Adaptively Secure Primitives in the Random Oracle Model

A THESIS
SUBMITTED FOR THE DEGREE OF
Master of Science (Engineering)
IN THE
Faculty of Engineering

BY

Pratik Sarkar

Computer Science and Automation
Indian Institute of Science
Bangalore – 560 012 (INDIA)

October, 2018

Declaration of Originality

I, **Pratik Sarkar**, with SR No. **04-04-00-10-21-15-1-12531** hereby declare that the material presented in the thesis titled

Adaptively Secure Primitives in the Random Oracle Model

represents original work carried out by me in the **Department of Computer Science and Automation** at **Indian Institute of Science** during the years **2015-2018**.
With my signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property. I have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I have understood that any false claim will result in severe disciplinary action.
- I have understood that the work may be screened for any form of academic misconduct.

Date: Student Signature

In my capacity as supervisor of the above-mentioned work, I certify that the above statements are true to the best of my knowledge, and I have carried out due diligence to ensure the originality of the report.

Advisor Name: Advisor Signature

DEDICATED TO

My Grandmother and Parents

who helped me achieve success and honor in life

Thank you for being there

Acknowledgements

It gives me immense pleasure to express a deep sense of gratitude to my research supervisor Dr. Arpita Patra, for accepting me into her lab and considering me as a part of her family. We have engaged ourselves in countless hours of discussions regarding different topics ranging from cryptography to societal problems and life in general. I would be forever grateful to her for teaching us the concepts of MPC through her course "Secure Computation" with utmost sincerity. It has always been a wonderful experience to work with such an energetic person like her who is so full of ideas.

I would like to extend my gratitude to my co-authors: Ajith Suresh, Divya Ravi, Megha Byali, Ashish Choudhury, Chaya Ganesh, Yashvanth Kondi and Gayathri Garimella. Due to them, I came to know about many other different topics in cryptography, apart from the ones present in this thesis. I would also like to thank my other labmates, Swati Singla, Nishat Koti, Arun Joseph, Harsh Choudhuri and Dheeraj Ram. It was a great time sharing the lab with all my labmates and I would like to specially thank Ajith and Divya for saving me, whenever I messed up in the lab. I would also like to extend my appreciation for Sruthi Sekar, Sayantan Mukherjee and Suvradip Chakraborty whom I consider as my extended labmates. I have troubled them all along with questions related to different aspects of cryptography.

Special regards to the organizing committee of TPMPC 2017 committee for supporting my travel to attend TPMPC'17. I am also grateful to Robert Bosch for funding my MTech project and allowing me to attend TPMPC'18. I would also like to thank Dr. Carmit Hazay for hosting me in Bar-Ilan University. It has been a great experience to work with her.

Finally, I would like to express my profound regards to my parents, Kaustuv Lahiri, Sohini Mukherjee and my friends in IISc for supporting me all throughout my Masters program. Thank you for all the kind words, unfailing support and indispensable advices, throughout my research.

Abstract

Adaptive security embodies one of the strongest notions of security that allows an adversary to corrupt parties at any point during protocol execution and gain access to its internal state. Since it models real-life situations such as "hacking", efficient adaptively-secure multiparty computation (MPC) protocols are desirable. Such protocols demand primitives such as zero knowledge (ZK), oblivious transfer (OT) and commitment schemes that are adaptively-secure as building blocks. Efficient realizations of these primitives have been found to be challenging, especially in the no erasure model. We make progress in this direction and provide efficient constructions that are Universally-Composable in the random oracle model.

The study of efficient ZK protocols for non-algebraic statements has seen rapid progress in recent times, relying on the techniques from secure computation. Our primary contribution in ZK lies in constructing efficient constant round ZK protocols from garbled circuits that are adaptively-secure, with communication linear in the size of the statement. We begin by showing that the practically efficient ZK protocol of Jawurek et al. (CCS 2013) is adaptively-secure when the underlying OT satisfies a mild adaptive security guarantee. We gain adaptive security with little to no overhead over the static case. A conditional verification technique is then used to obtain a three-round adaptively secure zero-knowledge argument in the non-programmable non-observable random oracle model.

We present the first *round optimal* framework for building adaptively-secure OT in the programmable random oracle (PRO) model, relying upon the framework of *Peikert* et al. (Crypto 2008). When instantiated with Decisional Diffie Hellman assumption, it incurs a minimal communication overhead of one κ bit string and computational overhead of 5 random oracle queries over its static counterpart, where κ is the security parameter. Additionally, we obtain a construction of adaptively-secure 1-out-of-N OT by extending the result of *Naor* et al. (Journal of Cryptology 2005) that transforms $\log \mathsf{N}$ copies of 1-out-of-2 OTs to one 1-out-of-N OT in the PRO model. We complete the picture of efficient OT constructions by presenting the first adaptively secure OT Extension, extending the protocol of Asharov et al. (Eurocrypt 2015) for the adaptive setting using PRO. Our OT extension enables us to obtain adaptive OTs at an amortized cost of 3 symmetric key operations and communication of 3κ bit strings.

We present an adaptively secure commitment scheme solely relying on observable random oracle (ORO). Our commitment scheme has a one-time offline setup phase, where a common reference string (crs) is generated between the parties using an ORO. In the online phase, the parties use the crs and ORO to generate commitments in a non-interactive fashion. Our construction incurs communication of 4κ bit strings and computation of 8 exponentiations and 4 random oracle queries for committing to an arbitrary length message. It finds applications in secure two-party computation (2PC) protocols that adopt offline-online paradigm, where the crs can be generated in the offline phase and the scheme can be used in the online phase.

Publications based on this Thesis

- Chaya Ganesh, Yashvanth Kondi, Arpita Patra and **Pratik Sarkar**. *Efficient Adaptively Secure Zero-Knowledge from Garbled Circuits*. In Proceedings of 21st Public Key Cryptography, Rio de Janeiro, Brazil, pages 499-529, 2018.
- Megha Byali, Arpita Patra, Divya Ravi and **Pratik Sarkar**. *Fast and Universally-Composable Oblivious Transfer and Commitment Scheme with Adaptive Security*. Under Submission.

Contents

Chapter 1

Introduction

Cryptography in its initial years have concentrated on enabling two parties to communicate a sensitive message with each other in a secure manner such that an eavesdropper who is tapping the communication channel does not obtain any information about the secret message. There has been extensive research in this domain over the years, using symmetric and public key encryption schemes, thereby addressing the problem of secure communication.

Modern-day cryptography aspires to achieve beyond secure communication. It aims to enable two or more parties to compute a function on their private inputs, without the involvement of any trusted third party. This is called Secure Multiparty Computation or popularly abbreviated as MPC. Secure MPC guarantees that an adversary corrupting one or more participating parties will not procure any additional information about the uncorrupted parties' private input, than what is already revealed from the output. Below, we provide a formal definition of MPC and discuss its various classifications based on the adversarial corruption. And then we give a brief overview of our results, which we elaborate in chapters 4, 3 and 5.

1.1 Introduction to Secure Multiparty Computation

An MPC protocol involves a set of n mutually distrustful parties $\{\mathsf{P}_1, \mathsf{P}_2, \ldots, \mathsf{P}_n\}$, where P_i possess a private input x_i for $i \in \{1, 2, \ldots, n\}$. The parties want to compute a publicly known function f on their private inputs and obtain the output $f(\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n)$. The protocol must satisfy two properties:

1. **Correctness:** The protocol should compute the correct output $f(\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n)$ and return it to the participating parties.

2. **Input Privacy:** The MPC protocol should preserve input privacy of each party from any subset of corrupted parties. This is captured by considering a central adversary A who corrupts a set

of parties. Security of the protocol ensures that given control over the corrupted parties, A does not obtain any extra information about the honest parties' input from the protocol.

Secure MPC has seen various applications in privacy preserving protocols, some of which have been listed below:

- **Private Set Intersection:** In private set intersection (PSI) two or more parties possess sets of private data and they want to compute the intersection of their sets without disclosing any other information. A secure PSI protocol ensures that a party only obtains the desired output, i.e. the set intersection, and no information is leaked about the private sets of other parties from the protocol. An example of PSI is when two hospitals want to share records of a patient who got treated in both hospitals.
- **Privacy Preserving Machine Learning:** In today's world different advertising companies possess different types of data about their customers. In order to improve their advertisements, the companies might want to train a machine learning algorithm on those joint data without revealing their private data to the other company. This can be performed by a secure multi-party computation protocol which trains the model on their private data in a privacy preserving fashion. It also allows parties
- **Secure Auctioning:** In an auctioning process, parties input their possess private bids and the highest bidder wins the auction. Only the winning bid is revealed without disclosing the other bids. A secure auctioning protocol performs the auction correctly, revealing the winning bid and hides the other bids.

1.2 Classification of MPC Protocols

The MPC literature can be classified based on the adversarial strategy and the model of corruption. Based on adversarial strategy, a party can continue to follow the protocol steps after it got corrupted. Such an adversary is called *semi-honest* adversary. Whereas, if the corrupt party arbitrarily deviates from the protocol steps and follow its own adversarial algorithm then such an adversary is called *malicious* adversary. The literature contains many efficient semi-honestly secure protocols [Yao82, Rab81, IKNP03, ALSZ13, KK13] whereas their maliciously secure counterparts [LP07, PVW08, KOS15, PSS17] incurs some overhead over the semi-honest protocols.

Based on model of corruption, the parties can be either statically corrupted or adaptively corrupted. *Static corruption* permits the adversary to corrupt the honest parties only at the outset of the protocol. Most widely known protocols of [Lin13, HKK$^+$14, MR17, ALSZ15], consider malicious security against static corruptions. Although static security is of interest, it is desirable to achieve security in the stronger model of adaptive security. *Adaptive corruption* allows the adversary to choose which parties to corrupt at the outset/during/after the protocol execution. It models real-life situations in a more comprehensive way. For instance, it captures the event of “hacking”, where a hacker can illegally capitalize on the system and corrupt any workstation while protocols are in execution. Adaptive security is further classified based on secure erasures of the memory. An adaptively-secure protocol assuming erasure allows secure erasure of a workstation’s internal memory once it is corrupted by a hacker. Canetti [Can01] argued that security relying on erasures often leads to problems and is impractical, especially for real-life systems. It requires an inherent trust assumption on the part of a workstation that it will erase its memory upon being corrupted. Hence, adaptive security without erasures (referred to simply as *adaptive-security* throughout the thesis) is preferable as it precisely models real world “hacking” attacks. However, the current literature of MPC dealing with adaptive adversaries is less explored compared to static security since it turns out to be considerably more challenging. In this paper, we explore the less traveled path of dealing with adaptive adversaries.

1.3 Brief Overview of Our Results

We focus on adaptively secure protocols by constructing efficient, adaptively-secure primitives. These primitives serve as important building blocks in the construction of practical adaptively-secure MPC protocols. Our focus centres around three such primitives in particular: Zero Knowledge (ZK), Oblivious Transfer (OT) and Commitment Schemes. We construct protocols for ZK, OT and commitment schemes which are proven secure in the Universal Composability (UC) security model of [Can01] (described later in Section. 2.4). Detailed discussion of our contributions has been deferred to chapters 3, 4 and 5, corresponding to zero knowledge, oblivious transfer and commitment scheme respectively.

1.4 Organization

The thesis begins by introducing the concept of MPC and its classification based on the model of corruption and the adversarial strategy followed by the corrupted parties. The next few chapters covers adaptively secure constructions for ZK, OT and commitment scheme. The thesis proceeds in a set of chapters, whose contents have been summarized below.

In chapter 2, we define the ZK, OT and commitment scheme functionality. We also describe the notations, cryptographic primitives and the random oracle functionality essential for our protocols. In the same chapter, we briefly recall the UC security model in the static and adaptive setting.

In chapter 3, we discuss the primitive of ZK and briefly elaborate on state-of-the-art literature. In regard of adaptive security, we highlight the different approaches employed in the literature and finally we provide our contribution alongside its security proof.

In chapter 4, we state different use cases of OT in MPC and motivate the need for an adaptively secure OT. Following this, we provide a brief survey of OT protocols in the adaptive setting. We also define a related primitive called OT extension and discuss its literature. Finally we present our adaptively secure OT and OT extension protocols and prove their security in the UC model.

In chapter 5, we motivate the primitive of commitment scheme by stating its usefulness in MPC. Following this, we elaborate on the commitment scheme literature and highlight the adaptively secure commitment schemes in a comprehensive manner. We provide our adaptively secure commitment scheme protocol alongside its proof of security in the UC model.

Finally, in chapter 6 we conclude the thesis with a summary of our results and a discussion of open questions, related to this work.

Chapter 2

Preliminaries

In this chapter, we formally define zero knowledge, oblivious transfer and extension, and commitment scheme functionalities. Next, we describe the notations, cryptographic primitives and the random oracle functionality that are required for our protocols. Lastly, we give an outline of Universal Composability security model.

Zero Knowledge. A Zero-knowledge (ZK) proof allows a prover to convince a verifier of the validity of a statement, without revealing any other information beyond that. Let R be an NP relation, and $\mathcal{L}$ be the associated language. $\mathcal{L} = \{z \mid \exists x : R(z, x) = 1\}$. A zero-knowledge proof for $\mathcal{L}$ lets the prover convince a verifier that $z \in \mathcal{L}$ for a common input z. A proof of knowledge captures not only the truth of a statement $z \in \mathcal{L}$, but also that the prover "possesses" a witness x to this fact. A proof of knowledge for a relation $R(\cdot, \cdot)$ is an interactive protocol where a prover P convinces a verifier V that P knows a x such that $R(z, x) = 1$, where z is a common input to P and V . The prover can always successfully convince the verifier if indeed P knows such a x. Conversely, if P can convince the verifier with high probability, then he "knows" such a x, that is, such a x can be efficiently computed given z and the code of P. When the soundness holds only for a probabilistically polynomial time (PPT) prover, it is called an *argument*. As in [JKO13], we define the ideal functionality for zero-knowledge $\mathcal{F}^{R}_{\mathsf{ZK}}$ in the framework of [Can01] in order to capture all the properties that we require, in Figure 2.1.

Figure 2.1: The ideal functionality $\mathcal{F}^{R}_{\mathsf{ZK}}$ for Zero-knowledge

$\mathcal{F}^{R}_{\mathsf{ZK}}$

On input $(\mathsf{prove}, \mathsf{sid}, z, x)$ from P and $(\mathsf{verify}, \mathsf{sid}, z')$ from V, output $(\mathsf{accept}, \mathsf{sid}, z)$ to V if $z = z'$ and $R(z, x) = 1$ else output $(\mathsf{reject}, \mathsf{sid}, z)$

Oblivious Transfer and Extension. Oblivious transfer (OT) is a protocol between a sender (S) and a receiver (R). In a 1-out-of-2 OT, the sender holds two inputs $a_0, a_1 \in \{0,1\}^n$ and the receiver holds a choice bit σ. At the end of the protocol, the receiver obtains a_σ. The sender learns nothing about the choice bit, and the receiver learns nothing about the sender's other input $a_{\overline{\sigma}}$, where $\overline{\sigma}$ denotes $1 - \sigma$. The ideal OT functionality $\mathcal{F}_{\mathsf{OT}}$ is recalled below in Figure 2.2. Similarly a 1-out-of-N OT can be defined as $\mathcal{F}_{\mathsf{N\text{-}OT}}$ functionality in Figure 2.3. For our protocols, we need an additional functionality $\mathcal{F}_{\mathsf{ROT}}$ called random OT functionality. $\mathcal{F}_{\mathsf{ROT}}$ is an OT functionality where R has an input whereas S does not have any input. The functionality returns random messages to S and one of the random messages to R based on his input. Formally stating, $\mathcal{F}_{\mathsf{ROT}}$ takes choice bit σ as input from receiver R, and generates two random pads (a_0, a_1) for S. It sends a_σ to R_{OT} and (a_0, a_1) to S_{OT}. We also define a primitive called Committing Oblivious Transfer (Fig. 2.5), borrowed from [JKO13]. The Committing OT functionality $\mathcal{F}_{\mathsf{COT}}$ has a similar structure that of $\mathcal{F}_{\mathsf{OT}}$. In addition, the $\mathcal{F}_{\mathsf{COT}}$ functionality reveals both messages to R upon receiving a signal $\mathsf{open\text{-}all}$ from S.

In the OT literature it is known [IR89] that OT cannot be implemented based on symmetric key operations, and it requires expensive public key operations. To circumvent this, Beaver came up with the concept of OT extension in [Bea96a]. An OT extension protocol generates large number of OTs using only cheap symmetric key operations, given only a smaller number of OTs. The amortized cost of each extended OT is just a few symmetric key operations, whereas the public key operations for the smaller number of OTs gets amortized over all the extended OTs. More formally speaking, an OT extension protocol generates $\mathsf{poly}(\kappa)$ OTs given only κ OTs and symmetric key operations, where κ is the security parameter.

Figure 2.2: The ideal functionality $\mathcal{F}_{\mathsf{OT}}$ for Oblivious Transfer

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

Choose: On input $(\mathsf{rec}, \mathsf{sid}, \sigma)$ from R where $\sigma \in \{0,1\}$; if no message of the form $(\mathsf{rec}, \mathsf{sid}, \sigma)$ has been recorded in the memory, store $(\mathsf{rec}, \mathsf{sid}, \sigma)$ and send $(\mathsf{rec}, \mathsf{sid})$ to S.

Transfer: On input $(\mathsf{sen}, \mathsf{sid}, (a_0, a_1))$ from S with $a_0, a_1 \in \{0,1\}^n$, if no message of the form $(\mathsf{sen}, \mathsf{sid}, (a_0, a_1))$ is recorded and a message of the form $(\mathsf{rec}, \mathsf{sid}, \sigma)$ is stored, send $(\mathsf{sent}, \mathsf{sid}, a_\sigma)$ to R and $(\mathsf{sent}, \mathsf{sid})$ to S.

Figure 2.3: The ideal functionality $\mathcal{F}_{\mathsf{N\text{-}OT}}$ for Oblivious Transfer

$\mathcal{F}_{\mathsf{N\text{-}OT}}$

Choose: On input $(\mathsf{rec}, \mathsf{sid}, \sigma)$ from R where $\sigma \in \{0,1\}^{\log \mathsf{N}}$; if no message of the form $(\mathsf{rec}, \mathsf{sid}, \sigma)$ is present in memory, store $(\mathsf{rec}, \mathsf{sid}, \sigma)$ and send $(\mathsf{rec}, \mathsf{sid})$ to S.

Transfer: On input $(\mathsf{sen}, \mathsf{sid}, \{a_j\}_{j=1}^{\mathsf{N}})$ from S with $a_j \in \{0,1\}^n$, if no message of the form $(\mathsf{sen}, \mathsf{sid}, \{a_j\}_{j=1}^{\mathsf{N}})$ is present in memory and a message of the form $(\mathsf{rec}, \mathsf{sid}, \sigma)$ is stored, send $(\mathsf{sent}, \mathsf{sid}, a_\sigma)$ to R and $(\mathsf{sent}, \mathsf{sid})$ to S.

Figure 2.4: The ideal functionality $\mathcal{F}_{\mathsf{ROT}}$ for Random Oblivious Transfer

$\mathcal{F}_{\mathsf{ROT}}$

Initiate: On input $(\mathsf{rec}, \mathsf{sid}, (n, \sigma))$ from R; if no message of the form $(\mathsf{sid}, (n, \sigma))$ is present in memory, store $(\mathsf{sid}, (n, \sigma))$. Send $(\mathsf{rec}, \mathsf{sid})$ to S.

Transfer: On input $(\mathsf{sen}, \mathsf{sid}, (\mathsf{transfer}, n))$ from S, if no message of the form $(\mathsf{sid}, (n, \sigma))$ is present in memory, then abort. Else sample $a_0, a_1 \leftarrow_R \{0,1\}^n$. Send $(\mathsf{sent}, \mathsf{sid}, a_\sigma)$ to R and $(\mathsf{sent}, \mathsf{sid}, (a_0, a_1))$ to S.

Corruption: If A corrupts S^* then receive (a_0, a_1) from A and continue execution as above using these values. If A corrupts R^* the usual execution continues.

Figure 2.5: The ideal functionality $\mathcal{F}_{\mathsf{COT}}$ for Committing Oblivious Transfer

$\mathcal{F}_{\mathsf{COT}}$

Choose: On input $(\mathsf{rec}, \mathsf{sid}, \sigma)$ from R where $\sigma \in \{0,1\}$; if no message of the form $(\mathsf{rec}, \mathsf{sid}, \sigma)$ has been recorded in the memory, store $(\mathsf{rec}, \mathsf{sid}, \sigma)$ and send $(\mathsf{rec}, \mathsf{sid})$ to S.

Transfer: On input $(\mathsf{sen}, \mathsf{sid}, (a_0, a_1))$ from S with $a_0, a_1 \in \{0,1\}^n$, if no message of the form $(\mathsf{sen}, \mathsf{sid}, (a_0, a_1))$ is recorded and a message of the form $(\mathsf{rec}, \mathsf{sid}, \sigma)$ is stored, send $(\mathsf{sent}, \mathsf{sid}, a_\sigma)$ to R and $(\mathsf{sent}, \mathsf{sid})$ to S.

Open-all: Receive $(\mathsf{open\text{-}all})$ from S. Send all messages of the form $(\mathsf{sent}, \mathsf{sid}, (a_0, a_1))$ to R and $(\mathsf{sent}, \mathsf{sid})$ to S.

Commitment Schemes. Commitment schemes allow a party S to commit to a message m using randomness r. It keeps the message hidden, while allowing S to reveal the committed message later. We denote an UC secure commitment to message m with randomness r as $\textsc{Com}(m; r)$. The ideal commitment functionality $\mathcal{F}_{\mathsf{COM}}$ has been depicted in Fig. 2.6.

Figure 2.6: Functionality $\mathcal{F}_{\mathsf{COM}}$

$\mathcal{F}_{\mathsf{COM}}$

Commit: On receiving input $(\textsc{Commit}, \mathsf{sid}, m)$ from S, if sid has been recorded, ignore the input. Else record the tuple $(\mathsf{sid}, \mathsf{S}, \mathsf{R}, m)$ and send $(\textsc{Receipt}, \mathsf{sid}, \mathsf{S}, \mathsf{R})$ to R.

Decommit: On receiving input $(\textsc{Decommit}, \mathsf{sid})$ from S, if there is a record of the form $(\mathsf{sid}, \mathsf{S}, \mathsf{R}, m')$ return $(\textsc{Decommit}, \mathsf{sid}, m')$ to R. Otherwise, ignore the input.

2.1 Notations

For the oblivious transfer and the commitment scheme, we denote the sender by S and receiver by R. Similarly for ZK, we denote the prover and verifier as P and V respectively. We denote by $a \leftarrow_R D$ the random sampling of a from a distribution D and the set of elements $\{1, \ldots, n\}$ is represented by $[n]$. We denote a polynomial on variable x as $\mathsf{poly}(x)$. A function $\mathsf{neg}(\cdot)$ is said to be negligible, if for every polynomial $p(\cdot)$, there exists a constant c, such that for all $n > c$, it holds that $\mathsf{neg}(n) < \frac{1}{p(n)}$. We denote a probabilistic polynomial time algorithm as PPT. We denote the statistical security parameter by μ and the computational security parameter by κ. We use $\overset{c}{\approx}$ and $\overset{s}{\approx}$ to denote computational and statistical indistinguishability, respectively. Let $\mathbb{Z}_p$ denote the field of order p, where p is a prime and $p > 2^\kappa$. Let $\mathbb{G}$ be the multiplicative group corresponding to $\mathbb{Z}_q^*$ with generator g, where $q = 2p + 1$ and q is a prime number. For a bit $b \in \{0, 1\}$, we denote $1 - b$ by $\bar{b}$.

2.2 Primitives

Pseudorandom Generator. A pseudorandom generator (PRG) $G : \{0,1\}^\kappa \rightarrow \{0,1\}^{\mathsf{poly}(\kappa)}$ takes as input a κ-bits random string as a seed and outputs a $\mathsf{poly}(\kappa)$ bit pseudorandom string which is indistinguishable from a random string of same length, to a PPT adversary.

Pseudorandom Function. A pseudorandom function (PRF) $F_k : \{0,1\}^\kappa \times \{0,1\}^\ell \rightarrow \{0,1\}^\ell$ is parametrized by a κ-bits random key and takes in input an ℓ-bit argument x and outputs an ℓ-bit string. The output of F_k at x is indistinguishable from the output of a randomly sampled function at x, to a PPT adversary.

Garbled Circuit. Bellare et al [BHR12b] gave an abstraction of garbling schemes for circuits and formalized several notions of security. Using the language of [BHR12b] for circuits; the circuit itself is a directed acyclic graph, where each gate g is indexed by its outgoing wire, and its left and right incoming wires $A(g)$ and $B(g)$ are numbered such that $g > B(g) > A(g)$. Also, a circuit output wire can not be an input wire to any gate. We denote the number of input wires, gates and output wires using n, q and m respectively in a circuit C.

At a high-level, a garbling scheme consists of the following algorithms: Gb takes a circuit C as input and outputs a garbled circuit $\mathbf{C}$, encoding information e, and decoding information d. En takes an input x and encoding information and outputs a garbled input $\mathbf{X}$. Ev takes a garbled circuit and garbled input $\mathbf{X}$ and outputs a garbled output $\mathbf{Y}$. Finally, De takes a garbled output $\mathbf{Y}$ and decoding information and outputs a plain circuit-output (or an error, $\perp$).

In [JKO13], there is an additional verification algorithm in the garbling scheme which when accepts a given $(\mathbf{C}, e)$ signifies that the $\mathbf{C}$ is correct, and that the garbled output corresponding to any clear output can be extracted. Formally, a *garbling scheme* is defined by a tuple of functions $\mathsf{Garble} = (\mathsf{Gb}, \mathsf{En}, \mathsf{Ev}, \mathsf{De}, \mathsf{Ve})$, described as follows:

– Garble $\mathsf{Gb}\left(1^\kappa, C\right)$: A randomized algorithm which takes as input the security parameter and a circuit $C : \{0,1\}^n \rightarrow \{0,1\}^m$ and outputs a tuple of strings $(\mathbf{C}, e, d)$, where $\mathbf{C}$ is the garbled circuit, e denotes the input-wire labels, and d denotes the decoding information.

– Encode $\mathsf{En}\left(x, e\right)$: a deterministic algorithm that outputs the garbled input $\mathbf{X}$ corresponding to input x.

– Evaluation $\mathsf{Ev}\left(\mathbf{C}, \mathbf{X}\right)$: A deterministic algorithm which evaluates garbled circuit $\mathbf{C}$ on garbled input $\mathbf{X}$, and outputs a garbled output $\mathbf{Y}$.

– Decode $\mathsf{De}\left(\mathbf{Y}, d\right)$: A deterministic algorithm that outputs the plaintext output corresponding to $\mathbf{Y}$ or $\perp$ signifying an error if the garbled output $\mathbf{Y}$ is invalid.

– Verify $\mathsf{Ve}\left(C, \mathbf{C}, e\right)$: A deterministic algorithm which takes as input a circuit $C : \{0,1\}^n \mapsto \{0,1\}^m$, a garbled circuit (possibly malicious) $\mathbf{C}$, encoding information e, and outputs d when $\mathbf{C}$ is a valid garbling of C, and $\perp$ otherwise.

A garbling scheme may satisfy several properties such as *correctness, privacy, obliviousness, authenticity and verifiability*. We review some of these notions below. The definitions for correctness and authenticity are standard: correctness enforces that a correctly garbled circuit, when evaluated, outputs the correct output of the underlying circuit; authenticity enforces that the evaluator can only learn the output label that corresponds to the value of the function. *Verifiability* [JKO13] allows one to check that the garbling of a circuit indeed implements the specified plaintext circuit C. Given that verification succeeds for a candidate $(C, \mathbf{C}, e)$, the garbled output corresponding to a given clear output can be extracted. We also need a decisional version of the authenticity property, which we denote as *decisional authenticity*. It ensures that the encoded output, not corresponding to adversary's output will be indistinguishable from a random string of same length. We provide definitions of correctness, privacy and verifiability as we need it for our π_{CRS} protocol.

Definition 2.2.1. Correctness: *A garbling scheme* Garble *is* **correct** *if for all input lengths* $n \leq \mathsf{poly}(\kappa)$*, circuits* $C : \{0,1\}^n \to \{0,1\}^m$ *and inputs* $x \in \{0,1\}^n$*, the following probability is negligible in* κ*:*

$$\Pr\Big(\mathsf{De}(\mathsf{Ev}(\mathbf{C}, \mathsf{En}(e,x)), d) \neq C(x) : (\mathbf{C}, e, d) \leftarrow \mathsf{Gb}(1^\kappa, C)\Big).$$

Definition 2.2.2. Privacy: *A garbling scheme* Garble *is* **private** *if for all input lengths* $n \leq \mathsf{poly}(\kappa)$*, circuits* $C : \{0,1\}^n \to \{0,1\}^m$*, there exists a PPT simulator* *Sim* *such that for all inputs* $x \in \{0,1\}^n$*, for all probabilistic polynomial-time adversaries* A*, the following two distributions are computationally indistinguishable:*

- $\mathrm{REAL}(C,x)$: *run* $(\mathbf{C}, e, d) \leftarrow \mathsf{Gb}(1^\kappa, C)$*, and output* $(\mathbf{C}, \mathsf{En}(x,e), d)$*.*

- $\mathrm{IDEAL}_{\mathsf{Sim}}(C, C(x))$*: output* $(\mathbf{C}', \mathbf{X}, d') \leftarrow \mathsf{Sim}(1^\kappa, C, C(x))$

Definition 2.2.3. *(Authenticity) A garbling scheme* Garble *is* **authentic** *if for all input lengths* $n \leq \mathsf{poly}(\kappa)$*, circuits* $C : \{0,1\}^n \to \{0,1\}^m$*, inputs* $x \in \{0,1\}^n$*, and all probabilistic polynomial-time adversaries* A*, the following probability is negligible in* κ :

$$\Pr\begin{pmatrix} \widehat{\mathbf{Y}} \neq \mathsf{Ev}(\mathbf{C}, \mathbf{X}) & : & \mathbf{X} = \mathsf{En}(x,e),\ (\mathbf{C}, e, d) \leftarrow \mathsf{Gb}(1^\kappa, C) \\ \wedge \mathsf{De}(\widehat{\mathbf{Y}}, d) \neq \perp & & \widehat{\mathbf{Y}} \leftarrow \mathsf{A}(C, x, \mathbf{C}, \mathbf{X}) \end{pmatrix}.$$

Definition 2.2.4. *(Decisional Authenticity) A garbling scheme* Garble *is* **decisional authentic** *if for all input lengths* $n \leq \mathsf{poly}(\kappa)$*, circuits* $C : \{0,1\}^n \to \{0,1\}^m$*, for all inputs* $x \in \{0,1\}^n$*, there exists a function* F*, s.t for all probabilistic polynomial-time adversaries* A*, the following two distributions are computationally indistinguishable:*

- $(\mathbf{C}, \mathbf{X}, \{F(\widehat{\mathbf{Y}}_i, i)\}_{i=1}^m)$ *s.t.* $(\mathbf{C}, e, d) \leftarrow \mathsf{Gb}(1^\kappa, C)$, $\mathbf{X} = \mathsf{En}(x,e)$, $\mathbf{Y} = \mathsf{Ev}(\mathbf{C}, \mathbf{X})$, $\widehat{\mathbf{Y}} = (\widehat{\mathbf{Y}_1 \ldots \mathbf{Y}_m})$, $\mathsf{De}(\widehat{\mathbf{Y}}, d) \neq \perp$, $\{\mathbf{Y}_i \neq \widehat{\mathbf{Y}_i}\}_{i=1}^m$.

- $(\mathbf{C}, \mathbf{X}, \{Z_i\}_{i=1}^m)$ *s.t* $(\mathbf{C}, e, d) \leftarrow \mathsf{Gb}(1^\kappa, C)$, $\mathbf{X} = \mathsf{En}(x,e)$, $\mathbf{Y} = \mathsf{Ev}(\mathbf{C}, \mathbf{X})$, $\widehat{\mathbf{Y}} = (Z_1 \ldots Z_m)$, $\{Z_i \leftarrow_R \{0,1\}^{|\mathbf{Y}_i|}\}_{i=1}^m$.

Definition 2.2.5. Verifiability: *A garbling scheme* Garble *is* **verifiable** *if for all input lengths* $n \leq$ *poly*(κ)*, circuits* $C : \{0,1\}^n \to \{0,1\}^m$*, inputs* $x \in \{0,1\}^n$*, and PPT adversaries* A*, the following probability is negligible in* κ*:*

$$\Pr\left(\mathsf{De}(\mathsf{Ev}(\mathbf{C}, \mathsf{En}(x,e)), d) \neq C(x) \ : \begin{array}{c} (\mathbf{C}, e, d) \leftarrow \mathsf{A}(1^\kappa, C) \\ \mathsf{Ve}\,(C, \mathbf{C}, e) = d \neq \perp \end{array}\right).$$

We are interested in a class of garbling schemes referred to as *projective* in [BHR12b]. When garbling a circuit $C : \{0,1\}^n \mapsto \{0,1\}^m$, a projective garbling scheme produces encoding information of the form $e = (K_i^0, K_i^1)_{i\in[n]}$, and the encoded input $\mathbf{X}$ corresponding to $x = (x_i)_{i\in[n]}$ can be interpreted as $\mathbf{X} = \mathsf{En}(x, e) = (K_i^{x_i})_{i\in[n]}$.

Let $\mathbf{C}_k$ denote the kth garbled circuit instantiating circuit C. We assume that the randomness used for generating circuit k is derived from a κ-bit random string seed_k using a PRF. We assume that the fan-in of each gate is 2. We can assume that each AND gate in the circuit has 2 ciphertexts and XOR gates have 0 ciphertexts, using the Half-Gate construction [ZRE15] as the garbling scheme. In the same [ZRE15], they have optimized privacy-free garbling schemes when only authenticity property is required from the scheme. The privacy-free garbling scheme of [ZRE15] requires 0 ciphertext for XOR gates and 1 ciphertext for AND gates. In a privacy-free GC, the evaluator has private input whereas the constructor does not possess any input. Such schemes are useful when the evaluator has to prove that he knows an input, s.t. the circuit when computed on his input gives a particular output. The authenticity property prevents a corrupt evaluator from obtaining wire labels corresponding to other output bits. We will demonstrate the usefulness of such a scheme in our ZK protocol.

2.3 Random Oracle Functionality

A random oracle functionality is parametrized by a domain and a range and it is denoted as $\mathcal{F}_{\mathsf{RO}}$ in Fig. 2.7. A random oracle is queried on a message m from its domain D. Its returns an uniformly sampled random string and it is denoted as $\mathcal{F}_{\mathsf{RO}}(\mathsf{sid}||m)$, where sid is the session ID. The random oracle functionality can be broadly classified [CDG+18] into three categories based on its features- plain RO, observable RO and programmable RO. A plain RO returns a random string, from its range, upon being queried on a message m, from its domain. A plain random oracle is also called non-programmable non-observable random oracle. An observable RO inherits the properties of the plain RO but in addition it grants the simulator to observe the queries made, to $\mathcal{F}_{\mathsf{RO}}$, by the adversary. A programmable RO allows the simulator to program $\mathcal{F}_{\mathsf{RO}}(m)$ to return any string from the range, upon being queried on m for the first time.

2.4 Universal Composability Model

We prove security of our protocol in the standard Universal Composability (UC) framework of Canetti [Can01], with static and adaptive corruptions. We provide the definition of $\mathcal{F}$-hybrid model, which is instrumental for security proofs in the UC model. Then, we formally define the UC model for the two party setting as we deal with two parties only.

Figure 2.7: Functionality $\mathcal{F}_{\mathsf{RO}}$

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

$\mathcal{F}_{\mathsf{RO}}$ is parameterized by a domain D and range R and it proceeds as follows, running on security parameter k:

- $\mathcal{F}_{\mathsf{RO}}$ maintains a list L (which is initially empty) of pairs of values $(\hat{m}, \hat{h})$, s.t. $\hat{m} \in D$ and $\hat{h} \in R$.
- Upon receiving a value (sid, m) (where $m \in D$) perform the following: If there is a pair $(m, \hat{h})$, for some $\hat{h} \in R$, in the list L, set $h := \hat{h}$. If there is no such pair, sample $h \leftarrow_R R$ and store the pair (m, h) in L. Once h is set, reply to the activating machine with (sid, h).

2.4.1 The $\mathcal{F}$-hybrid model

The UC composition theorem states that if a protocol ρ UC-securely implements $\mathcal{F}$ then execution of ρ can be replaced by "ideal calls" to $\mathcal{F}$ functionality. This enables invoking ρ as a blackbox subprotocol in a bigger protocol π, without leading into compositional issues. Specifically, when constructing a protocol π that uses a subprotocol ρ for securely computing some functionality $\mathcal{F}$, the parties run π and invoke $\mathcal{F}$ (instead of running ρ) with "ideal calls". The execution of π that invokes $\mathcal{F}$, for each execution of ρ, is called the *$\mathcal{F}$-hybrid execution of* π and is denoted as $\pi^{\mathcal{F}}$. The hybrid ensemble $\mathsf{HYB}_{\pi^{\mathcal{F}},\mathsf{A},\mathcal{Z}}(1^{\kappa}, z)$ describes $\mathcal{Z}$'s output after interacting with A and the parties running protocol $\pi^{\mathcal{F}}$. Whereas, the execution of π that considers execution of ρ is denoted as π^{ρ}. The hybrid ensemble $\mathsf{HYB}_{\pi^{\rho},\mathsf{A},\mathcal{Z}}(1^{\kappa}, z)$ describes $\mathcal{Z}$'s output after interacting with A and the parties running protocol π^{ρ}. By security of ρ, the two hybrids $\mathsf{HYB}_{\pi^{\mathcal{F}},\mathsf{A},\mathcal{Z}}(1^{\kappa}, z)$ and $\mathsf{HYB}_{\pi^{\rho},\mathsf{A},\mathcal{Z}}(1^{\kappa}, z)$ are indistinguishable. This permits replacing executions of ρ, in π, with ideal calls to $\mathcal{F}$ functionality; thereby allowing π to execute in the $\mathcal{F}$-hybrid model. It simplifies the security proof of $\pi^{\mathcal{F}}$ as it can be performed in the $\mathcal{F}$-hybrid model, instead of proving security of ρ within the proof of π^{ρ}.

