CvT95, IPS08, IKO$^+$11, BL18, GS18], which allows mutually distrusting parties to securely perform joint computations on their privately held data. MPC has a plethora of applications in practice, for example, in securely training machine learning models (e.g. [MR18]), private set intersection (e.g. [KKRT16, PRTY20]) etc. In fact, a significant body of practically efficient MPC protocols do rely primarily on the primitive of OT (e.g. [NNOB12, KOS16]), which makes *efficient secure OT* an important and very natural objective.

Within the last years, there has been significant progress in making OT more efficient. Chou and Orlandi [CO15] proposed a very efficient OT in the random oracle model [BR93, CGH98] based on the DDH assumption. It turned out, that it does not achieve UC security [GIR17, HL17], but only stand-alone security. Masny and Rindal [MR19] proposed an OT from public key encryption (PKE) with pseudorandom public keys that is as well very efficient but also UC secure and can be instantiated from a variety of assumptions such as LWE or code based assumptions. The construction makes it very easy to plug in PKE schemes such as the NIST PQC candidates [SAB$^+$20, DKR$^+$20, CDH$^+$20, ABC$^+$20] which is a significant advantage over more tailored construction of OT based on DDH [CSW20], LWE [PVW08, BD18, BDK$^+$20] or McEliece [DvMN08, DNM12].

McQuoid, Rosulek and Roy [MRR20, MRR21] extended this approach and suggested a concept called programmable once public functions (POPFs). This concept allows a more modular analysis of [MR19]. Intuitively, POPFs are based on random oracles (or ideal ciphers) and allow a receiver to freely chose one public key while a second public key determined by the random oracle. This approach unfortunately also

	UC	**Loss**	Model	$\mathrm{Com}(\mathcal{R})$	$\mathrm{Com}(\mathcal{S})$
[CO15]	✗	-	ROM	$\log \lvert\mathsf{G}\rvert$	$\log \lvert\mathsf{G}\rvert$
[MR19]	PKE A	$\mathbf{O(q)}$	ROM	$2\lvert\mathsf{pk}\rvert$	$2\lvert\mathsf{ct}\rvert$
[CSW20]	DDH	$\mathbf{O(q^2)}$	ROM,CRS	$2\log \lvert\mathsf{G}\rvert$	$\log \lvert\mathsf{G}\rvert$
[MRR20]	PKE B	$\mathbf{O(q)}$	ROM	$\lvert\mathsf{ct}\rvert + \lambda$	$\lvert\mathsf{pk}\rvert$
[MRR21]	PKE B	$\mathbf{O(q)}$	Ideal Cipher	$\lvert\mathsf{ct}\rvert + \lambda$	$\lvert\mathsf{pk}\rvert$
Ours	PKE A	$\mathbf{O(1)}$	ROM	$\lvert\mathsf{pk}\rvert + 2\lambda$	$2\lvert\mathsf{ct}\rvert$

Figure 1: We compare our construction with previous works. The depicted loss assumes tight multi-user security of the underlying PKE. We emphasize that the listed works realize different OT functionalities and therefore the comparison between the communication should be interpreted with caution. PKE A stands for PKE with pseudorandom public keys and PKE B stands for PKE with uniform ciphertexts. ICM stands for the Ideal Cipher model. q is the amount of adversarial hash evaluations.

has some drawbacks, namely the receiver could query the random oracle many times to find a POPF which defines public keys that might be easier to break. Further, when proving security against a malicious receiver, the simulation strategy using a known POPF causes a loss of at least q where q is the amount of adversarial oracle queries.

The first drawback can be easily resolved by using a PKE that is tightly secure in the multi-user setting. Bellare, Boldyreva and Micali [BBM00] showed that ElGamal is tightly secure even when multiple challenge ciphertexts are given to the adversary. There are numerous works that focus on tight multi-user security [Hås88, HJ12, Zav12, CKMS16, GKP18] and it is a rather well understood area. The tightness requirement does not seem to put significant restrictions on known PKEs. Tight multi-user security seems to be a very natural property that a PKE should typically have since usually the security of all users and not just of a single user needs to be considered. Non-tightness would demand an increase in the bit security level of the PKE when used across many users which would render the PKE significantly less efficient.

The second issue cannot be resolved that easily and requires a more in-depth analysis of the OT constructions. Both Masny and Rindal [MR19] as well as McQuoid, Rosulek and Roy [MRR20, MRR21] use guessing strategies that cause the loss. Therefore it seems hard to be resolved but it opens up the question whether a similar construction could achieve tight security. In this paper, we answer the following question:

Can we construct efficient OT that is tightly secure in the QROM from public-key encryption?

1.1 Our Contribution

We propose a new construction of OT in the random oracle model which can be proven tightly secure based on the multi-user security of the underlying PKE. We achieve this via a new construction of a programmable once public function (POPF) [MRR20] which can be used to adapt previous constructions of OT such as [MR19, MRR20, MRR21]. The proof requires a careful simulation of the random oracle and this is why we do not follow the more modular approach using an POPF.

We use a mild notion of multi-user security which is weaker than the notion proposed in previous literature such as [BBM00]. In our notion, we require that an adversary receives n user public keys and then decides for which he wants to see a challenge ciphertext. The notion of [BBM00] allows an adversary to see challenge ciphertext for all of the public keys. Nevertheless, there are many PKEs that even achieve the stronger notion of [BBM00] with a tight security proof under the DDH or LWE [Reg05] assumption. We recap the most basic PKEs and their tight reductions to DDH and LWE in Section 3. The results extend straightforwardly to code based schemes, the ring or module LWE [LPR10, BGV12, LS15] setting or elliptic curves.

For our OT, we require a second property that is the pseudorandomness of the public keys. This requirement is the same as in [MR19] we the exception that it holds tightly based on the underlying assumption even when n keys are seen. We recap this property as well in Section 3 for the PKEs of interest.

In Figure 1, we compare our result with previous works. Since the main difference of our construction to

[MR19] is how the random oracle is used, the efficiency of our OT is very similar to [MR19]. On one hand, we need to compute 3 additional hash evaluations. The hash evaluations are standard evaluations mapping onto $\{0,1\}^*$ and when using elliptic curves, not to curve points. On the other hand, we are actually, similar to [MRR20] able to reduce the communication complexity on the receivers side from $2|\mathsf{pk}|$ ([MRR20]) to $|\mathsf{pk}+2\lambda$. In particular when instantiating the OT with lattice or code based schemes [SAB+20, DKR+20, CDH+20, ABC+20] which have rather long keys, this is a significant reduction. Even when instantiating the OT with ElGamal encryption, we need to sample one random group element less which requires an exponentiation. In the elliptic curve setting, our construction is compatible with the performance optimizations of [MRR21] and would therefore be competitive with the currently fastest implementations of UC OT reported in [MRR21]. Further, our OT is based on a PKE with pseudorandom public keys which, unlike PKEs with uniform ciphertexts, can be efficiently instantiated with post-quantum PKEs, e.g. from codes or lattices. We could also use our techniques to construct an OT from a PKE with pseudorandom ciphertexts, though it is unclear whether the tightness would still hold and it might require stronger assumptions such as the interactive DDH assumption [MR19] or oracle assumptions [BCJ+19, MRR21].

