Cryptography with One-Way Communication

Sanjam Garg[*]   Yuval Ishai[†]   Eyal Kushilevitz[‡]   Rafail Ostrovsky[§]   Amit Sahai[¶]

Abstract

There is a large body of work on using noisy communication channels for realizing different cryptographic tasks. In particular, it is known that secure message transmission can be achieved unconditionally using only *one-way* communication from the sender to the receiver. In contrast, known solutions for more general secure computation tasks inherently require interaction, even when the entire input originates from the sender.

We initiate a general study of cryptographic protocols over noisy channels in a setting where only one party speaks. In this setting, we show that the landscape of what a channel is useful for is much richer. Concretely, we obtain the following results.

- **Relationships between channels.** The binary erasure channel (BEC) and the binary symmetric channel (BSC), which are known to be securely reducible to each other in the interactive setting, turn out to be qualitatively different in the setting of one-way communication. In particular, a BEC cannot be implemented from a BSC, and while the erasure probability of a BEC can be manipulated in both directions, the crossover probability of a BSC can only be manipulated in one direction.

- **Zero-knowledge proofs and secure computation of deterministic functions.** One-way communication over BEC or BSC is sufficient for securely realizing any deterministic (possibly reactive) functionality which takes its inputs from a sender and delivers its outputs to a receiver. This provides the first truly non-interactive solutions to the problem of zero-knowledge proofs.

- **Secure computation of randomized functions.** One-way communication over BEC or BSC *cannot* be used for realizing general randomized functionalities which take input from a sender and deliver output to a receiver. On the other hand, one-way communication over other natural channels, such as bursty erasure channels, can be used to realize such functionalities. This type of protocols can be used for distributing certified cryptographic keys without revealing the keys to the certification authority.

---

[*]UC Berkeley
[†]Technion
[‡]Technion
[§]UCLA
[¶]UCLA

1 Introduction

The seminal work of Wyner [Wyn75] demonstrated the usefulness of noise for secure communication. Since then, there has been a large body of work on basing various cryptographic primitives, such as key agreement and commitment [BBCM95, BBR88, Mau91, DKS99, WNI03, Wul09, RTWW11], on different types of noisy communication channels.

In 1988, Crépeau and Kilian [CK88] showed that noise in a communication channel can be used to realize essentially everything a cryptographer could wish for. In particular, they showed that any non-trivial *binary-symmetric channel* (BSC) can be used to realize *oblivious transfer* (OT) which is sufficient for realizing two-party secure computation. (More efficient construction were later considered in [KM01, SW02, IKO$^+$11b].) Finally, Crépeau, Morozov and Wolf [CMW04] generalized these results to arbitrary *discrete memory-less* channels. Other results towards characterizing the types of channels on which OT can be based appeared in [Kil88, DKS99, DFMS04, Wul07, Wul09].

Following the work of Crépeau and Kilian [CK88], the entire body of research on secure two-party computation over noisy channels requires parties to interact. In contrast, the present paper considers cryptographic protocols which only use *one-way communication*, namely ones in which only one party speaks. There has been a considerable amount of work on realizing information-theoretic secure message transmission in this setting. These works are motivated not only by the goal of achieving information-theoretic security, but also by the goal of efficiency; see [BTV12] for discussion. Our goal is to extend this study to more general cryptographic tasks, including useful special cases of secure two-party computation in which the input originates from only one party.

1.1 Our Model

We model a channel as an ideal functionality $\mathcal{C}$. This is done in order to capture the security properties of the channel in a clean way and in order to facilitate the use of composition theorems. A channel provides a communication medium between a *sender* and a *receiver*. The sender can invoke the channel $\mathcal{C}$ on an input of its choice. The channel "based on its nature" processes the input and outputs the processed value to the receiver. The correctness and secrecy requirements of a channel and the protocols we build on top of it can be specified in terms of UC security. For example, consider a binary erasure channel (BEC) parameterized by a probability $p \in (0, 1)$. For this channel, the sender inputs a bit $x \in \{0, 1\}$ and the channel outputs (for the receiver) x with a probability p and $\perp$ with a probability $1 - p$. [1] Even for this basic channel, stating the correctness and security properties is non-trivial. Correctness requires that if the sender sends x then the receiver outputs either x or $\perp$ with the right probability distribution. Security is a bit more involved; it requires that no malicious sender can figure out whether the receiver actually received the sent bit or not, and that a malicious receiver does not learn any partial information about the sent bit in the case of an erasure.