2.4.2 Static Security in the UC Model

In this model, the real world execution of protocol π is carried out between the honest parties P_1, P_2 and an adversary A, in the presence of an external entity called the environment $\mathcal{Z}$. All the parties are PPT Turing machines and $\mathcal{Z}$ has an auxiliary information z. At the outset of the protocol the environment initiates the parties with inputs and provides some initial information to A. $\mathcal{Z}$ is allowed to interact with A throughout the protocol. At the outset of the protocol, A may or may not corrupt a party. Upon corruption of a party, A gets access to the internal state and input of that party. From now on the party will behave according to A's instructions (since we are in the malicious model). At the end of the protocol, the honest parties send their output to $\mathcal{Z}$ while A outputs $\perp$ on behalf of the corrupted parties and its internal state to $\mathcal{Z}$. We denote the view of $\mathcal{Z}$ as $\text{REAL}_{\mathcal{F},\mathsf{A},\mathcal{Z}}(1^{\kappa}, z)$.

In the ideal world we consider the honest parties P_1, P_2, an ideal world PPT adversary Sim, $\mathcal{Z}$ and the functionality $\mathcal{F}$. Sim has a random tape r and security parameter κ. He simulates the role of A in the ideal world and whenever A corrupts a party in the real world Sim corrupts that party in the ideal world and gets access to its internal state. Sim invokes the algorithm of A, in his head, in another internal protocol execution where Sim simulates the view of the honest parties to A. We will denote this internal copy of A as $\mathsf{A}_{\mathsf{Int}}$. Based on the reply of $\mathsf{A}_{\mathsf{Int}}$ in the internal execution, Sim behaves accordingly in the ideal world execution. He extracts the inputs of the corrupted parties in the internal execution and invokes $\mathcal{F}$ in the ideal world with those inputs to obtain the output. In the internal execution he simulates the protocol in such a way that $\mathsf{A}_{\mathsf{Int}}$ obtains that output. At the end of the protocol, $\mathsf{A}_{\mathsf{Int}}$ forwards his view to Sim who forwards it to $\mathcal{Z}$. We denote the view of $\mathcal{Z}$ as $\text{IDEAL}_{\mathcal{F},\mathsf{Sim},\mathcal{Z}}(1^\kappa, z)$. We say that a protocol π UC-securely implements a functionality $\mathcal{F}$ in the presence of static adversaries if the real world and ideal world views are indistinguishable.

Definition 2.4.1. *Let π be a protocol for computing a functionality $\mathcal{F}$. We say that π UC-securely computes the two party protocol functionality $\mathcal{F}$ in the presence of static adversaries if for every PPT adaptive real-world adversary A and every environment $\mathcal{Z}$, there exists a PPT ideal-world adversary Sim, such that:*

$$\text{REAL}_{\mathcal{F},\mathsf{A},\mathcal{Z}}(1^\kappa, z) \overset{c}{\approx} \text{IDEAL}_{\mathcal{F},\mathsf{Sim},\mathcal{Z}}(1^\kappa, z)$$

2.4.3 Adaptive Security in the UC Model

In the adaptive setting, $\mathcal{Z}$ can ask the real world adversary A to corrupt an honest party during the real world execution of the protocol or after the execution completes. During the execution, A can observe the public transcript of the protocol and based on that he can adaptively corrupt an honest party. Once a party gets corrupted, A gets access to the input and private randomness of the party, thus controlling the party from thereon. In case of post execution corruption, A observes the output and the transcript of the protocol, and then he corrupts the honest party to get access to the input and private randomness of the party. After post execution corruption occurs, A forwards its view to $\mathcal{Z}$. Based on that, $\mathcal{Z}$ constructs its real world view, which we denote as $\text{REAL}_{\mathcal{F},\mathsf{A},\mathcal{Z}}(1^\kappa, z)$.

Similarly, in the ideal world $\mathcal{Z}$ can ask the ideal world adversary Sim to corrupt an honest party during the ideal world execution of the protocol or after the execution completes. When $\mathcal{Z}$ instructs Sim to corrupt an honest party in the ideal world, Sim obtains the input of the honest party, in the ideal world, and he instructs the internal world adversary $\mathsf{A}_{\mathsf{Int}}$ to corrupt the corresponding honest party in the internal world. Recall that Sim simulates the honest parties in the internal execution. When $\mathsf{A}_{\mathsf{Int}}$ corrupts an honest party in the internal world, Sim has to produce a private randomness for the simulated honest party such that it matches with the input of the honest party and the simulated transcript produced by Sim, in the internal world, on behalf of the honest party. Sim provides this matching randomness and the input of the simulated honest party to $\mathsf{A}_{\mathsf{Int}}$ in the internal world. In case of post execution corruption of an honest party, Sim obtains the honest party's input in the ideal world and produces the matching randomness (corresponding to the simulated transcript) in a similar fashion to $\mathsf{A}_{\mathsf{Int}}$ in the internal world. After post execution corruption occurs, A forwards its view to Sim, who forwards it to $\mathcal{Z}$. Based on that, $\mathcal{Z}$ constructs its ideal world view, which we denote as $\text{IDEAL}_{\mathcal{F},\mathsf{Sim},\mathcal{Z}}(1^{\kappa}, z)$. We say that a protocol π UC-securely implements a functionality $\mathcal{F}$ in the presence of adaptive adversaries if the real world and ideal world views are indistinguishable.

Definition 2.4.2. *Let π be a protocol for computing a functionality $\mathcal{F}$. We say that π UC-securely computes the two party protocol functionality $\mathcal{F}$ in the presence of adaptive adversaries if for every PPT adaptive real-world adversary A and every environment $\mathcal{Z}$, there exists a PPT ideal-world adversary Sim, such that:*

$$\text{REAL}_{\mathcal{F},\mathsf{A},\mathcal{Z}}(1^{\kappa}, z) \overset{c}{\approx} \text{IDEAL}_{\mathcal{F},\mathsf{Sim},\mathcal{Z}}(1^{\kappa}, z)$$

Challenges in Adaptive Security. It is challenging to construct adaptively-secure protocols since Sim has to provide matching randomness for the simulated transcripts (corresponding to an honest party) in the internal world, once the honest party is corrupted by $\mathsf{A}_{\mathsf{Int}}$. This makes the simulation procedure inherently difficult since the simulated transcripts were generated without access to the honest party's input. Whereas, later on they have to be equivocated to look consistent with the honest party's input when $\mathsf{A}_{\mathsf{Int}}$ corrupts the party. Such a requirement for equivocation yields inefficient constructions in the adaptive domain. However, in the later chapters we provide some efficient constructions of adaptively-secure primitives in the RO model.

Chapter 3

Adaptively Secure Zero Knowledge

Zero-knowledge (ZK) proofs introduced in [GMR85] provide a powerful tool in designing a variety of cryptographic protocols. Since then, they have been an important building block in various applications. Zero-knowledge proofs allow a prover to convince a verifier about the validity of a statement, while giving no information beyond the truth of the statement. Informally, an honest prover should always convince a verifier about a true statement (completeness). Moreover, a malicious verifier learns nothing beyond the validity of the statement (zero-knowledge) and a malicious prover cannot convince a verifier of a false statement (soundness). In addition to soundness, a ZK protocol in which the prover's witness can be extracted by a simulator offers *proof of knowledge*.

3.1 Related Work

It is known that every language in NP has a zero-knowledge proof system [GMW86]. Despite this, proving generic statements is inefficient in practice, and there are few techniques that allow efficient proofs. These techniques almost always apply to a restricted set of languages, with a series of works [Sch90, GQ88, CM99, GS08] on proving algebraic relationships like knowledge of roots, discrete logarithms etc. Kilian's zero-knowledge argument [Kil92] achieves sub-linear communication, but relies on Probabilistically Checkable Proofs and is of theoretical interest. Groth [Gro10] gave the first constant-size non-interactive ZK proofs. Since then, many constructions of succinct non-interactive arguments of knowledge (SNARKs) have been presented [GGPR13, Lip13, DFGK14, Gro16], and have been implemented as well [PHGR13, CFH$^+$15]. Though SNARKs have short proofs and allow efficient verification, they have shortcomings in prover efficiency. The prover performs public-key operations proportional to the size of the circuit representing the statement. In addition, they rely on a large trusted parameter; for example, a long crs.

An interesting line of recent works [IKOS07, BP12, JKO13, HMR15, CGM16, GMO16, HV16, AHIV17] establishes connections between MPC and ZK, and use the techniques of 2PC and MPC for truly efficient ZK protocols. The two main streams of works connecting MPC with efficient ZK protocols rely on "MPC-in-the-head" approach [IKOS07] and garbled circuit based approach [JKO13], as elaborated below.

Ishai et al. [IKOS07, IKOS09] show how to use an MPC protocol to obtain a ZK proof for an NP relation in the commitment-hybrid model. This approach, called "MPC-in-the-head", provides a powerful tool to obtain black-box constructions for generic statements without relying on expensive Karp reductions. Recently, this technique spurred progress in constructing practical ZK protocols [GMO16, CDG$^+$17] resulting in efficient ZK arguments tailored for Boolean circuits, known as 'ZKBoo' and 'ZKBoo++' respectively. They study variants of the "MPC-in-the-head" framework, plug in different MPC protocols, and provide concrete estimates of soundness. In yet another recent attempt, [AHIV17] proposes 'Ligero', a 4 round interactive ZK argument with sub-linear (in the circuit size) proof-size relying on interactive PCPs and plugging in a refined MPC of [DI06] in the "MPC-in-the-head" approach. Specifically, they achieve a proof size of $\mathcal{O}(\kappa\sqrt{|C|\log|C|})$. The construction uses Reed Solomon Codes from coding theory techniques. The marked improvement in the proof size is obtained by careful tweaking of the protocol parameters. The prover and verifier time is $\mathcal{O}(|C|\log|C|)$ symmetric key operations, and without any public key operations. The protocol does not require any setup and the security is proven in the stand-alone setting. The constructions of [GMO16, CDG$^+$17, AHIV17] can be made non-interactive using the Fiat-Shamir heuristic [FS86] in the PRO model.

Jawurek et al. [JKO13] construct a UC-secure ZK protocol (referred to as ZKGC henceforth) using garbled circuits as the primary building block. The communication required for their protocol is linear in the size of the circuit implementing the NP relation, and is also concretely efficient as it achieves malicious security with only one garbled circuit. However, the protocol is inherently interactive. ZKGC is essentially a version of Yao's original constant-round 2PC protocol where the GC constructor has no input; this yields full malicious security at little overhead over the semi-honest case as Yao's protocol in this case is already secure against a malicious evaluator. The protocol uses oblivious transfer (OT). The use of OT in ZK protocols dates back to [KMO89]. Notably, Zero-knowledge, when viewed as a special case of 2PC, allows for a relaxation in the properties required of the underlying GCs, as noted in [JKO13]. This led to the introduction of the notion of *privacy-free* garbling schemes [FNO15], which are optimized for the ZK setting of [JKO13]. A privacy-free garbling scheme only achieves authenticity, and leverages privacy-freeness in order to save on communication and computation costs of garbling. Privacy-free GCs are further studied by Zahur et al. [ZRE15], who construct a privacy-free scheme using the HalfGates approach. Their privacy-free scheme makes use of FreeXOR [KS08] to garble and evaluate XOR gates at no cost, and produces only one ciphertext when garbling an AND gate (along with two calls to a hash function). Their construction comprises the current state-of-the-art in privacy-free garbling for circuits. When formulaic circuits are of concern, [KP17] shows how to do privacy-free garbling with *zero* ciphertext and with information-theoretic security.

The interactive schemes based on garbled circuits allow for the flexibility of how the keys for the underlying GCs are constructed and how the garbled input (ie. witness) is encoded. This leads to interesting applications making non-blackbox use of ZKGC [CGM16, KKL$^+$16]. For instance, Kolesnikov et al. [KKL$^+$16] introduce a new primitive called "attribute selective encryption" as a method of input encoding in ZKGC in order to construct attribute-based key-exchange. This allows a client to prove to a server that it holds a certificate corresponding to its attributes issued by a trusted authority, and that these attributes satisfy a policy constructed by the server. Another point of comparison is that the PRO assumption required by non-interactive 'MPC-in-the-head' based ZK protocols can be used to construct highly efficient adaptively secure garbled circuits [BHR12a] allowing ZKGC and our protocol to be cast in the online-offline paradigm, with all circuit-dependent communication moved to a preprocessing stage.

Lastly, we note that all of the above protocols deal with *static* adversaries. In this work, we are interested in building efficient concurrently composable ZK protocols which are adaptively-secure. Next, we summarize the literature on practical ZK protocols for non-algebraic statements, and zero-knowledge protocols secure against adaptive adversaries.

Adaptively-Secure Zero-Knowledge. We recall that an adaptive adversary may dynamically decide which party to corrupt as the protocol progresses. Its choice of corruptions may be adapted according to the specific information it sees, possibly even corrupting both the parties. Tolerating an adaptive adversary in a ZK protocol in the UC setting requires a straight-line simulator that can generate a transcript on behalf of the prover without knowledge of the witness, and later be able to "explain" the transcript for any given witness (ie. concoct valid-looking corresponding local randomness). In [Bea96a], the authors show that the zero-knowledge proof system of GMW [GMW91] is not secure against adaptive adversaries or else the polynomial hierarchy collapses, and proceed to build ZK arguments. This work is further advanced in [CLOS02] where UC-secure ZK arguments are presented relying on adaptive commitments schemes. In [LZ11], it is shown that adaptive ZK proofs exist for all of NP assuming only one-way functions. They present constructions of adaptively secure ZK proofs from adaptive instance dependent commitment schemes.

We note that the "MPC-in-the-head" approach is likely to generate adaptively secure ZK protocols by relying on adaptive commitments and possibly adaptively secure MPC. An adaptive commitment scheme is used to commit to the views of the virtual parties. The adaptive commitment schemes from standard assumptions [HV16, HPV17] may be taxing in terms of both communication and round efficiency. Alternatively, the commitments used in IKOS-style protocols can be implemented in the programmable random oracle model, allowing the simulator to equivocate committed views, which yields adaptive security in a straightforward manner. Another related method is via non-committing encryption (NCE), an approach that has in other circumstances allowed circumvention of known lower bounds in the plain model. For instance, the adaptively secure garbling scheme of [BHR12a] uses a programmable RO to achieve NCE, which results in the circumvention of a lower bound in the online communication complexity of adaptively secure garbling schemes shown by Applebaum et al. [AIKW15].

The work of [HV16] uses the "MPC-in-the-head" technique [IKOS09] to construct adaptive ZK proofs. Their use of interactive hashing [NOVY98] to construct instance dependent commitments to equivocate committed views requires a non-constant number of rounds. The overall round complexity of their adaptive ZK protocol is $\mathcal{O}(\mu \log \mu)$, where μ is the soundness parameter. The proof size is $\mathcal{O}(\mu|\mathbf{C}|\mathsf{poly}(\kappa))$ and the $\mathsf{poly}(\kappa)$ factor is $\Omega(\kappa)$. While their scheme can be made constant round by plugging in the appropriate instance-dependent commitment scheme, it comes at the cost of proofs that are quadratic in the size of the circuit implementing the NP relation.

In this work, we explore the possibility of building protocols that lie at the intersection of all of these desirable qualities. Specifically we address the following question:

> Can we construct constant-round UC-secure ZK protocols that are secure against adaptive corruptions, with proof size linear in the size of the circuit that implements the NP relation?

3.2 Our Results

Inspired by the recent progress in the domain of garbling schemes as primitives and interesting applications of garbled circuit (GC) based ZK protocols, we revisit ZK protocols from GCs. Recent works including [CGM16, KKL$^+$16] make non-blackbox use of the GC-based ZK protocols of [JKO13], exploiting particularly the way the keys for the underlying GCs are constructed and the method by which the garbled input (i.e. witness) is encoded. Such applications will directly benefit from any improvement in the domain of garbled circuit based ZK protocols. Our contributions are listed below.

While security against static adversaries provides a convenient stepping-stone for designing protocols against strong malicious attacks, a general real-life scenario certainly calls for adaptive security where the adversary can use its resources in a gradual fashion, making dynamic corruption decisions as the protocol progresses. Our first contribution is to show that the ZK protocol of [JKO13] can be proven to be adaptively secure in the UC setting if the underlying oblivious transfer (OT) primitive satisfies a mild adaptive security guarantee. Namely, we require that the receiver's communication can be equivocated to any input of the receiver. Such an OT is referred to as receiver equivocal OT ($\mathsf{RE\text{-}OT}$). We show that the framework of [PVW08] itself, in one of its incarnation, provides $\mathsf{RE\text{-}OT}$. Specifically, the mode of [PVW08] that offers statistical security for the receiver also offers the flavor of adaptive security that we demand from $\mathsf{RE\text{-}OT}$. The main observation instrumental in crafting the adaptive proof of security for ZKGC is that the constructor of GC has no input. Therefore, the primary challenge of explaining the randomness of the GC construction in post-execution corruption case is bypassed.

Next, we focus on reducing the exact round complexity of ZKGC style protocols. We propose a three-round protocol. Since neither zero-knowledge proofs nor arguments can be achieved in less than four rounds without additional assumptions [GK96], we devise our protocols in the crs model where the crs is short unlike those used in SNARKs. Starting with ZKGC, our three-round protocol cuts down two rounds in [JKO13] using the idea of conditional opening [BP12] of a secret information that enables garbled circuit verification. That is, the key to GC verification can be unlocked only when the prover possesses a valid witness. Though fairly simple, implementing this idea makes the security proof of the resulting protocol challenging and subtle due to a circularity issue. Loosely speaking, when the prover does not hold a valid witness, the authenticity of GC should translate to the security of the key and at the same time, the security of the key should translate to the authenticity of the GC. We handle this issue by implementing the conditional disclosure via encryption in the observable RO model. If we further assume that the garbling scheme satisfies decisional authenticity then our proof holds even in the plain RO model. The state-of-the-art garbling scheme of [ZRE15] already satisfies decisional authenticity enabling us to reduce the assumption to plain RO only. While the ZKGC protocol requires at least 5 rounds in its most round-efficient instantiation, we improve the complexity to three at *no* additional cost of communication (in fact with slight improvement), and little change in computation (one hash invocation versus a commitment in [JKO13]). We show this protocol to be adaptively secure too, when plugged in with $\mathsf{RE\text{-}OT}$s. In table 3.1, we compare our protocol asymptotically with the existing efficient constructions. Let 'PKE' and 'SKE' denote the number of public key and respectively secret key operations. We note that $\mathsf{RE\text{-}OT}$ can be efficiently constructed assuming DDH assumption, with no overhead over the regular OT in the framework of [PVW08]. Moreover, we can instantiate the garbled circuit in our protocol with state-of-the-art privacy free garbling schemes [ZRE15] as the constructor of the circuit is the verifier and he does not possess any input. This would incur 1 ciphertext for AND gate and 0 ciphertexts for XOR gate.

Roadmap. We begin by discussing the definition of $\mathsf{RE\text{-}OT}$ in Section 3.3. In Section 3.4, we recall the ZK protocol of [JKO13] and prove that it is adaptively-secure, when plugged in with $\mathsf{RE\text{-}OT}$. Finally, in Section 3.5 we present our three-round adaptively-secure ZK protocol from conditional disclosure.

Table 3.1: Comparison among Zero Knowledge Protocols

Protocols	Proof Size	Prover Runtime	Verifier Runtime	Rounds	Assumptions	Security
ZKGC [JKO13]	$\mathcal{O}(\kappa \cdot \lvert C\rvert)$	$\mathcal{O}(\lvert C\rvert)$ SKE + $\mathcal{O}(n)$ PKE	$\mathcal{O}(\lvert C\rvert)$ SKE + $\mathcal{O}(n)$ PKE	5	Standard (OWF) +OT	Static (UC)
ZKBoo [GMO16]	$\mathcal{O}(\kappa \cdot \lvert C\rvert)$	$\mathcal{O}(\kappa\lvert C\rvert)$ SKE	$\mathcal{O}(\kappa\lvert C\rvert)$ SKE	1	PRO	Adaptive
ZKB++ [CDG+17]	$\mathcal{O}(\kappa \cdot \lvert C\rvert)$	$\mathcal{O}(\kappa\lvert C\rvert)$ SKE	$\mathcal{O}(\kappa\lvert C\rvert)$ SKE	1	PRO	Adaptive
Ligero (Arithmetic)	$\mathcal{O}(\kappa^{1.5}\sqrt{\lvert C\rvert})$	$\mathcal{O}(\lvert C\rvert \log \lvert C\rvert)$ SKE	$\mathcal{O}(\lvert C\rvert \log \lvert C\rvert)$ SKE	1	PRO	Adaptive
Ligero (Boolean)	$\mathcal{O}(\kappa\sqrt{\lvert C\rvert \log \lvert C\rvert})$	$\mathcal{O}(\lvert C\rvert \log \lvert C\rvert)$ SKE	$\mathcal{O}(\lvert C\rvert \log \lvert C\rvert)$ SKE	1	PRO	Adaptive
[HV16]	$\mathcal{O}(\mu\lvert C\rvert\mathsf{poly}(\kappa))$	$\mathcal{O}(\mu\lvert C\rvert\mathsf{poly}(\kappa))$ SKE	$\mathcal{O}(\mu\lvert C\rvert\mathsf{poly}(\kappa))$ SKE	$\mathcal{O}(\mu \log \mu)$	Standard (OWP)	Adaptive
ZKGC **(This paper)**	$\mathcal{O}(\kappa \cdot \lvert C\rvert)$	$\mathcal{O}(\lvert C\rvert)$ SKE + $\mathcal{O}(n)$ PKE	$\mathcal{O}(\lvert C\rvert)$ SKE + $\mathcal{O}(n)$ PKE	5	Standard (OWF) RE-OT (DDH)	Adaptive (UC)
This paper	$\mathcal{O}(\kappa \cdot \lvert C\rvert)$	$\mathcal{O}(\lvert C\rvert)$ SKE + $\mathcal{O}(n)$ PKE	$\mathcal{O}(\lvert C\rvert)$ SKE + $\mathcal{O}(n)$ PKE	3	plain RO RE-OT (DDH)	Adaptive (UC)

3.3 Receiver-Equivocal Oblivious Transfer

An oblivious transfer protocol is said to be receiver equivocal if it is possible to produce the receiver's message in the protocol *without committing to a choice bit*. For this to be meaningful, we also require that it be possible to efficiently generate the local randomness which when combined with either choice bit would make an honest receiver output the same message. This is formalized by requiring the existence of a simulator $\mathsf{Sim}^{\mathsf{RE}}$ which can perform this task, in Definition 3.3.1.

Definition 3.3.1. *($\mathsf{RE\text{-}OT}$) Let $\pi_{\mathsf{ot}} = (\pi^{\mathsf{S}}_{\mathsf{ot}}, \pi^{\mathsf{R}}_{\mathsf{ot}})$ be a 2-round OT protocol securely implementing the $\mathcal{F}_{\mathsf{OT}}$ functionality in the crs model where S and R run their respective algorithms as specified by $\pi^{\mathsf{S}}_{\mathsf{ot}}(\mathsf{crs}, a_0, a_1, m^{\mathsf{R}}; r^{\mathsf{S}})$ and $\pi^{\mathsf{R}}_{\mathsf{ot}}(\mathsf{crs}, \sigma; r^{\mathsf{R}})$ respectively. Here, a_0, a_1 are the sender's inputs, σ is the receiver's choice bit, $r^{\mathsf{S}}, r^{\mathsf{R}}$ are the sender's and receiver's respective local randomness, and m^{R} is the receiver's message. Let $(\mathsf{crs}, t) \leftarrow \mathsf{Setup}(1^n, \mu)$ be the output of the setup functionality which takes the security parameter and a mode $\mu \in \{0, 1\}$, and t is the corresponding trapdoor which is accessible only to the simulator Sim. Then π_{ot} is an $\mathsf{RE\text{-}OT}$ if the following conditions hold:*

- *Indistinguishability of modes: The crs of the two modes are computationally indistinguishable,*

$$\mathsf{crs}_0 \overset{c}{\equiv} \mathsf{crs}_1 \,\forall\, (\mathsf{crs}_0, t_0) \leftarrow \mathsf{Setup}(1^n, 0), (\mathsf{crs}_1, t_1) \leftarrow \mathsf{Setup}(1^n, 1)$$

- *$\mathcal{F}_{\mathsf{OT}}$ in mode 0: $\forall\, \mathsf{crs} \leftarrow \mathsf{Setup}(1^n, 0)$, $\pi_{\mathsf{ot}} = \left(\pi^{\mathsf{S}}_{\mathsf{ot}}(\mathsf{crs}, a_0, a_1, m^{\mathsf{R}}; r^{\mathsf{S}}), \pi^{\mathsf{R}}_{\mathsf{ot}}(\mathsf{crs}, \sigma; r^{\mathsf{R}})\right)$ securely implements the $\mathcal{F}_{\mathsf{OT}}$ functionality.*

- *Equivocation in mode 1: There exists an algorithm $\mathsf{Sim}^{\mathsf{RE}}\,(\mathsf{crs}, t)$ which outputs $\left(m^{\mathsf{R}}, r^{\mathsf{R}}_0, r^{\mathsf{R}}_1\right)$ such that $m^{\mathsf{R}} = \pi^{\mathsf{R}}_{\mathsf{ot}}(\mathsf{crs}, 0; r^{\mathsf{R}}_0) = \pi^{\mathsf{R}}_{\mathsf{ot}}(\mathsf{crs}, 1; r^{\mathsf{R}}_1)$, and $r^{\mathsf{R}}_0, r^{\mathsf{R}}_1 \overset{s}{\approx} r^{\mathsf{R}}$, $\forall\, \mathsf{crs} \leftarrow \mathsf{Setup}(1^n, 1)$.*

On the use of a crs. We note here that there is nothing inherent in receiver equivocation that demands a crs to implement $\mathsf{RE\text{-}OT}$. We are interested in achieving UC-security, and so as to allow the protocol of [PVW08] as an instantiation of our definition, we assume that the protocol realizing $\mathsf{RE\text{-}OT}$ will make use of a crs. There is a concurrent work by [GS17] which compiles any 2 round actively OT protocol into a 2 round $\mathsf{RE\text{-}OT}$ protocol using garbled circuits. We refer to their work for more details about the compiler.

Instantiation of $\mathsf{RE\text{-}OT}$. The OT framework of [PVW08] is already receiver equivocal as per Definition 3.3.1. The protocol can be constructed efficiently under the Decisional Diffie Hellman, Quadratic Residuosity, or Learning With Errors hardness assumptions. The constructions of [PVW08] operate in two modes: messy and decryption, that corresponds to mode 0 and 1 respectively of our definition.

Theorem 3.3.2. *The protocol π_{PVW} in Fig. 3.1 is a $\mathsf{RE\text{-}OT}$, assuming that DDH is hard for $\mathbb{G}$.*

Proof. The protocol π_{PVW} in Fig. 3.1 is proven to realize the $\mathcal{F}_{\mathsf{OT}}$ functionality in the UC model by Peikert et al. [PVW08]. It is easy to see how $\mathsf{Sim}^{\mathsf{RE}}_{\mathsf{PVW}}$ allows for receiver equivocation as per Def. 3.3.1 when the crs is generated in mode 1:

- The randomness r^{R}_{σ} provided is interpreted as R's secret exponent α.
- Recall that the message m^{R} is (g_0^r, h_0^r), and candidate randomness output by $\mathsf{Sim}^{\mathsf{RE}}_{\mathsf{PVW}}$ is $r^{\mathsf{R}}_0 = r$, and $r^{\mathsf{R}}_1 = r^{\mathsf{R}}_0 \cdot t^{-1} = r \cdot t^{-1}$
- Correctness of message m^{R} can be seen as follows:
  (a) $\pi_{\mathsf{PVW}}\left(\mathsf{crs}, 0; r^{\mathsf{R}}_0\right)$ will output $\left(g_0^{r^{\mathsf{R}}_0}, h_0^{r^{\mathsf{R}}_0}\right) = (g_0^r, h_0^r) = m^{\mathsf{R}}$
  (b) $\pi_{\mathsf{PVW}}\left(\mathsf{crs}, 1; r^{\mathsf{R}}_1\right)$ will output $\left(g_1^{r^{\mathsf{R}}_1}, h_1^{r^{\mathsf{R}}_1}\right) = \left(g_1^{(r\cdot t^{-1})}, h_1^{(r\cdot t^{-1})}\right)$
  Recall that the trapdoor t relates g_0 to g_1 as $g_0^t = g_1$ and similarly $h_0^t = h_1$. Therefore we have that $\left(g_1^{(r\cdot t^{-1})}, h_1^{(r\cdot t^{-1})}\right) = (g_0^r, h_0^r) = m^{\mathsf{R}}$
- Finally, $r^{\mathsf{R}}_0, r^{\mathsf{R}}_1 = r, r \cdot (t^{-1})$ are clearly uniformly random, as r is sampled uniformly at random.

□

The construction satisfies Definition 3.3.1 when instantiated in "decryption mode". In the simulation, when the receiver is corrupted before the first message is sent, the simulator sets the crs in the messy mode, and no equivocation is necessary. Otherwise, the simulator sets the crs in the decryption mode. Here we recall the instantiation of π_{PVW} under the DDH hardness assumption and describe $\mathsf{Sim}^{\mathsf{RE}}_{\mathsf{PVW}}$ in the decryption mode. (Fig. 3.1).

Also note that $\mathsf{RE\text{-}OT}$ is strictly weaker than OT with security against adaptive corruptions; any protocol satisfying the latter notion will necessarily be receiver-equivocal in order for the receiver's view to be fully simulatable in the event of a post-execution corruption.

Figure 3.1: $\mathsf{RE\text{-}OT}$ assuming DDH: as per [PVW08]

π_{PVW}

The parties have access to a common reference string $\mathsf{crs} \in \mathbb{G}^4$. Operations are over group $\mathbb{G}$.

$\mathsf{Setup}(1^n, 0)$:
$\mathsf{crs} = (g_0, h_0, g_1, h_1) \in \mathbb{G}^4$. The trapdoor available to the simulator is $t = (t_0, t_1)$ such that $g_0^{t_0} = h_0$ and $g_1^{t_1} = h_1$.

$\mathsf{Setup}(1^n, 1)$:
$\mathsf{crs} = (g_0, h_0, g_1, h_1) \in \mathbb{G}^4$. The trapdoor available to the simulator is t such that $g_0^t = g_1$ and $h_0^t = h_1$.

$\pi^{\mathsf{R}}_{\mathsf{PVW}}(\mathsf{crs}, \sigma)$:

- Sample $\alpha \in \mathbb{Z}_q$ uniformly at random.
- Compute $g = (g_\sigma)^\alpha$, $h = (h_\sigma)^\alpha$
- Send (g, h)

$\pi^{\mathsf{S}}_{\mathsf{PVW}}(\mathsf{crs}, a_0, a_1, m^{\mathsf{R}})$:

- Sample random elements r_0, s_0, r_1, s_1 from $\mathbb{Z}_q$.
- Compute $u_0 = g_0^{r_0} h_0^{s_0}$, $v_0 = g^{r_0} h^{s_0}$, $u_1 = g_1^{r_1} h_1^{s_1}$, $v_1 = g^{r_1} h^{s_1}$.
- Send $(u_0, w_0 = v_0 a_0)$, $(u_1, w_1 = v_1 a_1)$

R can retrieve the chosen message as $a_\sigma = w_\sigma \cdot (u_\sigma)^{-\alpha}$

$\mathsf{Sim}^{\mathsf{RE}}(\mathsf{crs}, t)$:

- Sample $r \in \mathbb{Z}_q$ and compute $m^{\mathsf{R}} = (g_0^r, h_0^r)$.
- Compute local randomness for both possible receiver inputs as $r_0^{\mathsf{R}} = r$ and $r_1^{\mathsf{R}} = r \cdot t^{-1}$.
- Output $(m^{\mathsf{R}}, r_0^{\mathsf{R}}, r_1^{\mathsf{R}})$

3.4 Adaptive Security of [JKO13]

In this section, we recall the construction of [JKO13] (the schematic diagram is given in Fig. 3.3). Next, we prove that it satisfies adaptive security if the underlying OT is receiver equivocal.

3.4.1 Recap of [JKO13]

We recall the ZKGC protocol below in the $(\mathcal{F}_{\mathsf{COT}}, \mathcal{F}_{\mathsf{COM}})$ hybrid model. In ZKGC, the verifier V constructs a garbled circuit $\mathbf{C}$ which computes circuit C implementing the relation $R(z, x)$. The wires corresponding to x is the private input of the evaluator whereas z is public statement. V sends $\mathbf{C}$ to P and P obtains the input wire labels corresponding to x through OTs. Upon obtaining the input wire labels, P computes the circuit and obtains the garbled output Z, which is the output wire label corresponding to bit 1. If V was honest, then P could have sent Z to V to prove knowledge of the witness. However, V can be malicious and he can manipulate the $\mathbf{C}$ s.t. if P's first witness bit is 0, then P obtains Z else he aborts. Based on P's behavior, a corrupt V^* can infer the first bit of x. In order to tackle that, ZKGC allows P to check the circuit before sending Z to V. However, upon checking a malicious P^* can obtain Z, even if he does not possess a valid witness. To tackle this, ZKGC makes the prover commit to Z after he evaluates the garbled circuit. Upon obtaining the commitment, V opens the randomness for OTs and garbled circuit. P runs the verification algorithm on the OT and garbled circuit using the randomness and aborts if inconsistency is detected. Else, P decommits to the commitment to Z and proves to V that he has indeed obtained the output wire label corresponding to bit 1. V outputs accept if the decommitment is correct, else he outputs reject. The original ZKGC protocol has been presented in Fig. 3.2 and a schematic diagram is given in Fig. 3.3).

3.4.2 Proof of Adaptive Security for [JKO13] from RE-OT

In this section we show that instantiating the ZKGC protocol with RE-OT satisfying Definition 3.3.1 yields a UC-secure protocol realizing $\mathcal{F}_{\mathsf{ZK}}^{R}$ (see Figure 2.1) tolerating adaptive adversaries. During the simulation, the simulator Sim plays the ideal world adversary role. It runs the protocol in an internal world with the parties corrupted by adversary A. Sim invokes the algorithm of the corrupted parties. Sim simulates the role of the honest parties in the internal world. Based on the interaction in the internal world, Sim creates an ideal world adversarial view by running the ZK functionality with the honest parties. At the end of the simulation, Sim forwards the view of the ideal world adversary to the environment $\mathcal{Z}$. Whereas, in the real world A corrupts the parties and runs the protocol with the honest parties to obtain a real world view. A protocol is secure if the view of the ideal world and real world adversaries are indistinguishable to $\mathcal{Z}$. Next, we recall the static proof of security for ZKGC.

Figure 3.2: Zero-knowledge from Garbled Circuits [JKO13]

π_{ZKGC}

- **Oracles and Cryptographic Primitives:** A *correct, authentic, verifiable* garbling scheme $\mathsf{Garble} = (\mathsf{Gb}, \mathsf{En}, \mathsf{De}, \mathsf{Ev}, \mathsf{Ve})$. A committing OT oracle $\mathcal{F}_{\mathsf{COT}}$.
- **Common Inputs of P and V:** Relation R realized by circuit C and statement z.
- **Input of P:** A witness x of size $n = \mathsf{poly}(\kappa)$ such that $R(z, x) = 1$.
- **Input of V:** Nothing.

Witness input phase:
For all $i \in [n]$, P sends $(\mathsf{rec}, \mathsf{sid}, x_i)$ to $\mathcal{F}_{\mathsf{COT}}$.

GC Construction and wire label transfer phase:
V garbles the circuit, $\left(\mathbf{C}, (K_i^0, K_i^1)_{i\in[n]}, Z\right) \leftarrow \mathsf{Gb}(1^\kappa, C)$[a]. V sends $\left(\mathsf{S}, \mathsf{sid}, (K_i^0, K_i^1)\right)$ as input to $\mathcal{F}_{\mathsf{COT}}$ for all $i \in [n]$.

GC Evaluation and output commitment phase:
P receives $(\mathsf{R}, \mathsf{sid}, K_i^{x_i})$ for $i \in [n]$ from $\mathcal{F}_{\mathsf{COT}}$, and parses $X = K_1^{x_1} \ldots K_i^{x_i} \ldots K_n^{x_n}$. P obtains $Z' = \mathsf{Ev}(\mathbf{C}, X)$ and sends $(\textsc{commit}, \mathsf{sid}, Z')$ to $\mathcal{F}_{\mathsf{COM}}$.

GC verification and conditional output disclosure phase:
On receiving $(\mathsf{Receipt}, \mathsf{sid}, \mathsf{P}, \mathsf{V})$ from $\mathcal{F}_{\mathsf{COM}}$, V sends the message $(\mathsf{open\text{-}all}, id)$ to $\mathcal{F}_{\mathsf{COT}}$. On receiving $\left(\mathsf{sent}, \mathsf{sid}, (K_i^0, K_i^1)\right)$ for all $i \in [n]$ from $\mathcal{F}_{\mathsf{COT}}$, P verifies if the garbled circuit $\mathbf{C}$, sent by the verifier earlier was correctly constructed.

i. if $\mathsf{Ve}\left(C, \mathbf{C}, \{K_i^0, K_i^1\}_{i\in[n']}\right) \neq 1$, P aborts.

ii. else P sends $(\textsc{Decommit}, \mathsf{sid})$ to $\mathcal{F}_{\mathsf{COM}}$.

Final verification phase: On receiving the message $(\textsc{Decommit}, \mathsf{sid}, Z')$ from $\mathcal{F}_{\mathsf{COM}}$, V outputs accept if $Z = Z'$, else output reject.

[a] Instead of returning d, Gb is tweaked to return the 1-key on the output wire.

Recalling Static Proof of Security. The simulator for a corrupt P^* invokes the algorithm of $\mathsf{P}^\star$ internally and simulates the role of honest verifier to $\mathsf{P}^\star$. It constructs and communicates a correct garbled circuit, extracts the witness acting on behalf of $\mathcal{F}_{\mathsf{COT}}$ functionality, and accepts the proof only if the extracted witness is a valid one. On the other hand the real verifier accepts when the opening of the commitment is the correct output wire key Z. In $\mathcal{F}_{\mathsf{COM}}$-hybrid model, we can show that a malicious prover who is able to make a real verifier output 'accept' (but not the simulator) can be used to break authenticity of the underlying garbling scheme. We can use such a malicious prover P^* to construct an adversary A for the authenticity game of [BHR12b] as follows:

1. A receives the invalid witness x^* from P^* on behalf of $\mathcal{F}_{\mathsf{COT}}$ and forwards it to the authenticity

challenger.