As shown in Figure 1, our OT is currently the only OT among the most efficient OTs that is tightly secure. The main challenge is typically security against a malicious receiver. Previous works suffer at least a loss of $O(q)$ where q is the amount of adversarial hash evaluations. For a conservative parameter choice, previous works need to start with a significantly higher security level of the PKE or elliptic curve which negatively impacts efficiency and communication complexity.

1.2 Technical Overview

We follow an approach by Masny and Rindal [MR19]. They construct a two round OT in which the receiver starts by sending a message r_0, r_1 from this message the sender can derrive two public keys under which he encrypts the two OT strings. The public keys are $\mathsf{pk}_0 \coloneqq r_1 + \mathsf{H}(r_0)$ and $\mathsf{pk}_1 \coloneqq r_0 + \mathsf{H}(r_1)$. When following this approach, proving security against a malicious sender is typically easy since the random oracle can be programmed such that the simulator knows the secret keys for both public keys which can then be used to extract the malicious sender's string. The more challening part is to prove security against a malicious receiver $\mathcal{R}^*$. Given that $\mathcal{R}^*$ makes only two random oracle queries, r_0 and r_1, the simulator can observe the first query, let it be r_b. Then, when the second query is made, the simulator could pick a public key pk^* of its choice and program the oracle H such that $\mathsf{H}(r_{1-b}) \coloneqq \mathsf{pk}^* - r_b$ and thus $\mathsf{pk}_{1-b} = \mathsf{pk}^*$. If $\mathcal{R}^*$ learns information about the OT string s_{1-b}, he would then break the security of the PKE.

Unfortunately, when the malicious receiver makes many queries, it is not clear how to program $\mathsf{H}(r_{1-b})$ since any of the q previous queries $\tilde{r}_1, \ldots, \tilde{r}_q$ could be the r_b query. This would lead to the potential public keys $\mathsf{pk}_{1-b,1} \coloneqq \tilde{r}_1 + \mathsf{H}(r_{1-b}), \ldots, \mathsf{pk}_{1-b,q} \coloneqq \tilde{r}_q + \mathsf{H}(r_{1-b})$. We could guess $j \in [q]$ such that $r_b = \tilde{r}_j$ but this would cause a loss of q.

Before explaining our construction, we first take an intermediate step. The MR OT has similarities with a sequential OR proof [RST01, AOS02]. Instead we could follow the parallel OR proof paradigm [CDS94]. The public keys would be then derrived from a message r, c_0, c_1 and defined as $\mathsf{pk}_0 \coloneqq r + \mathsf{H}(c_0)$ and $\mathsf{pk}_1 \coloneqq r + \mathsf{H}(c_1)$. This construction has similarities with the McQuoid, Rosulek and Roy OT [MRR20]. As an additional constraint, we ask that $\hat{\mathsf{H}}(r) = c_0 + c_1$, where $\hat{\mathsf{H}}$ is a second random oracle. When proving security against $\mathcal{R}^*$, whenever $\mathcal{R}^*$ makes a query to $\hat{\mathsf{H}}$, the simulator samples a random $\hat{c}$ and programs $\mathsf{H}(\hat{c} + c_j) = \mathsf{pk}_j^* - r$ for any previous query c_j to H for a public key of its choice. Since $\hat{c}$ is uniform, it is very unlikely that H has been programmed on this input for a previous query. Now we could just rely on the multi-user security of the PKE rather than trying to guess which of the previous queries corresponds to r_b. Nevertheless, $\mathcal{R}^*$ could first query $\hat{\mathsf{H}}$ for r and then query H for c_0, c_1 such that $\hat{\mathsf{H}}(r) = c_0 + c_1$. This would cause an issue in the programming strategy which assumes that the adversary queries first c_0 or c_1 to H. Further, this strategy does not seem to help $\mathcal{R}^*$ since by using a guessing strategy, we could show that by the security of the PKE, $\mathcal{R}^*$ cannot learn any of the OT strings. However, it seems that we cannot show this via a tight reduction.

We resolve the issue via the following approach. We let the receiver send (r, c_0, c_1) and the public keys are

defined as $\mathsf{pk}_0 \coloneqq r + \hat{\mathsf{H}}(\hat{c}_0)$ and $\mathsf{pk}_1 \coloneqq r + \hat{\mathsf{H}}(\hat{c}_1)$, where $\hat{c}_0 \coloneqq c_1 + \mathsf{H}(r, c_0)$ and $\hat{c}_1 \coloneqq c_0 + \mathsf{H}(r, c_1)$. $\hat{c}_0$ and $\hat{c}_1$ could be seen as the r_0, r_1 values of the MR OT. But rather than using them directly, we apply an additional random oracle on them as "correlation breaker". A PKE scheme is typically not tightly secure in a setting where an adversary A can first suggest q shifts $\tilde{r}_1, \ldots, \tilde{r}_q$, then receives public key pk and finally tries to break IND-CPA security under public key $\mathsf{pk} - \tilde{r}_j$ where $j \in [q]$ is chosen by A. Though a correlation robust hash function $\hat{\mathsf{H}}$ [IKNP03] is tailored to such a setting and maps all inputs $\mathsf{pk} - r_1, \ldots, \mathsf{pk} - r_q$ to strings that do not collide as long as pk is uniform and independent of $\tilde{r}_1, \ldots \tilde{r}_q$. In our setting, we need something stronger than correlation robustness since we also need programmability such that we can program these disjunct strings to different public keys. Fortunately, a random oracle provides both properties such that for any choice of r, c_0, c_1 among the random oracle queries of $\mathcal{R}^*$, at least one of the public keys pk_0 and pk_0 will correspond to a programmed key chosen by the simulator. When q is the total amount of random oracle queries, there are at most q^2 choices for r, c_0, c_1 among the queries. This is due to the fact, that for any $b \in \{0, 1\}$, c_b is uniquely defined by r and c_{1-b}. Therefore, there will be at most q^2 choices of public keys pk_0, pk_1 and hence the multi-user security of PKE for q^2 user is sufficient to prove security against a malicious receiver.

For the proof, it would sufficient to just hash r_0, r_1 of the MR OT, though in the actual protocol, we need to allow the receiver to control one of the public keys. For this reason we introduce r to the protocol. Interestingly, our protocol could be seen as a combination of sequential and parallel OR proof techniques.

2 Preliminaries

Notation. For $n \in \mathbb{N}$, we use $[n]$ to denote the set $\{1, \ldots, n\}$. We use λ to denote the security parameter. And $x \leftarrow \mathcal{X}$, $x \leftarrow X$ to sample x from a distribution $\mathcal{X}$ or uniformly random from a set X.