In this work, we consider various such channels. Two other channels that would be of great interest to us are the *binary symmetric channel* (BSC) and the *random oblivious transfer* (ROT) channel. A BSC is parameterized by a probability $p \in (\frac{1}{2}, 1)$. For this channel, the sent bit is transmitted correctly with probability p and is flipped with probability $1 - p$. An ROT channel takes as input two strings m_0 and m_1 from the sender and outputs either $(m_0, \perp)$ or $(\perp, m_1)$ to the receiver, with equal probability.

When considering protocols built on top of such channels, we distinguish between the weaker *semi-honest* model, where the sender follows the protocol but tries to learn information about the receiver's output from its random coins, and the *malicious model*, where the sender may send arbitrary information over the channel. When the sender follows the protocol, the receiver's output should be as specified by the functionality. When the sender deviates from the protocol, the security requirement uses the standard real-ideal paradigm, asserting that the sender's strategy can be simulated by a distribution over honest strategies.

---

[1] In the literature, p sometimes stands for the error probability, while in our paper it is the probability of the "no noise" event.
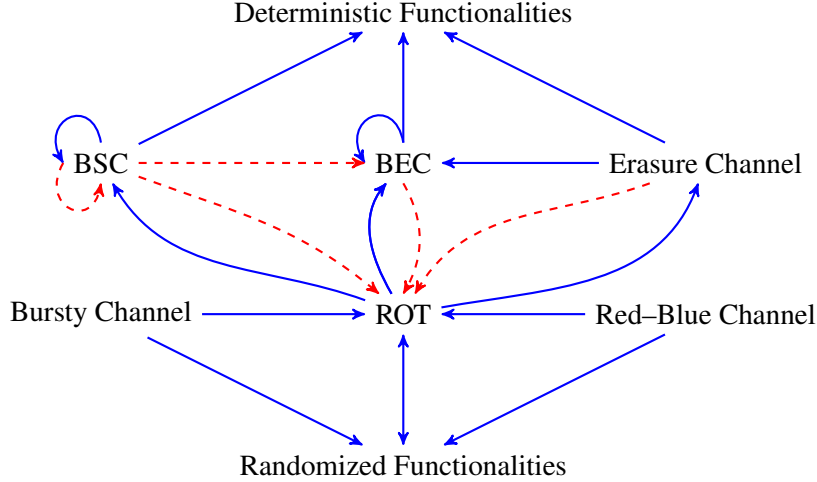
Figure 1: Relationships among different kinds of channels and their applications. Solid arrows are used to denote a positive reduction, i.e. $A \to B$ implies that B can be constructed given A. On the other hand, dashed arrows indicate negative results, i.e. $A \dashrightarrow B$ implies that B cannot be constructed given A. Solid self-edge of BEC indicates that the transmission probability of a BEC can be manipulated in both directions. On the other hand, the solid and dashed self-edges of BSC respectively indicate that the probability of correct transmission of a BSC can be diminished (and brought closer to $\frac{1}{2}$) but cannot be amplified.

It is important to note, however, that in this case the standard definition of "security with abort" also allows the sender to make the protocol fail, as long as the receiver can detect this failure. By default, the term "secure" refers to the malicious model, though most of our negative results apply also to the semi-honest model.

1.2 Our Results

We initiate a general study of one-way secure computation (OWSC) protocols over noisy channels in a setting where only one party speaks. Surprisingly, the one-way setting is strikingly different from the interactive setting. In the interactive setting, all finite channels are either trivial, equivalent to secure message transmission, or equivalent to oblivious transfer. On the other hand, in the setting of OWSC, the landscape of what a channel is useful for is much richer. Specifically, we obtain the following results. All the implications have been summarized in Figure 1.

- **Relationships between channels.** Binary erasure channel (BEC) and binary symmetric channel (BSC), which are known to be securely reducible to each other in the interactive setting, turn out to be qualitatively very different in the setting of one-way communication. In particular, we show that a BEC cannot be implemented given a BSC. Also, somewhat surprisingly, we show that while the erasure probability of a BEC can be manipulated in both directions the probability of correct transmission of a BSC can only be manipulated in one direction.

- **Deterministic functions.** We show that both BEC and and BSC are sufficient for securely realizing any deterministic (possibly reactive) functionality that takes input from a sender and delivers its output to a receiver with only one-way communication. This provides the *first* truly non-interactive solution to the problem of zero-knowledge. We extend our results to the Generalized Erasure Channel (GEC) which is a generalization of BEC (see Section 4 for formal definition).

2

- **Randomized functions.** We show that neither BEC nor BSC can be used (even assuming computational assumptions) for the task of realizing randomized functionalities which take input from a sender and deliver output to a receiver, in the setting of one-way communication. Nonetheless, one-way communications over natural channels, such as bursty erasure channels, can be used to realize such functionalities. This result is obtained by first constructing a random oblivious-transfer channel (ROT) and building on the techniques from [IPS08, IKO$^+$11a]. This provides the first non-trivial feasibility result for secure-computation in a setting where only one party speaks.