2. A receives $\mathbf{C}, X$ from the authenticity challenger and forwards it to P^*

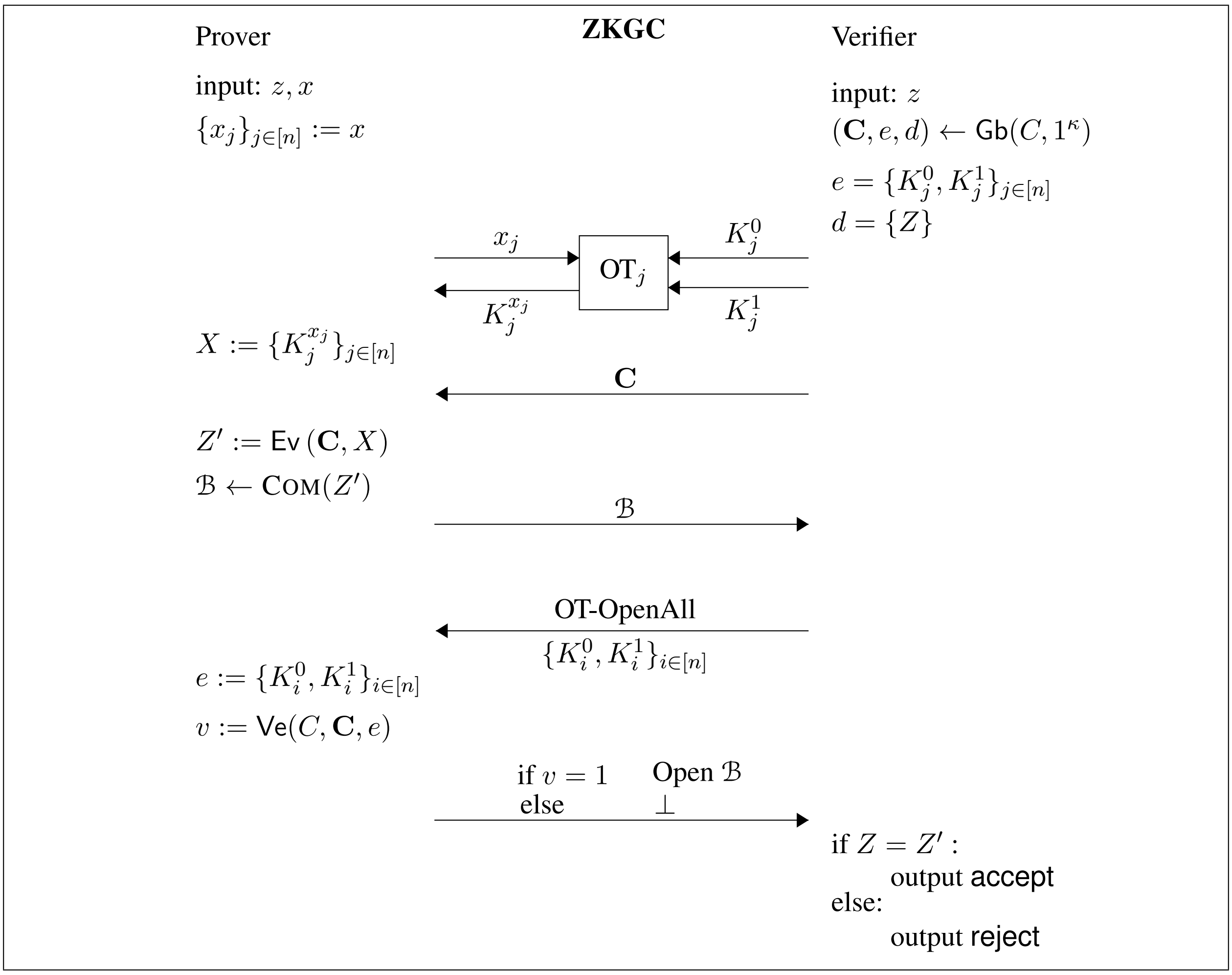
3. A receives forged key Z' from P^* on behalf of $\mathcal{F}_{\mathsf{COM}}$ and submits it to the authenticity challenger.

Clearly, the event that A successfully forges an output for the given $\mathbf{C}, X$ is equivalent to the event that P^* convinces a verifier to output 'accept' without a valid witness. By authenticity of the garbling scheme, this event occurs with negligible probability.

The simulator for a corrupt V^* receives the encoding information from V^* on behalf of the $\mathcal{F}_{\mathsf{COT}}$ functionality and extracts the the output 1-key Z using received garbled circuit and encoding information. It then sends Z to the verifier only after receiving the correct encoding information from V^* in the open-all phase. Otherwise, it sends $\perp$ to V^*. Security in this case follows from the verifiability (that allows extraction of the output key from encoding information) of the underlying garbling scheme.

Adaptive Proof of Security. The bottleneck faced in simulating garbled circuit based protocols for post-execution corruptions usually lies in "explaining" the randomness of the GC constructor once her input is known. In the case of two-party computation, equivocating the view of the garbled circuit constructor requires heavy machinery such as in Canetti et al. [CPV17]. However in the ZKGC protocol verifier V is the GC constructor and *has no input*. The simulator can therefore run the code of honest V, which includes being an honest sender in the OT protocol (this is also why our OT need not achieve full-fledged adaptive security). On the prover's side, receiver equivocality of the OT allows a simulator to equivocate an adaptively corrupted prover's view of the OT protocol, as per the witness once known. We make the observation that *every step of P following the OT is independent of the witness*. Specifically, once the output key Z has been obtained by evaluating the GC sent by V, P does not use the witness again. Note that the simulator does not need the witness to obtain Z; the ZKGC simulator invokes the π_{ot} simulator in order to extract all inputs of V and obtain all keys of the GC. Once the simulator obtains Z, the code of honest P can be run to complete the simulation. The implication of this for simulation of a post-execution corruption of P is that no additional work needs to be done besides equivocating the view of P in the OT. We now give a formal proof for all the cases:

Figure 3.3: ZKGC: Zero-knowledge from one GC [JKO13]


Prover
ZKGC
Verifier
input: z, x
$\{x_j\}_{j\in[n]} := x$
input: z
$(\mathbf{C}, e, d) \leftarrow \mathsf{Gb}(C, 1^\kappa)$
$e = \{K_j^0, K_j^1\}_{j\in[n]}$
$d = \{Z\}$
x_j
OT_j
K_j^0
$K_j^{x_j}$
K_j^1
$X := \{K_j^{x_j}\}_{j\in[n]}$
$\mathbf{C}$
$Z' := \mathsf{Ev}(\mathbf{C}, X)$
$\mathcal{B} \leftarrow \textsc{Com}(Z')$
$\mathcal{B}$
OT-OpenAll
$\{K_i^0, K_i^1\}_{i\in[n]}$
$e := \{K_i^0, K_i^1\}_{i\in[n]}$
$v := \mathsf{Ve}(C, \mathbf{C}, e)$
if $v = 1$ Open $\mathcal{B}$
else $\perp$
if $Z = Z'$:
output accept
else:
output reject

- **Simulation for V.** The verifier, until it is corrupted, can be simulated following the static simulator for the corrupt P, irrespective of when P is corrupted. As recalled above, the simulation can be carried out by running the code of honest verifier (constructing a correct garbled circuit, participating in the RE-OTs with the correct encoding information and sending the correctly constructed garbled circuit). Upon corruption, the simulator can explain to the corrupt V the communication by means of the randomness used in its honest execution of V's code. The indistinguishability follows from the proof in the static corrupt prover case.

- **Simulation for P.** If the prover is corrupted at the outset, then the crs is set in mode 0. Otherwise, we consider the worst scenario of post-execution corruption, and set the crs in mode 1. If the verifier is also not corrupt during the construction of the garbled circuit, then simulator acts on behalf of both the honest parties and runs the code of honest verifier. In the $\mathcal{F}_{\mathsf{COM}}$-hybrid model, the simulator, without having access to the actual witness, runs $\left(m^{\mathsf{R}}, r_0^{\mathsf{R}}, r_1^{\mathsf{R}}\right) \leftarrow \mathsf{Sim}^{\mathsf{RE}}\left(\mathsf{crs}, t\right)$

to generate the transcript that needs to be communicated on behalf of P in $\mathsf{RE\text{-}OT}$ instances. The rest of the simulation is straight-forward irrespective of whether the verifier is corrupt or not. In the final step, the simulator may have to communicate Z which it picked itself while simulating V in this case. When P is corrupt in the end, its input x_i to the i^{th} $\mathsf{RE\text{-}OT}$ instance can be explained as per any input using the randomness $r^{\mathsf{R}}_{x_i}$ returned by $\mathsf{Sim}^{\mathsf{RE}}$ of the $\mathsf{RE\text{-}OT}$s. On the other hand, if V was corrupt before the garbled circuit construction phase, then the simulator gets Z via unlocking the GC using encoding information extracted from the corrupt V's communication. The rest remains the same as the previous case. Security in the former case follows via receiver equivocality of $\mathsf{RE\text{-}OT}$. In the latter, it follows additionally from verifiability that ensures the encoding information leads to the correct Z with high probability.

3.5 Adaptively-Secure Zero Knowledge in Three Rounds

In this section, we present a *3-round ZK protocol* against a malicious verifier requiring just one GC in the non-programmable random oracle model, with *no increase in communication complexity*. Our protocol achieves this by a technique for non-interactive GC verification which allows us to remove the commitment and OT open-all phases from ZKGC. Our approach is reminiscent of the technique of *conditional disclosure of secrets* (CDS)[GIKM98]. CDS has since been generalized [IW14], and used in several works, including in applications to improve round complexity of protocols [AIR01, BCPW15]. We show that the protocol is adaptively secure when the underlying OTs are receiver equivocal.

3.5.1 High-Level Idea

The high round cost of ZKGC makes it undesirable for many applications. However its usage of only one GC for an actively secure protocol is an attractive feature, prompting us to examine whether we can improve on the number of rounds required to realize ZK with only one GC. We now describe our intuition behind the protocol, beginning with informal observations about the number of rounds in ZKGC. Assuming the ZKGC paradigm to be broadly characterized by a protocol where the verifier V constructs a GC which is then evaluated by prover P, we make the following (informal) observations:

1. As V constructs the GC, P's witness bits must be encoded as garbled input and delivered by means of an OT. The most efficient UC-secure OT in the literature [PVW08] requires 2 rounds to instantiate.
2. Assuming the underlying GC to be statically secure in the terminology of Bellare et al. [BHR12a], the GC can at best be sent to P along with the final message of the OT (if not after the OT).
3. P must communicate some information as a 'response' to V's GC 'challenge'; for instance the garbled output obtained as a result of evaluating the GC with her witness. This must necessarily

be after she receives the GC, adding at least one more round after the OT.

In summary, it appears that the ZKGC paradigm requires at least 2 rounds for the OT, plus the GC transmission, and one round following that. Therefore, a 3-round ZK protocol appears to be optimal in the ZKGC paradigm, informally suggesting the optimality of our protocol. In the following, we make several observations that are instrumental to our protocol.

Conditional Verification of Garbled Circuits. We begin by making the following observation about the original ZKGC protocol: even a prover who does not have a witness is given the chance to first commit to her garbled output and verify that the GC she received was correctly generated. Verification of the GC is a process that takes two additional rounds of interaction in their protocol. We ask, can we use conditional disclosure of secrets to reduce the number of rounds: *"can we provide some additional information with a GC that will allow an evaluator to non-interactively verify that the GC was correctly constructed only when it possesses a valid witness?"* We answer this question in the affirmative, at least for the ZKGC setting. An idea somewhat similar in spirit was proposed in [BP12] to construct a three-round 'weak' ZK protocol from a garbling scheme and point-obfuscation. That is, knowing the witness gives the prover access to a secret via a garbled circuit handed over by the verifier. The secret, then, can be used to unlock the seed that opens the garbled circuit and enables verifying the correct construction of the GC. Technique-wise, we depart from the work of [BP12] as follows. The secret is encoded in the circuit output in [BP12] and hence, privacy of the garbling circuit is one of the properties they rely on to achieve soundness. On the contrary, the secret, in our case is the output key corresponding to bit 1 and hence, soundness is achieved via authenticity. Qualitatively, their protocol is not a full-fledged ZK, is in the plain model, has a non-black-box simulator and relies on strong assumptions such as obfuscation. Our ZK protocol is proven UC-secure with a black-box simulator and relies on standard assumptions, albeit assuming a crs setup.

Interestingly, the intuition behind the ability of [JKO13] to achieve full black-box simulation was that the relaxation in round complexity rendered the four-round barrier in the plain model [GK96] inapplicable. However, our result demonstrates that the trusted setup required to implement a full black-box simulatable two-round OT is sufficient to construct a three round zero-knowledge argument using the concretely efficient [JKO13] technique and a non-programmable random oracle.

Our intuition is implemented as follows: Given $\left(\mathbf{C}, \left\{(K_j^0, K_j^1)\right\}_{j\in[n]}, (K^0, K^1)\right) \leftarrow \mathsf{Gb}\left(1^\kappa, C\right)$ and an honest P has obtained encoded input $X = \left(K_j^{x_j}\right)_{j\in[n]}$ for a witness $x = (x_1 \ldots, x_n)$, she can compute $Z = K^1 = \mathsf{Ev}\left(\mathbf{C}, X\right)$. Now that P has evaluated the GC, we wish to enable her to 'open' the GC and verify that it was constructed correctly. To do this, we provide her with a ciphertext encrypting some useful information. Concretely, the ciphertext $T = \mathcal{F}_{\mathsf{RO}}(\mathsf{sid}||K^1) \oplus r^{\mathsf{S}}$, where $\mathcal{F}_{\mathsf{RO}}$ is the random oracle functionality and r^{S} contains the randomness used by the sender in the OT instances. Once P

gets this randomness, she can unlock $\left\{K_j^0, K_j^1\right\}_{j\in[n]}$ and can verify if the circuit has been constructed correctly. In the following, we formalize the property needed from the OT protocol, namely that the randomness of the sender reveals the inputs of the sender.

Sender-Extractability of OT. Let $\pi_{\mathsf{ot}} = (\pi_{\mathsf{ot}}^{\mathsf{S}}, \pi_{\mathsf{ot}}^{\mathsf{R}})$ be a 2-round OT protocol securely implementing the $\mathcal{F}_{\mathsf{OT}}$ functionality in the crs model where S and R run their respective algorithm as specified by $\pi_{\mathsf{ot}}^{\mathsf{S}}$ and $\pi_{\mathsf{ot}}^{\mathsf{R}}$ respectively. Let crs be the string that both parties have access to. We denote the first message of the protocol sent by the receiver R by $m^{\mathsf{R}} = \pi_{\mathsf{ot}}^{\mathsf{R}}(\mathsf{crs}, \sigma; r^{\mathsf{R}})$ where σ is R's choice bit and r^{R} his randomness. Let the input of the sender S be (a_0, a_1); we denote the second message of the OT protocol, sent by S, by $m^{\mathsf{S}} = \pi_{\mathsf{ot}}^{\mathsf{S}}(\mathsf{crs}, (a_0, a_1), m^{\mathsf{R}}; r^{\mathsf{S}})$. The receiver can now compute the chosen message, $x_\sigma = \pi_{\mathsf{ot}}^{\mathsf{R}}(\mathsf{crs}, \sigma, m_{\mathsf{S}}; r^{\mathsf{R}})$. We assume that π_{ot} has the following sender-extractable property: revealing the randomness of the sender, allows the receiver to reconstruct the sender's messages correctly with high probability. That is, there exists a public efficiently computable function, Ext such that $\mathsf{Ext}(\mathsf{crs}, \mathcal{T}_{\mathsf{OT}}((a_0, a_1), \sigma), r^{\mathsf{S}})$ outputs (a_0, a_1) where $\mathcal{T}_{\mathsf{OT}}((a_0, a_1), \sigma)$ refers to the transcript of π_{ot} with sender's input as (a_0, a_1) and receiver's input as σ. Namely, $\mathcal{T}_{\mathsf{OT}}((a_0, a_1), \sigma) = (m^{\mathsf{R}}, m^{\mathsf{S}})$ where m^{R} and m^{S} are as defined above.

Definition 3.5.1. *A protocol π_{ot} is a secure sender-extractable OT protocol if*

- *it securely implements $\mathcal{F}_{OT}$ in the presence of malicious adversaries, and*
- $\forall$ $(a_0, a_1), \sigma$, *such that* $|a_0|, |a_1| \le$ *poly* (κ), $\sigma \in \{0, 1\}$, $\exists$ *a PPT algorithm* Ext *such that the following probability is negligible in κ.*

$$\Pr\left(a_0' \neq a_0 \cup a_1' \neq a_1 : \mathsf{Ext}(crs, \mathcal{T}_{\mathsf{OT}}((a_0, a_1), \sigma), r^{\mathsf{S}}) = (a_0', a_1')\right).$$

We note that the protocol of [PVW08] is UC-secure in the CRS model, is 2-rounds, and satisfies the sender-extractability property of Definition 3.5.1. We use such a protocol in our construction.

3.5.2 Our Construction

At a high-level, our construction proceeds as follows. The verifier constructs a garbled circuit of the circuit C implementing the relation. The prover obtains the wire keys corresponding to his witness via an OT protocol. Now, the verifier sends the garbled circuit to the prover, and, in addition, a ciphertext. This ciphertext allows the prover to open and verify the garbled circuit, but only if he possesses a valid witness. The complete description of our protocol π_{ZK3} is presented in Figure 3.4. We now prove security of π_{ZK3} in Universal Composability (UC) framework. First, we prove static security of our protocol and then we discuss the adaptive proof.

π_{ZK3}

- **Oracles and Cryptographic Primitives:** A *correct, authentic, verifiable* garbling scheme Garble $=$ $(\mathsf{Gb}, \mathsf{En}, \mathsf{De}, \mathsf{Ev}, \mathsf{Ve})$. A sender-extractable 2-round OT π_{ot} with the common reference string crs. Random Oracle $\mathcal{F}_{\mathsf{RO}} : \{0,1\}^* \rightarrow \{0,1\}^{\mathsf{poly}(\kappa)}$.
- **Common Inputs of P and V:** A security parameter κ, relation R realized by circuit C, statement z, common reference string crs for π_{ot}.
- **Input of P:** A witness x of size $n = \mathsf{poly}(\kappa)$ such that $R(z, x) = 1$.
- **Input of V:** Nothing.

OT First Message Phase:
P plays the role of the receiver R in n instances of π_{ot} and provides his witness bit x_j as input to the jth instance of π_{ot}. Specifically, it:

- Chooses $r_j^{\mathsf{R}} \xleftarrow{R} \{0,1\}^\kappa$, and computes $m_j^{\mathsf{R}} = \pi_{\mathsf{ot}}^{\mathsf{R}}(\mathsf{crs}, x_j; r_j^{\mathsf{R}}), \forall j \in [n]$ as the first message in the jth instance of π_{ot}
- Sends $\{m_j^{\mathsf{R}}\}_{j \in [n]}$ to V.

GC Construction and OT Second Message Phase:
V constructs a garbled circuit $\mathbf{C}$ for C as $(\mathbf{C}, \{K_j^0, K_j^1\}_{j\in[n]}, (K^0, K^1)) \leftarrow \mathsf{Gb}(1^\kappa, C)$. V now provides the wire labels for the input wires of $\mathbf{C}$ by playing the role of the sender S in n instances of π_{ot}. Specifically, it

- Samples randomness $r_j^{\mathsf{S}} \xleftarrow{R} \{0,1\}^\kappa$, $\forall j \in [n]$ and parses $r^{\mathsf{S}} = r_1^{\mathsf{S}} || \cdots || r_n^{\mathsf{S}}$
- Computes $m_j^{\mathsf{S}} = \pi_{\mathsf{ot}}^{\mathsf{S}}(\mathsf{crs}, K_j^0, K_j^1, m_j^{\mathsf{R}}; r_j^{\mathsf{S}}), \forall j \in [n]$ and $T = \mathcal{F}_{\mathsf{RO}}\left(\mathsf{sid}||K^1\right) \oplus r^{\mathsf{S}}$ and
- Sends $(\mathbf{C}, \{m_j^{\mathsf{S}}\}_{j\in[n]}, T)$ to P.

P computes the wire-keys corresponding to his input: $K_j^{x_j} = \pi_{\mathsf{ot}}^{\mathsf{R}}(\mathsf{crs}, m_j^{\mathsf{R}}, m_j^{\mathsf{S}}, r_j^{\mathsf{R}}), \forall j \in [n]$.

GC Evaluation, Verification and Output Disclosure Phase: P evaluates $\mathbf{C}$ and obtains the garbled output. He then recovers the randomness used by the sender (namely, V) using the output-wire key he obtained. By the sender-extractability of π_{ot}, P recovers the input-wire labels which are the OT inputs of V. P can now verify that the garbled circuit was correctly constructed using the recovered wire keys. Specifically, it:

- Executes $\mathbf{Y} = \mathsf{Ev}(\mathbf{C}, \{K_j^{x_j}\}_{j\in[n]})$
- Recovers $r^{\mathsf{S}} = \mathcal{F}_{\mathsf{RO}}\left(\mathsf{sid}||\mathbf{Y}\right) \oplus T$, and parses $r^{\mathsf{S}} = r_1^{\mathsf{S}} || \cdots || r_n^{\mathsf{S}}$
- Aborts if $\exists j$ such that $\mathsf{Ext}(\mathsf{crs}, m_j^{\mathsf{R}}, m_j^{\mathsf{S}}, r_j^{\mathsf{S}}) = \perp$. Else, extracts $(K_j^0, K_j^1) = \mathsf{Ext}(\mathsf{crs}, m_j^{\mathsf{R}}, m_j^{\mathsf{S}}, r_j^{\mathsf{S}}), \forall j \in [n]$ otherwise
- Aborts if $\mathsf{Ve}(C, \mathbf{C}, \{K_j^0, K_j^1\}_{j\in[n]}) = 0$. Else, sends $\mathbf{Y}$ to V otherwise.

Output Phase:
If $\mathbf{Y} = K^1$, then V outputs accept, else he outputs reject.

Figure 3.4: 3-round GC based Zero Knowledge protocol

Theorem 3.5.2. *Let* Garble *be a correct, authentic, verifiable garbling scheme, π_{ot} be a sender-extractable OT protocol, and $\mathcal{F}_{RO}$ be an extractable random oracle. The protocol π_{ZK3} in Figure 3.4 securely implements $\mathcal{F}_{ZK}^{R}$ in the presence of static malicious adversaries.*

Proof. To prove the static security of our protocol, we describe two simulators.

Security against a Corrupt Prover $\mathsf{P}^\star$. The simulator Sim_P simulates the view of a corrupt prover and appears in Fig. 3.5. We now prove that $\text{IDEAL}_{\mathcal{F}^R_{\mathsf{ZK}},\mathsf{Sim}_\mathsf{P},\mathcal{Z}} \overset{c}{\approx} \text{REAL}_{\pi_{\mathsf{ZK3}},\mathsf{A},\mathcal{Z}}$ when A corrupts $\mathsf{P}^\star$. We begin by noting that the simulated and the real worlds are identical when $\mathsf{P}^\star$ uses a valid witness x. The view of a malicious $\mathsf{P}^\star$ who does not possess a valid witness x is proven to be computationally close to the simulation through an intermediate hybrid HYB_1. The hybrid HYB_1 is constructed identically to $\text{IDEAL}_{\mathcal{F}^R_{\mathsf{ZK}},\mathsf{Sim}_\mathsf{P},\mathcal{Z}}$ with the exception of the criterion to output accept. In HYB_1, the verifier accepts if $\mathsf{P}^\star$ outputs the correct K^1 (as in the REAL view) regardless of the witness used. We begin our analysis by noting that unless a $\mathsf{P}^\star$ queries the correct K^1 to the random oracle $\mathcal{F}_{\mathsf{RO}}$, the string T appears completely random. Therefore, given that a $\mathsf{P}^\star$ attempting to distinguish between the REAL view and the view generated by HYB_1, we branch our analysis into the following cases:

Figure 3.5: Simulator Sim_P against corrupt P^*

Simulator Sim_P

The simulator plays the role of the honest V and simulates each step of the protocol π_{ZK3} as follows. The communication of the $\mathcal{Z}$ with the adversary A who corrupts P^* is handled as follows: Every input value received by the simulator from $\mathcal{Z}$ is written on A's input tape. Likewise, every output value written by A on its output tape is copied to the simulator's output tape (to be read by the environment $\mathcal{Z}$).

OT First Message Phase: Sim_P invokes the simulator of π_{ot} for corrupt receiver and extracts P^*'s input bit to the jth instance of π_{ot}, namely the jth witness bit x_j.

GC Construction and OT Second Message Phase: Sim_P emulates an honest V if the extracted witness x is valid i.e. $R(z,x) = 1$. Otherwise, Sim_P does the following:

- It constructs a garbled circuit $\mathbf{C}$ for C as $(\mathbf{C}, \{(K^0_j, K^1_j)\}_{j\in[n]}, (K^0, K^1)) \leftarrow \mathsf{Gb}(1^\kappa, C)$.
- It samples r^S uniformly at random and parses it as $r^\mathsf{S} = r^\mathsf{S}_1 || \cdots || r^\mathsf{S}_n$,
- It computes $m^\mathsf{S}_j = \pi^\mathsf{S}_{\mathsf{ot}}(\mathsf{crs}, K^{x_j}_j, 0^\kappa, m^\mathsf{R}_j; r^\mathsf{S}_j)$, if $x_j = 0$ else it computes $m^\mathsf{S}_j = \pi^\mathsf{S}_{\mathsf{ot}}(\mathsf{crs}, 0^\kappa, K^{x_j}_j, m^\mathsf{R}_j; r^\mathsf{S}_j), \forall j \in [n]$.
- It samples T uniformly at random.
- It sends $(\mathbf{C}, \{m^\mathsf{S}_j\}_{j\in[n]}, T)$ to $\mathsf{P}^\star$.

GC Evaluation, Verification and Output Disclosure Phase: Sim_P does nothing in this step.

Output Phase: Sim_P sends x to $\mathcal{F}^R_{\mathsf{ZK}}$ on behalf of $\mathsf{P}^\star$ if $R(z,x) = 1$. Otherwise, it sends $\perp$.

- **Case 1: $\mathsf{P}^\star$ does not output the correct K^1 in either world.** Here we assume that a $\mathsf{P}^\star$ also does not query the correct K^1 to $\mathcal{F}_{\mathsf{RO}}$ to be able to unlock ciphertext T. If the prover does indeed query the correct K^1 to $\mathcal{F}_{\mathsf{RO}}$ with non-negligible probability, we move on to the next case. A $\mathsf{P}^\star$ who is successful in distinguishing $\text{REAL}_{\pi_{\mathsf{ZK3}},\mathsf{A},\mathcal{Z}}$ from HYB_1 in this case can be used to break OT sender security. The reduction computes a garbled circuit $\mathbf{C}$ and sends the input keys to the OT challenger (by means of the environment for the OTs) as the sender's input. The reduction then extracts the input x of $\mathsf{P}^\star$ and forwards to the OT challenger as the choice bits of the receiver. The response of OT challenger who computes the sender's message either by invoking a real sender i.e. as $m_j^{\mathsf{S}} = \pi_{\mathsf{ot}}^{\mathsf{S}}(\mathsf{crs}, K_j^0, K_j^1, m_j^{\mathsf{R}}; r_j^{\mathsf{S}}), \forall j \in [n]$ or by invoking a simulator i.e. as $m_j^{\mathsf{S}} = \pi_{\mathsf{ot}}^{\mathsf{S}}(\mathsf{crs}, K_j^{x_j}, 0^\kappa, m_j^{\mathsf{R}}; r_j^{\mathsf{S}}), \forall j \in [n]$ is sent to the reduction who further forwards the message to $\mathsf{P}^\star$ along with $\mathbf{C}$ and a random T. In case the OT challenger invokes a simulator the view of $\mathsf{P}^\star$ is identical to HYB_1, whereas when the OT challenger uses a real execution of π_{ot} the view of $\mathsf{P}^\star$ is identical to REAL (T is random given that the correct K^1 is never queried to $\mathcal{F}_{\mathsf{RO}}$). Therefore, the probability of distinguishing between the REAL and HYB_1 view translates to the probability of distinguishing between the real and the simulated view of the OT protocols for the case when the receiver is corrupt.

- **Case 2: $\mathsf{P}^\star$ outputs the correct K^1 in $\textsc{real}_{\pi_{\mathsf{ZK3}},\mathsf{A},\mathcal{Z}}$ with significantly higher probability than in HYB_1.** This case is similar to the previous case in that $\mathsf{P}^\star$ can be used to break sender security of the OT by computing $\mathbf{C}$ locally in the reduction. If $\mathsf{P}^\star$ outputs a correct K^1, the reduction is interacting with π_{ot} whereas if not, the challenger must have invoked the simulator for π_{ot}. The advantage of this reduction is the difference in probabilities with which $\mathsf{P}^\star$ forges K^1 successfully in the REAL and HYB_1 worlds.
- **Case 3: $\mathsf{P}^\star$ outputs the correct K^1 in both worlds with almost the same probability.** The corrupt $\mathsf{P}^\star$ can be used directly to break authenticity of the garbling scheme. Clearly the OT message corresponding to inactive input keys are not used by the corrupt P; the ability to output the correct K^1 must be derivative of the ability to forge a key for the garbled circuit alone. It is therefore straightforward to use $\mathsf{P}^\star$ to forge K^1 for a given garbled circuit $\mathbf{C}$, as its view can be generated as per HYB_1, which does not require the inactive garbled circuit keys to compute the OT messages.

Note that in Cases 2 and 3, we consider a $\mathsf{P}^\star$ who outputs K^1 to be equivalent to a $\mathsf{P}^\star$ who queries the random oracle on K^1 to unlock T in its effort to distinguish REAL from HYB_1. Instead of receiving K^1 directly from $\mathsf{P}^\star$, our reductions will observe its query to the random oracle. Note that our simulation does not rely on the observability property of $\mathcal{F}_{\mathsf{RO}}$ rather the reduction for indistinguishability between hybrids require the observability property. Hence, we can claim that our protocol requires only non-programmable non-observable random oracle whereas the proof requires observability.

Finally $\textsc{ideal}_{\mathcal{F}_{\mathsf{ZK}}^R,\mathsf{Sim}_\mathsf{P},\mathcal{Z}}$ deviates from HYB_1 only in its criteria to output accept. Only a corrupt $\mathsf{P}^\star$ who is able to output K^1 will be able to distinguish HYB_1 from $\textsc{ideal}_{\mathcal{F}_{\mathsf{ZK}}^R,\mathsf{Sim}_\mathsf{P},\mathcal{Z}}$. Such a $\mathsf{P}^\star$ can be used directly to forge an output key for a given $\mathbf{C}$ with the same probability (which by authenticity of the garbling scheme, must be negligible).

Security against a Corrupt Verifier $\mathsf{V}^{\star}$. The simulator $\mathsf{Sim}_{\mathsf{V}}$ simulates the view of a corrupt verifier and is presented in Fig. 3.6. We now argue that the ideal and real world views are indistinguishable by proving $\textsc{ideal}_{\mathcal{F}_{\mathsf{ZK}}^{R},\mathsf{Sim}_{\mathsf{V}},\mathcal{Z}} \overset{c}{\approx} \textsc{real}_{\pi_{\mathsf{ZK3}},\mathsf{A},\mathcal{Z}}$ when A corrupts V. The above two views of $\mathcal{Z}$ are shown to be indistinguishable via a series of intermediate hybrids.

Figure 3.6: Simulator $\mathsf{Sim}_{\mathsf{V}}$ against corrupt V^{*}

Simulator $\mathsf{Sim}_{\mathsf{V}}$

The simulator plays the role of the honest P and simulates each step of the protocol π_{ZK3} as follows. The communication of the $\mathcal{Z}$ with the adversary A who corrupts V^{*} is handled as follows: Every input value received by the simulator from $\mathcal{Z}$ is written on A's input tape. Likewise, every output value written by A on its output tape is copied to the simulator's output tape (to be read by the environment $\mathcal{Z}$).

OT First Message Phase: $\mathsf{Sim}_{\mathsf{V}}$ invokes the simulator of π_{ot} for corrupt receiver to simulate the first OT message.

GC Construction and OT Second Message Phase: $\mathsf{Sim}_{\mathsf{V}}$ uses the OT simulator to extract V's inputs to the jth instance of π_{ot}, namely (K_j^0, K_j^1).

GC Evaluation, Verification and Output Disclosure Phase: On receiving the garbled circuit $\mathbf{C}$ and T from V^{*}, $\mathsf{Sim}_{\mathsf{V}}$ runs $\mathsf{Ve}(C, \mathbf{C}, \{K_j^0, K_j^1\}_{j\in[n]})$. It aborts if the output of Ve is 0. Else, it sends K^1 to V^{*} where $K^1 \leftarrow \mathsf{Ve}\left(\mathbf{C}, e, 1\right)$.

Output Phase: It does nothing in this step.

1. $\mathbf{HYB}_0$: Same as $\textsc{real}_{\pi_{\mathsf{ZK3}},\mathsf{A},\mathcal{Z}}$.
2. $\mathbf{HYB}_1$: Same as HYB_0, except that **OT First Message phase** is emulated by invoking the simulator of π_{ot} for corrupt receiver.
3. $\mathbf{HYB}_2$: Same as HYB_1, except that K^1 is computed in the following way instead of running $\mathsf{Ev}(\mathbf{C}, X)$. The simulator of π_{ot} for corrupt receiver is used to extract (K_j^0, K_j^1) for $j \in [n]$. Then $\mathsf{Ve}(C, \mathbf{C}, \{K_j^0, K_j^1\}_{j\in[n]})$ is run. If the output is 0, the prover aborts. Otherwise $\mathsf{Ve}\left(\mathbf{C}, e, 1\right)$ is run to extract K^1 and the prover runs the rest of the protocol using K^1.
4. $\mathbf{HYB}_3$: Same as HYB_2, except that the following check for abort in **GC Evaluation, Verification and Output Disclosure Phase** is removed: On computing $r_1^{\mathsf{S}}||\cdots||r_n^{\mathsf{S}} = r^{\mathsf{S}} = T \oplus \mathcal{F}_{\mathsf{RO}}\left(\mathsf{sid}||K^1\right)$, the prover aborts if any call to the extractor Ext of the sender's input to OT returns $\perp$.

Clearly, $\mathsf{HYB}_3 = \textsc{ideal}_{\mathcal{F}_{\mathsf{ZK}}^{R},\mathsf{Sim}_{\mathsf{V}},\mathcal{Z}}$. Our proof will conclude, as we show that every two consecutive hybrids are computationally indistinguishable.

$\mathbf{HYB}_0 \overset{c}{\approx} \mathbf{HYB}_1$: The difference between these hybrids lies in the way OT first message is generated. In HYB_0, the message is generated by a real receiver that possesses the choice bits x, whereas in HYB_1, the simulator for π_{ot} for the corrupt receiver generates the message. The indistinguishability follows via reduction to the sender security of n instances of OT.

$\mathbf{HYB}_1 \overset{c}{\approx} \mathbf{HYB}_2$: The difference between these hybrids lies in the way K^1 is computed. In HYB_1, K^1 is computed as a real prover does. On the other hand, K^1 is extracted using Ve and the encoding information extracted from the OTs in HYB_2. By the verifiability property of the garbling scheme, the view of $\mathsf{V}^\star$ in HYB_2 and HYB_1 are indistinguishable.

$\mathbf{HYB}_2 \overset{c}{\approx} \mathbf{HYB}_3$: The difference between these hybrids lies in the conditions checked by P for abort in **GC Evaluation, Verification and Output Disclosure Phase**. In the former, the protocol is aborted when one of the invocations to Ext returns messages different from corresponding input labels which does not happen in the latter as the check is removed. By the sender extractability of the OT protocol (Definition 3.5.1), the hybrids are indistinguishable except with negligible probability.

□

3.5.3 Making π_{ZK3} Adaptively Secure

The challenge in achieving adaptive security for π_{ZK3} is essentially the same as ZKGC; once the GC output key Z has been retrieved, all of P's steps are independent of the witness.

Simulation for P. Consider the worst case scenario of post-execution corruption. The simulator runs $\left(m^{\mathsf{R}}, r_0^{\mathsf{R}}, r_1^{\mathsf{R}}\right) \leftarrow \mathsf{Sim}^{\mathsf{RE}}\left(\mathsf{crs}, t\right)$ to generate the first message of P, and obtains the GC output key Z either by extracting the encoding information from V's response (if V is corrupt) or using the key it picked itself when simulating V. The rest of the simulation is straightforward, as the code of honest P can be run from this point. In case the adversary chooses to corrupt P, the simulator hands over the randomness $r_{x_i}^{\mathsf{R}}$ for each OT instance encoding witness bit x_i.

Simulation for V. As V has no input, the simulator proceeds by running the code of the honest verifier, with the only difference being that it accepts a proof by checking whether P has input a valid witness in the OT. A malicious P can distinguish between the real protocol and the simulation only by forging Z, for which there is no advantage afforded by adaptive corruptions; a dishonest P who is successful in this setting can be used to break authenticity of the garbling scheme just as in the static case.

This concludes the adaptive security proof of our 3 round ZK protocol. The proof has been summarized in Thm. 3.5.3.