Let Π be a protocol between two parties $\mathcal{S}$ and $\mathcal{R}$. For two (interactive) algorithms $\mathcal{S}', \mathcal{R}'$ that do not necessarily follow the protocol description of Π, we use $[\mathcal{S}', \mathcal{R}']_\Pi$ to denote the interaction between $\mathcal{S}'$ and $\mathcal{R}'$ in protocol Π, where $\mathcal{S}'$ takes the role of $\mathcal{S}$ and $\mathcal{R}'$ the role of $\mathcal{R}$. For an environment D, we use $\mathsf{D}([\mathcal{S}', \mathcal{R}']_\Pi)$ to denote an interaction of D with $\mathcal{S}', \mathcal{R}'$ who interact in Π. Here, we follow the simple UC framework of [CCL15].

For a cyclic group $\mathcal{G}$ of order $p \in \mathbb{N}$ with generator g, we use $[\![1]\!]$ to denote g and for $a, b \in \mathbb{N}$, $[\![a]\!] + b[\![1]\!] = [\![a + b]\!]$. For $a, b \in \mathbb{Z}_q^\eta$, we use $\langle a, b \rangle$ to denote the inner product between a and b. For an oracle $\mathcal{O}$ and an algorithm A, we use $\mathsf{A}^{\mathcal{O}}$ to denote A when A has query access to $\mathcal{O}$.

Cryptographic Assumptions. We recap the DDH and LWE problems below. Since we consider the UC setting, we need to consider non-uniform algorithms which receive an auxiliary input.

Definition 2.1 (Decisional Diffie-Hellman (DDH))**.** *A ppt algorithm* A *solves the decisional Diffie-Hellman (DDH) problem for a group* $\mathcal{G}$ *of order* $p \in \mathbb{N}$ *with generator* $[\![1]\!]$ *with probability* ϵ *if for any polynomial auxiliary input* z*,*

$$|\Pr[\mathsf{A}(z, [\![1]\!], [\![a]\!], [\![b]\!], [\![ab]\!]) = 1] - \Pr[\mathsf{A}(z, [\![1]\!], [\![a]\!], [\![b]\!], [\![c]\!]) = 1]| \geq \epsilon,$$

where $a, b, c \leftarrow \mathbb{Z}_p$*.*

Definition 2.2 (Learning with Errors (LWE))**.** *A ppt algorithm* A *solves the Learning with Errors (LWE) problem for parameters* $q, \eta \in \mathbb{N}$ *and noise distribution* $\mathcal{X}$ *with probability* ϵ *if for any polynomial auxiliary input* z

$$|\Pr[\mathsf{A}^{\mathcal{O}_{\mathsf{LWE}}}(z) = 1] - \Pr[\mathsf{A}^{\mathcal{O}_{\mathsf{U}}}(z) = 1]| \geq \epsilon,$$

where $\mathcal{O}_{\mathsf{LWE}}$ *is a oracle that outputs samples of the form* $a, \langle a, s \rangle + e$ *with* $a \leftarrow \mathbb{Z}_q^\eta$*,* $e \leftarrow \mathcal{X}$ *and each sample uses the same secret* $s \leftarrow \mathbb{Z}_q^\eta$*.* $\mathcal{O}_{\mathsf{U}}$ *is the oracle that outputs* a, u *with* $a \leftarrow \mathbb{Z}_q^\eta$*,* $u \leftarrow \mathbb{Z}_q$*.*

Public Key Encryption. We define public key encryption and its multi-user security below. We emphasize that we consider a setting with only a single challenge ciphertext which is a weaker security notion than the commonly used multi-user security setting in which an adversary receives a challenge ciphertext for each public key.

Definition 2.3 (Public Key Encryption)**.** *A public key encryption (PKE) is a triplet of algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *and a message space* M *with the following syntax.*

Gen**:** *Takes as input* 1^λ *and outputs a key pair* $(\mathsf{sk}, \mathsf{pk})$*.*

Enc**:** *Takes as input* pk *and a message* $\mathsf{m} \in \mathsf{M}$ *and outputs a ciphertext* ct*.*

Dec**:** *Takes as input* sk *and a ciphertext* ct *and outputs a message* m*.*

We require correctness and M-IND-CPA *security.*

Correctness: *For any* $\mathsf{m} \in \mathsf{M}$

$$\Pr[\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{m})) = \mathsf{m}] \geq 1 - \mathsf{negl},$$

where $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(1^\lambda)$*.*

n**-Multi-User IND-CPA** (M-IND-CPA)**:** *For any ppt adversay* $\mathsf{A} := (\mathsf{A}_1, \mathsf{A}_2)$ *and any polynomial auxiliary input* z

$$|\Pr[\mathsf{A}_2(\mathsf{st}, \mathsf{ct}_0^*) = 1] - \Pr[\mathsf{A}_2(\mathsf{st}, \mathsf{ct}_1^*) = 1]| \leq \mathsf{negl},$$

where for all $i \in [n]$*,* $(\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow \mathsf{Gen}(1^\lambda)$*,* $(\mathsf{st}, i^*, m_0, m_1) \leftarrow \mathsf{A}_1(z, \mathsf{pk}_1, \ldots, \mathsf{pk}_n)$ *and for all* $b \in \{0,1\}$ $\mathsf{ct}_b^* \leftarrow \mathsf{Enc}(\mathsf{pk}_{i^*}, m_b)$*.*

In addition to the multi-user IND-CPA security, we also need that public keys are indistinguishable from uniform in the multi-user setting.

Definition 2.4 (PKE with Pseudorandom Public Keys)**.** *For* $n \in \mathbb{N}$*, we call a PKE scheme* n*-multi-user public key indistinguishable* (M-IND-PK) *over group* G *if for any ppt* A *and polynomial auxiliary input* z

$$|\Pr[\mathsf{A}(z, \mathsf{pk}_1, \ldots, \mathsf{pk}_n) = 1] - \Pr[\mathsf{A}(z, u_1, \ldots, u_n) = 1]| \leq \mathsf{negl},$$

where for all $i \in [n]$*,* $(\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow \mathsf{Gen}(1^\lambda)$ *and* $u_i \leftarrow \mathsf{G}$*.*

Oblivious Transfer.

Definition 2.5 (Ideal Oblivious Transfer Functionality)**.** *An ideal OT functionality* $\mathcal{F}_{\mathsf{OT}}$ *interacts with two ppt parties* $\mathcal{S}$ *and* $\mathcal{R}$ *as follows.* $\mathcal{F}_{\mathsf{OT}}$ *takes* s_0, s_1 *from* $\mathcal{S}$*.* $\mathcal{F}_{\mathsf{OT}}$ *takes* b *from* $\mathcal{R}$ *and returns* s_b*.*

Definition 2.6 (Oblivious Transfer)**.** *We call a protocol* Π *between two ppt parties, a sender* $\mathcal{S}$ *and a receiver* $\mathcal{R}$*, oblivious transfer (OT) if at the end of the protocol they have established a correlation in which* $\mathcal{S}$ *holds strings* (s_0, s_1) *and* $\mathcal{R}$ *holds* (b, s_b)*. For security, we require two properties with respect to a functionality* $\mathcal{F}_{\mathsf{OT}}$*.*