Theorem 3.5.3. *Let* Garble *be a correct, authentic, verifiable garbling scheme,* π_{ot} *be an sender-extractable OT protocol, and* $\mathcal{F}_{\mathsf{RO}}$ *be an observable random oracle. The protocol* π_{ZK3} *in Figure 3.4 securely implements* $\mathcal{F}_{\mathsf{ZK}}^{R}$ *in the presence of adaptive malicious adversaries.*

3.5.4 Reducing the Random Oracle Assumption

We have proved adaptive security of our protocol in Thm. 3.5.3 by relying on the observability property of the random oracle. Our proof for a corrupt $\mathsf{P}^{\star}$ case relied on the observability property, where $\mathsf{P}^{\star}$ forges the correct K^1 in both worlds without having the correct witness. In such a case, the reduction uses the forged K^1 key to break the authenticity of the garbling scheme. However, if we assume that the garbling scheme is decisionally authentic then we can use a non-observable random oracle for the reduction instead of the observable random oracle. A decisionally authentic garbling scheme is one where the wire label for an output wire, looks random if it does not correspond to the actual output bit on that wire. The decisional authenticity game is a decisional one and an adversary participating in the game is not required to compute the other output wire label. And hence in our proof, the reduction plays the role of the adversary in the decisional authenticity game. If a corrupt prover $\mathsf{P}^{\star}$ can distinguish between two hybrids, one in which T is correctly formed and another in which T is completely random, then $\mathsf{P}^{\star}$ can be used to break the decisional authenticity property of the garbling scheme. Note that, we need $\mathcal{F}_{\mathsf{RO}}$ to behave as a plain RO and not an observable one as the reduction is required to distinguish the inactive output wire label from random string, and not compute the inactive wire label. This reduces our assumption, for the reduction, to plain RO instead of observable RO.

Chapter 4

Adaptively Secure Oblivious Transfer and its Extension

In the literature, oblivious transfer has been regarded as the fundamental primitive [Rab81, BCR86, Kil88, NP05, IPS08], known to be complete for MPC [Kil88, GMW87, GV87, Yao86]. It has been widely used in various applications, ranging from two party computation [MR17, WMK17, LR15, Lin13, FJN$^+$13, NO09, IPS08], private set intersection [RR17, PSSZ15, PSZ14, DCW13], zero knowledge [JKO13, GKPS18] and even multiparty computation [KOS16, LOS14, GMW87]. Following this, many flavors of OT such as 1-out-of-2 OT [PVW08, CO15], 1-out-of-N OT [NP05], k-out-of-N OT [GH08], have evolved in past. In its most basic form, a 1-out-of-2 OT consists of two parties, sender S and a receiver R. S has input messages say m_0, m_1 and R has a choice bit b. At the end of the protocol, R obtains message m_b corresponding to his choice bit and nothing else. S remains oblivious to the message obtained by R. Various constructions of OT (mainly 1-out-of-2) have been proposed providing both static and adaptive security. The protocols of [CO15, PVW08, NP05] deal with malicious adversaries in the static model whereas those of [BCG17, BC16, CKWZ13, GWZ09] address the same in adaptive model. However, designing adaptively-secure protocols has been a challenging task. The OT protocols that achieve adaptive security lack optimality in terms of efficiency-rounds, computation and communication. We have round optimal, i.e. 2 rounds, and efficient protocols in the literature for the static case whereas there is no such round-optimal protocol in the adaptive setting. In this direction, we present the first round optimal OT protocol which is secure against adaptive adversaries without erasures.

Another interesting direction to consider for OTs is to reduce the number of public key operations. The impossibility result by [IR89] states that its highly unlikely that OTs can be constructed without public key operations. In order to circumvent this limitation, the concept of OT Extension [Bea96b, IKNP03, ALSZ13, ALSZ15, KOS15] was introduced and explored. It allows the parties to execute a small number of OTs, called seed OTs, and then extend them to obtain large number of OTs using cheap symmetric key operations. The amortized cost of generating one single OT reduces to a constant number of symmetric key operations. However, there is no known OT extension protocol in the adaptive setting. An effort towards obtaining adaptively secure OT Extension would be of quite interest as it would open the gates towards constructing large number of efficient adaptively secure OTs using small number of seed OTs. Our paper presents one such result for adaptively-secure OT extension.

4.1 Related Work

In this section we outline the relevant literature of OT and OT extension schemes.

Oblivious Transfer. The literature of OT is vast and quite diverse in terms of assumptions and security. We highlight few works that are closely related to ours. Firstly, in the standalone model, the works of [NP01, AIR01, HK12, Lin08] are statically-secure against malicious adversary. Secondly, in the UC model, [CLOS02] proposed the first UC secure OT protocol based on general assumptions. Their work includes construction of both static and adaptively UC-secure OTs using Cook-Levin reductions. Despite being of theoretical interest, [CLOS02] motivated research towards obtaining OTs in the UC model. In the setting of static security, [GMY04] presented a constant round committed bit-OT under Decisional Diffie Hellman (DDH) and RSA assumptions. The work of [JS07] proposed a four round, UC-secure protocol under the Decisional Composite Residue (DCR) assumption. [HK07] provided the first round-optimal protocol that is UC-secure, assuming common reference string (crs). This was followed by the seminal work of Peikert *et al.* [PVW08], that provided a general framework for round optimal UC-secure OT protocols along with efficient instantiations based on DDH, Quadratic Residue and Learning With Errors (LWE) assumptions in the crs model. We denote the popular DDH based construction of the [PVW08] paper as PVW protocol in rest of the paper.

In the setting of adaptive security, the literature can be divided based on the erasure model. The works of [BCG17, BC16, BC15, ABB$^+$13, CKWZ13] rely on secure erasures of the memory, whereas [BDD$^+$17, GWZ09, CDMW09a, CDMW09b] consider the stronger model of no erasures. [CKWZ13] presented a framework for adaptively-secure OT in the global crs model. They provide instantiations under various assumptions- DLIN, Symmetric External Diffie Hellman (SXDH), DDH and DCR. However, their protocols are not round optimal and achieve adaptive security at the cost of significant overhead in communication and computation compared to the PVW protocol. [CKWZ13] also provided two constructions (Appendix A of [CKWZ13]) of [GWZ09] framework under Decisional Linear (DLIN) assumptions. These instantiations are adaptively-secure with erasures, with computational overhead reduced to constant number of exponentiations. There has been a separate line of work [BCG17, BC16, BC15, ABB+13] based on password-authenticated key exchange (PAKE) and smooth projective hash functions. They require atleast 3 rounds of communication, assuming erasures for adaptivity. On the other hand, achieving adaptivity with no erasures is a challenging task. [GWZ09] followed the compiler approach to transform the [PVW08] framework into an adaptively-secure OT using adaptively secure commitments. They transformed the [PVW08] framework into a actively secure semi-adaptive one by adaptively generating the crs within the protocol. The crs is generated by running an adaptively secure coin-tossing subprotocol using an adaptively-secure commitment scheme. Then they compile their semi-adaptive OT into an actively secure adaptive OT by using a primitive called somewhat non-committing encryption (NCE). They proceed to show that this transformation can be performed in constant number of rounds and it would incur an overhead of $\mathcal{O}(n)$ exponentiations, where n is the size of sender input messages. It was further improved by a recent work of [ABP17] though the computation overhead still continued to be $\mathcal{O}(n)$ exponentiations. Concurrently, [CDMW09a, CDMW09b] proposed theoretical constructions of adaptively secure OT. [CDMW09b] presented a compiler for transforming a protocol that is secure against a semi-honest adaptive adversary into one that is secure against a malicious adaptive adversary. The popular technique of cut-and-choose [Lin13, LP11, LP07] is applied on the semi-honest OT protocol and their adaptively secure OT protocol uses trapdoor simulatable public key encryption and blackbox access to semi-honest OT. Their scheme involves $\mathcal{O}(\kappa^2 n)$ copies of the underlying semi-honest OT protocol to achieve the transformation. On the other hand, [CDMW09a] presented an optimized NCE scheme based on trapdoor simulatable cryptosystem. Their work uses the NCE scheme to transform a semi-honest adaptively secure protocol to one secure against active sender. Then the compiler transformation of [CDMW09b] is applied to attain security against active receiver thereby obtaining a 6 round 1-out-of-N OT protocol. Both protocols require atleast $\mathcal{O}(n)$ exponentiations for a 1-out-of-2 OT on n-bit message.

The work of [CO15], the “simplest OT” protocol explored 3-round OT constructions in the PRO model. Although the paper claimed adaptive security, several bugs have been identified [GIR17, BDD+17, HL17] recently in their static security proof thereby rendering the protocol of [CO15] insecure in the UC model. Recently two more works ([HL17],[BDD+17][1]) have claimed to achieve adaptive security in the same model. However, in Section 4.3, we give a justification that these protocols are not UC-secure. The authors of [BDD+17] have updated their protocol and their new version presents a 3 round OT framework which can be instantiated under Learning from Parity with Noise, McEliece cryptosystem, QC-MDPC, LWE and Computational Diffie Hellman (CDH) in the PRO model. Their most efficient instantiation (under CDH) incurs twice the amount of communication,

[1] previous version of their paper

while maintaining the same computation cost as ours. Consequently, the problem of attaining an *round-optimal, efficient and adaptively-secure OT* continued to remain open, which we try to address through our work. Table 4.1 summarizes the literature on adaptively-secure OT protocols and our result. We do not compare with [GWZ09, CDMW09a, CDMW09b, ABP17] in the table since they require atleast $\mathcal{O}(n)$ exponentiations, where sender's input message is of n bits, whereas the other protocols in the table require constant number of exponentiations. Among the PAKE-based schemes we compare with the most efficient works of [ABB$^+$13, BCG17].

Oblivious Transfer Extension. Next, we consider the problem of OT Extension, which was introduced by the work of [Bea96b], followed by the seminal work of [IKNP03]. The paper of [IKNP03] presented an efficient 1-out-of-2 semi-honest OT Extension protocol which was secure against static adversaries. An optimized version of this protocol appeared in [ALSZ13]. The paper of [ALSZ15, KOS15] presented the actively secure versions. The paper of [KK13] gave constructions for 1-out-of-N case, which were made actively secure by [PSS17, OOS17]. However, all of these protocols are in the static setting and it was not known *whether adaptively-secure OT Extension protocol is possible.* Our work answers it in an affirmative way by proving that existing static OT extension [ALSZ13, ALSZ15] schemes satisfy adaptive security under the PRO assumption, while preserving the same efficiency.

Table 4.1: Comparison among UC secure Oblivious Transfer Protocols

Protocol	Communication (κ-bit strings / Group elements)	Computation				Rounds	Assumptions	Setup	Security
		Sender		Receiver					
		SKE	PKE	SKE	PKE				
[GWZ09] + [FLM11][1]	83	3	26	3	72	4	DLIN	crs	Adaptive with erasures
[CKWZ13]	59	3	$\geq$14	2	$\geq$27	3	DLIN	crs	Adaptive with erasures
[CKWZ13]	43	3	$\geq$8	2	$\geq$15	3	SXDH	crs	Adaptive with erasures
[CKWZ13]	35	4	19	3	37	4	DDH	crs	Adaptive with erasures
[CKWZ13]	28	4	13	3	26	4	DCR	crs	Adaptive with erasures
[ABB$^+$13]	15	2	13	1	11	3	SXDH	crs	Adaptive with erasures
[BCG17]	10	4	18	4	9	3	SXDH	crs	Adaptive with erasures
PVW	6	-	8	-	3	2	DDH	crs	Static
[BDD$^+$17]	15	5	6	2	5	3	CDH	PRO	Adaptive (GUC model)
Our scheme	7	3	8	2	3	2	DDH	PRO	Adaptive (GUC model)
Our scheme (after OT Extension)	3	2	-	2	-	3	Static Receiver Equivocal OT	PRO	Adaptive (GUC model)

Notations:
SKE - symmetric key encryptions, PKE - exponentiations, GUC - Generalized UC ,
DDH - Decisional Diffie Hellman, DLIN - Decisional Linear, SXDH - symmetric external Diffie Hellman,
CDH - Computational Diffie Hellman, DCR - Decisional Composite Residuosity, PRO - programmable random oracle
1 The commitment scheme used for instantiation is of [FLM11].

4.2 Our Results

We initiate our discussion by demonstrating an attack in concurrent adaptive OT papers [BDD$^+$17, HL17]. Next, we present our schemes which focus on optimizing the round complexity while attaining adaptive security in an efficient manner for OT and its extension. We also present an adaptively secure well-defined transformation from $\log \mathsf{N}$ 1-out-of-2 OTs to 1-out-of-N OT, restricting the number of exponentiations to $\mathcal{O}(\log \mathsf{N})$. Our contributions are briefly stated below.

Adaptively Secure 1-out-of-2 Oblivious Transfer. We construct the first OT framework that is round-optimal and adaptively-secure assuming no erasures. Our construction is motivated by the vital observation of [CO15] that the crs in OT can be replaced with the PRO. We apply the same observation on the static OT framework of [PVW08]. At the heart of their framework lies the Dual Mode Encryption Scheme (DME) which requires a crs for its functioning. We generate the crs of the Dual Mode Encryption (DME) scheme using the PRO. During simulation the crs can be suitably modified, to equivocate R's view, by programming the PRO. However, for our scheme it should be possible to generate the crs of the DME using an RO. Hence, we customize the definition of DME, based on our requirements, to obtain a stronger version, called Samplable DME. Once the crs has been generated it can be suitably modified, to extract/equivocate R's view, by programming the PRO. On the other hand, S's messages are encrypted using another PRO, such that it enables equivocation of S's view when required. Thus, we replace the crs in the round-optimal [PVW08] framework with PRO to achieve adaptive security. A similar observation was made by the work of [CJS14] where they tried to generate the crs using the observability property of the Global Random Oracle (GRO) [CJS14]. However, their goal was to obtain one-sided simulatable static OT in the GUC model. Whereas, we aim for adaptive security relying on the programmability feature. In our framework, the DME scheme can be instantiated under the DDH and LWE assumptions. Additionally, when instantiated with DDH assumption, our protocol incurs a computation overhead of 5 random oracle queries and a minimal communication overhead of one κ-bit string over the static protocol of PVW (the DDH-based instantiation of [PVW08]).Tab. 4.1 compares our scheme with various other schemes.

Adaptively Secure 1-out-of-N Oblivious Transfer. The work of [NP05] established that $\log \mathsf{N}$ copies of 1-out-of-2 OTs can be transformed to obtain one 1-out-of-N OT, which is statically-secure against active adversaries. This transformation implies existence of statically-secure 1-out-of-N OT at the expanse of $\mathcal{O}(\log \mathsf{N})$ exponentiations. We extend their result to provide a formal proof that the transformation satisfies adaptive security under PRO assumption. At present, one adaptive 1-out-of-N OT protocol [ABB$^+$13, BC15, BC16, BCG17] incurs atleast $\mathcal{O}(\mathsf{N})$ exponentiations. Our adaptive transformation brings down the number of exponentiations to $\mathcal{O}(\log \mathsf{N})$; thereby matching the efficiency of statically-secure 1-out-of-N OT. Interestingly, it can be shown that for the semi-honest setting the seed OTs can be statically secure, if we consider the simulation of the 1-out-of-2 OTs in a non-blackbox manner. For the active setting, we can show that if the 1-out-of-2 OTs is $\mathsf{RE\text{-}OT}$ then it is possible to generate adaptively-secure 1-out-of-N OT from our transformation, if the simulation of the 1-out-of-2 OTs is performed in a non-blackbox manner. This implies that we can plug-in statically-secure 1-out-of-2 $\mathsf{RE\text{-}OT}$.

Oblivious Transfer Extension. We provide the first adaptively-secure protocols for OT Extension solely relying on the PRO assumption. In this regard we present two results, one corresponding to semi-honest setting and the other for the active setting. Our first result proves that the semi-honest protocol of [ALSZ13] can be made adaptively-secure. Interestingly, we show that the seed OTs can be statically secure, if we invoke them in a non-blackbox way. We know that for adaptive security, blackbox-usage of the seed OTs is not possible in our construction since it would violate the results of [LZ13, IR89]. [LZ13] proves that the existence of OT extension protocol, secure against semi-honest adaptive adversaries, imply OT protocol secure against static semi-honest adversaries. In that case, blackbox usage of seed OTs establishes that PRO would imply static semi-honest OT, contradicting the result of [IR89] which states that public key operations are necessary for static OT. Our second result proves that the 3 round (actively secure) protocol of [ALSZ15] can be made adaptively-secure against active adversaries. The seed OTs in this case can be replaced with receiver equivocal static OTs which are secure against active adversaries. Our OT Extension protocols preserve the efficiency of the original static protocols, yielding adaptive 1-out-of-2 OTs at an amortized cost of 3 symmetric key operations and 3κ bits communication per OT. Moreover, if we combine the OT Extension protocol with our 1-out-of-N Transformation, then we obtain 1-out-of-N adaptive OTs at an amortized cost of $\mathsf{N} + 3\log\mathsf{N} + 1$ symmetric key operations per OT. The other adaptive protocols [ABB$^+$13, BC15, BC16, BCG17, BDD$^+$17] require $\mathcal{O}(\mathsf{N})$ public key operations instead.

Roadmap. In Section 4.3 we explain the bugs present in the security proofs of concurrent adaptive OT papers[CO15, BDD$^+$17, HL17] in the PRO model. We proceed to the definition of Samplable DME (referred as DME only) in Section 4.4. Then we present our OT protocol framework in Section 4.5. The 1-out-of-2 to 1-out-of-N OT transformation is elaborated in Section 4.6. Finally, we conclude this chapter with our results in OT extension in Section 4.7.

4.3 Attack in Concurrent Works on UC-secure Adaptive OT

Concurrent to our work, the works of [BDD$^+$17, HL17] on OT, claim adaptive UC security in the PRO model. The previous version of [BDD$^+$17] proposed a general framework for 2-round adaptive OT and provides instantiations under various assumptions such as Learning from Parity with Noise, McEliece cryptosystem, QC-MDPC, Learning With Errors and Computational Diffie Hellman

(CDH). We note however that the authors of [BDD+17] have fixed their protocol, making it UC secure. [HL17] proposes a construction of 1-out-of-N OT under the CDH assumption. However, both protocols as well as the 'simplest OT' construction of [CO15] are prone to a bug when we consider UC-security even against a statically corrupt receiver R^*. The attack stems from the late input extraction of a statically-corrupt R as detailed below. The simulator for the case of static corruption of R, playing the role of honest S, can only extract R^*'s input by observing R^*'s query to RO, made in an attempt to decrypt its chosen message on receiving the last OT message from the sender. This implies that a corrupt R^* can indefinitely delay the input extraction causing composition-related issues.

The delayed input extraction allows us to demonstrate that their constructions do not realise the OT functionality $\mathcal{F}_{\mathsf{OT}}$ presented in Fig. 2.2 where S obtains a notification from the functionality, denoting the end of ideal world execution. Rather, they realise only a weaker version of OT functionality $\mathcal{F}^{\mathsf{w}}_{\mathsf{OT}}$ as depicted in Fig. 4.1. In the weaker variant $\mathcal{F}^{\mathsf{w}}_{\mathsf{OT}}$, S does not obtain any notification from the functionality. Instead its role is limited to sending (sid, a_0, a_1) to $\mathcal{F}^{\mathsf{w}}_{\mathsf{OT}}$, after which S halts. However, $\mathcal{F}^{\mathsf{w}}_{\mathsf{OT}}$ is not composable and cannot be used in a bigger protocol to implement the oblivious transfer functionality. Our observation aligns with the work of [LM16], which states (in page 2, last para of Section 1) that the naive OT functionality (Fig. 1 in their paper), same as $\mathcal{F}^{\mathsf{w}}_{\mathsf{OT}}$, is not composable whereas the modified/revised OT functionality (Fig. 3 of their paper), same as our $\mathcal{F}_{\mathsf{OT}}$, can be proven to be composable. The late input extraction problem is referred to as "timing bug" in their paper (page 3, first paragraph). They explain the issue of composability due to timing bug in the naive OT functionality with an example of OT Extension protocol in Section 3. In Section 4, they address the issue by plugging in the revised OT functionality ($\mathcal{F}_{\mathsf{OT}}$ in our case). Interestingly, all the currently known UC-secure OT protocols, barring the protocols of [CO15, HL17], implement both $\mathcal{F}_{\mathsf{OT}}$ and $\mathcal{F}^{\mathsf{w}}_{\mathsf{OT}}$ functionalities and hence they are composable. In what follows, we first show that the protocols of [CO15, HL17] do not realise $\mathcal{F}_{\mathsf{OT}}$ functionality, but realise only $\mathcal{F}^{\mathsf{w}}_{\mathsf{OT}}$. Next, we demonstrate the compositional issue of using $\mathcal{F}^{\mathsf{w}}_{\mathsf{OT}}$ in (yet another example) of 2PC protocol based on garbled circuit (GC) approach.

Figure 4.1: The ideal functionality $\mathcal{F}^{\mathsf{w}}_{\mathsf{OT}}$ for weaker OT

$\mathcal{F}^{\mathsf{w}}_{\mathsf{OT}}$

Choose: On input $(\mathsf{rec}, \mathsf{sid}, \sigma)$ from R where $\sigma \in \{0, 1\}$; if no message of the form $(\mathsf{rec}, \mathsf{sid}, \sigma)$ has been recorded in the memory, store $(\mathsf{rec}, \mathsf{sid}, \sigma)$ and send $(\mathsf{rec}, \mathsf{sid})$ to S.

Transfer: On input $(\mathsf{sen}, \mathsf{sid}, (a_0, a_1))$ from S with $a_0, a_1 \in \{0, 1\}^n$, if no message of the form $(\mathsf{sen}, \mathsf{sid}, (a_0, a_1))$ is recorded and a message of the form $(\mathsf{rec}, \mathsf{sid}, \sigma)$ is stored, send $(\mathsf{sent}, \mathsf{sid}, a_\sigma)$ to R.

To show that the constructions that feature delayed input extraction (a.k.a timing bug) do not realise $\mathcal{F}_{\mathsf{OT}}$ functionality, we consider an adversarial strategy where R^* does not decrypt the last OT message. In the ideal world, Sim will not be able to extract R^*'s input as R^* does not proceed to decrypting its chosen message from the last OT message. Consequently, Sim fails to invoke $\mathcal{F}_{\mathsf{OT}}$ functionality with R^*'s input and as a result, the $\mathcal{F}_{\mathsf{OT}}$ functionality keeps running. This causes a honest S in the ideal world keep waiting for notification from the $\mathcal{F}_{\mathsf{OT}}$ functionality in order to terminate. Therefore, while honest S *does not halt* in the ideal world, it halts in the real world immediately after sending its last message. This difference in the behavior of honest S can be used to distinguish between the two worlds by an environment. With $\mathcal{F}^{\mathsf{w}}_{\mathsf{OT}}$ functionality, in the scenario mentioned above, an honest S would halt in both the worlds, preserving indistinguishability.

We now illustrate the compositional issue resulted from using $\mathcal{F}^{\mathsf{w}}_{\mathsf{OT}}$ by means of a GC based 2PC protocol in the OT-hybrid model, where the evaluator E first involves with the constructor in a set of OT functionalities to choose the circuits that will be used for evaluation and respectively for checking and on completion of OTs, the constructor sends the GCs to the evaluator. When $\mathcal{F}_{\mathsf{OT}}$ was used, a simulator against a corrupt evaluator E^*, extracts the inputs of E^* from the OT and either constructs a fake/simulated GC or a real GC based on the extracted input of E^*. However in $\mathcal{F}^{\mathsf{w}}_{\mathsf{OT}}$-hybrid model, the simulator for a corrupt evaluator E^* (playing the role of R^*) cannot exact the E^*'s input bits to OT and hence cannot substitute certain (evaluation to be specific) GCs with simulated ones (without getting caught with high probability). This difference would enable the environment $\mathcal{Z}$ to distinguish between both worlds based on E^*'s view.

4.4 Samplable Dual-Mode Encryption

The seminal paper of [PVW08] introduced the primitive of *dual mode encryption* (DME). It works like a regular public key encryption scheme, alongside a notion of encryption branches. The key generation algorithm, takes a branch $\sigma \in \{0,1\}$ as an input, alongside the crs, to generate the public and secret key pair $(\mathsf{pk}, \mathsf{sk})$. Messages encrypted using pk, on branch σ, can be decrypted using sk, whereas the messages encrypted on the other branch remains hidden under certain conditions, described next. The encryption scheme can be further initialized in either of the two modes - *messy* or *decryption* mode, based on the setup phase. The setup phase is invoked with the mode and it returns (crs, t) to the invoking party, where t is the trapdoor for the crs. If the mode is initialized to messy, then the message encrypted on branch $1-\sigma$ remains statistically hidden. Also, it is possible to extract the branch value $1-\sigma$ (and σ can be computed) given pk and t. Whereas, if the scheme is set to decryption mode, then it is possible to generate secret keys sk_0 and sk_1 which decrypts ciphertexts on branches 0 and 1 respectively. Formally, a dual mode encryption scheme is defined as follows:

- $(\mathsf{crs}, t) \leftarrow \mathsf{SetupMessy}(1^\kappa)$: It is a randomized algorithm that takes as input the security parameter κ and outputs the crs and the trapdoor information t, for the messy mode. It enables the invocation of $\mathsf{FindMessy}$ algorithm.

  $(\mathsf{crs}, t) \leftarrow \mathsf{SetupDec}(1^\kappa)$: It is a randomized algorithm that takes as input the security parameter κ and outputs the crs and the trapdoor information t, for the decryption mode. It enables the invocation of $\mathsf{DecKeyGen}$ algorithm.

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{crs}, \sigma)$ It is a randomized algorithm that takes as input the crs and the branch σ and returns the public/encryption and secret/decryption key pair $(\mathsf{pk}, \mathsf{sk})$ for branch σ.

- $(y, r) \leftarrow \mathsf{Enc}(\mathsf{pk}, \sigma, m)$: It is a randomized algorithms that takes as input pk, the branch σ and the message $m \in \{0,1\}^\kappa$. It returns the ciphertext y, encrypted on branch σ and the randomness r, used for the encryption.

- $m \leftarrow \mathsf{Dec}(\mathsf{sk}, y)$: It is a deterministic algorithm that takes in input sk and y and it returns m only if sk and y correspond to the same branch.

- $b \leftarrow \mathsf{FindMessy}(\mathsf{pk}, t)$: It returns the branch value $b \in \{0,1\}$ given pk and the trapdoor t, when the mode is set to messy. The message encrypted on this branch is statistically hidden.

- $(\mathsf{pk}, \mathsf{sk}_0, \mathsf{sk}_1) \leftarrow \mathsf{DecKeyGen}(t)$: It is a randomized algorithm that generates a key pair $(\mathsf{pk}, \mathsf{sk}_0, \mathsf{sk}_1)$ when it is invoked with the trapdoor t as input and the mode is set to decryption. The secret key sk_b, $b \in \{0,1\}$, enables decryption of ciphertexts on branch b.

The above defined scheme satisfies correctness and four security properties, as mentioned in the [PVW08] paper. In addition, the dual mode encryption scheme must satisfy another property for our OT protocol. It must be possible to generate a crs for the messy mode from a random oracle query with overwhelming probability. We outline the correctness and security properties as follows:

- **Correctness**: For every mode, for all $(\mathsf{crs}, t) \leftarrow \mathsf{SetupMessy}(1^\kappa)$ / $\mathsf{SetupDec}(1^\kappa)$, for each $\sigma \in \{0,1\}$, for each $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{crs}, \sigma)$, and for all $m \in \{0,1\}^\kappa$, the following holds $\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, \sigma, m)) = m$.

- **Property 1** (Indistinguishability of modes): The crs generated from either modes are indistinguishable, i.e. for all $(\mathsf{crs}, t) \leftarrow \mathsf{SetupMessy}(1^\kappa)$ and for all $(\mathsf{crs}', t') \leftarrow \mathsf{SetupDec}(1^\kappa)$, the following holds $\mathsf{crs} \overset{c}{\approx} \mathsf{crs}'$.

- **Property 2** (Indistinguishability of Branches in Messy mode): In messy mode, the public key does not leak about the branch value against a computational adversary i.e., for all crs generated by $\mathsf{SetupMessy}$, the following condition holds true - $\mathsf{KeyGen}(\mathsf{crs}, 0) \overset{c}{\approx} \mathsf{KeyGen}(\mathsf{crs}, 1)$.

- **Property 3** (Messy Branch Identification): For each crs generated by $\mathsf{SetupMessy}$ and for each pk returned by $\mathsf{KeyGen}(\mathsf{crs}, \sigma)$, $\mathsf{FindMessy}(\mathsf{pk}, t)$ returns the messy branch b, s.t. $b = 1 - \sigma$. Moreover, any message encrypted on branch b is statistically hidden, i.e. for every $m_0, m_1 \in \{0, 1\}^\kappa$, $\mathsf{Enc}(\mathsf{pk}, b, m_0) \overset{s}{\approx} \mathsf{Enc}(\mathsf{pk}, b, m_1)$.

- **Property 4** (Dual Decryptable Branches in decryption mode): For each $(\mathsf{crs}, t) \leftarrow \mathsf{SetupDec}(1^\kappa)$ and for each $(\mathsf{pk}, \mathsf{sk}_0, \mathsf{sk}_1) \leftarrow \mathsf{DecKeyGen}(t)$, the following three conditions hold for all $m, m_0, m_1 \in \{0, 1\}^\kappa$:

    i. $(\mathsf{pk}, \mathsf{sk}_0) \overset{s}{\approx} \mathsf{KeyGen}(\mathsf{crs}, 0)$, $(\mathsf{pk}, \mathsf{sk}_1) \overset{s}{\approx} \mathsf{KeyGen}(\mathsf{crs}, 1)$ and $(\mathsf{pk}, \mathsf{sk}_0) \overset{s}{\approx} (\mathsf{pk}, \mathsf{sk}_1)$.

    ii. $\mathsf{Dec}(\mathsf{sk}_b, \mathsf{Enc}(\mathsf{pk}, b, m)) = m$ for all $b \in \{0, 1\}$.

    iii. $\mathsf{Enc}(\mathsf{pk}, b, m_0) \overset{c}{\approx} \mathsf{Enc}(\mathsf{pk}, b, m_1)$, for all $b \in \{0, 1\}$.

- **Property 5** (Samplable crs in Messy mode): for all $c \in \{0, 1\}^\kappa$, $(\mathsf{crs}, t) \leftarrow \mathsf{SetupMessy}(1^\kappa)$, the random oracle query $\mathcal{F}_{\mathsf{RO}}(c)$ is identically distributed to the crs of messy mode except with negligible probability i.e., $\mathcal{F}_{\mathsf{RO}}(c) \overset{s}{\approx} \mathsf{crs}$.

Next, we provide concrete instantiations of our samplable DME scheme based on the LWE and DDH assumption.

4.4.1 Instantiation under LWE assumption

In this section we provide an instantiation of our DME based on LWE. Before describing the DME instantiation we recall the definition of an LWE encryption scheme from [GPV08], which will be instrumental in the instantiation. The encryption scheme is a collection of following three algorithms:

- $\mathsf{LWESetup}$: Choose a matrix $\mathbf{A} \leftarrow \mathbb{Z}_q^{m \times \kappa}$ uniformly at random.

- $\mathsf{LWEKeyGen}$: Choose a secret decryption key $\mathbf{s} \leftarrow \mathbb{Z}_q^\kappa$ uniformly at random. The public key is the vector $\mathbf{p} = \mathbf{A}^\mathsf{T}\mathbf{s} + \mathbf{x} \in \mathbb{Z}_q^m$, where $\mathbf{x} = (x_1, \ldots, x_m)$ and each x_i is chosen independently from the error distribution χ for $i \in [m]$.

- $\mathsf{LWEEnc}(\mathbf{p}, b)$: To encrypt a bit $b \in \{0, 1\}$, choose a vector $\mathbf{e} \in \mathbb{Z}^m$ uniformly at random and set the ciphertext as $(\mathbf{u}, c) = (\mathbf{A}\mathbf{e}, \mathbf{p}^\mathsf{T}\mathbf{e} + b.\lceil q/2 \rfloor) \in \mathbb{Z}_q^{\kappa+1}$.

- $\mathsf{LWEDec}(\mathbf{s},(\mathbf{u},c))$: Compute $b' = c - \mathbf{s}^{\mathsf{T}}\mathbf{u} \in \mathbb{Z}_q$. Output 0 if b' is closer to 0 than to $\lceil q/2 \rceil \bmod q$, otherwise output 1.

The above encryption scheme satisfies the notion of IND-CPA security if we assume that the LWE problem is hard for parameters $q = \mathcal{O}(\kappa^3), m = \mathcal{O}(\kappa \log \kappa)$ (Lemma. 4.4.1). The proof appears in the work of [GPV08] and we refer to their paper for more details.

Lemma 4.4.1. *The cryptosystem above is CPA-secure, assuming that LWE is hard for parameters* q, m.

Next, we will use the encryption scheme in our DME instantiation. In our instantiation we need an additional algorithm, $\mathsf{IsMessy}$, besides the usual algorithms of DME. $\mathsf{IsMessy}(t, \mathsf{pk})$ answers whether pk is messy or not, when it is invoked with trapdoor t on public key pk. Now we are ready to instantiate the algorithms for the DME scheme based on the LWE assumption. The description of the algorithms has been borrowed from the paper of [PVW08].

- $(\mathsf{crs}, t) \leftarrow \mathsf{SetupMessy}(1^\kappa)$: In the messy mode the crs is generated as follows: Sample a matrix $\mathbf{A} \leftarrow \mathbb{Z}_q^{\kappa \times m}$ uniformly at random, along with a trapdoor $t = (\mathbf{S}, \mathbf{A})$ (as described in Section 5.3.2 of [GPV08]). Sample an independent row vector $\mathbf{v}_b \leftarrow \mathbb{Z}_q^{l \times m}$ uniformly at random for every $b \in \{0, 1\}$. Set $\mathsf{crs} = (\mathbf{A}, \mathbf{v}_0, \mathbf{v}_1)$ and $t = (\mathbf{S}, \mathbf{A})$.

- $(\mathsf{crs}, t) \leftarrow \mathsf{SetupDec}(1^\kappa)$: In the decryption mode the crs is generated as follows: Sample a matrix $\mathbf{A} \leftarrow \mathbb{Z}_q^{\kappa \times m}$ uniformly at random. Choose a row vector $\mathbf{w} \leftarrow \mathbb{Z}_q^{1 \times m}$ uniformly at random. For every $b \in \{0, 1\}$, sample a secret $\mathbf{s}_b \leftarrow \mathbb{Z}_q^n$ uniformly at random and an error row vector $\mathbf{x}_b \leftarrow \chi^{1 \times m}$ (i.e., the m entries are chosen independently from error distribution χ). Let $\mathbf{v}_b = \mathbf{s}_b^{\mathsf{T}}\mathbf{A} + \mathbf{x}_b\mathbf{w}$. Set $\mathsf{crs} = (\mathbf{A}, \mathbf{v}_0, \mathbf{v}_1)$ and $t = (\mathbf{w}, \mathbf{s}_0, \mathbf{s}_1)$.

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{crs}, \sigma)$: Given crs and σ, sample a secret $\mathbf{r} \leftarrow \mathbb{Z}_q^n$ and a row vector $\mathbf{x} \leftarrow \chi^{l \times m}$. Set $\mathsf{pk} = \mathbf{r}^{\mathsf{T}}\mathbf{A} + \mathbf{x}\mathbf{v}$ and $\mathsf{sk} = \mathbf{r}$.

- $y \leftarrow \mathsf{Enc}(\mathsf{pk}, b, m)$: Given $\mathsf{pk} = \mathbf{A}, m$ and $b \in \{0, 1\}$, m is encrypted as $y = \mathsf{LWEEnc}((\mathbf{A}, \mathsf{pk} + \mathbf{v}_b), m)$.

- $m \leftarrow \mathsf{Dec}(\mathsf{sk}, y)$: Given $\mathsf{sk} = \mathbf{r}$ and ciphertext v, the underlying plaintext message is decrypted as $m = \mathsf{LWEDec}(\mathsf{sk}, y)$.

- $b \leftarrow \mathsf{FindMessy}(\mathsf{pk}, t)$: Given pk and $t = (\mathbf{S}, \mathbf{A})$ in messy mode, invoking $\mathsf{IsMessy}((\mathbf{S}, \mathbf{A}), \mathsf{pk} + \mathbf{v}_b)$ for each $b \in \{0, 1\}$, outputs the messy branch value for b, and it is correct with overwhelming probability.

- $(\mathsf{pk}, \mathsf{sk}_0, \mathsf{sk}_1) \leftarrow \mathsf{DecKeyGen}(t)$: Given the $\mathsf{crs} = (\mathbf{A}, \mathbf{v}_0, \mathbf{v}_1)$ in decryption mode and the trapdoor $t = (\mathbf{w}, \mathrm{s}_0, \mathrm{s}_1)$, the public and secret keys pair is formed as follows: $(\mathsf{pk}, \mathsf{sk}_0, \mathsf{sk}_1) = (\mathbf{w}, \mathrm{s}_0, \mathrm{s}_1)$.

The paper of [PVW08] has proven that the above instantiation satisfies correctness and Properties 1-4 of DME. It has also shown in Lemma 7.4 that most of the keys are messy, proving Property 5. In particular, if the crs in the messy mode is generated as $\mathsf{crs} = (\mathbf{A}, \mathbf{v}_0, \mathbf{v}_1) \leftarrow \mathcal{F}_{\mathsf{RO}}(\mathsf{sid}||c)$, then $\mathcal{F}_{\mathsf{RO}}(\mathsf{sid}||c)$ returns a messy key except with negligible probability for a random value c, where $\mathcal{F}_{\mathsf{RO}} : \{0,1\}^{2\kappa} \to \mathbb{Z}_q^{\kappa \times m} \times \mathbb{Z}_q^{2l \times m}$.

4.4.2 Instantiation under DDH assumption

In this section we present an instantiation of the DME based on the DDH assumption:

- $(\mathsf{crs}, t) \leftarrow \mathsf{SetupMessy}(1^\kappa)$: In messy mode, the crs is a non-DDH tuple and it is generated as follows: Sample $g_0, g_1 \leftarrow_R \mathbb{G}$, $x, y \leftarrow_R \mathbb{Z}_p$ and initialize $h_0 = g_0^x$, $h_1 = g_1^y$. Set $\mathsf{crs} = (g_0, g_1, h_0, h_1)$ and the trapdoor t is set to (x, y).

- $(\mathsf{crs}, t) \leftarrow \mathsf{SetupDec}(1^\kappa)$: In decryption mode, the crs is a DDH tuple and it is generated as follows: Sample $g_0 \leftarrow_R \mathbb{G}$, $x, y \leftarrow_R \mathbb{Z}_p$ and initialize $g_1 = g_0^y$, $h_0 = g_0^x$, $h_1 = g_1^x$. Set $\mathsf{crs} = (g_0, g_1, h_0, h_1)$ and the trapdoor t is set to (x, y).