Security Against a Malicious Sender: *For any ppt adversary* A*, there exists a ppt adversary* A*' such that for any ppt environment* D *and any polynomial size auxiliary input* z

$$|\Pr[\mathsf{D}(z, [\mathsf{A}, \mathcal{R}]_\Pi) = 1] - \Pr[\mathsf{D}(z, [\mathsf{A}', \mathcal{F}_{\mathsf{OT}}]_\Pi) = 1]| = \mathsf{negl},$$

where all algorithms receive input 1^λ*.* $\mathcal{R}$ *additionally receives input* b*.*

Security Against a Malicious Receiver: *For any ppt adversary* A*, there exists a ppt adversary* A*' such that for any ppt environment* D *and any polynomial size auxiliary input* z

$$|\Pr[\mathsf{D}(z, [\mathcal{S}, \mathsf{A}]_\Pi) = 1] - \Pr[\mathsf{D}(z, [\mathcal{F}_{\mathsf{OT}}, \mathsf{A}']_\Pi) = 1]| = \mathsf{negl},$$

where all algorithms receive input 1^λ*.*

3 Public Key Encryption in the Multi User Setting

We use this section to recap commonly known public key encryption schemes that are tightly secure in the multi-user setting. As a proof of concept, we consider ElGamal, Regev encryption and dual Regev e ncryption.

Definition 3.1 (ElGamal)**.** *The ElGamal PKE over group $\mathcal{G}$ with order $p \in \mathbb{N}$ and generator $[\![1]\!]$ with message space $\mathsf{M} := \mathcal{G}$ has the following syntax.*

$\mathsf{Gen}([\![1]\!]) \to (\mathsf{pk}, \mathsf{sk})$**:** *Sample $x \leftarrow \mathbb{Z}_p$ and output $\mathsf{pk} := [\![x]\!]$ and $\mathsf{sk} := x$.*

$\mathsf{Enc}([\![1]\!], \mathsf{pk}, \mathsf{m}) \to (\mathsf{ct}_1, \mathsf{ct}_2)$**:** *Sample $r \leftarrow \mathbb{Z}_p$ and output $\mathsf{ct}_1 := [\![r]\!]$, $\mathsf{ct}_2 := r\mathsf{pk} + \mathsf{m}$.*

$\mathsf{Dec}([\![1]\!], \mathsf{sk}, \mathsf{ct}) \to \mathsf{m}$**:** *Output $\mathsf{m} := \mathsf{ct}_2 - \mathsf{sk} \cdot \mathsf{ct}_1$.*

It is straightforward to see that ElGamal is perfectly correct. Let us recap that it is tightly secure in the multi-user setting. Due to the fact that the public keys are uniform over $\mathcal{G}$, ElGamal is perfectly n-M-IND-PK secure.

Lemma 3.1. *Let $\mathcal{G}$ be of prime order and DDH be ϵ hard over $\mathcal{G}$ and n polynomial, then ElGamal over $\mathcal{G}$ is 2ϵ n-M-IND-CPA secure.*

Proof. The proof follows straightforwardly from the random selfreducibility of the DDH assumption. The reduction for parameter $d \in \{0,1\}$ receives a DDH challenge $[\![a]\!], [\![b]\!], [\![c]\!]$ and samples for all $i \in [n]$ $r_i \leftarrow \mathbb{Z}_p$. It forwards z and $\mathsf{pk}_1 := r_1[\![a]\!], \ldots, \mathsf{pk}_n := r_n[\![a]\!]$ to A that tries to break ElGamal. When A send $i^*, \mathsf{m}_0, \mathsf{m}_1$, the reduction sends $\mathsf{ct} := ([\![b]\!], r_i[\![c]\!] \cdot \mathsf{m}_d)$. The reduction outputs the output of A.

When $[\![c]\!] = [\![ab]\!]$, ct is an encryption of m_d, i.e. $\mathsf{ct} := \mathsf{ct}_d$, while when c is uniform, ct encrypts a uniform message, i.e. $\mathsf{ct} := \mathsf{ct}_\mathsf{U}$. If A distinguishes ct_d from ct_U with probability ϵ', the reduction solves DDH with probability ϵ'. Assuming that DDH is ϵ hard, A cannot distinguish ct_d from ct_U with $\epsilon' > \epsilon$ for any $d \in \{0,1\}$ and it cannot distinguish ct_0 from ct_1 with $\epsilon' > 2\epsilon$. □

Definition 3.2 (Regev Encryption [Reg05])**.** *Regev encryption with the parameters $q, \eta, m \in \mathbb{N}$ with $m \geq \eta \log q$ and message space $\{0,1\}^m$ has the following syntax.*

$\mathsf{Gen}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$**:** *Sample $s \leftarrow \mathbb{Z}_q^\eta$, $A \leftarrow \mathbb{Z}_q^{m \times \eta}$, $e \leftarrow \mathcal{X}^m$ and output $\mathsf{pk} := (A, As + e)$ and $\mathsf{sk} := s$.*

$\mathsf{Enc}(\mathsf{pk}, \mathsf{m}) \to (\mathsf{ct}_1, \mathsf{ct}_2)$**:** *Sample $R \leftarrow \mathbb{Z}_q^{m \times m}$ and output $\mathsf{ct}_1 := R\mathsf{pk}_1$, $\mathsf{ct}_2 := R\mathsf{pk}_2 + \mathsf{m}\lfloor \frac{q}{2} \rceil$.*

$\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \to \mathsf{m}$**:** *Compute $\hat{\mathsf{m}} := \mathsf{ct}_2 - \mathsf{ct}_1 \cdot \mathsf{sk}$ and output $\mathsf{m} := ||\lfloor \frac{2}{q} \hat{\mathsf{m}} \rceil||$.*

For a proper choice of q, m and $\mathcal{X}$, Regev encryption will be correct.

Lemma 3.2. *Let LWE be ϵ hard and n polynomial, then Regev encryption is 2ϵ n-M-IND-CPA and ϵ n-M-IND-PK secure.*

Proof. We first show M-IND-CPA security. The reduction for parameter $d \in \{0,1\}$ receives access to an oracle $\mathcal{O}$ that it uses to generate A_i, b_i for all $i \in [n]$. It sets $\mathsf{pk}_i := (A_i, b_i + A_i s_i)$ for $s_i \leftarrow \mathbb{Z}_q^\eta$ and forwards them to A. After A sends (i^*, m_0, m_1), the reduction samples $R \leftarrow \mathbb{Z}_q^{m \times m}$ and sends $\mathsf{ct} := (RA_i, R(b_i + A_i s_i) + \mathsf{m}\lfloor \frac{q}{2} \rceil)$. The reduction outputs the output of A.

When $\mathcal{O} = \mathcal{O}_{\mathsf{LWE}}$, ct is an encryption of m_d, i.e. $\mathsf{ct} := \mathsf{ct}_d$, while when $\mathcal{O} = \mathcal{O}_\mathsf{U}$, ct is by the leftover hash lemma uniform, i.e. $\mathsf{ct} := \mathsf{ct}_\mathsf{U}$. If A distinguishes ct_d from ct_U with probability ϵ', the reduction solves LWE with probability ϵ'. Assuming that LWE is ϵ hard, A cannot distinguish ct_d from ct_U with $\epsilon' > \epsilon$ for any $d \in \{0,1\}$ and it cannot distinguish ct_0 from ct_1 with $\epsilon' > 2\epsilon$.