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{crs}, \sigma)$ Given $\sigma \in \{0,1\}$ and $\mathsf{crs} = (g_0, g_1, h_0, h_1)$, set $\mathsf{pk} = (g, h) = (g_\sigma^\alpha, h_\sigma^\alpha)$, where $\alpha \leftarrow_R \mathbb{Z}_p$. The secret key sk is set to α.

- $y \leftarrow \mathsf{Enc}(\mathsf{pk}, b, m)$: Given $\mathsf{pk} = (g, h)$, m and $b \in \{0,1\}$, m is encrypted as follows: Sample $s, r \leftarrow_R \mathbb{Z}_p$ and set $u = g_b^s h_b^r, v = g^s h^r$. The ciphertext is $y = (u, v.m)$ and the corresponding randomness is (s, r).

- $m \leftarrow \mathsf{Dec}(\mathsf{sk}, y)$: Given $\mathsf{sk} = \alpha$ and ciphertext $y = (c_0, c_1)$, the corresponding plaintext s obtained as $m = c_1/c_0^\alpha$.

- $b \leftarrow \mathsf{FindMessy}(\mathsf{pk}, t)$: Given $\mathsf{pk} = (g, h)$ and $t = (x, y)$, if $h = g^x$ then output that branch $b = 0$ is messy else output that branch $b = 1$ is messy.

- $(\mathsf{pk}, \mathsf{sk}_0, \mathsf{sk}_1) \leftarrow \mathsf{DecKeyGen}(t)$: Given $\mathsf{crs} = (g_0, g_1, h_0, h_1)$ in decryption mode and $t = (x, y)$ generate public and secret key pairs as follows: Sample $r_0 \leftarrow_R \mathbb{Z}_p$ and compute $r_1 = r_0/y$. Set $\mathsf{pk} = (g, h) = (g_0^{r_0}, h_0^{r_0})$, $\mathsf{sk}_0 = r_0$ and $\mathsf{sk}_1 = r_1$.

The above DDH-based instantiation correctly implements a DME scheme and it satisfies Properties 1-5 (Section 4.4). Proof of Property 1-4 follows from the paper of [PVW08]. Next, we show that it also satisfies Property 5, i.e. the crs in the messy mode can be sampled using a random oracle. The crs for the messy mode is required to be a non-DDH tuple. A random oracle query result of size $4|\mathbb{G}|$ bits returns a DDH tuple, with probability :

$$\frac{|\mathbb{Z}_p|^3}{|\mathbb{Z}_p|^4} = \frac{1}{|\mathbb{Z}_p|}$$

With $1 - \frac{1}{|\mathbb{Z}_p|}$ probability the tuple will be non-DDH and hence a valid crs for messy mode will be generated, satisfying Property 5. The construction of $\mathsf{RE\text{-}OT}$ in 3.1 is the DDH-based instantiation of the [PVW08] framework.

4.5 Framework for Adaptive Oblivious Transfer

In this section, we present our round-optimal framework for adaptive OT given a DME scheme, random oracles $\mathcal{F}_{\mathsf{RO1}}$ and $\mathcal{F}_{\mathsf{RO2}}$. We first present a brief overview of our protocol and then we present our proof. R generates the crs by sampling a random string c and invoking the random oracle on c. R obtains a valid crs, in messy mode, except with negligible probability. He invokes the KeyGen with the crs and his choice bit σ to obtain a key pair $(\mathsf{pk}, \mathsf{sk})$ which would allow decryption on branch σ using sk. R sends c and pk to S. Property 2 ensures that pk does not leak about σ. S generates the crs using c and encrypts two random pads on both branches using pk. Finally, S encrypts his messages using RO queries on the pads. R can decrypt the pad corresponding to branch σ, due to correctness of the DME scheme, and thus obtain the message corresponding to choice bit σ. The framework has been presented as protocol π_{OT} in Fig. 4.2.

4.5.1 Static Security

To make the proof of adaptive security more comprehensible, we first prove that π_{OT} realizes the ideal functionality $\mathcal{F}_{\mathsf{OT}}$ (Fig. 2.2) in presence of static adversaries. This is extended to adaptive security in Section 4.5.2.

In order to prove static security, we describe a simulator Sim who behaves as the ideal world adversary and generates a view of $\mathcal{Z}$ which is indistinguishable from the view generated by the real world adversary A in the real world. It does so by invoking $\mathcal{F}_{\mathsf{OT}}$, on behalf of the adversary in ideal world, and running a copy of A internally, in the head. We denote this internal adversary as $\mathsf{A}_{\mathsf{Int}}$. Sim simulates the role of the honest parties and the environment to $\mathsf{A}_{\mathsf{Int}}$ in the internal execution. Whenever A corrupts a party in the real world, $\mathsf{A}_{\mathsf{Int}}$ also corrupts that party in the internal execution and Sim corrupts that party in the ideal world. At the end of the protocol $\mathsf{A}_{\mathsf{Int}}$ forwards its view

Figure 4.2: Adaptively-Secure Oblivious Transfer Protocol

π_{OT}

- **Functionalities:** Random oracles $\mathcal{F}_{\mathsf{RO1}} : \{0,1\}^{2\kappa} \to \{0,1\}^*$ and $\mathcal{F}_{\mathsf{RO2}} : \{0,1\}^{2\kappa} \to \{0,1\}^{\ell}$ respectively.
- **Private Inputs:** S has input messages (a_0, a_1) and R has an input bit σ.

Choose:

- R samples $c \leftarrow_R \{0,1\}^{\kappa}$.
- R generates $\mathsf{crs} \leftarrow \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||c)$.
- R computes $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{crs}, \sigma)$.
- R sends (c, pk) to S.

Transfer:

- S generates $\mathsf{crs} \leftarrow \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||c)$.
- S samples $r_0, r_1 \leftarrow_R \{0,1\}^{\kappa}$.
- S computes $s_b \leftarrow \mathsf{Enc}(\mathsf{pk}, b, r_b)$, for $b \in \{0,1\}$.
- S sets $y_0 = \mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||r_0) \oplus a_0$ and $y_1 = \mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||r_1) \oplus a_1$.
- S sends $\{(s_0, y_0), (s_1, y_1)\}$ to R.

Local Computation by R:

- Computes $r_\sigma = \mathsf{Dec}(\mathsf{sk}, s_\sigma)$ and outputs $a_\sigma = y_\sigma \oplus \mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||r_\sigma)$.

to Sim who forwards it to $\mathcal{Z}$ as its ideal world view. We refer to Appendix 2.4 for clarity of $\mathsf{A}_{\mathsf{Int}}$ notation and details of static security in the UC model [Can01]. For static security, we prove Theorem 4.5.1 by considering the four exhaustive corruption cases namely : (1) Both S and R are honest (2) S^* is corrupt while R is honest (3) S is honest while R^* is corrupt (4) Both S^* and R^* are corrupt. In each of the above cases we describe a simulator Sim and show that the real world view of $\mathcal{Z}$ is indistinguishable from its ideal world view.

We first give a brief intuition of the security proof. Revisiting the proof of security for a static adversary [PVW08], note that Sim sets the crs in accordance with which party is corrupt, to enable input extraction of $\mathsf{A}_{\mathsf{Int}}$ in the internal execution. More specifically, while crs of the internal execution is set to messy mode when *R is corrupted*, it is set to decryption mode when *S is corrupted*. Sim can perform this by invoking $\mathsf{SetupMessy}$ (or $\mathsf{SetupDec}$) to obtain (crs, t) and then program the random oracle to return the same crs. In the former case, $\mathsf{FindMessy}$ can be suitably invoked to extract σ. In the latter case, Sim can invoke $\mathsf{DecKeyGen}$ to obtain secret keys on both branches and unlock both messages using the secret keys. The simulation of our protocol for static security is similar to that of PVW.

We are ready to present the formal proof of Theorem. 4.5.1. We design an ideal world adversary Sim who creates an ideal world view $\text{IDEAL}_{\mathcal{F},\mathsf{Sim},\mathcal{Z}}(1^{\kappa}, z)$ of $\mathcal{Z}$ which is indistinguishable from the real world view $\text{REAL}_{\mathcal{F},\mathsf{A},\mathcal{Z}}(1^{\kappa}, z)$ of $\mathcal{Z}$.

Theorem 4.5.1. *If DME is a samplable dual mode encryption scheme then protocol π_{OT} securely realizes the $\mathcal{F}_{\mathsf{OT}}$ functionality against static active adversaries in $(\mathcal{F}_{\mathsf{RO1}}, \mathcal{F}_{\mathsf{RO2}})$- hybrid model.*

The Simulator: We describe the simulator Sim for each possible case of corruption.

Case 1. S and R are honest: In this case Sim acts on behalf of both parties in the internal execution. At the end, $\mathsf{A}_{\mathsf{Int}}$ generates his view without corrupting any party and sends it to Sim who forwards it to $\mathcal{Z}$.

(i) *Simulating the crs and R's message:* Sim obtains $(\mathsf{crs}, t) \leftarrow \mathsf{SetupDec}$. Sim samples $c \leftarrow_R \{0,1\}^\kappa$ and programs $\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||c)$ to return crs. Sim invokes $\mathsf{DecKeyGen}(t)$ to obtain $(\mathsf{pk}, \mathsf{sk}_0, \mathsf{sk}_1)$ and sends (c, pk) as the first OT message to S on behalf of R in the internal execution.

(ii) *Simulating S's message:* Sim sets (s_0, s_1) as per the protocol and (y_0, y_1) are set randomly. Sim sends the simulated message on behalf of S.

(iii) *Simulating R's computation:* Sim completes the simulation on behalf of R in the internal world.

Case 2. S^* is corrupted and R is honest: In this case, Sim acts on behalf of R in the internal execution. At the end, $\mathsf{A}_{\mathsf{Int}}$ generates the view of S^* and sends it to Sim who forwards it to $\mathcal{Z}$.

(i) *Simulating the crs and R's message:* Same as Case 1.

(ii) $\mathsf{A}_{\mathsf{Int}}$ plays the role of S^* and computes the sender message in the internal world. $\mathsf{A}_{\mathsf{Int}}$ sends the second OT message to R in the internal world.

(iii) *Simulating R's computation:* Sim decrypts r_0 and r_1 using secret keys sk_0 and sk_1 and obtains a_0 and a_1 from y_0 and y_1. Sim invokes $\mathcal{F}_{\mathsf{OT}}$ with (a_0, a_1) and completes the simulation on behalf of R in the internal world.

Case 3. S is honest and R^* is corrupted: In this case, Sim acts on behalf of S in the internal execution. At the end, $\mathsf{A}_{\mathsf{Int}}$ generates the view of R^* and sends it to Sim who forwards it to $\mathcal{Z}$.

(i) *Simulating the crs:* $\mathsf{A}_{\mathsf{Int}}$ plays the role of R^* and computes the receiver message in the internal world. Whenever $\mathsf{A}_{\mathsf{Int}}$ queries $\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||c)$, Sim invokes $\mathsf{SetupMessy}$ to obtain (crs, t) and programs $\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||c)$ to return crs. Sim stores the tuple in a list Q as $(\mathsf{sid}, c, \mathsf{crs}, t)$. If a query is repeated then Sim returns the crs corresponding to the entry in Q indexed by the sid and c values. $\mathsf{A}_{\mathsf{Int}}$ sends the first OT message to S in the internal world.

(ii) *Simulating S's message:* Sim extracts the the choice bit, i.e. σ, of R^* by invoking $\mathsf{FindMessy}(\mathsf{pk}, t)$. Sim sends σ to $\mathcal{F}_{\mathsf{OT}}$ on behalf of R in the ideal world, obtains a_σ and constructs (s_σ, y_σ) honestly. On the other hand, $s_{\overline{\sigma}}$ is set honestly and $y_{\overline{\sigma}}$ is set randomly. Finally, Sim computes the sender's message and sends it to R^* in the internal world.

(iii) $\mathsf{A}_{\mathsf{Int}}$ computes on behalf of R^* and completes the protocol in the internal world.

Case 4. Both S^* and R^* are corrupted: This is a trivial case of corruption. Sim invokes $\mathsf{A}_{\mathsf{Int}}$, who simulates messages of both parties and generates the view internally. At the end of execution, $\mathsf{A}_{\mathsf{Int}}$ sends the generated view to Sim who forwards it to $\mathcal{Z}$.

Indistinguishability: Here we show that the ideal world view $\text{IDEAL}_{\mathcal{F},\mathsf{Sim},\mathcal{Z}}(1^\kappa, z)$ of $\mathcal{Z}$ is indistinguishable from the real world view $\text{REAL}_{\mathcal{F},\mathsf{A},\mathcal{Z}}(1^\kappa, z)$ of $\mathcal{Z}$. We denote $\text{REAL}_{\mathcal{F},\mathsf{A},\mathcal{Z}}(1^\kappa, z)$ as hybrid HYB_R. The ideal world view of $\mathcal{Z}$ varies based on the case of corruption. For case D ($\text{D} \in [4]$), we denote the ideal world view as $\mathsf{HYB}_{I.d}$. We prove that HYB_R is indistinguishable from the corresponding ideal world view for each of the four exhaustive cases of corruption.

Case 1. S and R are honest: We prove that HYB_R and the ideal world view i.e $\mathsf{HYB}_{I.1}$ is indistinguishable through a series of intermediate hybrids.

- **HYB_1 :** We consider a hybrid HYB_1 which is same as HYB_R except that here, the crs is generated using $\mathsf{SetupDec}$. Indistinguishability follows from Property 1 of DME, i.e. indistinguishability of the two modes, and random sampling of c. A distinguisher for the hybrids can be used to break Property 1 or guess the exact value of c, both of which happen with negligible probability.

- **HYB_2 :** We consider a hybrid HYB_2 which is same as HYB_1 except that Sim generates pk using $\mathsf{DecKeyGen}$. Indistinguishability follows statistically from Property 4i of DME, i.e. the branch remains statistically hidden when the DME is set to decryption mode.

- **HYB_3 :** We consider a hybrid HYB_3 similar to HYB_2 except that in HYB_3 Sim constructs s_0 and s_1 using r_0 and r_1 whereas y_0 and y_1 are formed using different random pads, r'_0 and r'_1. Indistinguishability between the hybrids follows from Property 4iii of DME, i.e. indistinguishability of ciphertexts in decryption mode.

- **$\mathsf{HYB}_{I.1}$:** We consider the ideal world hybrid $\mathsf{HYB}_{I.1}$ similar to HYB_3 except that in $\mathsf{HYB}_{I.1}$, Sim sets y_0 and y_1 randomly. $\mathsf{A}_{\mathsf{Int}}$ can distinguish between the hybrids only if the values r'_0 or r'_1 are guessed precisely and queried to the random oracle. This event occurs with negligible probability in the random oracle model and as a result indistinguishabilty between the hybrids follows.

Case 2. S^* is corrupted and R is honest: We prove that HYB_R and the ideal world view i.e $\mathsf{HYB}_{I.2}$ is indistinguishable through an intermediate hybrid.

- HYB_1 **:** We consider hybrid HYB_1 which is same as HYB_1 in previous case. Indistinguishability between HYB_R and HYB_1 follows (similar to the previous case) from Property 1 of DME.

- $\mathsf{HYB}_{I.2}$ **:** We consider a hybrid $\mathsf{HYB}_{I.2}$ similar to HYB_1 except that in $\mathsf{HYB}_{I.2}$ Sim decrypts r_0 and r_1 and obtains a_0 and a_1 respectively using secret keys on both branches. Indistinguishability follows from Property 4ii of DME, which ensures that Sim can decrypt messages on both branches using the secret keys in the decryption mode.

Case 3. S is honest and R^* is corrupted: We prove that HYB_R and the ideal world view i.e $\mathsf{HYB}_{I.3}$ is indistinguishable through a series of intermediate hybrids.

- HYB_1 **:** We consider a hybrid HYB_1 which is same as HYB_R except here the crs is generated using $\mathsf{SetupMessy}$. Indistinguishability follows from Property 5 and 1, i.e. the crs in the messy mode can be sampled using the random oracle and the crs in the messy mode is indistinguishable from the crs in decryption mode. Another possible way of distinguishing is if the distinguisher can guess the value of the crs without querying c to the random oracle, but that happens with negligible probability, due to the random oracle assumption.

- HYB_2 **:** We consider a hybrid HYB_2 which is same as HYB_1 except that, here, Sim extracts the value of σ, by invoking $\mathsf{FindMessy}(\mathsf{crs}, t)$, and sends it to $\mathcal{F}_{\mathsf{OT}}$ and obtains a_σ. Indistinguishability follows from Property 3 of DME.

- HYB_3 **:** We consider a hybrid HYB_3 similar to HYB_2 except that in HYB_3, Sim constructs $s_{\overline{\sigma}}$ using $r_{\overline{\sigma}}$, whereas $y_{\overline{\sigma}}$ is formed using different random pad, $r'_{\overline{\sigma}}$. Indistinguishability between the hybrids follows from the fact that $r'_{\overline{\sigma}}$ remains statistically hidden, due to Property 3, in the messy mode.

- $\mathsf{HYB}_{I.3}$ **:** Finally we consider our ideal world hybrid $\mathsf{HYB}_{I.3}$ where $y_{\overline{\sigma}}$ is set randomly. Indistinguishability follows from the random oracle assumption since the distinguisher has to guess $r'_{\overline{\sigma}}$ to distinguish between the two hybrids and this happens with negligible probability.

Case 4. Both S^* and R^* are corrupted: In this case HYB_R and $\mathsf{HYB}_{I.4}$ are generated by A and $\mathsf{A}_{\mathsf{Int}}$ after being in control of both honest parties in the real world and internal world respectively. As a result, the two views are identical.

4.5.2 Adaptive Security

Building upon the proof of static security in the previous section, we now prove that π_{OT} securely implements $\mathcal{F}_{\mathsf{OT}}$ in the presence of adaptive adversaries. We refer to Appendix 2.4 for details about the security model. We give a brief overview of the proof and then we present it formally. Following the lines of the static proof, Sim programs the crs of the internal execution to be in messy mode, when the receiver is corrupt in the first round, or in the decryption mode otherwise, to enable extraction of R's input or S's input respectively.

In addition, Sim has to equivocate the view of R (resp. S), in the internal execution, when R (resp. S) gets corrupted adaptively by $\mathsf{A}_{\mathsf{Int}}$. The proof demands equivocation only when R gets corrupted after sending the first OT message and/or S gets corrupted after sending the second OT message. This is done to ensure that the ideal world views (messages and internal state) of the simulated honest parties (in the internal execution) are consistent with the real world views (messages and internal state) of the actual honest parties else $\mathcal{Z}$ can distinguish between the two views. In the first scenario, when R is corrupted after the first OT message is sent, the mode is set to decryption, and Sim can extract secrets keys for both branches, corresponding to pk, by invoking $\mathsf{DecKeyGen}$. When R gets corrupted and Sim obtains σ as R's input, Sim can provide sk_σ as its secret key on branch σ. Equivocation is successful due to Property 4ii of DME. In the second case when S is corrupted after the second OT message is sent, the random values sent corresponding to (y_0, y_1) by Sim have to be made consistent with sender's actual input (a_0, a_1). For this, Sim exploits the programmability of $\mathcal{F}_{\mathsf{RO2}}$ to enforce that (y_0, y_1) decrypts to (a_0, a_1). We are ready to present the proof of Theorem 4.5.2.

Theorem 4.5.2. *If DME is a samplable dual mode encryption scheme then protocol π_{OT} securely realizes $\mathcal{F}_{\mathsf{OT}}$ functionality against adaptive active adversaries in $(\mathcal{F}_{\mathsf{RO1}}, \mathcal{F}_{\mathsf{RO2}})$- hybrid model.*

Proof. We describe the simulator corresponding to the protocol π_{OT}, for each possible case of adaptive corruption.

The Simulator: The simulator Sim that generates the ideal world view, is initialized with input values from $\mathcal{Z}$ based on which party is corrupted to facilitate simulation.

Outset of the Protocol: Whenever $\mathsf{A}_{\mathsf{Int}}$ queries $\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||c)$, Sim invokes the $\mathsf{SetupMessy}$ algorithm to obtain (crs, t) and programs $\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||c)$ to return crs. Sim stores the tuple in a list Q as $(\mathsf{sid}, c, \mathsf{crs}, t)$. If a query is repeated then Sim returns the crs corresponding to the entry in Q indexed by the sid and c values.

R is honest in the first round: Sim computes R's message similar to case 1(i) (R's message for (S, R) case) of the static proof. The crs is set in the decryption mode by programming $\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||c)$.

Sim invokes $\mathsf{DecKeyGen}$ to obtain $(\mathsf{pk}, \mathsf{sk}_0, \mathsf{sk}_1)$ and he sends c and pk as the first OT message to $\mathsf{A}_{\mathsf{Int}}$ on behalf of R in the internal execution.

- **S is honest in the second round:** Sim acts on behalf of S in the internal execution. Sim simulates according to Case 1(ii) (S's message for (S, R) case) of static proof.

  - **Case 1(A). R is honest in the first round, S is honest in the second round, R^* is corrupted after second OT message:** Sim obtains σ and a_σ in the ideal world. Sim equivocates pk by setting sk_σ as the secret key on branch σ. Additionally, Sim equivocates y_σ s.t. a_σ can be obtained from it. For this, Sim programs $\mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||r_\sigma) = a_\sigma \oplus y_\sigma$. At the end of the protocol, $\mathsf{A}_{\mathsf{Int}}$ outputs $\perp$ and sends its internal state to Sim who forwards it to $\mathcal{Z}$.

    Post Execution. In case of post execution corruption of S^*, Sim obtains (a_0, a_1) and needs to provide the internal randomness of S^* s.t. y_0 and y_1 open to a_0 and a_1. We note that y_σ was previously equivocated. Equivocation of $y_{\overline{\sigma}}$ is performed by programming $\mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||r_{\overline{\sigma}}) = a_{\overline{\sigma}} \oplus y_{\overline{\sigma}}$. Sim sends the internal state of S^* to $\mathcal{Z}$ who halts with an output.

  - **Case 1(B). R is honest in the first round, S is honest in the second round, R is honest after second OT message:** In this case, Sim acts on behalf of both parties throughout the protocol in the internal execution. At the end of the protocol, $\mathsf{A}_{\mathsf{Int}}$ outputs his random tape as its internal state to Sim who forwards it to $\mathcal{Z}$.

    Post Execution. In case of post execution corruption of R^* and S^*, Sim obtains (σ, a_σ) and (a_0, a_1), and has to provide the internal randomness of R^* and S^* to $\mathcal{Z}$. Equivocation of both views is similar to the previous case (Case 1(A) of adaptive simulation) where the view of R^* is equivocated first and then the view of S^* is equivocated. Sim sends the equivocated views of R^* and S^* to $\mathcal{Z}$, who halts with an output.

- **Case 2. R is honest in the first round, S^* is corrupted in the second round:** Sim receives the second message on behalf of R from S^* in the internal execution. Simulation is performed as in Case 2(iii) (R's computation for (S^*, R) case) of static proof. $\mathsf{A}_{\mathsf{Int}}$ outputs $\perp$ at the end of the protocol and sends its internal state to Sim who forwards it to $\mathcal{Z}$.

  Post Execution. In case of post execution corruption of R^*, Sim obtains σ and a_σ and he proceeds like the simulator for case 1(A) of adaptive simulation.

R^* is corrupted in the first round: Whenever $\mathsf{A}_{\mathsf{Int}}$ queries $\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||c)$, Sim invokes the $\mathsf{SetupMessy}$ algorithm to obtain crs and programs $\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||c)$ to return crs. Sim stores the tuple in a list Q as $(\mathsf{sid}, c, \mathsf{crs}, t)$. If a query is repeated then Sim returns the the crs corresponding to entry in Q indexed by the sid and c values. $\mathsf{A}_{\mathsf{Int}}$ generates the first OT message which is sent to S in the internal execution. This is similar to Case 3(i) (R's message for (S, R*) case) in static proof.

- **Case 3. R* is corrupted in the first round, S is honest in the second round:** Sim receives the first OT message, on behalf of S, from $\mathsf{A}_{\mathsf{Int}}$ controlling R* in the internal execution. Sim continues simulation as in Case 3(ii) (S's message for (S, R*) case) of static proof.

  Post Execution. In case of post execution corruption of S*, Sim obtains (a_0, a_1) and needs to equivocate $y_{\overline{\sigma}}$ such that it opens to $a_{\overline{\sigma}}$. Sim performs this by programming $\mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||r_{\overline{\sigma}}) = y_{\overline{\sigma}} \oplus a_{\overline{\sigma}}$. The internal state of S* is revealed to $\mathcal{Z}$ who halts with an output.

- **Case 4. R* is corrupted in the first round, S* is corrupted in the second round:** This is a trivial case since both parties are corrupted by the adversary. The parties are controlled by A/Sim/$\mathsf{A}_{\mathsf{Int}}$ in the real/ideal/internal world. $\mathsf{A}_{\mathsf{Int}}$ generates the second OT message similar to Case 4 ((S*, R*) case) of static proof. At the end of execution, $\mathsf{A}_{\mathsf{Int}}$ outputs a special symbol $\perp$ on behalf of the corrupted parties and hands over the internal state to Sim who in turn forwards it to $\mathcal{Z}$.

  Post Execution. There is no post execution corruption since both parties are corrupted and $\mathcal{Z}$ halts with an output computed from the internal state of $\mathsf{A}_{\mathsf{Int}}$.

Indistinguishability: Here we show that the ideal world view $\text{IDEAL}_{\mathcal{F},\mathsf{Sim},\mathcal{Z}}(1^\kappa, z)$ of $\mathcal{Z}$ is indistinguishable from the real world view $\text{REAL}_{\mathcal{F},\mathsf{A},\mathcal{Z}}(1^\kappa, z)$ of $\mathcal{Z}$. We denote $\text{REAL}_{\mathcal{F},\mathsf{A},\mathcal{Z}}(1^\kappa, z)$ as hybrid HYB_R and show that in each simulation case HYB_R is indistinguishable from the ideal world by relying on the static indistinguishability proof.

Case 1(A). R is honest in the first round, S is honest in the second round, R* is corrupted after second OT message: Simulation of both OT messages follows along the same direction as Case 1 of static proof. When R* gets corrupted after second OT message Sim obtains σ and he can provide sk_σ as the secret key for branch σ. Equivocation is successful due to Property 4ii of DME, i.e. σ is statistically hidden in pk when the crs is generated in decryption mode. In case of post execution corruption, Sim successfully equivocates $y_{\overline{\sigma}}$, s.t. it unlocks $a_{\overline{\sigma}}$ by programming $\mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||r_{\overline{\sigma}})$. A can query $r_{\overline{\sigma}}$ to $\mathcal{F}_{\mathsf{RO2}}$ only with negligible probability and hence equivocation is successful. This follows from Property 4iii of DME, i.e. indistinguishability of ciphertexts in the decryption mode.

Case 1(B). R is honest in the first round, S is honest in the second round, R is honest after second OT message: In this case we can consider that the post execution corruption occurs in two phases - first R* gets corrupted and only after that S* gets corrupted. Then it becomes identical to the previous case (case 1(A) of adaptive proof) and indistinguishability follows from the previous argument.

Case 2. R is honest in the first round, S^* is corrupted in the second round: The first message is computed according to Case 1(i) of static proof, which is identical to Case 2(i) of static proof. The rest of the simulation proceeds as Case 2 of static proof. Indistinguishability follows from the indistinguishability of $\mathsf{HYB}_{I.2}$ from HYB_R in the static proof. In case of post execution corruption, equivocation of R's view is possible since the crs is in decryption mode and Sim has secret keys for both branches. sk_σ can be provided as the secret key for branch σ. Indistinguishability follows from Property 4ii of DME.

Case 3. R^* is corrupted in the first round, S is honest in the second round: Indistinguishability follows from the indistinguishability of $\mathsf{HYB}_{I.3}$ from HYB_R in the static proof. In case of post execution corruption equivocation can fail if during the simulation $\mathsf{A}_{\mathsf{Int}}$ unlocks $r_{\overline{\sigma}}$ from $s_{\overline{\sigma}}$ and queries it to $\mathcal{F}_{\mathsf{RO2}}$. However, this occurs with negligible probability due to Property 3 of DME, i.e. ciphertext indistinguishability in messy mode, and the random oracle assumption.

Case 4. R^* is corrupted in the first round, S^* is corrupted in the second round: This is identical to case 4 (both $(\mathsf{S}^*, \mathsf{R}^*)$ are corrupted) in static proof. Hence, indistinguishability of simulation for this case follows from indistinguishability of $\mathsf{HYB}_{I.4}$ from HYB_R in the static proof. □

4.5.3 Instantiation of the Framework

Our framework can be instantiated by instantiating the underlying DME scheme under the LWE and DDH assumptions. Furthermore, we observe that our protocol can be further optimized, while considering the DDH-based instantiation of the DME. In Figure. 4.3 we present our optimized protocol $\pi_{\mathsf{OT}}^{\mathrm{DDH}}$. It differs from the original framework since protocol $\pi_{\mathsf{OT}}^{\mathrm{DDH}}$ does not send separate messages (s_0, s_1) for encrypting the random pads. Rather, the random pads and the messages are encrypted in the same ciphertext (w_0, w_1). This protocol incurs a computation cost of 11 exponentiations and 5 random oracle queries and it has an optimal round complexity of 2. It requires sending a κ bit string - c, two ℓ bit strings - (w_0, w_1); and 4 group elements– (u_0, u_1) and (g, h). Interestingly, it is the first round-optimal adaptively-secure OT protocol and it has an overhead of 5 random oracle queries and κ-bit string communication overhead over the static protocol of [PVW08]. The security of our protocol follows from the security of the OT framework and the properties of the DDH-based DME instantiation presented in Section. 4.4.2. We summarize the security proof in Theorem. 4.5.3.

Theorem 4.5.3. *If the Decisional Diffie Hellman problem is hard in group $\mathbb{G}$ then protocol π_{OT}^{DDH} securely realizes the $\mathcal{F}_{OT}$ functionality against adaptive active adversaries in $(\mathcal{F}_{RO1}, \mathcal{F}_{RO2})$- hybrid model.*

Figure 4.3: Optimized Adaptively-Secure Oblivious Transfer Protocol from DDH

$\pi_{\mathsf{OT}}^{\mathrm{DDH}}$

- **Functionalities:** Random oracles $\mathcal{F}_{\mathsf{RO1}} : \{0,1\}^{2\kappa} \rightarrow 4\mathbb{G}$ and $\mathcal{F}_{\mathsf{RO2}} : \{0,1\}^{\kappa} \times \mathbb{G} \rightarrow \{0,1\}^{\ell}$.
- **Private Inputs:** S has ℓ-bit input messages (a_0, a_1) and R has an input bit σ.

Choose:

- R samples $c \leftarrow_R \{0,1\}^{\kappa}$.
- R generates $\mathsf{crs} = (g_0, g_1, h_0, h_1)$, s.t. $(g_0, g_1, h_0, h_1) \leftarrow \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||c)$.
- R samples $\alpha \leftarrow_R \mathbb{Z}_p$ and computes $(g, h) = (g_\sigma^\alpha, h_\sigma^\alpha)$.
- R sends $\{c, (g, h)\}$ to S.

Transfer:

- S generates the $\mathsf{crs} = (g_0, g_1, h_0, h_1)$, s.t. $(g_0, g_1, h_0, h_1) \leftarrow \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||c)$.
- S samples $r_0, r_1, s_0, s_1 \leftarrow_R \mathbb{Z}_p$.
- S computes $u_0 = g_0^{r_0} h_0^{s_0}$ and $u_1 = g_1^{r_1} h_1^{s_1}$.
- S sets $w_0 = \mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||g^{r_0} h^{s_0}) \oplus a_0$ and $w_1 = \mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||g^{r_1} h^{s_1}) \oplus a_1$.
- S sends $\{(u_0, w_0), (u_1, w_1)\}$ to R.

Local Computation by R:

- Computes a_σ as $a_\sigma = w_\sigma \oplus \mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||u_\sigma^\alpha)$.

4.5.4 Receiver Equivocal Oblivious Transfer

We can optimize our adaptive π_{OT} protocol to obtain an efficient static OT protocol, denoted as π'_{OT}. It is similar to the [PVW08] framework, except here we generate the crs using a PRO. We can obtain π'_{OT} from π_{OT} by removing the random oracle invocation $\mathcal{F}_{\mathsf{RO2}}$. $\mathcal{F}_{\mathsf{RO2}}$ is not required since static corruption does not demand equivocation, of sender's message, by the Sim. The security of our protocol is summarized in Theorem 4.5.4 and the proof is similar to the static security proof of π_{OT}.

Theorem 4.5.4. *If DME is a samplable dual mode encryption scheme then protocol π_{OT} securely realizes the $\mathcal{F}_{OT}$ functionality against static active adversaries in $\mathcal{F}_{RO1}$- hybrid model.*

Interestingly, π'_{OT} behaves like an $\mathsf{RE\text{-}OT}$, i.e. the view of the receiver can be equivocated if the receiver gets adaptively corrupted during/after the protocol execution. We analyze this property as two exhaustive subcases as follows : (1) R gets corrupted before the first OT message is sent (2) R gets corrupted after the first OT message is sent. For the first case, the simulator has not sent any message on behalf of R and hence equivocation is not required. In this case the crs is set to messy mode. For the second case, the simulator sets the crs in the decryption mode which enables him to equivocate R's view on obtaining σ, after R gets corrupted.

When instantiated under the DDH assumption, the receiver equivocal OT can be further improved by applying the optimizations from Section. 4.5.3. The protocol requires 11 exponentiations and 2 random oracle queries. The communication involves sending a κ bit string and 6 group elements.

4.6 Adaptively Secure 1-out-of-N Oblivious Transfer

The work of [NP05] implements $\mathcal{F}_{\mathsf{N\text{-}OT}}$, against static adversaries in the $\mathcal{F}_{\mathsf{OT}}$-hybrid model by relying on pseudorandom functions. S has N strings $(a_1, a_2, \ldots, a_{\mathsf{N}})$ as input and R has a $\log \mathsf{N}$ bit input choice string σ. S samples $2 \log \mathsf{N}$ random pads $p_0^i, p_1^i \leftarrow \{0,1\}^\kappa$ for $i \in [\log \mathsf{N}]$. The two parties invoke $\log \mathsf{N}$ copies of $\mathcal{F}_{\mathsf{OT}}$ with S's input as (p_0^i, p_1^i) and R's input as σ_i respectively. The ith invocation of $\mathcal{F}_{\mathsf{OT}}$ outputs $p_{\sigma_i}^i$ to R, i.e the pad corresponding to his σ_i choice bit. Finally, S encrypts a_j as w_j using the pads corresponding to the bit representation of j, $j \in [\mathsf{N}]$. The message a_j is encrypted as follows :

$$w_j = a_j \oplus \bigoplus_{i=1}^{\log \mathsf{N}} \mathrm{PRF}_{p_c^i}(j),$$

where $c = j_i$, i.e. the ith bit of string j. R obtains $(w_0, w_1, \ldots, w_{\mathsf{N}})$ and decrypts w_σ for which he possesses all the $\log \mathsf{N}$ pads. For other w values, R lacks at least one random pad and security follows by applying PRF on that secret random pad. The transformation communicates N ciphertexts, and requires $\log \mathsf{N}$ invocations of $\mathcal{F}_{\mathsf{OT}}$ and $\mathsf{N} \log \mathsf{N}$ evaluations of a PRF. It guarantees security against a statically corrupted active adversary.

We show in Fig. 4.4 that the same transformation can be made adaptively-secure (by protocol $\pi_{\mathsf{N\text{-}OT}}$) by implementing the underlying $\mathcal{F}_{\mathsf{OT}}$ functionality in an adaptive secure manner. In addition, we replace PRF with a programmable random oracle $\mathcal{F}_{\mathsf{RO}}$ and modify the formation of w_j as follows, where j_i denotes the ith bit of j.

$$w_j = a_j \oplus \mathcal{F}_{\mathsf{RO}}(\mathsf{sid}||j, p_{j_1}^1||p_{j_2}^2||\ldots||p_{j_{\log \mathsf{N}}}^{\log \mathsf{N}}) = a_j \oplus \mathcal{F}_{\mathsf{RO}}(\mathsf{sid}||j, v_j),$$

where $v_j = (p_{j_1}^1||p_{j_2}^2||\ldots||p_{j_{\log \mathsf{N}}}^{\log \mathsf{N}})$. This reduces the $\mathsf{N} \log \mathsf{N}$ PRF evaluations to N random oracle queries. Later, in this section we further demonstrate that the underlying $\mathcal{F}_{\mathsf{OT}}$ need not be full adaptively secure and $\mathsf{RE\text{-}OT}$ suffices for the transformation. Hence, we obtain adaptively-secure 1-out-of-N from $\log \mathsf{N}$ 1-out-of-2 $\mathsf{RE\text{-}OT}$s.

4.6.1 Security

The security of the protocol is proven by constructing a simulator Sim for $\pi_{\mathsf{N\text{-}OT}}$. We first present the static proof and then discuss the adaptive proof. For a statically corrupted S^*, Sim can extract the pads by invoking the simulator for $\mathcal{F}_{\mathsf{OT}}$. Sim forms $\{v_j\}_{j \in [\mathsf{N}]}$ and decrypts $\{w_j\}_{j \in [\mathsf{N}]}$ to unlock all the input messages of S^*, i.e. $\{a_j\}_{j \in [\mathsf{N}]}$, and completes the simulation by invoking $\mathcal{F}_{\mathsf{N\text{-}OT}}$ with the

Figure 4.4: Adaptively-Secure 1-out-of-N Oblivious Transfer Protocol

$\pi_{\mathsf{N\text{-}OT}}$

- **Functionalities:** Random oracle $\mathcal{F}_{\mathsf{RO}} : \{0,1\}^{\kappa+\log\mathsf{N}(1+\kappa)} \rightarrow \{0,1\}^n$. $\mathcal{F}_{\mathsf{OT}}$ denotes a 1-out-of-2 OT functionality.
- **Private Inputs:** S has input messages $\{a_j\}_{j=1}^{\mathsf{N}}$ and R has an input choice string σ, where $|\sigma| = \log\mathsf{N}$.

Choose:

- R invokes $\mathcal{F}_{\mathsf{OT}}$ functionality $\log\mathsf{N}$ times. He invokes the ith copy of $\mathcal{F}_{\mathsf{OT}}$ with input $(\mathsf{rec}, i, \sigma_i)$ for $i \in [\log\mathsf{N}]$.

Transfer:

- S samples $2\log\mathsf{N}$ random pads (p_0^i, p_1^i), where $p_0^i, p_1^i \in \{0,1\}^\kappa$.
- S invokes the ith copy of $\mathcal{F}_{\mathsf{OT}}$ with input $(\mathsf{sen}, i, (p_0^i, p_1^i))$.
- S encrypts his input messages as $w_j = a_j \oplus \mathcal{F}_{\mathsf{RO}}(\mathsf{sid}||j, p_{j_1}^1||p_{j_2}^2||\ldots||p_{j_{\log\mathsf{N}}}^{\log\mathsf{N}})$, for $j \in [\mathsf{N}]$ where $j = j_1 j_2 \ldots j_{\log\mathsf{N}}$.

Local Computation by R:

- Computes a_σ as $a_\sigma = w_\sigma \oplus \mathcal{F}_{\mathsf{RO}}(\mathsf{sid}||\sigma, p_{\sigma_1}^1||p_{\sigma_2}^2||\ldots||p_{\sigma_{\log\mathsf{N}}}^{\log\mathsf{N}})$.

input messages. Indistinguishability follows from the $\mathcal{F}_{\mathsf{OT}}$ hybrid model. For a statically corrupted R^*, Sim can extract the choice bit σ_i by invoking the simulator for ith copy of $\mathcal{F}_{\mathsf{OT}}$. Sim reconstructs σ and invokes $\mathcal{F}_{\mathsf{N\text{-}OT}}$ with σ to obtain a_σ. Sim concludes the simulation by setting w_σ correctly while $\{w_j\}_{j\in[\mathsf{N}]/\sigma}$ are set as random strings. R^* obtains all the random pads (corresponding to σ) from $\log\mathsf{N}$ invocations of $\mathcal{F}_{\mathsf{OT}}$ and constructs v_σ to unlock a_σ. The other $\{a_j\}_{j\in[\mathsf{N}]/\sigma}$ values remain hidden since R^* lacks at least one random pad for $\{v_j\}_{j\in[\mathsf{N}]/\sigma}$, and hence $\mathcal{F}_{\mathsf{RO}}(\mathsf{sid}||j, v_j)$ will be indistinguishable from a random string, except with negligible probability, due to the random oracle assumption.

In order to prove adaptivity, we need the property of equivocation. The simulated messages have to be equivocated appropriately so that they are indistinguishable from the real world messages of honest parties. Sim can equivocate the simulated message (consisting of only $\mathcal{F}_{\mathsf{OT}}$ messages) of R by invoking the adaptive simulator for $\mathcal{F}_{\mathsf{OT}}$. The simulated message of S consists of the $\mathcal{F}_{\mathsf{OT}}$ messages and the simulated ciphertexts. The $\mathcal{F}_{\mathsf{OT}}$ messages can be trivially equivocated by invoking the adaptive simulator for $\mathcal{F}_{\mathsf{OT}}$. The simulated ciphertexts can be equivocated by programming $\mathcal{F}_{\mathsf{RO}}(\mathsf{sid}||j, v_j) = w_j \oplus a_j$, for $j \in [\mathsf{N}]/\sigma$. Adversary can query $\mathcal{F}_{\mathsf{RO}}(\mathsf{sid}||j, v_j)$ with negligible probability since he lacks atleast one pad in v_j and hence simulator can program $\mathcal{F}_{\mathsf{RO}}(\mathsf{sid}||j, v_j)$ successfully to equivocate correctly. The proof of indistinguishability is similar to the one for statically corrupted R^*.

4.6.2 Efficiency

Our protocol invokes $\mathcal{F}_{\mathsf{OT}}$ functionality $\log \mathsf{N}$ times and queries $\mathcal{F}_{\mathsf{RO}}$ for N times. It incurs communication of $\log \mathsf{N}$ copies of $\mathcal{F}_{\mathsf{OT}}$ and N ciphertexts of size n bits.

4.6.3 Optimized Version

We observe that our transformation continues to be adaptively-secure, despite replacing the adaptively-secure 1-out-of-2 OTs in the above result with RE-OT. When S^* or R^* is corrupted, inputs are extracted by invoking the simulator for the actively-secure static OT protocol. For adaptive security, we need equivocation of S's and R's views if corruption occurs during the course of execution or at the end of protocol. In the transformation, S's view consists of the OT messages and ciphertexts, i.e. $\{w_j\}_{j\in[\mathsf{N}]}$. A simulator, playing the role of S can trivially simulate the OT messages by running the honest S algorithm with random pads, which are independent of S's inputs. The ciphertexts are equivocated by programming $\mathcal{F}_{\mathsf{RO}}(\mathsf{sid}||j, v_j)$ (see Section 4.6.1). On the other hand, R's view consists only of the OT messages which can be trivially equivocated by relying on the receiver equivocal property of the OT. Our π'_{OT} protocol (Section 4.5.4) is a statically-secure protocol satisfying receiver equivocal property; hence we can plug in our protocol to obtain adaptively-secure 1-out-of-N OTs. The security of $\pi_{\mathsf{N}\text{-}\mathsf{OT}}$ is summarized in Theorem 4.6.1.

Theorem 4.6.1. *If $\mathcal{F}_{\mathsf{RO}}$ is a programmable random oracle and $\mathcal{F}_{\mathsf{OT}}$ is an actively secure RE-OT, then $\pi_{\mathsf{N}\text{-}\mathsf{OT}}$ UC-securely realizes the $\mathcal{F}_{\mathsf{N}\text{-}\mathsf{OT}}$ functionality in the $\mathcal{F}_{\mathsf{OT}}$ hybrid model against adaptive (without erasures) active adversaries.*

4.7 Adaptively Secure Oblivious Transfer Extension

In this section we present our semi-honest and actively secure OT Extension protocols which are secure against adaptive adversaries, without assuming erasures.

4.7.1 Adaptively Secure OT Extension against Semi-Honest Adversaries

We prove that the semi-honest OT Extension protocol of [ALSZ13] (denoted as ALSZ13 hereby) can be made adaptively secure using PRO. We initiate our proof by briefly recalling the protocol of ALSZ13. In ALSZ13, S has m pairs of messages - $\left\{\mathbf{x}_{j,0}, \mathbf{x}_{j,1}\right\}$ for $j \in [m]$, and R has m selection bits, denoted as vector $\mathbf{r} = (r_1, \cdots, r_m)$. R samples two random strings $(\mathbf{k}_i^0, \mathbf{k}_i^1)$ and S samples a random string s, $i \in [\kappa]$. R and S performs the seed OT phase by invoking $\mathcal{F}_{\mathsf{OT}}$ on inputs $(\mathbf{k}_i^0, \mathbf{k}_i^1)$ and s_i as sender and receiver respectively. S obtains $\mathbf{k}_i^{\mathsf{s}_i}$ from $\mathcal{F}_{\mathsf{OT}}$. R computes matrices $\mathbf{B}$ and $\mathbf{E}$ s.t. $\mathbf{B}^i = G(\mathbf{k}_i^0)$ and $\mathbf{E}^i = \mathbf{r}$, where G is a PRG. R sends a correction matrix $\mathbf{D}$ to S, s.t. $\mathbf{D}^i = \mathbf{B}^i \oplus G(k_i^1) \oplus \mathbf{E}^i$. S forms the matrix $\mathbf{Q}$ using $G(\mathbf{k}_i^{s_i})$, s and $\mathbf{D}$. The jth row of $\mathbf{Q}$ is used to encrypt the jth message pair $\{\mathbf{x}_{j,0}, \mathbf{x}_{j,1}\}_{j\in[m]}$ using pads $H(j, \mathbf{Q}_j)$ and $H(j, \mathbf{Q}_j \oplus \mathbf{s})$ respectively, where H is a correlation robust function [IKNP03]. S sends the ciphertexts to R, who decrypts and obtains $\mathbf{x}_{j,r_j}$ using $H(j, \mathbf{B}_j)$. The ALSZ13 is similar to the protocol presented in Fig. 4.5, except $\mathcal{F}_{\mathsf{RO1}}$ and $\mathcal{F}_{\mathsf{RO2}}$ will be replaced by G and H. Security against a statically corrupt S^* follows from the sender privacy of $\mathcal{F}_{\mathsf{OT}}$, which ensures that one of R's input (acting as sender in $\mathcal{F}_{\mathsf{OT}}$) to $\mathcal{F}_{\mathsf{OT}}$ remains hidden from S. On the other hand, a statically corrupt R can break honest sender privacy if he obtains both $\mathbf{x}_{j,0}$ and $\mathbf{x}_{j,1}$ for some j. However, this happens with negligible probability since R has to either guess the value of $\mathbf{s}$ or obtain a collision in the random oracle query, s.t. $H(j, \mathbf{B}_j) = H(j, \mathbf{B}_j \oplus \mathbf{s})$. We refer to [ALSZ13] for the full proof.

While considering adaptive security, the messages of S and R require equivocation. These messages can be classified into three exhaustive cases. Below, we give a high level overview for equivocating each case :

- $\mathcal{F}_{\mathsf{OT}}$ messages: It can be observed that S and R invoke $\mathcal{F}_{\mathsf{OT}}$ on random inputs - $\mathbf{s}_i$ and $(\mathbf{k}_i^0, \mathbf{k}_i^1)$ respectively, independent of their actual inputs. This can be leveraged to reduce the assumption on $\mathcal{F}_{\mathsf{OT}}$. More specifically, $\mathcal{F}_{\mathsf{OT}}$ can be secure only against static adversaries, since the simulator can simulate the view of an honest sender (resp. receiver) by running the honest sender (resp. receiver) algorithm on random inputs. When sender (resp. receiver) gets corrupted, the simulator can produce the view, generated using random inputs, which is indistinguishable from the honest sender's (resp. receiver's) view.

- R's messages : It consists of matrix $\mathbf{D}$ which depends on R's input $\mathbf{r}$. When R gets corrupted, the simulator has to equivocate R's message, i.e. matrix $\mathbf{D}$, based on R's input. This is ensured by replacing G with a PRO $\mathcal{F}_{\mathsf{RO1}}$.

- S's messages: It consists of messages - $(\mathbf{y}_{j,0}, \mathbf{y}_{j,1})$, which are dependent on S's inputs $(\mathbf{x}_{j,0}, \mathbf{x}_{j,1})$. When S gets corrupted, the simulator has to equivocate S's message based on S's inputs. This is ensured by replacing H with a PRO $\mathcal{F}_{\mathsf{RO2}}$.

The explanation for the last two subcases will be clear from the next few paragraphs, where we elaborately discuss the equivocation cases. We consider equivocation of S and R in two separate cases. Our passively secure adaptive OT extension protocol $\pi^{\mathsf{p}}_{\mathsf{ALSZ}}$ is presented in Fig. 4.5 and the security proof has been summarized in Theorem 4.7.1.

Figure 4.5: Adaptive OT Extension Protocol secure against passive adversaries

Protocol $\pi^{\mathsf{p}}_{\mathsf{ALSZ}}$ for obtaining $\binom{2}{1}$-OT^{m}_{ℓ} from $\binom{2}{1}$-OT^{κ}_{m}

- **Input of S:** m pairs $\{\mathbf{x}_{j,0}, \mathbf{x}_{j,1}\}_{j \in [m]}$ of ℓ bit strings.
- **Input of R:** m selection bit vector $\mathbf{r} = (r_1, \cdots, r_m)$ such that each $r_j \in \{0,1\}$.
- **Functionalities:** Random Oracles $\mathcal{F}_{\mathsf{RO1}} : \{0,1\}^{2\kappa} \to \{0,1\}^m$ and $\mathcal{F}_{\mathsf{RO2}} : \{0,1\}^{\kappa} \times [m] \times \{0,1\}^{\kappa} \to \{0,1\}^{\ell}$ respectively. $\mathcal{F}_{\mathsf{OT}}$ denotes a 1-out-of-2 OT functionality.
- **Notations:** In the protocol $j \in [m]$ and $i \in [\kappa]$, unless specified otherwise.

Seed OT Phase:

1. S chooses $\mathbf{s} \leftarrow \{0,1\}^{\kappa}$ at random.
2. S computes the first message of π_{OT} as $\pi^{1}_{\mathsf{OT}}(\mathbf{s})$.
3. R chooses κ pairs of seeds $(\mathbf{k}^0_i, \mathbf{k}^1_i)$ each of length κ.
4. S and R invokes the $\mathcal{F}_{\mathsf{OT}}$ functionality for κ times with inputs $(\mathsf{rec}, \mathsf{sid}, \mathbf{s}_i)$ and $(\mathsf{rec}, \mathsf{sid}, (\mathbf{k}^0_i, \mathbf{k}^0_i))$ respectively. S receives $\{\mathbf{k}^{s_i}_i\}$ as output.

OT Extension Phase :

1. R forms three $m \times \kappa$ matrices $\mathbf{B}$, $\mathbf{E}$ and $\mathbf{C}$ in the following way and sends $\mathbf{D}$ to S:
   - Set $\mathbf{B}^i = \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||\mathbf{k}^0_i)$.
   - Set $\mathbf{E}_j = (r_j|| \dots ||r_j)$. Clearly, $\mathbf{E}^i = \mathbf{r}$.
   - Set $\mathbf{D}^i = \mathbf{B}^i \oplus \mathcal{F}_{\mathsf{RO1}}(\mathbf{k}^1_i) \oplus \mathbf{E}^i$.
2. On receiving $\mathbf{D}$, S forms $m \times \kappa$ matrix $\mathbf{Q}$ with the jth column of $\mathbf{Q}$ set as $\mathbf{Q}^i = \big(s_i \odot \mathbf{D}^i\big) \oplus \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||\mathbf{k}^{s_i}_i)$. Clearly, (i) $\mathbf{Q}^i = \big(\mathbf{B}^i \oplus (s_i \odot \mathbf{E}^i)\big)$ and (ii) $\mathbf{Q}_j = \big(\mathbf{B}_j \oplus (\mathbf{s} \odot \mathbf{E}_j)\big) = \big(\mathbf{B}_j \oplus (\mathbf{s} \odot r_j)\big)$.
3. For every $j \in [m]$, S computes $\mathbf{y}_{j,0} = \mathbf{x}_{j,0} \oplus \mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||j, \mathbf{Q}_j)$ and $\mathbf{y}_{j,1} = \mathbf{x}_{j,1} \oplus \mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||j, \mathbf{Q}_j \oplus \mathbf{s})$. S sends $\{\mathbf{y}_{j,0}, \mathbf{y}_{j,1}\}_{j \in [m]}$ to R.
4. For every $j \in [m]$, R recovers $\mathbf{z}_j = \mathbf{y}_{j,r_j} \oplus \mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||j, \mathbf{B}_j)$. R outputs $\{\mathbf{z}_j\}_{j \in [m]}$.

Theorem 4.7.1. *The protocol π^{p}_{ALSZ} UC-securely realizes the Oblivious Transfer Extension functionality in the $(\mathcal{F}_{OT}, \mathcal{F}_{RO1}, \mathcal{F}_{RO2})$ hybrid model against adaptive (without erasures) passive adversaries.*

Proof. We refer to the original paper of [ALSZ13] for the security proof against static adversaries. For the adaptive proof, we require equivocation of the honest party's view. We consider two cases separately - one for S's view and another for R's view.

- Equivocation of S's view: When S is honest, Sim plays the role of S in the internal world.

The view of S consists of his input $\left\{\mathbf{x}_{j,0}, \mathbf{x}_{j,1}\right\}$, random vector s, matrix $\mathbf{Q}$ and the ciphertexts $(\mathbf{y}_{j,0}, \mathbf{y}_{j,1})$. The s vector and $\mathbf{Q}$ matrix is independent of S's inputs and can be simulated correctly without equivocation. The ciphertexts are simulated by relying on the programmability property of $\mathcal{F}_{\mathsf{RO2}}$. We first consider the case where both parties are honest. The formal simulation for S is presented as follows. Sim invokes the $\mathcal{F}_{\mathsf{OT}}$ simulator with random input s_i to simulate the ith seed-OT. Sim generates $\mathbf{Q}$ following honest sender algorithm. In the OT Extension phase, Sim sends random values for $\mathbf{y}_{j,b}$, where $b \in \{0,1\}$. In case of post execution corruption of S, Sim obtains the inputs of S. Sim provides s and $\mathbf{Q}$ as part of the view, and this is indistinguishable from the real world view. In addition, Sim opens $\mathbf{y}_{j,b}$ to $\mathbf{x}_{j,b}$ by programming $\mathcal{F}_{\mathsf{RO2}}$ as follows:

$$\mathcal{F}_{\mathsf{RO2}}(j, \mathbf{Q}_j) = \mathbf{y}_{j,0} \oplus \mathbf{x}_{j,0},$$

$$\mathcal{F}_{\mathsf{RO2}}(j, \mathbf{Q}_j \oplus \mathrm{s}) = \mathbf{y}_{j,1} \oplus \mathbf{x}_{j,1}.$$

Equivocation is successful due to the randomness of s and random oracle assumption. An adversary $\mathsf{A}_{\mathsf{Int}}$ can prevent Sim from programming $\mathcal{F}_{\mathsf{RO2}}$ if he queries $\mathbf{Q}_j$ or $\mathbf{Q}_j \oplus \mathrm{s}$ to $\mathcal{F}_{\mathsf{RO2}}$. This happens with negligible probability as $\mathbf{Q}$ and s are randomly chosen. This proves that real and ideal world are indistinguishable. Now, we consider the case where $\mathsf{A}_{\mathsf{Int}}$ corrupts R^*. Sim obtains the choice vector $\mathbf{r}$. Sim invokes the OT Extension ideal world protocol with $\mathbf{r}$ and obtains $\{\mathbf{x}_{j,r_j}\}_{j\in[m]}$. Sim computes y_{j,r_j} honestly like an honest sender, s.t. y_{j,r_j} opens to $\mathbf{x}_{j,r_j}$. In case of post execution corruption, Sim obtains $(\mathbf{x}_{j,0}, \mathbf{x}_{j,1})$ and he programs $\mathcal{F}_{\mathsf{RO2}}(j, \mathbf{Q}_j \oplus (\mathrm{s} \odot \overline{r_j}))$, s.t. $y_{j,\overline{r_j}}$ opens to $\mathbf{x}_{j,\overline{r_j}}$. To prevent equivocation $\mathsf{A}_{\mathsf{Int}}$ has to guess the value of s and query $\mathbf{Q}_j \oplus (\mathrm{s} \odot \overline{r_j}) = \mathbf{B}_j \oplus \mathrm{s}$ to $\mathcal{F}_{\mathsf{RO2}}$ which happens with negligible probability, proving indistinguishability of the two worlds.

– Equivocation of R's view: When R is honest, Sim plays the role of R in the internal world. The view of R consists of his inputs, the seeds $(\mathbf{k}_i^0, \mathbf{k}_i^1)$ and the matrices, $\mathbf{B}$, $\mathbf{E}$ and $\mathbf{D}$. The seeds and $\mathbf{B}$ matrix can be simulated without any equivocation since they are independent of R's input. The other two matrices are simulated by programming $\mathcal{F}_{\mathsf{RO1}}$, since they depend on R's input. We first consider the case where both parties are uncorrupted. The formal simulation is presented as follows. Sim randomly generates $(\mathbf{k}_i^0, \mathbf{k}_i^1)$ and simulates ith $\mathcal{F}_{\mathsf{OT}}$ by invoking $\mathcal{F}_{\mathsf{OT}}$ with inputs $(\mathbf{k}_i^0, \mathbf{k}_i^1)$ like an honest R. Sim sets $\mathbf{D}$ matrix randomly and the $\mathbf{B}$ and $\mathbf{E}$ matrices are constructed later as they are not required for the simulation. Sim sends $\mathbf{D}$ to S and ends simulation as both parties are honest. The adversary $\mathsf{A}_{\mathsf{Int}}$ (without corrupting S) observes R's message, which consists only of $\mathbf{D}$ matrix. Based on that, he cannot distinguish between real and ideal world since in ideal world $\mathbf{D}$ is a random matrix, whereas in real world, it is padded with $\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}, \mathbf{k}_i^0)$ and $\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}, \mathbf{k}_i^1)$, where $\mathbf{k}_i^0$ and $\mathbf{k}_i^1$ remains hidden from $\mathsf{A}_{\mathsf{Int}}$. Thus, indistinguishability follows from the random oracle assumption. In case of post execution corruption of R, Sim obtains $\mathbf{r}$ and he programs $\mathcal{F}_{\mathsf{RO}}(\mathsf{sid}||\mathbf{k}_i^0)$ and $\mathcal{F}_{\mathsf{RO}}(\mathsf{sid}||\mathbf{k}_i^1)$ s.t. the following holds true:

$$\mathcal{F}_{\mathsf{RO1}}(\mathbf{k}_i^0) \oplus \mathcal{F}_{\mathsf{RO1}}(\mathbf{k}_i^1) = \mathbf{D}^i \oplus \mathbf{r}. \tag{4.1}$$

The $\mathbf{B}$ matrix is constructed after $\mathcal{F}_{\mathsf{RO1}}$ has been programmed according to Eq. 4.1. Equivocation is successful since $\mathsf{A}_{\mathsf{Int}}$ does not obtain $\mathbf{k}_i^0$ and $\mathbf{k}_i^1$ from $\mathcal{F}_{\mathsf{OT}}$ and he has to guess it, which occurs with negligible probability, in order to prevent equivocation. If $\mathsf{A}_{\mathsf{Int}}$ corrupts S^*, then Sim obtains s from $\mathcal{F}_{\mathsf{OT}}$ simulator (since S^* is the OT receiver) and he sets $\mathcal{F}_{\mathsf{RO1}}(\mathbf{k}_i^{\mathbf{s}_i})$ randomly concluding the simulation. In case of post execution corruption of R^*, Sim obtains $\mathbf{r}$ and constructs $\mathbf{E}$ matrix s.t. $\mathbf{E}^i = \mathbf{r}$. Sim equivocates $\mathbf{D}$ matrix by programming $\mathcal{F}_{\mathsf{RO1}}(\mathbf{k}_i^{\overline{\mathbf{s}_i}})$ as follows:

$$\mathcal{F}_{\mathsf{RO1}}(\mathbf{k}_i^{\overline{\mathbf{s}_i}}) = \mathcal{F}_{\mathsf{RO1}}(\mathbf{k}_i^{\mathbf{s}_i}) \oplus \mathbf{D}^i \oplus \mathbf{r}. \tag{4.2}$$

The $\mathbf{B}$ matrix is constructed after $\mathcal{F}_{\mathsf{RO1}}$ has been programmed according to Eq. 4.2. Equivocation is successful since $\mathsf{A}_{\mathsf{Int}}$ does not obtain $\mathbf{k}_i^{\overline{\mathbf{s}_i}}$ from $\mathcal{F}_{\mathsf{OT}}$ and he has to guess it, in order to prevent equivocation. However, it occurs with negligible probability and hence the two worlds are indistinguishable.

□

4.7.2 Adaptively Secure OT Extension against Active Adversaries

The ALSZ13 protocol is actively secure against S^* and passively secure against R^*, assuming seed OTs to be actively secure. An active R^* can send a bad $\mathbf{D}$ matrix such that $\mathbf{Q}$ is malformed and it leaks the s vector. We refer to [ALSZ15] (denoted as ALSZ henceforth) for the exact attack. To address this attack, the ALSZ15 protocol introduces column wise consistency check for every pair of columns. Moreover, for active security we require that the simulator Sim can extract a corrupted party's input from the protocol - s from S^* and $(\mathbf{k}_i^0, \mathbf{k}_i^1)$ from R^*, even when the parties are adaptively corrupted. For adaptive security, the simulator also has to simulate the consistency checks when Sim plays the role of R in the ideal world, in addition to the usual equivocation of views of honest parties.

In order to extract corrupted parties' inputs and simulate the consistency checks properly, we observe that the $\mathcal{F}_{\mathsf{OT}}$ messages are random in the OT extension protocol of ALSZ13 and ALSZ15. Hence, we can modify the protocol s.t. the parties invoke $\mathcal{F}_{\mathsf{ROT}}$ (Fig. 2.4) instead of $\mathcal{F}_{\mathsf{OT}}$. Recall, that $\mathcal{F}_{\mathsf{ROT}}$ is an OT functionality where R has an input whereas S does not have any input. The functionality returns random messages to S and one of the random messages to R based on his input. Formally stating, $\mathcal{F}_{\mathsf{ROT}}$ takes choice bit σ as input from receiver of $\mathcal{F}_{\mathsf{ROT}}$, i.e. R_{OT}, and generates two random pads (a_0, a_1) for sender (i.e. S_{OT}) of $\mathcal{F}_{\mathsf{ROT}}$. It sends a_σ to R_{OT} and (a_0, a_1) to S_{OT}. On a high level, when $\mathcal{F}_{\mathsf{ROT}}$ is invoked in our OT Extension protocol the input of S (playing as R_{OT}) to the $\mathcal{F}_{\mathsf{ROT}}$ can be extracted by invoking the static simulator for $\mathcal{F}_{\mathsf{ROT}}$. The seeds for R (playing the role of S_{OT}) are obtained from $\mathcal{F}_{\mathsf{ROT}}$.

For equivocation of R's view, Sim runs the honest S_{OT} algorithm to obtain random seeds. The consistency checks are simulated by programming $\mathcal{F}_{\mathsf{RO1}}$ and $\mathcal{F}_{\mathsf{RO2}}$. For equivocation of S's view, Sim invokes the $\mathcal{F}_{\mathsf{ROT}}$ simulator for R_{OT} with a random s vector. Indistinguishability follows since the value of s, sampled by honest S, is random in real world as well. The ciphertexts $(\mathbf{y}_{j,0}, \mathbf{y}_{j,1})$ are equivocated by programming $\mathcal{F}_{\mathsf{RO3}}$. Invoking $\mathcal{F}_{\mathsf{ROT}}$ has a significant advantage - $\mathcal{F}_{\mathsf{ROT}}$ can be adaptively implemented by a static receiver equivocal OT (Sec. 3.3), since S does not have any message, whereas implementing $\mathcal{F}_{\mathsf{OT}}$ in an adaptive way is expensive.

Now we formally describe our protocol $\pi^{\mathsf{a}}_{\mathsf{ALSZ}}$, which has been presented in Fig. 4.6 and 4.7. Our protocol is same as ALSZ15, except here the PRG, hash and correlation robust function invocations of ALSZ15 are replaced by PRO invocations. Our protocol has a seed OT phase similar to the $\pi^{\mathsf{p}}_{\mathsf{ALSZ}}$ protocol. The OT extension phase is divided into two parts, based on R's and S's role. R's role consist of phase I of OT extension, followed by the consistency checks and finally phase II of OT extension consists of S's role in the OT extension. Similar to ALSZ15, our protocol also contains column wise consistency check for every pair of columns. It has been recently identified by [ALSZ15] team that the checks leak κ^2 bits of randomness, on R's end. In order to counter that, they add κ^2 bits of randomness by adding κ columns and κ rows, as dummy choice bits τ. As a result the new choice bit string is $\mathbf{r}' = \mathbf{r}||\tau$. The checks ensure that the same $\mathbf{r}'$ has been used in all the columns of $\mathbf{D}$, i.e. $\mathbf{E}^i = \mathbf{r}'$ for all $i \in [k]$, where $k = 2\kappa$. Our static proof of security follows from the security proof of [ALSZ15] and we refer to their paper for the static proof against an active adversary. Adaptive security for our protocol can be proved against active adversaries in the PROM model, provided the underlying seed OTs are adaptive actively secure implementation of $\mathcal{F}_{\mathsf{ROT}}$ (Fig. 2.4) functionality. Security of $\pi^{\mathsf{a}}_{\mathsf{ALSZ}}$ has been summarized in Theorem 4.7.2.

Theorem 4.7.2. *If $\mathcal{F}_{RO1}$, $\mathcal{F}_{RO2}$ and $\mathcal{F}_{RO3}$ are programmable random oracles then π^{a}_{ALSZ} UC-securely realizes the Oblivious Transfer Extension functionality in the $\mathcal{F}_{ROT}$ hybrid model against adaptive (without erasures) active adversaries.*

Figure 4.6: Adaptive OT Extension Protocol secure against active adversaries

Protocol $\pi^{\mathsf{a}}_{\mathsf{ALSZ}}$ for obtaining $\binom{2}{1}$-OT^{m}_{ℓ} from $\binom{2}{1}$-OT^{κ}_{m}

- **Input of S:** m pairs $\{\mathbf{x}_{j,0}, \mathbf{x}_{j,1}\}_{j \in [m]}$ of ℓ bit strings.
- **Input of R:** m selection bit vector $\mathbf{r} = (r_1, \cdots, r_m)$ such that each $r_j \in \{0,1\}$.
- **Security Parameter:** $k = 2\kappa$.
- **Functionalities:** $\mathcal{F}_{\mathsf{RO1}} : \{0,1\}^{2\kappa} \to \{0,1\}^{m+\kappa}$, $\mathcal{F}_{\mathsf{RO2}} : \{0,1\}^{m+\kappa} \to \{0,1\}^{\kappa}$ and $\mathcal{F}_{\mathsf{RO3}} : \{0,1\}^{\kappa} \times [m] \times \{0,1\}^{k} \to \{0,1\}^{\ell}$ respectively. $\mathcal{F}_{\mathsf{OT}}$ denotes OT functionality.
- **Notations:** In the protocol $j \in [m+\kappa]$ and $i \in [k]$, unless specified otherwise.

Seed OT Phase:

1. S samples $\mathbf{s} \in \{0,1\}^{k}$ and invokes $\mathcal{F}_{\mathsf{ROT}}$ with message $(\mathsf{rec}, \mathsf{sid}, (\kappa, \mathbf{s}_i))$ for k times to obtain $\{\mathbf{k}_i^{\mathbf{s}_i}\}$ seeds.

2. R invokes $\mathcal{F}_{\mathsf{ROT}}$ with message $(\mathsf{sen}, \mathsf{sid}, (\mathsf{transfer}, \kappa))$ for k times to obtain $\{(\mathbf{k}_i^0, \mathbf{k}_i^1)\}$ seeds.

OT Extension Phase I:

1. R forms three $(m+\kappa) \times k$ matrices $\mathbf{B}$, $\mathbf{E}$ and $\mathbf{C}$ in the following way and sends $\mathbf{D}$ to S:
   - Sets $\mathbf{B}^i = \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||\mathbf{k}_i^0)$.
   - Samples $\tau \leftarrow \{0,1\}^{\kappa}$ and sets $\mathbf{r}' = \mathbf{r}||\tau$.
   - Sets $\mathbf{E}_j = (r'_j|| \ldots ||r'_j)$. Clearly, $\mathbf{E}^i = \mathbf{r}'$.
   - Set $\mathbf{D}^i = \mathbf{B}^i \oplus \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||\mathbf{k}_i^1) \oplus \mathbf{E}^i$.

2. On receiving $\mathbf{D}$, S forms $(m+\kappa) \times k$ matrix $\mathbf{Q}$ with the jth column of $\mathbf{Q}$ set as $\mathbf{Q}^i = \big(s_i \odot \mathbf{D}^i\big) \oplus \mathcal{F}_{\mathsf{RO1}}(\mathbf{k}_i^{s_i})$. Clearly, (i) $\mathbf{Q}^i = \big(\mathbf{B}^i \oplus (s_i \odot \mathbf{E}^i)\big)$ and (ii) $\mathbf{Q}_j = \big(\mathbf{B}_j \oplus (\mathbf{s} \odot \mathbf{E}_j)\big) = \big(\mathbf{B}_j \oplus (\mathbf{s} \odot r_j)\big)$.

Check Phase:

1. For every pair $\alpha, \beta \subseteq [k^2]$ and $a, b \in \{0,1\}$, R defines the following four values:

$$h^{a,b}_{\alpha,\beta} = \mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||\mathbf{k}^a_\alpha) \oplus \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||\mathbf{k}^b_\beta))$$

$$h^{0,1}_{\alpha,\beta} = \mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||(\mathbf{k}^0_\alpha) \oplus \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||\mathbf{k}^1_\beta)),$$

$$h^{1,0}_{\alpha,\beta} = \mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||\mathbf{k}^1_\alpha) \oplus \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||\mathbf{k}^0_\beta)),$$

$$h^{1,1}_{\alpha,\beta} = \mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||\mathbf{k}^1_\alpha) \oplus \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||\mathbf{k}^1_\beta)).$$

R sends $h^{0,0}_{\alpha,\beta}$, $h^{0,1}_{\alpha,\beta}$, $h^{1,0}_{\alpha,\beta}$ and $h^{1,1}_{\alpha,\beta}$ to S.

Figure 4.7: Adaptive OT Extension Protocol secure against active adversaries (cont.)

Protocol $\pi^{\mathsf{a}}_{\mathsf{ALSZ}}$ for obtaining $\binom{2}{1}$-OT^{m}_{ℓ} from $\binom{2}{1}$-OT^{κ}_{m}

2. For every pair $\alpha, \beta \subseteq [k^2]$, S knows $\mathbf{s}_\alpha$, $\mathbf{s}_\beta$, $\mathbf{k}_\alpha^{\mathbf{s}_\alpha}$, $\mathbf{k}_\beta^{\mathbf{s}_\beta}$, $\mathbf{D}^\alpha$ and $\mathbf{D}^\beta$. S checks that:

   - $h_{\alpha,\beta}^{s_\alpha,s_\beta} = \mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||\mathbf{k}_\alpha^{s_\alpha}) \oplus \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||\mathbf{k}_\beta^{s_\beta}))$.
   - $h_{\alpha,\beta}^{\overline{s_\alpha},\overline{s_\beta}} = \mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||\mathbf{k}_\alpha^{s_\alpha}) \oplus \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||\mathbf{k}_\beta^{s_\beta}) \oplus \mathbf{D}^\alpha \oplus \mathbf{D}^\beta) = \mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||\mathbf{k}_\alpha^{\overline{s_\alpha}}) \oplus \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||\mathbf{k}_\beta^{\overline{s_\beta}}) \oplus \mathbf{r}^\alpha \oplus \mathbf{r}^\beta)$, where $\mathbf{r}^\alpha$, $\mathbf{r}^\beta$ denotes $\mathbf{r}$ used in $\mathbf{D}^\alpha$, $\mathbf{D}^\beta$ respectively.
   - $\mathbf{D}^\alpha \neq \mathbf{D}^\beta$.

   In case one of these checks fails, S aborts and outputs $\perp$.

OT Extension Phase II:

1. For every $j \in [m]$, S computes $\mathbf{y}_{j,0} = \mathbf{x}_{j,0} \oplus \mathcal{F}_{\mathsf{RO3}}(\mathsf{sid}||j, \mathbf{Q}_j)$ and $\mathbf{y}_{j,1} = \mathbf{x}_{j,1} \oplus \mathcal{F}_{\mathsf{RO3}}(\mathsf{sid}||j, \mathbf{Q}_j \oplus \mathsf{s})$. S sends $\{\mathbf{y}_{j,0}, \mathbf{y}_{j,1}\}_{j\in[m]}$ to R.

2. For every $j \in [m]$, R recovers $\mathbf{z}_j = \mathbf{y}_{j,r_j} \oplus \mathcal{F}_{\mathsf{RO3}}(\mathsf{sid}||j, \mathbf{B}_j)$. R outputs $\{\mathbf{z}_j\}_{j\in[m]}$.

Proof. We refer to the original paper of [ALSZ15] for the security proof against static adversaries. For the adaptive proof, we require equivocation of the honest party's view. For proving adaptive security, the views of S and R require equivocation. Equivocation of S's view follows from the equivocation of S's view in ALSZ13 protocol. Equivocation of R's view is similar to that of ALSZ13 but in addition it requires simulating the consistency checks correctly. We consider two cases separately - one for S's view and another for R's view.

- Equivocation of S's view: Sim simulates the seed-OT by invoking the $\mathcal{F}_{\mathsf{ROT}}$ functionality with a random s to obtain $\{\mathbf{k}_i^{\mathbf{s}_i}\}$, for $i \in [k]$. If both parties are honest then the simulation proceeds similar to the simulation for "Equivocation of S's view" (passive case) in Sec. 4.7.1. The consistency checks can be trivially simulated since S's role is limited to verification of the checks. If $\mathsf{A}_{\mathsf{Int}}$ corrupts R^* then Sim extracts the choice vector $\mathbf{r}'$. More specifically, Sim extracts R^*'s input, $(\mathbf{k}_i^0, \mathbf{k}_i^1)$, by invoking the $\mathcal{F}_{\mathsf{ROT}}$ simulator and computes $\mathbf{r}' = \mathbf{E}^i = \mathbf{r}||\tau$ from $\mathbf{D}^i$ using the seeds. Sim invokes the OT Extension functionality with $\mathbf{r}$ and obtains $\{\mathbf{x}_{j,r_j}\}_{j\in[m+\kappa]}$. Sim programs $\mathcal{F}_{\mathsf{RO3}}(\mathsf{sid}||j, \mathbf{Q}_j \oplus (\mathsf{s} \odot r_j))$ s.t. y_{j,r_j} opens to $\mathbf{x}_{j,r_j}$. In case of post execution corruption, Sim obtains $(\mathbf{x}_{j,0}, \mathbf{x}_{j,1})$ and he programs $\mathcal{F}_{\mathsf{RO3}}(\mathsf{sid}||j, \mathbf{Q}_j \oplus (\mathsf{s} \odot \overline{r_j}))$, s.t. $y_{j,\overline{r_j}}$ opens to $\mathbf{x}_{j,\overline{r_j}}$. To prevent equivocation $\mathsf{A}_{\mathsf{Int}}$ has to guess the value of s and query $\mathbf{Q}_j \oplus (\mathsf{s} \odot \overline{r_j}) = \mathbf{B}_j \oplus \mathsf{s}$ to $\mathcal{F}_{\mathsf{RO3}}$ which happens with negligible probability.