Let us now consider the M-IND-PK security. The reduction defines pk_i as previously. When $\mathcal{O} = \mathcal{O}_{\mathsf{LWE}}$, then pk_i is a proper public key and when $\mathcal{O} = \mathcal{O}_\mathsf{U}$, then the public key is uniform. If A can distinguish them, it solves LWE. □

Definition 3.3 (Dual Regev Encryption [GPV08])**.** *Dual Regev encryption with the parameters $q, \eta, m \in \mathbb{N}$ with $m \geq \eta \log q$ and message space $\{0,1\}^m$ has the following syntax.*

$\mathsf{Gen}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$**:** *Sample $R \leftarrow \mathbb{Z}_q^{m \times m}$, $A \leftarrow \mathbb{Z}_q^{m \times \eta}$ and output $\mathsf{pk} \coloneqq (A, RA)$ and $\mathsf{sk} \coloneqq R$.*

$\mathsf{Enc}(\mathsf{pk}, \mathsf{m}) \to (\mathsf{ct}_1, \mathsf{ct}_2)$**:** *Sample $s \leftarrow \mathbb{Z}_q^n$, $e_1, e_2 \leftarrow \mathcal{X}^m$, $R' \leftarrow \mathbb{Z}_q^{m \times m}$ and outputs $\mathsf{ct}_1 \coloneqq \mathsf{pk}_1 \cdot s + e_1$, $\mathsf{ct}_2 \coloneqq \mathsf{pk}_2 \cdot s + R' e_2 + \mathsf{m} \lfloor \frac{q}{2} \rceil$.*

$\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \to \mathsf{m}$**:** *Compute $\hat{\mathsf{m}} \coloneqq \mathsf{ct}_2 - \mathsf{sk} \cdot \mathsf{ct}_1$ and output $\mathsf{m} \coloneqq \lfloor \lfloor \frac{2}{q} \hat{\mathsf{m}} \rceil \rceil$.*

Correctness follows in the same way as in Regev encryption. By the leftover hash lemma, the public key is statistically indistinguishable from uniform and therefore dual Regev encryption is M-IND-PK secure.

Lemma 3.3. *Let LWE be ϵ hard and n polynomial, then dual Regev encryption is 2ϵ n-M-IND-CPA secure.*

Proof. The reduction for parameter $d \in \{0,1\}$ receives access to an oracle $\mathcal{O}$ that it uses to generate A_i, b_i for all $i \in [n]$. It sets $\mathsf{pk}_i \coloneqq (A_i, R_i A_i)$ for $R_i \leftarrow \mathbb{Z}_q^{m \times m}$ and forwards them to A. After A sends (i^*, m_0, m_1), the reduction sends $\mathsf{ct} \coloneqq (b_i, R_i b_i + \mathsf{m} \lfloor \frac{q}{2} \rceil)$. The reduction outputs the output of A.

When $\mathcal{O} = \mathcal{O}_{\mathsf{LWE}}$, ct is an encryption of m_d, i.e. $\mathsf{ct} \coloneqq \mathsf{ct}_d$, while when $\mathcal{O} = \mathcal{O}_{\mathsf{U}}$, ct is by the leftover hash lemma (with leakage Re_2) uniform, i.e. $\mathsf{ct} \coloneqq \mathsf{ct}_{\mathsf{U}}$. If A distinguishes ct_d from ct_{U} with probability ϵ', the reduction solves LWE with probability ϵ'. Assuming that LWE is ϵ hard, A cannot distinguish ct_d from ct_{U} with $\epsilon' > \epsilon$ for any $d \in \{0,1\}$ and it cannot distinguish ct_0 from ct_1 with $\epsilon' > 2\epsilon$. □

4 Oblivious Transfer from PKE

Theorem 4.1. *Let PKE be a* M-IND-CPA *and* M-IND-PK *secure and correct. Then Protocol 2 is a UC secure OT in the ROM.*

Proof. Given the correctness of PKE, an honest sender and receiver will establish correlation $(s_0, s_1), (b, s_b)$ with overwhelming probability.

We now focus on security against a malicious sender.

Claim 4.1. *Let PKE be ϵ_u* 1-M-IND-PK *secure. Then, for any ppt adversary* A*, there exists a ppt adversary* A*' such that for any ppt environment* D *and any polynomial size auxiliary input z*

$$|\Pr[\mathsf{D}(z, [\mathsf{A}, \mathcal{R}]_\Pi) = 1] - \Pr[\mathsf{D}(z, [\mathsf{A}', \mathcal{F}_{\mathsf{OT}}]_\Pi) = 1]| \leq \epsilon_u,$$

where all algorithms receive input 1^λ. $\mathcal{R}$ additionally receives input b.

Proof. We construct a receiver $\mathcal{R}'$ follows the description of $\mathcal{R}$ by sampling $(\mathsf{pk}_b, \mathsf{sk}_b) \leftarrow \mathsf{Gen}(1^\lambda)$, $\hat{c}_b \leftarrow \{0,1\}^\lambda$, $c_b \leftarrow \{0,1\}^\lambda$, computing $r \coloneqq \mathsf{pk}_b - \hat{\mathsf{H}}_b(\hat{c}_b)$, $c_{1-b} \coloneqq \hat{c}_b \oplus H(r, c_b)$. Unlike $\mathcal{R}$, $\mathcal{R}'$ computes $\hat{c}_{1-b} \coloneqq c_b \oplus \mathsf{H}_{1-b}(r, c_{1-b})$, samples $(\mathsf{pk}_{1-b}, \mathsf{sk}_{1-b}) \leftarrow \mathsf{Gen}(1^\lambda)$ and programs $\hat{\mathsf{H}}_{1-b}(\hat{c}_{1-b}) \coloneqq \mathsf{pk}_{1-b} - r$. Otherwise, $\mathcal{R}'$ follows the description of $\mathcal{R}$.

Notice that in case of $\mathcal{R}$, $r + \hat{\mathsf{H}}_{1-b}(\hat{c}_{1-b})$ is uniform while in case of $\mathcal{R}'$, it has the distribution of a public key generated by Gen. If D can distinguish $[\mathsf{A}, \mathcal{R}']$ from $[\mathsf{A}, \mathcal{R}]$, then D can be used to break the 1-M-IND-PK security of PKE with probability ϵ_u as follows. The reduction receives a 1-M-IND-PK challenge pk and sets $\mathsf{pk}_{1-b} \coloneqq \mathsf{pk}$. When pk is uniform, it simulates $\mathcal{R}$ and otherwise $\mathcal{R}'$. Therefore,

$$|\Pr[\mathsf{D}(z, [\mathsf{A}, \mathcal{R}]_\Pi) = 1] - \Pr[\mathsf{D}(z, [\mathsf{A}, \mathcal{R}']_\Pi) = 1]| \leq \epsilon_u.$$