– Equivocation of R's view: The view of R consists of his inputs, the seeds $(\mathbf{k}_i^0, \mathbf{k}_i^1)$ and the matrices, $\mathbf{B}$, $\mathbf{E}$ and $\mathbf{D}$. When R is honest, Sim plays the role of R in the internal world where Sim obtains $(\mathbf{k}_i^0, \mathbf{k}_i^1)$ from $\mathcal{F}_{\mathsf{ROT}}$. If both parties are honest then simulation proceeds similar to the simulation for "Equivocation of R's view" (passive case) in Sec. 4.7.1. The consistency checks are simulated by setting $h_{\alpha,\beta}^{a,b}$ values randomly, for all $a, b \in \{0,1\}$ and all pairs $\alpha, \beta \subseteq [k^2]$. In case of post execution corruption, Sim obtains $\mathbf{r}$, and then based on the equivocated $\mathbf{B}$, $\mathbf{E}$ and $\mathbf{D}$ matrices, Sim can equivocate the $h_{\alpha,\beta}^{a,b}$ values correctly by programming $\mathcal{F}_{\mathsf{RO2}}$. $\mathsf{A}_{\mathsf{Int}}$ lacks $\mathbf{k}_i^0$ and $\mathbf{k}_i^1$, hence he cannot query the required argument to $\mathcal{F}_{\mathsf{RO2}}$ s.t. it would prevent equivocation of $h_{\alpha,\beta}^{a,b}$. Thus, the consistency checks can be correctly simulated. If $\mathsf{A}_{\mathsf{Int}}$ corrupts S^* then Sim extracts s vector from $\mathcal{F}_{\mathsf{ROT}}$ simulator. S computes $\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||\mathbf{k}_i^{\mathbf{s}_i})$ and simulates the $\mathbf{B}$, $\mathbf{E}$ and $\mathbf{D}$ matrices according to the simulation strategy in Section 4.7.1. For simulating the consistency checks the $h_{\alpha,\beta}^{a,b}$ values are randomly set, for all $a, b \in \{0,1\}$. For every pair $\alpha, \beta \subseteq [k^2]$, Sim programs $\mathcal{F}_{\mathsf{RO2}}$ as follows :

$$\mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||\mathbf{k}_\alpha^{\mathbf{s}_\alpha}) \oplus \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||\mathbf{k}_\beta^{\mathbf{s}_\beta})) = h_{\alpha,\beta}^{\mathbf{s}_\alpha,\mathbf{s}_\beta},$$
$$\mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||\mathbf{k}_\alpha^{s_\alpha}) \oplus \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||\mathbf{k}_\beta^{s_\beta}) \oplus \mathbf{D}^\alpha \oplus \mathbf{D}^\beta) = h_{\alpha,\beta}^{\overline{\mathbf{s}_\alpha},\overline{\mathbf{s}_\beta}}.$$

If $\mathcal{F}_{\mathsf{RO2}}$ is queried on other values by the adversary $\mathsf{A}_{\mathsf{Int}}$ then Sim answers it with a random value. When R^* gets corrupted, Sim programs $\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||\mathbf{k}_i^{\overline{\mathbf{s}_i}})$ according to Eq. 4.1. Furthermore, he programs $\mathcal{F}_{\mathsf{RO2}}$ as follows:

$$\begin{aligned}\mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||\mathbf{k}_\alpha^{\mathbf{s}_\alpha}) \oplus \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||\mathbf{k}_\beta^{\overline{\mathbf{s}_\beta}})) &= h_{\alpha,\beta}^{\mathbf{s}_\alpha,\overline{\mathbf{s}_\beta}}\\ \mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||\mathbf{k}_\alpha^{\overline{\mathbf{s}_\alpha}}) \oplus \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||\mathbf{k}_\beta^{\mathbf{s}_\beta})) &= h_{\alpha,\beta}^{\overline{\mathbf{s}_\alpha},\mathbf{s}_\beta}\end{aligned} \tag{4.3}$$

$\mathsf{A}_{\mathsf{Int}}$ corrupting S^* does not obtain $\mathbf{k}_\alpha^{\overline{s_\alpha}}$ and $\mathbf{k}_\beta^{\overline{s_\beta}}$ except with negligible probability as we are in the $\mathcal{F}_{\mathsf{ROT}}$ hybrid model. It allows Sim to correctly program $\mathcal{F}_{\mathsf{RO2}}$ involving $\mathbf{k}_\alpha^{\overline{s_\alpha}}$ and $\mathbf{k}_\beta^{\overline{s_\beta}}$, according to Eq. 4.3, permitting equivocation of R's view and completing the simulation for the consistency checks.

□

4.7.3 Efficiency and Implications

Our protocols $\pi_{\mathsf{ALSZ}}^{\mathsf{p}}$ and $\pi_{\mathsf{ALSZ}}^{\mathsf{a}}$ requires one extra round, compared to the seed OTs, and it matches with the round complexity of the state-of-the-art static OT extension protocols [ALSZ13, KOS15, PSS17]. They preserve the efficiency of the semi-honest and actively secure protocols of [ALSZ13] and [ALSZ15] respectively. In concrete terms, the amortized cost for generating $m = \mathsf{poly}(\kappa)$ copies of both semi-honest and active adaptively secure 1-out-of-2 OTs is 3κ bits communication and 3 symmetric key operations per OT. The other costs - seed OTs, PRG expansion and consistency checks, are independent of m. We present a few implications resulting from our OT Extension protocols:

- Our semi-honest OT Extension protocol requires static semi-honest OT protocol for the seed-OTs in a non-blackbox fashion, where we explicitly modified the simulation strategy for the static semi-honest OT. Thus, static semi-honest OT implies adaptive semi-honest OT Extension under the PROM assumption in a non-blackbox manner. Our theorem is summarized in Thm. 4.7.3. [LZ11] states that adaptive semi-honest oblivious transfer cannot be constructed from static semi-honest oblivious transfer in a black-box manner and we demonstrate that its possible in a non-blackbox way.

- If the random OT is instantiated using our receiver equivocal static OT protocol (Sec. 4.5.4) then our actively secure OT Extension protocol can efficiently generate large number of OTs at a cost of 3κ bits communication and 3 symmetric key operations per OT. However, the state-of-the-art adaptively secure 1-out-of-2 OT protocols [BDD⁺17] requires 11 exponentiations and communication of 15 κ bits. Thus, we drastically reduce the cost of generating adaptive OTs.

- We can efficiently generate $m \log \mathsf{N}$ copies of 1-out-of-2 OTs using our OT Extension protocol (following the previous approach) and then apply our adaptive transformation (Sec. 4.6) to obtain m copies of 1-out-of-N OTs. The amortized cost for each 1-out-of-N OT is $\mathsf{N} + 3 \log \mathsf{N} + 1$ symmetric key operations. Whereas the state-of-the-art adaptively secure 1-out-of-N OT protocols [BCG17, BDD⁺17] require atleast $\mathcal{O}(\mathsf{N})$ exponentiations.

Theorem 4.7.3. *If there exists an Oblivious Transfer protocol that is secure in the presence of static semi-honest adversaries, then there exists an Oblivious Transfer Extension protocol from κ to $\mathsf{poly}(\kappa)$ that is secure in the presence of adaptive semi-honest adversaries, in a non-blackbox way, under the programmable random oracle assumption.*

Our work closes the efficiency gap between static and adaptive OT and OT extension domain assuming PRO as a setup assumption.

Chapter 5

Adaptively Secure Commitment Scheme

Commitment Scheme is another fundamental primitive in the MPC literature that draws attention. Informally, we describe a commitment scheme as follows: The sender S commits to a message m in a commitment c and sends c to the receiver R in the commit phase. In the decommit phase, R learns the message m, along with some decommitment information, such that R is convinced that indeed m was committed in the commit phase. Besides its involvement in actively secure 2PC protocols [MR17, Lin15, LR15, Lin13, GWZ09], commitment schemes in the UC model also have implications in key exchange [DG03] and are non-malleable [CF01] in nature. In this work, we explore UC secure commitment protocols which are secure against adaptive adversaries.

5.1 Related Work

The study of UC secure commitment schemes was initiated by the seminal work of [CF01]. It was followed by the works of [CLOS02, DG03, HM04, Lin11, Fuj16, ABP17] and many more. We highlight some of the notable works in the relevant literature classified based on their round complexity, the security they achieve and the security model they are proven to be secure in.

The contributions of [DG03, Lin11, GIKW14, Fuj16, FLM11, CJS14, ABP17] based on hardness assumptions such as DDH, DLIN and Discrete Log (DLP) are interactive (involve either an interactive commit or decommit phase) in nature. In contrast, the contributions of [CLOS02, CF01, HM04, FLM11, NFT09, ABP17] present non-interactive commitment (NICOM) schemes, where both commit and decommit phases are non-interactive. The offline-online paradigm also forms an interesting flavour of NICOM schemes. These schemes [CDD$^+$15, DDGN14] consist of an input independent setup phase, which is run in the offline phase, while the NICOM is efficiently used in the online phase. The cost of the setup phase can be substantial but it gets amortized over multiple commitments. Our scheme also follows this particular model.

Similar to OT literature, the literature regarding commitment schemes is concentrated mainly around static security [Lin11, BCPV13, CJS14, GIKW14, DDGN14, CDD$^+$15, Fuj16, CDD$^+$16]. Building commitment schemes against adaptive adversaries has been a challenging task, since it involves equivocation of the internal states of the parties in case corruption occurs. There have been few contributions in the past addressing adaptive security. Adaptively-secure schemes can be broadly classified into two categories based on their ability to erase the internal state of the party when corruption occurs. [NFT09, FLM11, BCPV13, Fuj16] proposed adaptively-secure protocols which rely on secure erasure of the party's internal state whereas the constructions of [CF01, CLOS02, HM04, ABB$^+$13, HV15, ABP17, HPV17] achieved the same level of security without erasures. We skim through the protocols in the non-erasure model and compare all the above mentioned works with our protocol in Table 5.1.

[HV15] presented a theoretical construction of an interactive adaptive commitment scheme based on the minimal assumption of trapdoor simulatable public-key encryption. Since our focus is on adaptively-secure NICOM schemes, we elaborate the relevant NICOM results. [CF01, CLOS02, ABB$^+$13, ABP17] provided schemes for bit commitments, communicating at least $\mathcal{O}(\kappa^2)$ bits for committing to a κ bit string. [HM04] provided the first efficient NICOM for an arbitrary length message in the RO model and involves communication of constant number of bits for commitment. Programmability of RO is used to attain the property of equivocation in [HM04].

[HPV17] presents a theoretical construction of an adaptively secure commitment scheme relying on one way function and Global Random Oracle (GRO) [CJS14] in the tamper-proof hardware model. Also, the work of [CDG$^+$18] showed that the folklore commitment scheme $\mathcal{F}_{\mathsf{RO}}(m; r)$ is Generalized UC [CDPW07] secure assuming programmability from $\mathcal{F}_{\mathsf{RO}}$, where $\mathcal{F}_{\mathsf{RO}}$ is the RO functionality, m is the message and r is the randomness. And it can be trivially shown that it achieves adaptive security too. From the literature we can observe that, adaptive commitments either asks for programmability from the RO or it incurs a blowup in terms of efficiency. Hence, we can ask the following question:

"Can we obtain an efficient, non-interactive commitment scheme based on the non-programmable random oracle which is adaptively secure in the UC model?"

Our paper answers it in affirmative by presenting an adaptively-secure NICOM in the offline-online model, relying solely on the observable property of the RO. Table 5.1 consolidates the comparison of various UC secure commitment schemes alongside our protocol.

5.2 Our Results

We construct a NICOM, in the offline-online model, that is secure against an adaptive adversary without erasures. First, we generate a crs for Pedersen commitment [Ped91] using an ORO in the setup phase. The crs is of the form (g, h), where $h = g^x$ for $g, h \in \mathbb{G}$ and $x \in \mathbb{Z}_p$. Recall that

Table 5.1: Comparison among UC secure commitment schemes

Protocols	Message Size (bits)	Communication (κ-bit strings / Group elements)	Rounds (Commit/ Decommit)	Assumptions	Setup	Security
[Lin11]	κ	14	1/4	DDH + CRHF	crs	Static
[Fuj16]	κ	10	1/3	DDH + CRHF	crs	Static
[BCPV13]	κ	12	1/3	DDH+CRHF	crs	Static
[CDD$^+$16]	κ	1 + $o(1)$	5/1	OT	crs	Static
[CDD$^+$15]	κ	≥ 9 + $\mathcal{O}(\kappa^2)$ (one-time)	1/1 + 5 (one-time)	OT	crs	Static
[CJS14]	$\mathsf{poly}(\kappa)$	7	2/3	DLP	ORO	Static
[FLM11]	κ	21	1/1	DLIN + CRHF	crs	Adaptive with erasures
[NFT09]	κ	7	1/1	DDH + sEUF-OT	crs (Non-reusable)	Adaptive with erasures
[Fuj16]	κ	10	3/1	DDH + CRHF	crs	Adaptive with erasures
[BCPV13]	κ	14	3/1	DDH+CRHF	crs	Adaptive with erasures
[CF01]	κ	$\mathcal{O}(\kappa)$	1/1	DDH + UOWHF	crs	Adaptive
[CLOS02]	κ	$\mathcal{O}(\kappa)$	1/1	TDP	crs	Adaptive
[HM04]	$\mathsf{poly}(\kappa)$	5	1/1	DLP	PRO	Adaptive
[ABB$^+$13]	κ	$\mathcal{O}(\kappa)$	1/1	SXDH	crs	Adaptive
[ABP17]	κ	$\mathcal{O}(\kappa)$	1/1	DDH	crs	Adaptive
Our Scheme	$\mathsf{poly}(\kappa)$	4 + $\mathcal{O}(\mu\lvert C\rvert)$ (one-time)	1/1 + 4 (one-time)	DLP	ORO	Adaptive

Notations:
DDH - Decisional Diffie Hellman, CRHF - collision resistant hash function, DLP - Discrete Log Problem, DLIN - Decisional Linear, sEUF-OT - strongly unforgeable one-times signature, TDP - trapdoor permutations, UOWHF - universal one-way hash functions, OWF - one-way functions, ORO - observable random oracle, PRO - programmable random oracle,
ORO - observable random oracle, circuit C computes g^x
Note : The protocol of [CJS14] and [HM04] requires a non-interactive trapdoor commitment scheme. It has been instantiated with Pedersen commitment since to the best of our knowledge such a commitment scheme does not exist based on OWF.

$\mathbb{G}$ and $\mathbb{Z}_p$ denote a multiplicative group of prime order p and a prime field of order p respectively. The setup phase is a 4 round protocol where the parties perform a coin tossing protocol and a Zero Knowledge Proof of Knowledge (ZKPoK) protocol, relying on garbled circuits. Once generated, the crs alongwith the ORO can be reused to construct several instances of the NICOM. Under the hood, the NICOM relies on the Pedersen Commitment for equivocation and the ORO for extraction of a corrupted committer's input. Moreover, ORO permits committing to a message of length ℓ while incurring the overhead of committing to a κ bit string, where $\ell = \mathsf{poly}(\kappa)$. Compressing the message from ℓ to κ bits does not break binding since we are in the RO model, where it is hard to find two different messages of arbitrary length s.t. the RO returns the same result upon being queried on those messages. Our protocol involves communication of 4 group elements (or 4 κ bit string), computation of 8 exponentiations and 4 random oracle queries to commit to ℓ bits. This yields an efficient adaptively secure commitment scheme which is practically motivated for 2PC/MPC protocols based on offline-online paradigm [LR15, RR16]. This renders our commitment scheme useful in real-life situations where concurrently many protocols are run and share the same hash function. Table 5.1 compares our commitment schemes with the recent literature.

5.3 Non-Interactive UC-Secure Commitment Scheme

In this section we present our adaptively-secure NICOM scheme $\textsc{Com}$, implemented by protocol π_{COM}. The protocol π_{COM} (described in Fig. 5.2) is universally composable and securely realizes the functionality $\mathcal{F}_{\mathsf{COM}}$ (described in Fig. 2.6) in the $\mathcal{F}_{\mathsf{CRS}}$ model under ORO assumption and Discrete Log (over a group $\mathbb{G}$) assumption. The parties obtain the crs by invoking the $\mathcal{F}_{\mathsf{CRS}}$ (Fig. 5.1) functionality, which returns an instance of the Discrete Log problem. Later in this section, we demonstrate a protocol π_{CRS} which implements $\mathcal{F}_{\mathsf{CRS}}$ functionality in 4 rounds against adaptive adversaries. Once the crs is generated, it can be used for subsequent instances of $\textsc{Com}$. Our crs generation algorithm is independent of the parties' inputs and hence it is adaptively-secure, rendering the whole protocol adaptively-secure under the ORO assumption.

Figure 5.1: The ideal functionality $\mathcal{F}_{\mathsf{CRS}}$ for generating crs

$\mathcal{F}_{\mathbf{CRS}}$

On input $(\textsc{CRSGen}, \mathsf{sid})$ from party P_i, if $(\mathsf{sid}, ((s_1, s_2), s_3))$ is present in memory then send $(\mathsf{sid}, (s_1, s_2))$ to P_i. Else sample $x \leftarrow_R \mathbb{Z}_p$, compute $h = g^x$, store $(\mathsf{sid}, ((g, h), x))$ in the memory and return $(\mathsf{sid}, (g, h))$ to P_i.

5.3.1 Protocol Overview

We build upon the commitment scheme of Pedersen [Ped91], that relies on hardness of Discrete Log problem. The Pedersen commitment inherently supports equivocation as the message is statistically hidden in their case. However, for UC security the simulator, acting on behalf of R, has to extract the message, of a corrupted S^*, from the commitment. Our first approach was to apply an observable RO, i.e. $\mathcal{F}_{\mathsf{RO1}} : \{0,1\}^{\mathsf{poly}(\kappa)} \to \mathbb{Z}_p$, on the message being committed, i.e. $\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||m)$, and then commit the response of the ORO query in the Pedersen commitment. This would allow the simulator to observe the queries and obtain candidate message values. However, the simulator cannot uniquely identify the message committed. It is necessary to extract the randomness, say r_1, used in the commit phase so that the simulator can match the (message, randomness) with the commitment value. We achieve this by enforcing S to bind to r_1 using a second ORO, i.e. $\mathcal{F}_{\mathsf{RO2}} : \{0,1\}^{\kappa} \times \mathbb{Z}_p \to \{0,1\}^{\kappa}$. S commits to r_1 by means of the query $\mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||r_1)$. The hardness of the Discrete Log Problem ensures that a corrupted S^* is unable construct more than one such (message, randomness) pair that matches the Pedersen commitment.

However, the above technique demands binding to r_1 using RO which in turn prevents equivocation by simulator, acting on behalf of S. In order to restore the equivocal property, S is required to commit to $\mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||r_1)$ using another Pedersen commitment with fresh randomness r_2. This allows the simulator to equivocate the commitments by equivocating the first part of the commitment c_1 (Pedersen commitment on the message) separately, fixing r_1 to a new value. Now, the second part of the commitment c_2 can be equivocated by using r_1 and setting r_2 accordingly.

Figure 5.2: Non-Interactive UC-Secure Commitment Scheme COM

π_{COM}

- **Public Inputs:** The generator of group $\mathbb{G}$ is g.
- **Functionalities:** Random Oracles $\mathcal{F}_{\mathsf{RO1}} : \{0,1\}^{\mathsf{poly}(\kappa)} \rightarrow \mathbb{Z}_p$ and $\mathcal{F}_{\mathsf{RO2}} : \mathbb{Z}_p \rightarrow \{0,1\}^{\kappa}$ denote two random oracles.
- **Private Inputs:** S has input message m and R does not have any input.

Commit Phase:
On receiving input $(\textsc{Commit}, \mathsf{sid}, m)$ S performs the following:

- Invokes $\mathcal{F}_{\mathsf{CRS}}$ with input $(\text{CRSG}\textsc{en}, \mathsf{sid})$ to obtain $(g, h) = (\mathsf{g}, \mathsf{g}^x)$.
- Computes $a = \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||m)$.
- Samples $r_1, r_2 \leftarrow_R \mathbb{Z}_p$ and forms $\textsc{Com}(m; r_1, r_2) = (g^a h^{r_1} \bmod \mathrm{p}, g^{\mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||r_1)} h^{r_2} \bmod \mathrm{p}) = (c_1, c_2)$.
- Sends $\textsc{Com}(m; r_1, r_2)$ to R as commitment to m.

Decommitment Phase:
On receiving input $(\textsc{Decommit}, \mathsf{sid})$, S sends (m, r_1, r_2) to R. R invokes $\mathcal{F}_{\mathsf{CRS}}$ with input $(\text{CRSG}\textsc{en}, \mathsf{sid})$ to obtain $(g, h) = (\mathsf{g}, \mathsf{g}^x)$. R recomputes (c_1, c_2) using (m, r_1, r_2) and outputs accept if verification succeeds, else outputs reject.

5.3.2 Static Security

We show that our non-interactive commitment scheme COM is secure against static active adversaries and securely realizes the functionality $\mathcal{F}_{\mathsf{COM}}$ in the UC model by proving theorem 5.3.1.

Theorem 5.3.1. *If $\mathcal{F}_{\mathsf{RO1}}$ and $\mathcal{F}_{\mathsf{RO2}}$ are observable random oracles and solving the Discrete Log Problem is hard in multiplicative group $\mathbb{G}$, then π_{COM} UC-securely realizes the $\mathcal{F}_{\mathsf{COM}}$ functionality in the $\mathcal{F}_{\mathsf{CRS}}$ model against static active adversaries.*

Proof. Our proof is in the $\mathcal{F}_{\mathsf{CRS}}$ model where it is assumed that Sim knows the trapdoor x s.t. $h = g^x$ for the $\mathsf{crs} - (g; h)$. The proof proceeds in two cases - first, where A corrupts R^* and second, where the A corrupts S^*.

R^* is corrupted: A corrupts R^* in the real world, Sim corrupts R^* in the ideal world and $\mathsf{A}_{\mathsf{Int}}$ corrupts R^* in the internal execution. During the commit phase, Sim commits to a random message

m' using (r'_1, r'_2) as randomness, thereby computing $\textsc{Com}$, similar to an honest sender and sends $\textsc{Com}(m'; r'_1, r'_2)$ to $\mathsf{A}_{\mathsf{Int}}$. Sim further invokes $\mathcal{F}_{\mathsf{COM}}$ on behalf of R^* and obtains $(\textsc{Receipt}, \mathsf{sid}, \mathsf{S}, \mathsf{R})$. In the decommit phase, Sim invokes $\mathcal{F}_{\mathsf{COM}}$ to obtain the message $(\textsc{Decommit}, \mathsf{sid}, m)$. On obtaining the committed message m, Sim provides randomness (r_1, r_2) s.t. $\textsc{Com}$ decommits to m. The randomness (r_1, r_2) is computed by Sim as follows:

- Let $a = \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||m)$ and $a' = \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||m')$.
- The trapdoor x is known to Sim. Sim generates (r_1, r_2) s.t the values of (c_1, c_2) remain unchanged while the commitment is being equivocated. Sim performs it by solving equations 5.1 and 5.2 as follows:

$$\begin{aligned} & g^a h^{r_1} \text{ mod p} = g^{a'} h^{r'_1} \text{ mod p} \\ \implies & g^{a + r_1 x} \text{ mod p} = g^{a' + r'_1 x} \text{ mod p} \\ \implies & a + r_1 x = a' + r'_1 x \\ \implies & r_1 = (a' - a + r'_1 x) x^{-1} \end{aligned} \tag{5.1}$$

$$\begin{aligned} & g^{\mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||r_1)} h^{r_2} \text{ mod p} = g^{\mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||r'_1)} h^{r'_2} \text{ mod p} \\ \implies & g^{\mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||r_1) + r_2 x} \text{ mod p} = g^{\mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||r'_1) + r'_2 x} \text{ mod p} \\ \implies & \mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||r_1) + r_2 x = \mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||r'_1) + r'_2 x \\ \implies & r_2 = (\mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||r'_1) - \mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||r_1) + r'_2 x) x^{-1} \end{aligned} \tag{5.2}$$

  Sim provides (m, r_1, r_2), as opening to the commitment $\textsc{Com}$, to $\mathsf{A}_{\mathsf{Int}}$.

At the end of the protocol, $\mathsf{A}_{\mathsf{Int}}$ sends its view to Sim. Sim forwards the view to $\mathcal{Z}$ who halts with an output.

Indistinguishbaility : We show that the real world view of $\mathcal{Z}$ is indistinguishable from the ideal world view by showing that the following two hybrids are statistically indistinguishable.

- $\mathbf{HYB}_0$: Real world execution of the protocol.
- $\mathbf{HYB}_1$: Same as HYB_0, except that, Sim commits to a random message m' using (r'_1, r'_2) as randomness and opens to m in the decommit phase using different randomness (r_1, r_2). HYB_1 corresponds to the ideal world view of $\mathcal{Z}$. It follow from Eq. 5.1 and 5.2 that the committed message remains statistically hidden in $\textsc{Com}$. Hence, $\forall m, m', r'_1, r'_2$, Sim can always find a

consistent pair of randomness (r_1, r_2) s.t the commitment opens to m, provided Sim knows the trapdoor value x. This proves statistical indistinguishability of the two worlds.

S^* is corrupted: A corrupts S^* in the real world, Sim corrupts S^* in the ideal world and $\mathsf{A}_{\mathsf{Int}}$ corrupts S^* in the internal execution. Sim emulates the role of an honest R against $\mathsf{A}_{\mathsf{Int}}$ in the internal execution. Sim plays the role of S^* in $\mathcal{F}_{\mathsf{COM}}$. During the commit phase, Sim obtains the commitment $\textsc{Com}(m)$ from $\mathsf{A}_{\mathsf{Int}}$ in the internal execution. He observes the random oracle queries (both $\mathcal{F}_{\mathsf{RO1}}$ and $\mathcal{F}_{\mathsf{RO2}}$) made by $\mathsf{A}_{\mathsf{Int}}$ and tries to extract the committed message m. Sim aborts if it fails to extract the message. Let us assume that $\mathsf{A}_{\mathsf{Int}}$ makes s random oracle queries during the commit phase and Sim records them as $(q_1, q_2, \dots, q_s)$. We denote a (q_i, q_j) pair as valid, if q_i was queried to $\mathcal{F}_{\mathsf{RO1}}$, q_j was queried to $\mathcal{F}_{\mathsf{RO2}}$ and the following holds:

$$g^{\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||q_i)} h^{q_j} \bmod \mathrm{p} = c_1. \tag{5.3}$$

where $\textsc{Com}(m) = (c_1, c_2)$ is received from $\mathsf{A}_{\mathsf{Int}}$ by Sim. Sim runs over all possible pairs of (q_i, q_j) to find the valid pair(s). Based on the number of valid pair(s) discovered, Sim performs the following:

- If there does not exist any valid pair then Sim samples $m' \leftarrow_R \mathbb{G}$ and sends $(\textsc{Commit}, \mathsf{sid}, m')$ to $\mathcal{F}_{\mathsf{COM}}$.

- If there exists a unique valid (q_i, q_j) pair then Sim sends $(\textsc{Commit}, \mathsf{sid}, q_i)$ to $\mathcal{F}_{\mathsf{COM}}$.

- If there exists more than one valid pair then Sim samples $m' \leftarrow_R \mathbb{G}$ and sends $(\textsc{Commit}, \mathsf{sid}, m')$ to $\mathcal{F}_{\mathsf{COM}}$.

In the decommitment phase, $\mathsf{A}_{\mathsf{Int}}$ sends (m, r_1, r_2) to Sim. Sim verifies the commitment and aborts if verification fails in the internal execution. Whereas, Sim aborts in the ideal world as well as in $\mathcal{F}_{\mathsf{COM}}$ if the following holds:

- **Case 1:** If there was no valid pair.

- **Case 2:** If there was one valid (q_i, q_j) pair, and $q_i \neq m$.

- **Case 3:** If there exists more than one valid pair.

If none of the above conditions hold, then Sim has an unique valid (q_i, q_j) pair, s.t. $q_i = m$ and $q_j = r_1$. He sends $(\textsc{Decommit}, \mathsf{sid})$ to $\mathcal{F}_{\mathsf{COM}}$ to complete simulation of $\mathcal{F}_{\mathsf{COM}}$. At the end of the protocol, $\mathsf{A}_{\mathsf{Int}}$ sends its view to Sim. Sim forwards the view to $\mathcal{Z}$ who halts with an output.

Indistinguishability : We show that the real world view of $\mathcal{Z}$ is indistinguishable from the ideal world view by showing that the following two hybrids are computationally indistinguishable.

- HYB_0 : Real world execution of the protocol.
- HYB_1 : It represents the ideal world execution of the protocol $\mathcal{Z}$ can distinguish between the two hybrids (or worlds) if Sim aborts in ideal world and $\mathcal{F}_{\mathsf{COM}}$, while the honest R completes the protocol in real world. This occurs when the decommitment to COM provided by $\mathsf{A}_{\mathsf{Int}}$ verifies correctly but Sim fails to extract the underlying committed message in the internal execution. This event has been captured as an union of three exhaustive cases presented in the simulation. We will show that each case occurs with negligible probability:
  - **Case 1:** This case demonstrates that $\mathsf{A}_{\mathsf{Int}}$ obtained $\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||m)$ and $\mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||r_1)$ without querying m or r_1 to the random oracle during the commit phase. The random oracle assumption ensures that the query results would be random. Hence $\mathsf{A}_{\mathsf{Int}}$ can guess $\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||m)$ (or $\mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||r_1)$) without querying m (or r_1) with negligible probability. Another possible way to attain this case is when $\mathsf{A}_{\mathsf{Int}}$ commits to a junk message in c_2 and later he opens it to $\mathcal{F}_{\mathsf{RO2}}$ (after querying $\mathcal{F}_{\mathsf{RO2}}$) by finding the corresponding matching randomness r_2. However, this would contradict the binding property of the Pedersen Commitment c_2.
  - **Case 2:** This case demonstrates that either $\mathsf{A}_{\mathsf{Int}}$ obtained $\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||m)$ and $\mathcal{F}_{\mathsf{RO2}}(\mathsf{sid}||r_1)$ without querying m or r_1 to the random oracle during the commit phase, or the $\mathsf{A}_{\mathsf{Int}}$ possesses two valid pairs (q_i, q_j) and (m, r). The first event occurs with negligible probability as we are in the RO model. And the occurrence of the second event implies that the DLP problem can be solved by using $\mathsf{A}_{\mathsf{Int}}$ as a blackbox. The adversary $\mathsf{A}_{\mathsf{DLP}}$ will obtain a challenge instance $(g, h) = (g, g^x)$ from the challenger $\mathsf{Chall}_{\mathsf{DLP}}$ of the DLP game. $\mathsf{A}_{\mathsf{DLP}}$ will invoke the $\mathsf{A}_{\mathsf{Int}}$ while simulating the role of an honest R in the commitment scheme with $\mathsf{crs} = (g, h)$. $\mathsf{A}_{\mathsf{DLP}}$ will observe the queries made by $\mathsf{A}_{\mathsf{Int}}$ to obtain a valid pair (q_i, q_j). He will receive the commitment COM from $\mathsf{A}_{\mathsf{Int}}$. Upon obtaining the decommitment, (m, r), to COM, $\mathsf{A}_{\mathsf{DLP}}$ will find x as follows:

$$\begin{aligned}
& g^{\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||q_i)} h^{q_j} = g^{\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||m)} h^r \bmod \mathrm{p} \\
\Longrightarrow \quad & g^{\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||q_i)+q_j x} = g^{\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||m)+rx} \bmod \mathrm{p} \\
\Longrightarrow \quad & \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||q_i) + q_j x = \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||m) + rx \\
\Longrightarrow \quad & x = (\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||q_i) - \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||m))(r - q_j)^{-1} \qquad (5.4)
\end{aligned}$$

However, we assume that the DLP problem is hard in group $\mathbb{G}$ and hence this case occurs with negligible probability.

- **Case 3:** This case indicates that $\mathsf{A}_{\mathsf{Int}}$ obtains two or more valid pairs. This again implies that either the DLP problem has been solved by $\mathsf{A}_{\mathsf{Int}}$ or $\mathsf{A}_{\mathsf{Int}}$ found a collision in the random oracle queries. Let us denote two such valid pairs as (q_i, q_j) and (q'_i, q'_j). We will further split this case into two more subcases for analysis based on the equality of $\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||q_i)$ and $\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||q'_i)$ values.

  i. $q_i \neq q'_i, \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||q_i) = \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||q'_i), q_j = q'_j$: This indicates that $\mathsf{A}_{\mathsf{Int}}$ found a collision in the random oracle queries as $q_i \neq q'_i$. However, this event occurs with negligible probability as we are in the random oracle model.

  ii. $q_i \neq q'_i, \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||q_i) \neq \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||q'_i), q_j \neq q'_j$: In this case, $\mathsf{A}_{\mathsf{Int}}$ can be used as black-box by $\mathsf{A}_{\mathsf{DLP}}$ to solve the DLP problem. $\mathsf{A}_{\mathsf{DLP}}$ will invoke the $\mathsf{A}_{\mathsf{Int}}$ while simulating the role of an honest R in the commitment scheme with $\mathsf{crs} = (g, h)$. $\mathsf{A}_{\mathsf{DLP}}$ will observe the queries made by $\mathsf{A}_{\mathsf{Int}}$ to obtain two valid pairs (q_i, q_j) and (q'_i, q'_j) s.t. $q_i \neq q'_i$ and $q_j \neq q'_j$. $\mathsf{A}_{\mathsf{DLP}}$ solves the DLP problem by finding x as follows:

$$\begin{aligned} & g^{\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||q_i)} h^{q_j} = g^{\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||q'_i)} h^{q'_j} \text{ mod p} \\ \implies \quad & g^{\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||q_i) + q_j x} = g^{\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||q'_i) + q'_j x} \text{ mod p} \\ \implies \quad & \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||q_i) + q_j x = \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||q'_i) + q'_j x \\ \implies \quad & x = (\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||q_i) - \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||q'_i))(q'_j - q_j)^{-1} \end{aligned} \tag{5.5}$$

  However, we assume that the DLP problem is hard in group $\mathbb{G}$ and hence this case occurs with negligible probability.

  The other two cases involve $\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||q_i) = \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||q'_i), q_j \neq q'_j$ and $\mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||q_i) \neq \mathcal{F}_{\mathsf{RO1}}(\mathsf{sid}||q'_i), q_j = q'_j$ for $q_i \neq q'_i$. However, in these cases, it is not possible for both (q_i, q_j), (q'_i, q'_j) to be valid pairs (refer condition for valid pair in Eq 5.3) as $g^{\mathcal{F}_{\mathsf{RO1}}(q_i)} h^{q_j} \neq g^{\mathcal{F}_{\mathsf{RO1}}(q'_i)} h^{q'_j}$. There maybe atmost one valid pair, for which the analysis follows from Case 1 and 2.

□

5.3.3 Adaptive Security

In this section we show that our commitment scheme π_{COM} (Fig 5.2) also satisfies the stronger security notion of adaptivity under the observable random oracle assumption in the $\mathcal{F}_{\mathsf{CRS}}$ model. We briefly discuss the proof in this section.

Theorem 5.3.2. *If $\mathcal{F}_{\mathsf{RO1}}$ and $\mathcal{F}_{\mathsf{RO2}}$ are observable random oracles and solving the Discrete Log Problem is hard in multiplicative group $\mathbb{G}$, then protocol π_{COM} UC-securely realizes the $\mathcal{F}_{\mathsf{COM}}$ functionality in the $\mathcal{F}_{\mathsf{CRS}}$ model against adaptive active adversaries (without erasures).*

Proof. To prove adaptive security, we require Sim to equivocate the views of R and S appropriately on adaptive corruption in addition to static security. We divide our simulation into cases based on the party being corrupted:

R^* is corrupted: R does not have any private input or input randomness, and so the role of R in the protocol is restricted to verifying the commitments upon obtaining the message (m) and randomness $(r_1$ and $r_2)$ during the decommitment phase. When $\mathsf{A}_{\mathsf{Int}}$ corrupts R^* at any stage, i.e. commit phase, decommitment phase or post execution of the protocol, Sim returns a random tape as the internal randomness of R^*.

S^* is corrupted: Sim closely imitates the role of the simulator for static corruption when S^* gets corrupted adaptively. If $\mathsf{A}_{\mathsf{Int}}$ corrupts S^* in the beginning of the protocol or before the commitment is sent, Sim returns a random tape as the internal randomness of S^*. If $\mathsf{A}_{\mathsf{Int}}$ corrupts S^* after the commitment is sent, i.e. in decommitment phase or post execution, then Sim needs to equivocate. Sim initially commits to a dummy message m' and upon corruption of S^*, Sim obtains message m and successfully equivocates the commitment to open to m using randomness (r_1, r_2) (computed as described in the proof of Theorem 5.3.1). $\mathcal{Z}$ cannot distinguish between the commitment to m' and commitment to the actual value due to the statistical hiding property of the scheme. Equivocation follows from the equivocal property of π_{COM} as proven for the static case.

□

5.3.4 Implementing $\mathcal{F}_{\mathsf{CRS}}$ using Observable Random Oracle

For our protocol π_{COM}, the involved parties require a crs of the form (g, h), where $g = \mathsf{g}, h = g^x$. They obtained it by invoking the $\mathcal{F}_{\mathsf{CRS}}$ functionality. For our proof, the simulator should have the knowledge of the trapdoor x in order to perform correct simulation. The $\mathcal{F}_{\mathsf{CRS}}$ functionality can be trivially implemented if we assume a PRO. The parties can generate the crs as $\mathcal{F}_{\mathsf{RO}}(\mathsf{sid}||\text{“}\mathsf{com}\text{”})$. Sim samples x and programs the RO to return (g, h) s.t. $h = g^x$. This preserves adaptive security of π_{COM} when the crs generation algorithm is included as part of π_{COM}. However we are interested in implementing $\mathcal{F}_{\mathsf{CRS}}$ without relying on the programmability of the RO. This can be achieved by executing a 2PC protocol π_{CRS} (Fig. 5.3) relying solely on the observability property of the RO. Once the crs is generated it can be reused for subsequent commitments between the parties.

Intuition: Our π_{CRS} protocol proceeds in two phases - coin tossing and zero knowledge proof of knowledge (ZKPoK). The parties perform coin tossing to generate random shares h_{S} and h_{R}. These

Figure 5.3: Implementing $\mathcal{F}_{\mathsf{CRS}}$ using $\mathcal{F}_{\mathsf{RO}}$

π_{CRS}

- **Public Inputs:** The generator of group $\mathbb{G}$ is g.
- **Functionality:** Random oracle $\mathcal{F}_{\mathsf{RO}} : \{0,1\}^{\mathsf{poly}(\kappa)} \to \{0,1\}^{\kappa}$.
- **Private Inputs:** The parties do not have any input.

Coin Tossing:

- Round 1:
  - S samples $x_{\mathsf{S}} \leftarrow_R \mathbb{Z}_p$, computes $h_{\mathsf{S}} = g^{x_{\mathsf{S}}}$ mod p and sends $\mathcal{F}_{\mathsf{RO}}(\mathsf{sid}||h_{\mathsf{S}})$ to R.
  - R samples $x_{\mathsf{R}} \leftarrow_R \mathbb{Z}_p$, computes $h_{\mathsf{R}} = g^{x_{\mathsf{R}}}$ mod p and sends $\mathcal{F}_{\mathsf{RO}}(\mathsf{sid}||h_{\mathsf{R}})$ to S.
- Round 2:
  - S sends h_{S} to R.
  - R sends h_{R} to S.
- Computation:
  - S verifies $\mathcal{F}_{\mathsf{RO}}(\mathsf{sid}||h_{\mathsf{R}})$ and computes $h = h_{\mathsf{S}}.h_{\mathsf{R}}$ mod p, else aborts.
  - R verifies $\mathcal{F}_{\mathsf{RO}}(\mathsf{sid}||h_{\mathsf{S}})$ and computes $h = h_{\mathsf{S}}.h_{\mathsf{R}}$ mod p, else aborts.

Zero Knowledge Proof of Knowledge:

S and R perform the following steps in parallel with their roles interchanged.

- Round 1:
  - R samples a challenge string $c \leftarrow_R \{0,1\}^{\mu}$ and sends $\mathcal{F}_{\mathsf{RO}}(\mathsf{sid}||c)$ to S.
- Round 2:
  - S computes μ garbled circuits, by sampling $\mathsf{seed}_i \leftarrow_R \{0,1\}^{\kappa}$ and $(\mathbf{C}_i, e_i, d_i) \leftarrow \mathsf{Gb}(\mathrm{PRG}(\mathsf{seed}_i), C)$ where C computes g^x, for $i \in [\mu]$.
  - S sends $\mathcal{F}_{\mathsf{RO}}(\mathsf{sid}||\mathsf{seed}_i)$, $\mathcal{F}_{\mathsf{RO}}(\mathsf{sid}||\mathbf{C}_i)$ and d_i to R.
- Round 3:
  - R reveals c to S. S verifies $\mathcal{F}_{\mathsf{RO}}(\mathsf{sid}||c)$ and aborts if verification fails.
- Round 4: (Let c_i denote i^{th} bit of c)
  - If $c_i = 0$, then $\mathbf{C}_i$ is a check circuit and S sends seed_i to R.
  - If $c_i = 1$, then $\mathbf{C}_i$ is an evaluation circuit and S sends $\mathbf{X} = \mathsf{En}(x_{\mathsf{S}}, e_i)$ to R.
- Computation:
  - R verifies the check circuit $\mathbf{C}_i$ as $\mathsf{Ve}(C, \mathbf{C}_i, e_i)$, if $c_i = 0$, else he aborts.
  - R computes $y = \mathsf{De}(\mathsf{Ev}(\mathbf{C}_i, \mathbf{X}))$, if $c_i = 1$, and aborts if $y \neq h_{\mathsf{S}}$.
  - R stores (g, h) as the crs.

shares are then used to obtain $h = h_{\mathsf{S}}.h_{\mathsf{R}}$. Once the coin tossing is performed, they engage in ZKPoK in order to prove the knowledge of trapdoors to their respective shares, i.e. h_{S} and h_{R} respectively. The ZKPoK enables the simulator of π_{CRS} to extract the corrupted party's share in order to obtain the trapdoor to (g, h). Our coin tossing protocol requires 2 rounds and ZKPoK consumes 4 rounds. However, the first 2 rounds of the ZKPoK can be parallelized with the coin tossing protocol, thus yielding a 4 round protocol for crs generation. The coin tossing is performed using the random oracle and the

ZKPoK is performed by plugging a simplified ZK version of [HV16] which uses garbled circuits and random oracle only. Note that we need a ZKPoK protocol which can be implemented using an ORO and without relying on any other setup assumption. This rules out the possibilities of using efficient ZKPoK protocol of [JKO13] since it uses an OT, which would further require other setup assumptions. However, the interactive ZKPoK variants of "MPC-in-the-head" protocols [GMO16, AHIV17] also suffices for our purpose.

Static Security: We prove that π_{CRS} implements $\mathcal{F}_{\mathsf{CRS}}$ in the presence of static active adversaries. Since our protocol is symmetric, we consider the case where S^* is corrupted and simulator Sim plays the role of honest R. Sim behaves like an honest R throughout the protocol and extracts the trapdoor x_{S} from the ZKPoK sent by S^*. Sim plays the role of an evaluator in the ZKPoK whereas S^* plays the role of a constructor. Sim extracts the seeds, for all circuits sent by S^*, by observing the queries made to $\mathcal{F}_{\mathsf{RO}}$ and matching them with $\mathcal{F}_{\mathsf{RO}}(\mathsf{sid}||\mathsf{seed}_i)$. Sim generates the encoding information of all evaluation circuits from the seeds. Finally, Sim extracts the input witness x_{S} of S^*, by matching the input wire labels, of the evaluation circuit, with the encoding information, of the evaluation circuit. On the other hand, Sim can simulate the ZKPoK, on behalf of constructor, by behaving like an honest R since he knows the trapdoor x_{R} sampled by him. A corrupt S^* cannot extract any information about x_{R} due to the privacy of the garbling scheme.

Adaptive Security: We prove that protocol π_{CRS} is adaptively-secure by observing that the parties do not have any private inputs at the outset of the protocol. The simulator can simulate by running honest sender/receiver algorithms as required. Upon corruption of a party, the simulator can reveal its random tape for the corresponding party. The simulated honest party's view will be indistinguishable from the real honest party's view since in both cases the protocol transcripts are honestly generated using a random tape. This proves that our protocol π_{CRS} UC-securely implements $\mathcal{F}_{\mathsf{CRS}}$ and generates the crs required for our commitment scheme.

5.3.5 Final Commitment Scheme $\pi = \pi_{\mathsf{CRS}} + \pi_{\mathsf{COM}}$

We can combine π_{CRS} and π_{COM} to obtain an UC-secure protocol π which implements $\mathcal{F}_{\mathsf{COM}}$ functionality in the presence of adaptive adversaries solely relying on ORO. The security has been summarized in Theorem. 5.3.3. does not require a local crs.

Theorem 5.3.3. *If* $\mathsf{Garble} = (\mathsf{Gb}, \mathsf{En}, \mathsf{Ev}, \mathsf{De}, \mathsf{Ve})$ *is a private, verifiable garbling scheme and solving the Discrete Log Problem is hard in multiplicative group* $\mathbb{G}$*, if* π_{COM} *implements* $\mathcal{F}_{\mathsf{COM}}$ *functionality against active adaptive adversaries in the* crs*, ORO-model and* π_{CRS} *generates the* crs *against active adaptive adversaries in the ORO model, then* $\pi = \pi_{\mathsf{CRS}} + \pi_{\mathsf{COM}}$ *implements* $\mathcal{F}_{\mathsf{COM}}$ *functionality against active adaptive adversaries (without erasures) in the ORO-model.*

5.3.6 Efficiency of π

We analyze the efficiency of π by analyzing the efficiency of π_{CRS} and π_{COM} separately. The π_{CRS} protocol requires 4 rounds and has a computation cost of 2 exponentiations, $8\mu+8$ oracle queries, construction and evaluation of 2μ circuits. The communication cost is 2μ circuits + $(8+4\mu+2\kappa)$ strings of κ bits. However, it is a one-time cost which would get amortized when multiple commitments are performed using the same crs. Next, we analyze the efficiency of π_{COM}. The length of our commitment is two group elements, independent of the message length. Decommitment incurs communication of two group elements. The computation is also minimal, incurring one random oracle query on $|m|$ bits, one oracle query on a κ bits string and four exponentiations on sender's side for committing. Decommitment incurs similar computation overhead on the receiver's end. It is non-interactive in both commitment and decommitment phase. This yields an efficient adaptively secure commitment scheme which is practically motivated for offline-online 2PC/MPC protocols [HKK+14, LR14, LR15, RR16]. The π_{CRS} protocol can be run in the offline phase while the commitment scheme can be conveniently used in the online phase.

Chapter 6

Conclusion

In this thesis we have presented UC-secure protocols for cryptographic primitives in the random oracle model, which are secure against adaptive active adversaries. The first part of the thesis deals with adaptive zero knowledge protocols. We briefly elaborate on the various ZK paradigms employed in the literature. Then we focus on adaptively-secure ZK, where we talk about the techniques of MPC-in-the-head and [HV16]. Finally, we focus on the 5 round static protocol of [JKO13]. We show that it can be made adaptively-secure without any further blowup in communication or computation. The underlying OT scheme needs to be secure against an adaptive receiver. Next, we try to reduce its round complexity to 3 by relying on a plain RO. For this, we employ techniques of conditional disclosure of secret and apply those techniques using the RO to obtain the required protocol.

The second part of the thesis deals with adaptive OT. We initiate it by discussing the current adaptive OT literature. We show attacks in concurrent adaptive OT protocols and point out the dearth of round-optimal adaptive OT. Through our work, we try to solve the issue of round optimality. We address it by reinforcing the dual mode encryption scheme of [PVW08] to obtain a samplable dual mode encryption scheme and then we use this new scheme to construct a round-optimal adaptive OT framework in the PRO model. The framework can be instantiated under DDH and LWE assumptions. We compare our protocol with the state-of-the-art 3 round adaptive protocol of [BDD$^+$17] and demonstrate that our protocol outperforms it in terms of computation and communication. Infact, the runtime and communication of our protocol is almost same as the state-of-the-art static protocol of [PVW08], instantiated under the DDH assumption. Next, we show that $\log \mathsf{N}$ 1-out-of-2 static receiver equivocal OTs can be used to obtain one 1-out-of-N adaptive OT in the PRO model. The cost for such a conversion is $\mathcal{O}(\log \mathsf{N})$ public key operations, as opposed to $\mathcal{O}(\mathsf{N})$ public key operations in [BDD$^+$17, BCG17]. We also show that the OT extension protocols of [ALSZ13, ALSZ15] can be proven to be adaptively secure in the PRO model. Furthermore, the underlying seed OTs can be secure only against an adaptive receiver and a static sender. This gives us adaptive OTs at an amortized cost of few symmetric key operations, similar to static OT extension protocols.

The third part of the thesis involves adaptive commitment schemes. We briefly discuss the adaptive commitment literature and try to obtain a non-interactive adaptive commitment in the ORO model, whose efficiency is comparable to a non-interactive adaptive commitment [HM04] in the PRO model. Basically, we aim to reduce the assumption of programmability to observability, that is required from the random oracle for an efficient adaptive NICOM. To achieve this, we construct an adaptive NICOM using the Pedersen commitment scheme in the ORO + $\mathcal{F}_{\mathsf{CRS}}$ model. Next, we implement the $\mathcal{F}_{\mathsf{CRS}}$ functionality in an input-independent setup phase using a 4-round adaptively secure protocol using the ORO. This reduces the setup assumption of our commitment scheme to only ORO. Our commitment scheme is in the offline-online paradigm where the crs can be generated once in the offline phase and it can be reused multiple times in the online phase to generate the adaptive NICOM.

There a few open questions that result from our work. One of them is to obtain a provably secure OT protocol based on random oracle whose efficiency is similar to [NP05, CO15]. This is of practical interest as these two protocols [NP05, CO15] are more efficient than [PVW08] but they are not provably secure. Another interesting area to explore is adaptive OT extension based on non-programmable random oracle assumption. Currently, there are no other known OT extension protocols even in the weaker models of adaptive security, like one-sided adaptive security [HP14] and semi-adaptive security [GWZ09]. Whereas, in the static world the state-of-the-art OT extension protocols rely on non-observable non-programmable random oracle and there is a known OT extension protocol [Bea96b] based on non-blackbox access of one way functions. Obtaining an OT extension in the static semi-honest case based on blackbox one way function is also a longstanding open problem. In the area of adaptive commitment schemes, we saw that all the schemes either require random oracle or are bit commitments. The problem of obtaining an adaptive commitment for strings based on assumptions weaker than random oracle seems intriguing.

Bibliography

[ABB$^+$13] Michel Abdalla, Fabrice Benhamouda, Olivier Blazy, Céline Chevalier, and David Pointcheval. Sphf-friendly non-interactive commitments. In *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part I*, pages 214–234, 2013. 37, 38, 39, 40, 41, 71, 72

[ABP17] Michel Abdalla, Fabrice Benhamouda, and David Pointcheval. Removing erasures with explainable hash proof systems. In *Public-Key Cryptography - PKC 2017 - 20th IACR International Conference on Practice and Theory in Public-Key Cryptography, Amsterdam, The Netherlands, March 28-31, 2017, Proceedings, Part I*, pages 151–174, 2017. 38, 39, 70, 71, 72

[AHIV17] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight sublinear arguments without a trusted setup. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 2087–2104, 2017. 16, 81

[AIKW15] Benny Applebaum, Yuval Ishai, Eyal Kushilevitz, and Brent Waters. Encoding functions with constant online rate, or how to compress garbled circuit keys. *SIAM J. Comput.*, 44(2):433–466, 2015. 18

[AIR01] William Aiello, Yuval Ishai, and Omer Reingold. Priced oblivious transfer: How to sell digital goods. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 119–135. Springer, Heidelberg, May 2001. 27, 37

[ALSZ13] Gilad Asharov, Yehuda Lindell, Thomas Schneider, and Michael Zohner. More efficient oblivious transfer and extensions for faster secure computation. In *2013 ACM*

SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013, pages 535–548, 2013. 2, 37, 39, 41, 60, 61, 62, 68, 69, 83

[ALSZ15] Gilad Asharov, Yehuda Lindell, Thomas Schneider, and Michael Zohner. More efficient oblivious transfer extensions with security for malicious adversaries. In *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, pages 673–701, 2015. 2, 37, 39, 41, 64, 65, 67, 69, 83

[BC15] Olivier Blazy and Céline Chevalier. Generic construction of uc-secure oblivious transfer. In *Applied Cryptography and Network Security - 13th International Conference, ACNS 2015, New York, NY, USA, June 2-5, 2015, Revised Selected Papers*, pages 65–86, 2015. 37, 38, 40, 41

[BC16] Olivier Blazy and Céline Chevalier. Structure-preserving smooth projective hashing. In *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part II*, pages 339–369, 2016. 36, 37, 38, 40, 41

[BCG17] Olivier Blazy, Céline Chevalier, and Paul Germouty. Almost optimal oblivious transfer from QA-NIZK. In *Applied Cryptography and Network Security - 15th International Conference, ACNS 2017, Kanazawa, Japan, July 10-12, 2017, Proceedings*, pages 579–598, 2017. 36, 37, 38, 39, 40, 41, 69, 83

[BCPV13] Olivier Blazy, Céline Chevalier, David Pointcheval, and Damien Vergnaud. Analysis and improvement of lindell's uc-secure commitment schemes. In *Applied Cryptography and Network Security - 11th International Conference, ACNS 2013, Banff, AB, Canada, June 25-28, 2013. Proceedings*, pages 534–551, 2013. 71, 72

[BCPW15] Fabrice Benhamouda, Geoffroy Couteau, David Pointcheval, and Hoeteck Wee. Implicit zero-knowledge arguments and applications to the malicious setting. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, pages 107–129, 2015. 27

[BCR86] Gilles Brassard, Claude Crépeau, and Jean-Marc Robert. All-or-nothing disclosure of secrets. In *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, pages 234–238, 1986. 36

[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. *IACR Cryptology ePrint Archive*, abs/1710.08256, 2017. 37, 38, 39, 40, 41, 42, 69, 83

[Bea96a] Donald Beaver. Adaptive zero knowledge and computational equivocation (extended abstract). In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 629–638, 1996. 6, 17

[Bea96b] Donald Beaver. Correlated pseudorandomness and the complexity of private computations. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 479–488, 1996. 36, 39, 84

[BHR12a] Mihir Bellare, Viet Tung Hoang, and Phillip Rogaway. Adaptively secure garbling with applications to one-time programs and secure outsourcing. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 134–153. Springer, Heidelberg, December 2012. 17, 18, 27

[BHR12b] Mihir Bellare, Viet Tung Hoang, and Phillip Rogaway. Foundations of garbled circuits. In *the ACM Conference on Computer and Communications Security, CCS'12, Raleigh, NC, USA, October 16-18, 2012*, pages 784–796, 2012. 8, 11, 24

[BP12] Nir Bitansky and Omer Paneth. Point obfuscation and 3-round zero-knowledge. In *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings*, pages 190–208, 2012. 15, 19, 28

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 136–145, 2001. 3, 5, 11, 49

[CDD$^+$15] Ignacio Cascudo, Ivan Damgård, Bernardo Machado David, Irene Giacomelli, Jesper Buus Nielsen, and Roberto Trifiletti. Additively homomorphic UC commitments with optimal amortized overhead. In *Public-Key Cryptography - PKC 2015 - 18th IACR International Conference on Practice and Theory in Public-Key Cryptography, Gaithersburg, MD, USA, March 30 - April 1, 2015, Proceedings*, pages 495–515, 2015. 70, 71, 72

[CDD$^+$16] Ignacio Cascudo, Ivan Damgård, Bernardo David, Nico Döttling, and Jesper Buus Nielsen. Rate-1, linear time and additively homomorphic UC commitments. In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part III*, pages 179–207, 2016. 71, 72

[CDG$^+$17] Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. Post-quantum zero-knowledge and signatures from symmetric-key primitives. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 1825–1842, 2017. 16, 20

[CDG$^+$18] Jan Camenisch, Manu Drijvers, Tommaso Gagliardoni, Anja Lehmann, and Gregory Neven. The wonderful world of global random oracles. Cryptology ePrint Archive, Report 2018/165, 2018. `https://eprint.iacr.org/2018/165`. 11, 71

[CDMW09a] Seung Geol Choi, Dana Dachman-Soled, Tal Malkin, and Hoeteck Wee. Improved non-committing encryption with applications to adaptively secure protocols. In *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*, pages 287–302, 2009. 37, 38, 39

[CDMW09b] Seung Geol Choi, Dana Dachman-Soled, Tal Malkin, and Hoeteck Wee. Simple, black-box constructions of adaptively secure protocols. In *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings*, pages 387–402, 2009. 37, 38, 39

[CDPW07] Ran Canetti, Yevgeniy Dodis, Rafael Pass, and Shabsi Walfish. Universally composable security with global setup. In *Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings*, pages 61–85, 2007. 71

[CF01] Ran Canetti and Marc Fischlin. Universally composable commitments. In *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, pages 19–40, 2001. 70, 71, 72

[CFH$^+$15] Craig Costello, Cédric Fournet, Jon Howell, Markulf Kohlweiss, Benjamin Kreuter, Michael Naehrig, Bryan Parno, and Samee Zahur. Geppetto: Versatile verifiable computation. In *2015 IEEE Symposium on Security and Privacy*, pages 253–270. IEEE Computer Society Press, May 2015. 15

[CGM16] Melissa Chase, Chaya Ganesh, and Payman Mohassel. Efficient zero-knowledge proof of algebraic and non-algebraic statements with applications to privacy preserving credentials. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 499–530. Springer, Heidelberg, August 2016. 15, 17, 18

[CJS14] Ran Canetti, Abhishek Jain, and Alessandra Scafuro. Practical UC security with a global random oracle. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014*, pages 597–608, 2014. 40, 70, 71, 72

[CKWZ13] Seung Geol Choi, Jonathan Katz, Hoeteck Wee, and Hong-Sheng Zhou. Efficient, adaptively secure, and composable oblivious transfer with a single, global CRS. In *Public-Key Cryptography - PKC 2013 - 16th International Conference on Practice and Theory in Public-Key Cryptography, Nara, Japan, February 26 - March 1, 2013. Proceedings*, pages 73–88, 2013. 36, 37, 39

[CLOS02] Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. Universally composable two-party and multi-party secure computation. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 494–503, 2002. 17, 37, 70, 71, 72

[CM99] Jan Camenisch and Markus Michels. Proving in zero-knowledge that a number is the product of two safe primes. In Jacques Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 107–122. Springer, Heidelberg, May 1999. 15

[CO15] Tung Chou and Claudio Orlandi. The simplest protocol for oblivious transfer. In *Progress in Cryptology - LATINCRYPT 2015 - 4th International Conference on Cryptology and Information Security in Latin America, Guadalajara, Mexico, August 23-26, 2015, Proceedings*, pages 40–58, 2015. 36, 38, 40, 41, 42, 84

[CPV17] Ran Canetti, Oxana Poburinnaya, and Muthuramakrishnan Venkitasubramaniam. Equivocating yao: constant-round adaptively secure multiparty computation in the plain model. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory* *of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 497–509, 2017. 25

[DCW13] Changyu Dong, Liqun Chen, and Zikai Wen. When private set intersection meets big data: an efficient and scalable protocol. In *Proceedings of the 2013 ACM SIGSAC conference on Computer &#38; communications security*, CCS '13, pages 789–800, 2013. 36

[DDGN14] Ivan Damgård, Bernardo Machado David, Irene Giacomelli, and Jesper Buus Nielsen. Compact VSS and efficient homomorphic UC commitments. In *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II*, pages 213–232, 2014. 70, 71

[DFGK14] George Danezis, Cédric Fournet, Jens Groth, and Markulf Kohlweiss. Square span programs with applications to succinct NIZK arguments. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 532–550. Springer, Heidelberg, December 2014. 15

[DG03] Ivan Damgård and Jens Groth. Non-interactive and reusable non-malleable commitment schemes. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing, June 9-11, 2003, San Diego, CA, USA*, pages 426–437, 2003. 70

[DI06] Ivan Damgård and Yuval Ishai. Scalable secure multiparty computation. In *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*, pages 501–520, 2006. 16

[FJN$^+$13] Tore Kasper Frederiksen, Thomas Pelle Jakobsen, Jesper Buus Nielsen, Peter Sebastian Nordholt, and Claudio Orlandi. Minilego: Efficient secure two-party computation from general assumptions. In *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, pages 537–556, 2013. 36

[FLM11] Marc Fischlin, Benoît Libert, and Mark Manulis. Non-interactive and re-usable universally composable string commitments with adaptive security. In *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, pages 468–485, 2011. 39, 70, 71, 72

[FNO15] Tore Kasper Frederiksen, Jesper Buus Nielsen, and Claudio Orlandi. Privacy-free garbled circuits with applications to efficient zero-knowledge. In *Advances in Cryptology-EUROCRYPT 2015*, pages 191–219. Springer, 2015. 16

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, pages 186–194, 1986. 16

[Fuj16] Eiichiro Fujisaki. Improving practical uc-secure commitments based on the DDH assumption. In *Security and Cryptography for Networks - 10th International Conference, SCN 2016, Amalfi, Italy, August 31 - September 2, 2016, Proceedings*, pages 257–272, 2016. 70, 71, 72

[GGPR13] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct nizks without pcps. In *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, pages 626–645, 2013. 15

[GH08] Matthew Green and Susan Hohenberger. Universally composable adaptive oblivious transfer. In *Advances in Cryptology - ASIACRYPT 2008, 14th International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, Australia, December 7-11, 2008. Proceedings*, pages 179–197, 2008. 36

[GIKM98] Yael Gertner, Yuval Ishai, Eyal Kushilevitz, and Tal Malkin. Protecting data privacy in private information retrieval schemes. In *30th ACM STOC*, pages 151–160. ACM Press, May 1998. 27

[GIKW14] Juan A. Garay, Yuval Ishai, Ranjit Kumaresan, and Hoeteck Wee. On the complexity of UC commitments. In *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, pages 677–694, 2014. 70, 71

[GIR17] Ziya Alper Genç, Vincenzo Iovino, and Alfredo Rial. ”the simplest protocol for oblivious transfer” revisited. *IACR Cryptology ePrint Archive*, 2017:370, 2017. 38

[GK96] Oded Goldreich and Hugo Krawczyk. On the composition of zero-knowledge proof systems. *SIAM Journal on Computing*, 25(1):169–192, 1996. 19, 28

[GKPS18] Chaya Ganesh, Yashvanth Kondi, Arpita Patra, and Pratik Sarkar. Efficient adaptively secure zero-knowledge from garbled circuits. In *Public-Key Cryptography - PKC 2018 - 21st IACR International Conference on Practice and Theory of Public-Key Cryptography, Rio de Janeiro, Brazil, March 25-29, 2018, Proceedings, Part II*, pages 499–529, 2018. 36

[GMO16] Irene Giacomelli, Jesper Madsen, and Claudio Orlandi. Zkboo: Faster zero-knowledge for boolean circuits. In *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016.*, pages 1069–1083, 2016. 15, 16, 20, 81

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA*, pages 291–304, 1985. 15

[GMW86] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to prove all np-statements in zero-knowledge, and a methodology of cryptographic protocol design. In *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, pages 171–185, 1986. 15

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*, pages 218–229, 1987. 36

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(3):691–729, 1991. 17

[GMY04] Juan A. Garay, Philip MacKenzie, and Ke Yang. *Efficient and Universally Composable Committed Oblivious Transfer and Applications*, pages 297–316. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. 37

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 197–206, 2008. 45, 46

[GQ88] Louis C. Guillou and Jean-Jacques Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both trasmission and memory. In C. G. Günther, editor, *EUROCRYPT'88*, volume 330 of *LNCS*, pages 123–128. Springer, Heidelberg, May 1988. 15

[Gro10] Jens Groth. Short non-interactive zero-knowledge proofs. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 341–358. Springer, Heidelberg, December 2010. 15

[Gro16] Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 305–326. Springer, Heidelberg, May 2016. 15

[GS08] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, pages 415–432, 2008. 15

[GS17] Sanjam Garg and Akshayaram Srinivasan. Two-round multiparty secure computation from minimal assumptions. Cryptology ePrint Archive, Report 2017/1156, 2017. `https://eprint.iacr.org/2017/1156`. 21

[GV87] Oded Goldreich and Ronen Vainish. How to solve any protocol problem - an efficiency improvement. In *Advances in Cryptology - CRYPTO '87, A Conference on the Theory and Applications of Cryptographic Techniques, Santa Barbara, California, USA, August 16-20, 1987, Proceedings*, pages 73–86, 1987. 36

[GWZ09] Juan A. Garay, Daniel Wichs, and Hong-Sheng Zhou. Somewhat non-committing encryption and efficient adaptively secure oblivious transfer. In *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, pages 505–523, 2009. 36, 37, 38, 39, 70, 84

[HK07] Omer Horvitz and Jonathan Katz. Universally-composable two-party computation in two rounds. In *Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007, Proceedings*, pages 111–129, 2007. 37

[HK12] Shai Halevi and Yael Tauman Kalai. Smooth projective hashing and two-message oblivious transfer. *J. Cryptology*, 25(1):158–193, 2012. 37

[HKK$^+$14] Yan Huang, Jonathan Katz, Vladimir Kolesnikov, Ranjit Kumaresan, and Alex J. Malozemoff. Amortizing garbled circuits. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II*, pages 458–475, 2014. 2, 82

[HL17] Eduard Hauck and Julian Loss. Efficient and universally composable protocols for oblivious transfer from the cdh assumption. *IACR Cryptology ePrint Archive*, 2017:1011, 2017. 38, 40, 41, 42

[HM04] Dennis Hofheinz and Jörn Müller-Quade. Universally composable commitments using random oracles. In *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings*, pages 58–76, 2004. 70, 71, 72, 84

[HMR15] Zhangxiang Hu, Payman Mohassel, and Mike Rosulek. Efficient zero-knowledge proofs of non-algebraic statements with sublinear amortized cost. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, pages 150–169, 2015. 15

[HP14] Carmit Hazay and Arpita Patra. One-sided adaptively secure two-party computation. In *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014, San Diego, CA, USA, February 24-26, 2014. Proceedings*, pages 368–393, 2014. 84

[HPV17] Carmit Hazay, Antigoni Polychroniadou, and Muthuramakrishnan Venkitasubramaniam. Constant round adaptively secure protocols in the tamper-proof hardware model. In *Proceedings, Part II, of the 20th IACR International Conference on Public-Key Cryptography — PKC 2017 - Volume 10175*, pages 428–460, 2017. 17, 71

[HV15] Carmit Hazay and Muthuramakrishnan Venkitasubramaniam. On black-box complexity of universally composable security in the CRS model. In *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, pages 183–209, 2015. 71

[HV16] Carmit Hazay and Muthuramakrishnan Venkitasubramaniam. On the power of secure two-party computation. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 397–429. Springer, Heidelberg, August 2016. 15, 17, 18, 20, 81, 83

[IKNP03] Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank. Extending oblivious transfers efficiently. In *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, pages 145–161, 2003. 2, 37, 39, 61

[IKOS07] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge from secure multiparty computation. In David S. Johnson and Uriel Feige, editors, *39th ACM STOC*, pages 21–30. ACM Press, June 2007. 15, 16

[IKOS09] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge proofs from secure multiparty computation. *SIAM J. Comput.*, 39(3):1121–1152, 2009. 16, 18

[IPS08] Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. Founding cryptography on oblivious transfer - efficiently. In *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*, pages 572–591, 2008. 36

[IR89] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 44–61, 1989. 6, 36, 41

[IW14] Yuval Ishai and Hoeteck Wee. Partial garbling schemes and their applications. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *ICALP 2014, Part I*, volume 8572 of *LNCS*, pages 650–662. Springer, Heidelberg, July 2014. 27

[JKO13] Marek Jawurek, Florian Kerschbaum, and Claudio Orlandi. Zero-knowledge using garbled circuits: how to prove non-algebraic statements efficiently. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 955–966. ACM, 2013. v, vi, 5, 6, 9, 15, 16, 18, 19, 20, 23, 24, 26, 28, 36, 81, 83

[JS07] Stanislaw Jarecki and Vitaly Shmatikov. Efficient two-party secure computation on committed inputs. In *Advances in Cryptology - EUROCRYPT 2007, 26th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Barcelona, Spain, May 20-24, 2007, Proceedings*, pages 97–114, 2007. 37

[Kil88] Joe Kilian. Founding cryptography on oblivious transfer. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 20–31, 1988. 36

[Kil92] Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *24th ACM STOC*, pages 723–732. ACM Press, May 1992. 15

[KK13] Vladimir Kolesnikov and Ranjit Kumaresan. Improved OT extension for transferring short secrets. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, pages 54–70, 2013. 2, 39

[KKL$^+$16] Vladimir Kolesnikov, Hugo Krawczyk, Yehuda Lindell, Alex J. Malozemoff, and Tal Rabin. Attribute-based key exchange with general policies. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 1451–1463, 2016. 17, 18

[KMO89] Joe Kilian, Silvio Micali, and Rafail Ostrovsky. Minimum resource zero-knowledge proofs (extended abstract). In *30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA, 30 October - 1 November 1989*, pages 474–479, 1989. 16

[KOS15] Marcel Keller, Emmanuela Orsini, and Peter Scholl. Actively secure OT extension with optimal overhead. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, pages 724–741, 2015. 2, 37, 39, 68

[KOS16] Marcel Keller, Emmanuela Orsini, and Peter Scholl. MASCOT: faster malicious arithmetic secure computation with oblivious transfer. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 830–842, 2016. 36

[KP17] Yashvanth Kondi and Arpita Patra. Privacy-free garbled circuits for formulas: Size zero and information-theoretic. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part I*, pages 188–222, 2017. 17

[KS08] Vladimir Kolesnikov and Thomas Schneider. Improved garbled circuit: Free XOR gates and applications. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *ICALP 2008, Part II*, volume 5126 of *LNCS*, pages 486–498. Springer, Heidelberg, July 2008. 16

[Lin08] Yehuda Lindell. Efficient fully-simulatable oblivious transfer. *Chicago J. Theor. Comput. Sci.*, 2008, 2008. 37

[Lin11] Yehuda Lindell. Highly-efficient universally-composable commitments based on the DDH assumption. In *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, pages 446–466, 2011. 70, 71, 72

[Lin13] Yehuda Lindell. Fast cut-and-choose based protocols for malicious and covert adversaries. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, pages 1–17, 2013. 2, 36, 38, 70

[Lin15] Yehuda Lindell. An efficient transform from sigma protocols to NIZK with a CRS and non-programmable random oracle. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I*, pages 93–109, 2015. 70

[Lip13] Helger Lipmaa. Succinct non-interactive zero knowledge arguments from span programs and linear error-correcting codes. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 41–60. Springer, Heidelberg, December 2013. 15

[LM16] Baiyu Li and Daniele Micciancio. Equational security proofs of oblivious transfer protocols. Cryptology ePrint Archive, Report 2016/624, 2016. 42

[LOS14] Enrique Larraia, Emmanuela Orsini, and Nigel P. Smart. Dishonest majority multiparty computation for binary circuits. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II*, pages 495–512, 2014. 36

[LP07] Yehuda Lindell and Benny Pinkas. An efficient protocol for secure two-party computation in the presence of malicious adversaries. In *Advances in Cryptology - EUROCRYPT 2007, 26th Annual International Conference on the Theory and Applications* *of Cryptographic Techniques, Barcelona, Spain, May 20-24, 2007, Proceedings*, pages 52–78, 2007. 2, 38

[LP11] Yehuda Lindell and Benny Pinkas. Secure two-party computation via cut-and-choose oblivious transfer. In *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings*, pages 329–346, 2011. 38

[LR14] Yehuda Lindell and Ben Riva. Cut-and-choose yao-based secure computation in the online/offline and batch settings. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II*, pages 476–494, 2014. 82

[LR15] Yehuda Lindell and Ben Riva. Blazing fast 2pc in the offline/online setting with security for malicious adversaries. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-6, 2015*, pages 579–590, 2015. 36, 70, 72, 82

[LZ11] Yehuda Lindell and Hila Zarosim. Adaptive zero-knowledge proofs and adaptively secure oblivious transfer. *J. Cryptology*, 24(4):761–799, 2011. 17, 69

[LZ13] Yehuda Lindell and Hila Zarosim. On the feasibility of extending oblivious transfer. In *TCC*, pages 519–538, 2013. 41

[MR17] Payman Mohassel and Mike Rosulek. Non-interactive secure 2pc in the offline/online and batch settings. In *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part III*, pages 425–455, 2017. 2, 36, 70

[NFT09] Ryo Nishimaki, Eiichiro Fujisaki, and Keisuke Tanaka. Efficient non-interactive universally composable string-commitment schemes. In *Provable Security, Third International Conference, ProvSec 2009, Guangzhou, China, November 11-13, 2009. Proceedings*, pages 3–18, 2009. 70, 71, 72

[NO09] Jesper Buus Nielsen and Claudio Orlandi. LEGO for two-party secure computation. In *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings*, pages 368–386, 2009. 36

[NOVY98] Moni Naor, Rafail Ostrovsky, Ramarathnam Venkatesan, and Moti Yung. Perfect zero-knowledge arguments for NP using any one-way permutation. *Journal of Cryptology*, 11(2):87–108, 1998. 18

[NP01] Moni Naor and Benny Pinkas. Efficient oblivious transfer protocols. In *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms, January 7-9, 2001, Washington, DC, USA.*, pages 448–457, 2001. 37

[NP05] Moni Naor and Benny Pinkas. Computationally secure oblivious transfer. *J. Cryptology*, 18(1):1–35, 2005. 36, 40, 58, 84

[OOS17] Michele Orrù, Emmanuela Orsini, and Peter Scholl. Actively secure 1-out-of-n OT extension with application to private set intersection. In *Topics in Cryptology - CT-RSA 2017 - The Cryptographers' Track at the RSA Conference 2017, San Francisco, CA, USA, February 14-17, 2017, Proceedings*, pages 381–396, 2017. 39

[Ped91] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*, pages 129–140, 1991. 71, 73

[PHGR13] Bryan Parno, Jon Howell, Craig Gentry, and Mariana Raykova. Pinocchio: Nearly practical verifiable computation. In *2013 IEEE Symposium on Security and Privacy*, pages 238–252. IEEE Computer Society Press, May 2013. 15

[PSS17] Arpita Patra, Pratik Sarkar, and Ajith Suresh. Fast actively secure OT extension for short secrets. In *24th Annual Network and Distributed System Security Symposium, NDSS*, 2017. 2, 39, 69

[PSSZ15] Benny Pinkas, Thomas Schneider, Gil Segev, and Michael Zohner. Phasing: Private set intersection using permutation-based hashing. In *Proceedings of the 24th USENIX Conference on Security Symposium*, SEC'15, 2015. 36

[PSZ14] Benny Pinkas, Thomas Schneider, and Michael Zohner. Faster private set intersection based on ot extension. In *Proceedings of the 23rd USENIX Conference on Security Symposium*, SEC'14, 2014. 36

[PVW08] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In *Advances in Cryptology - CRYPTO 2008, 28th*

Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings, pages 554–571, 2008. 2, 19, 21, 22, 27, 29, 36, 37, 38, 40, 43, 44, 46, 47, 48, 49, 56, 57, 83, 84

[Rab81] Michael O. Rabin. How to exchange secrets with oblivious transfer, 1981. Harvard University Technical Report 81 talr@watson.ibm.com 12955 received 21 Jun 2005. 2, 36

[RR16] Peter Rindal and Mike Rosulek. Faster malicious 2-party secure computation with online/offline dual execution. In *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016.*, pages 297–314, 2016. 72, 82

[RR17] Peter Rindal and Mike Rosulek. Malicious-secure private set intersection via dual execution. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 1229–1242, 2017. 36

[Sch90] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 239–252. Springer, Heidelberg, August 1990. 15

[WMK17] Xiao Wang, Alex J. Malozemoff, and Jonathan Katz. Faster secure two-party computation in the single-execution setting. In *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part III*, pages 399–424, 2017. 36

[Yao82] Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*, pages 160–164, 1982. 2

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pages 162–167, 1986. 36

[ZRE15] Samee Zahur, Mike Rosulek, and David Evans. Two halves make a whole - reducing data transfer in garbled circuits using half gates. In *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications* *of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II*, pages 220–250, 2015. 11, 16, 19