Based on $\mathcal{R}'$, we can construct an adversary A' which interacts with A, relays all interaction between A and D and needs to submit s_0 and s_1 to $\mathcal{F}_{\mathsf{OT}}$. A' follows the process of $\mathcal{R}'$ when constructing r, c_0, c_1 that defines pk_0 and pk_1. As $\mathcal{R}'$, A' knows both, sk_0 and sk_1 which A' uses to decrypt ct_0 and ct_1 to obtain s_0 and s_1. Since, A' follows the description of $\mathcal{R}'$, it leads to the same interaction between A and D. Therefore

$$\Pr[\mathsf{D}(z, [\mathsf{A}, \mathcal{R}']_\Pi) = 1] = \Pr[\mathsf{D}(z, [\mathsf{A}', \mathcal{F}_{\mathsf{OT}}]_\Pi) = 1],$$

which concludes the proof of the claim. □

Protocol 2

Primitives:

- PKE scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with pseudorandom public keys in G.
- Random oracles
  - $\mathsf{H}_0, \mathsf{H}_1 : \mathsf{G} \times \{0,1\}^\lambda \to \{0,1\}^\lambda$.
  - $\hat{\mathsf{H}}_0, \hat{\mathsf{H}}_1 : \{0,1\}^\lambda \to \mathsf{G}$.

Common input: 1^λ.

Sender $\mathcal{S}$ input: s_0, s_1.

Receiver $\mathcal{R}$ input: $b \in \{0,1\}$.

1. $\mathcal{R}$ samples $(\mathsf{pk}_b, \mathsf{sk}_b) \leftarrow \mathsf{Gen}(1^\lambda), \hat{c}_b \leftarrow \{0,1\}^\lambda, c_b \leftarrow \{0,1\}^\lambda$, computes
   - $r \coloneqq \mathsf{pk}_b - \hat{\mathsf{H}}_b(\hat{c}_b)$
   - $c_{1-b} \coloneqq \hat{c}_b \oplus \mathsf{H}_b(r, c_b)$

   and sends (r, c_0, c_1).
2. $\mathcal{S}$ computes
   - $\hat{c}_0 \coloneqq c_1 \oplus \mathsf{H}_0(r, c_0), \hat{c}_1 \coloneqq c_0 \oplus \mathsf{H}_1(r, c_1)$,
   - $\mathsf{pk}_0 \coloneqq r + \hat{\mathsf{H}}_0(\hat{c}_0), \mathsf{pk}_1 \coloneqq r + \hat{\mathsf{H}}_1(\hat{c}_1)$,
   - $\mathsf{ct}_0 \coloneqq \mathsf{Enc}(\mathsf{pk}_0, s_0), \mathsf{ct}_1 \coloneqq \mathsf{Enc}(\mathsf{pk}_1, s_1)$,

   and sends $(\mathsf{ct}_0, \mathsf{ct}_1)$.
3. $\mathcal{R}$ computes $s_b \coloneqq \mathsf{Dec}(\mathsf{sk}_b, \mathsf{ct}_b)$.

Figure 2: Oblivious Transfer in the Random Oracle Model. $(+, -)$ are used to denote the operations in G. $\oplus$ is the xor operation over $\{0,1\}^*$.

We conclude the theorem with the following claim that establishes security against a malicious receiver.

Claim 4.2. *Let PKE be ϵ_u q^2-M-IND-PK and ϵ_t q^2-M-IND-CPA secure. Then, for any ppt adversary A making at most q random oracle queries to H_0, H_1, $\hat{\mathsf{H}}_0$ and $\hat{\mathsf{H}}_1$ combined, there exists a ppt adversary A' such that for any ppt environment D and any polynomial size auxiliary input z*

$$|\Pr[\mathsf{D}(z, [\mathcal{S}, \mathsf{A}]_\Pi) = 1] - \Pr[\mathsf{D}(z, [\mathcal{F}_{\mathsf{OT}}, \mathsf{A}']_\Pi) = 1]| \leq \epsilon_u + \epsilon_t + \frac{q^2}{2^\lambda},$$

where all algorithms receive input 1^λ.

Proof. For simplicity, we assume that when A sends r, c_0, c_1, it has queried the random oracles for $\mathsf{H}_0(r, c_0)$, $\mathsf{H}_1(r, c_1)$, $\hat{\mathsf{H}}_0(\hat{c}_0)$ and $\hat{\mathsf{H}}_1(\hat{c}_1)$. We can assume this without loss of generality by making at most 4 additional queries and setting the amount of queries to $\hat{q} = q + 4$. Since this is not significant for our overall bound, we identify $\hat{q}$ with q in the following. We also assume without loss of generality that A queries an oracle only once per input.

We define three intermediate algorithms $\mathcal{S}_1$, $\mathcal{S}_2$, $\mathcal{S}_3$ playing the role of sender $\mathcal{S}$. $\mathcal{S}_1$ is identical to $\mathcal{S}$ except that it simulates random oracles $\mathsf{H}_0, \mathsf{H}_1$ as follows. For all $i \in [q]$ and $j \in [q]$, it samples $\mathsf{pk}_{i,j} \leftarrow \mathsf{G}$.

Whenever A makes a query r_i, c_i to H_d for $i \in [q]$ and $d \in \{0,1\}$, $\mathcal{S}_1$ samples $\mathsf{H}_d(r_i, c_{i,d}) \leftarrow \{0,1\}^\lambda$ and does the following for any $j \in [q]$ with $j < i$ and the jth query is a query $r_j, c_{j,1-d}$ to H_{1-d} with $r_j = r_i$.

1. Compute $\hat{c}_{i,j,d} \coloneqq c_{j,1-d} \oplus \mathsf{H}_d(r_i, c_{i,d})$.
2. If $\hat{\mathsf{H}}_d(\hat{c}_{i,j,d})$ is defined (through programming or a query), abort. Otherwise, program $\hat{\mathsf{H}}_d(\hat{c}_{i,j,d}) \coloneqq \mathsf{pk}_{i,j} - r_i$.

Afterwards, $\mathcal{S}_1$ answers the query with $\mathsf{H}_d(r_i, c_{i,d})$.

When A sends r, c_0, c_1, $\mathcal{S}_1$ computes pk_0, pk_1 in the same way as $\mathcal{S}$. $\mathcal{S}_1$ defines b^* such that $\mathsf{pk}_{1-b^*} = \mathsf{pk}_{i,j}$ for a $i \in [q]$ and $j \in [q]$. If no such b^*, i, j exists, $\mathcal{S}_1$ aborts. Otherwise, it conludes the protocol according to the description of $\mathcal{S}$.

Let us now consider whether an environment D can distinguish $[\mathsf{A}, \mathcal{S}]$ from $[\mathsf{A}, \mathcal{S}_1]$. Since $\mathsf{pk}_{i,j}$ are uniform in G, the output distribution of $\hat{\mathsf{H}}_d$, in particular for every point $\hat{\mathsf{H}}_d(\hat{c}_{i,j,d}) \coloneqq \mathsf{pk}_{i,j} - r_i$ is uniform over G, in both settings. Other than that, $\mathcal{S}_1$ differs from $\mathcal{S}$ by two abort conditions - one during queries to H_d and one after seeing (r, c_0, c_1). Let us assume that $\mathcal{S}_1$ aborts during a query to H_d. This implies that either A has queried $\hat{\mathsf{H}}_d$ for $\hat{c}_{i,j,d} = c_{j,1-d} \oplus \mathsf{H}_d(r_i, c_{i,d})$ for an $j \in [q]$ or there exists a $j \in [q]$ and a $j' \in [q] \setminus \{j\}$ with $c_{j,1-d} \oplus \mathsf{H}_d(r_i, c_{i,d}) = c_{j',1-d} \oplus \mathsf{H}_d(r_i, c_{i,d})$. In the former case, A would predict $\mathsf{H}_d(r_i, c_{i,d}) = c_{j,1-d} \oplus \hat{c}_{i,j,d}$ which happens for each query with probability at most $\frac{q}{2^\lambda}$. In the latter case, $c_{j,1-d} = c_{j',1-d}$ and thus A would make the same query twice which does not happen.

The second abort condition never triggers for the following reason. Since A sends r, c_0, c_1, he will query r, c_0 to H_0 and r, c_1 to H_1. Let $b^* \in \{0,1\}$ such that A makes query r, c_{b^*} before r, c_{1-b^*}. When A makes query c_{1-b^*}, c_{b^*} will therefore be defined and $\mathcal{S}_1$ will program $\hat{\mathsf{H}}_{1-b^*}(c_{b^*} \oplus \mathsf{H}_d(r, c_{1-b^*})) = \mathsf{pk}_{i,j} - r$ for some $i, j \in [q]$. By the definition of pk_{1-b^*}, $\mathsf{pk}_{1-b^*} = \mathsf{pk}_{i,j}$. Thus, we obtain the bound

$$|\Pr[\mathsf{D}(z, [\mathcal{S}, \mathsf{A}]_\Pi) = 1] - \Pr[\mathsf{D}(z, [\mathcal{S}_1, \mathsf{A}]_\Pi) = 1]| \leq \frac{q^2}{2^\lambda}.$$

$\mathcal{S}_2$ is identical to $\mathcal{S}_1$ except that it samples $(\mathsf{pk}_{i,j}, \mathsf{sk}_{i,j}) \leftarrow \mathsf{Gen}(1^\lambda)$ for any $i, j \in [q]$. If there is an environment D that can distinguish $[\mathsf{A}, \mathcal{S}_2]$ from $[\mathsf{A}, \mathcal{S}_1]$, then we can break the q^2-M-IND-PK security or PKE as follows. The reduction receives q^2 challenge public keys $\widehat{\mathsf{pk}}_{i,j}$ for $i, j \in [q]$. Instead of sampling $\mathsf{pk}_{i,j}$, it sets $\mathsf{pk}_{i,j} \coloneqq \widehat{\mathsf{pk}}_{i,j}$.

When the challenge public keys are uniform, the reduction simulates $\mathcal{S}_1$ and otherwise (when the challenge public keys are distributed according to Gen) $\mathcal{S}_2$. Therefore,

$$|\Pr[\mathsf{D}(z, [\mathcal{S}_1, \mathsf{A}]_\Pi) = 1] - \Pr[\mathsf{D}(z, [\mathcal{S}_2, \mathsf{A}]_\Pi) = 1]| \leq \epsilon_u.$$

Our next intermediate sender $\mathcal{S}_3$ follows the description of $\mathcal{S}_2$ except that after receiving r, c_0, c_1 from A, it defines $\mathsf{ct}_{1-b^*} \coloneqq \mathsf{Enc}(\mathsf{pk}_{1-b^*}, 0)$. If there is an environment D that can distinguish $[\mathsf{A}, \mathcal{S}_2]$ from $[\mathsf{A}, \mathcal{S}_3]$, we can break the q^2-M-IND-CPA security of PKE as follows. The reduction receives q^2 challenge public keys $\widehat{\mathsf{pk}}_{i,j}$ for $i, j \in [q]$. As previously, it sets $\mathsf{pk}_{i,j} \coloneqq \widehat{pk}_{i,j}$. It then follows the description of $\mathcal{S}_2$ until it defines b^* and can compute $\mathsf{pk}_{1-b^*} = \mathsf{pk}_{i,j}$ for some $i, j \in [q]$. The reduction sends $(i^* = (i,j), \mathsf{m}_0 = s_{1-b^*}, \mathsf{m}_1 = 0)$ to the M-IND-CPA challenger and receives back ct^*. It then sets $\mathsf{ct}_{1-b^*} \coloneqq \mathsf{ct}^*$. When ct^* encrypts s_{1-b^*}, the reduction simulates $\mathcal{S}_2$ and otherwise $\mathcal{S}_3$. Therefore,

$$|\Pr[\mathsf{D}(z, [\mathcal{S}, \mathsf{A}]_\Pi) = 1] - \Pr[\mathsf{D}(z, [\mathcal{S}_1, \mathsf{A}]_\Pi) = 1]| \leq \epsilon_t.$$

Based on $\mathcal{S}_3$, we can define A' which interacts with A, relays all interaction between A and D and submits b^* to $\mathcal{F}_{\mathsf{OT}}$ and then receives s_{b^*} which is used to generate ct_{b^*}. Since A' follows the description of $\mathcal{S}_3$, it leads to the same interaction between A and D. Therefore, we can conclude the claim with

$$\Pr[\mathsf{D}(z, [\mathcal{S}_3, \mathsf{A}]_\Pi) = 1] = \Pr[\mathsf{D}(z, [\mathcal{F}_{\mathsf{OT}}, \mathsf{A}']_\Pi) = 1].$$

□

□

References

[ABC+20] Martin R. Albrecht, Daniel J. Bernstein, Tung Chou, Carlos Cid, Jan Gilcher, Tanja Lange, Varun Maram, Ingo von Maurich, Rafael Misoczki, Ruben Niederhagen, Kenneth G. Paterson, Edoardo Persichetti, Christiane Peters, Peter Schwabe, Nicolas Sendrier, Jakub Szefer, Cen Jung Tjhai, Martin Tomlinson, and Wen Wang. Classic McEliece. Technical report, National Institute of Standards and Technology, 2020. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions`.

[AOS02] Masayuki Abe, Miyako Ohkubo, and Koutarou Suzuki. 1-out-of-n signatures from a variety of keys. In Yuliang Zheng, editor, *ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 415–432. Springer, Heidelberg, December 2002.

[BBM00] Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 259–274. Springer, Heidelberg, May 2000.

[BCJ+19] Tatiana Bradley, Jan Camenisch, Stanislaw Jarecki, Anja Lehmann, Gregory Neven, and Jiayu Xu. Password-authenticated public-key encryption. In Robert H. Deng, Valérie Gauthier-Umaña, Martín Ochoa, and Moti Yung, editors, *ACNS 19*, volume 11464 of *LNCS*, pages 442–462. Springer, Heidelberg, June 2019.

[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.

[BDK+20] Niklas Büscher, Daniel Demmler, Nikolaos P. Karvelas, Stefan Katzenbeisser, Juliane Krämer, Deevashwer Rathee, Thomas Schneider, and Patrick Struck. Secure two-party computation in a quantum world. In Mauro Conti, Jianying Zhou, Emiliano Casalicchio, and Angelo Spognardi, editors, *ACNS 20, Part I*, volume 12146 of *LNCS*, pages 461–480. Springer, Heidelberg, October 2020.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In Shafi Goldwasser, editor, *ITCS 2012*, pages 309–325. ACM, January 2012.

[BL18] Fabrice Benhamouda and Huijia Lin. k-round multiparty computation from k-round oblivious transfer via garbled interactive circuits. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 500–532. Springer, Heidelberg, April / May 2018.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *ACM CCS 93*, pages 62–73. ACM Press, November 1993.

[CCL15] Ran Canetti, Asaf Cohen, and Yehuda Lindell. A simpler variant of universally composable security for standard multiparty computation. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 3–22. Springer, Heidelberg, August 2015.

[CDH+20] Cong Chen, Oussama Danba, Jeffrey Hoffstein, Andreas Hulsing, Joost Rijneveld, John M. Schanck, Peter Schwabe, William Whyte, Zhenfei Zhang, Tsunekazu Saito, Takashi Yamakawa, and Keita Xagawa. NTRU. Technical report, National Institute of Standards and Technology, 2020. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions`.

[CDS94] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Yvo Desmedt, editor, *CRYPTO'94*, volume 839 of *LNCS*, pages 174–187. Springer, Heidelberg, August 1994.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *30th ACM STOC*, pages 209–218. ACM Press, May 1998.

[CKMS16] Sanjit Chatterjee, Neal Koblitz, Alfred Menezes, and Palash Sarkar. Another look at tightness II: Practical issues in cryptography. Cryptology ePrint Archive, Report 2016/360, 2016. `https://eprint.iacr.org/2016/360`.

[CO15] 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.

[CSW20] Ran Canetti, Pratik Sarkar, and Xiao Wang. Efficient and round-optimal oblivious transfer and commitment with adaptive security. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part III*, volume 12493 of *LNCS*, pages 277–308. Springer, Heidelberg, December 2020.

[CvT95] Claude Crépeau, Jeroen van de Graaf, and Alain Tapp. Committed oblivious transfer and private multi-party computation. In Don Coppersmith, editor, *CRYPTO'95*, volume 963 of *LNCS*, pages 110–123. Springer, Heidelberg, August 1995.

[DKR+20] Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, Frederik Vercauteren, Jose Maria Bermudo Mera, Michiel Van Beirendonck, and Andrea Basso. SABER. Technical report, National Institute of Standards and Technology, 2020. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions`.

[DNM12] Bernardo Machado David, Anderson C. A. Nascimento, and Jörn Müller-Quade. Universally composable oblivious transfer from lossy encryption and the McEliece assumptions. In Adam Smith, editor, *ICITS 12*, volume 7412 of *LNCS*, pages 80–99. Springer, Heidelberg, August 2012.

[DvMN08] Rafael Dowsley, Jeroen van de Graaf, Jörn Müller-Quade, and Anderson C. A. Nascimento. Oblivious transfer based on the McEliece assumptions. In Reihaneh Safavi-Naini, editor, *ICITS 08*, volume 5155 of *LNCS*, pages 107–117. Springer, Heidelberg, August 2008.

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

[GIR17] Ziya Alper Genç, Vincenzo Iovino, and Alfredo Rial. "The simplest protocol for oblivious transfer" revisited. Cryptology ePrint Archive, Report 2017/370, 2017. `https://eprint.iacr.org/2017/370`.

[GKP18] Federico Giacon, Eike Kiltz, and Bertram Poettering. Hybrid encryption in a multi-user setting, revisited. In Michel Abdalla and Ricardo Dahab, editors, *PKC 2018, Part I*, volume 10769 of *LNCS*, pages 159–189. Springer, Heidelberg, March 2018.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008.

[GS18] Sanjam Garg and Akshayaram Srinivasan. Two-round multiparty secure computation from minimal assumptions. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 468–499. Springer, Heidelberg, April / May 2018.

[Hås88] Johan Håstad. Solving simultaneous modular equations of low degree. *SIAM J. Comput.*, 17(2):336–341, 1988.

[HJ12] Dennis Hofheinz and Tibor Jager. Tightly secure signatures and public-key encryption. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 590–607. Springer, Heidelberg, August 2012.

[HL17] Eduard Hauck and Julian Loss. Efficient and universally composable protocols for oblivious transfer from the CDH assumption. Cryptology ePrint Archive, Report 2017/1011, 2017. `https://eprint.iacr.org/2017/1011`.

[IKNP03] Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank. Extending oblivious transfers efficiently. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 145–161. Springer, Heidelberg, August 2003.

[IKO+11] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, Manoj Prabhakaran, and Amit Sahai. Efficient non-interactive secure computation. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 406–425. Springer, Heidelberg, May 2011.

[IPS08] Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. Founding cryptography on oblivious transfer - efficiently. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 572–591. Springer, Heidelberg, August 2008.

[Kil88] Joe Kilian. Founding cryptography on oblivious transfer. In *20th ACM STOC*, pages 20–31. ACM Press, May 1988.

[KKRT16] Vladimir Kolesnikov, Ranjit Kumaresan, Mike Rosulek, and Ni Trieu. Efficient batched oblivious PRF with applications to private set intersection. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016*, pages 818–829. ACM Press, October 2016.

[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.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 1–23. Springer, Heidelberg, May / June 2010.

[LS15] Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptogr.*, 75(3):565–599, 2015.

[MR18] Payman Mohassel and Peter Rindal. ABY^3: A mixed protocol framework for machine learning. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018*, pages 35–52. ACM Press, October 2018.

[MR19] Daniel Masny and Peter Rindal. Endemic oblivious transfer. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019*, pages 309–326. ACM Press, November 2019.

[MRR20] Ian McQuoid, Mike Rosulek, and Lawrence Roy. Minimal symmetric PAKE and 1-out-of-N OT from programmable-once public functions. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *ACM CCS 2020*, pages 425–442. ACM Press, November 2020.

[MRR21] Ian McQuoid, Mike Rosulek, and Lawrence Roy. Batching base oblivious transfers. *IACR Cryptol. ePrint Arch.*, 2021:682, 2021.

[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.

[PRTY20] Benny Pinkas, Mike Rosulek, Ni Trieu, and Avishay Yanai. PSI from PaXoS: Fast, malicious private set intersection. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 739–767. Springer, Heidelberg, May 2020.

[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, Harvard University, 1981.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.

[RST01] Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 552–565. Springer, Heidelberg, December 2001.

[SAB+20] Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, and Damien Stehlé. CRYSTALS-KYBER. Technical report, National Institute of Standards and Technology, 2020. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions`.

[Yao82] Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *23rd FOCS*, pages 160–164. IEEE Computer Society Press, November 1982.

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *27th FOCS*, pages 162–167. IEEE Computer Society Press, October 1986.

[Zav12] G.M. Zaverucha. Hybrid encryption in the multi-user setting. Cryptology ePrint Archive, Report 2012/159, 2012. `https://eprint.iacr.org/2012/159`.