1.3 Applications

OWSC both for deterministic and randomized functionalities enable a number of applications for which there are no known solutions.

Truly non-interactive zero-knowledge. Non-interactive zero-knowledge proof systems (NIZKs) [BFM90, FLS99] are a fundamental tool in cryptography with widespread applications. However, all known constructions rely on a common random string (or a random oracle)[2] and inherently fail to achieve useful features such as non-transferability or deniability [Pas03]. OWSC for deterministic functions provides the *first* truly non-interactive solution to the problem of zero-knowledge. This solution does not rely on a shared string between parties or a random oracle and achieves non-transferability and deniability properties. Furthermore, this solution achieves information theoretic and composable security.

Oblivious certification of cryptographic keys. Public-key cryptography relies on the existence of certification authorities (like Verisign) who sign the public keys of different parties. All known implementations of this certification procedure rely on interaction. Our OWSC for randomized functionalities provides for the *first* candidate to realize this procedure with just one-way communication. More specifically, our protocol allows the certification authority to send a public-key secret-key pair along with a certificate on the public key with just one-way communication. We stress that in this setting the certification authority itself does not learn the secret key of the recipient party, as the randomness used in its generation is derived from the channel. However, if the certificate authority deviates from the protocol, the recipient may detect failure rather than output a pair of keys.

Fair puzzle distribution. Consider a Sudoku Puzzle competition where the organizer of the competition would like to generate signed puzzles for all the participants. However the participants do not trust the organizer and would like their challenge Sudoku puzzles to be of the same difficulty. More specifically, we would like to have a mechanism that allows the competition organizer to provide independent puzzles of a pre-specified difficulty level (along with a signature on this puzzle) to each of the participants. The participants should be assured not only that the puzzles were generated independently from the correct distribution, but also that the organizers do not have an edge in solving the puzzles they generated (e.g., by generating random solved puzzles). There are no known solutions for this problem in a setting with just one-way communication. Our OWSC protocol for randomized functions gives the first such solution.

2 Preliminaries

Let λ denote a security parameter. We say that a function is *negligible* in λ if it is asymptotically smaller than the inverse of any fixed polynomial in λ. Otherwise, the function is said to be *non-negligible* in λ. We

---

[2]The result of Barak and Pass [BP04] is an exception to this. However they only achieve a weaker notion where security is only guaranteed against uniform provers. We, on the other hand, are interested in the standard notion of zero-knowledge.

say that an event happens with *overwhelming* probability if it happens with probability $p(\lambda) = 1 - \nu(\lambda)$, where $\nu(\lambda)$ is a negligible function in λ. We use $[n]$ to denote the set $\{1, \ldots, n\}$.

Monotone Sets. Let $X_1, X_2 \ldots X_n$ be independent Bernoulli variables with $\Pr[X_i = 1] = p_i$. We define $Q_n = \{0,1\}^n$ (the *n-cube*) and identify each element $a \in Q_n$ with the corresponding subset of $[n]$; i.e., $\{i \mid a_i = 1\}$. We define a probability measure $\Pr$ on Q_n by:

$$\Pr(a) = \prod_{i \in a} p_i \prod_{i \notin a} (1 - p_i) \,.$$

A set $A \subseteq Q_n$ is said to be a *monotone* if $a \in A$ and $a \subseteq b$ implies that $b \in A$.

Lemma 1 (Harris [Har60], Kleitman [Kle66]) *If A and B are two monotone subsets of Q_n then A and B are* positively correlated*; namely,*

$$\Pr[A \cap B] \geq \Pr[A] \Pr[B].$$

Chernoff bounds. Let $X_1, X_2 \ldots X_n$ be independent Bernoulli variables with $\Pr[X_i = 1] = p_i$. Let $X = \sum_{i=1}^{n} X_i$ and μ be the expectation of X. Then,

$$\Pr(X \geq (1 + \delta)\mu) \leq e^{-\frac{\delta^2 \mu}{3}}, \text{ for } 0 < \delta < 1.$$

$$\Pr(X \leq (1 - \delta)\mu) \leq e^{-\frac{\delta^2 \mu}{2}}, \text{ for } 0 < \delta < 1.$$

3 Oblivious ZK-PCP

An NP-relation $R \subseteq \{0,1\}^* \times \{0,1\}^*$ is given by a deterministic algorithm $W(\cdot, \cdot)$ that runs in time polynomial in the length of its first input. The relation is

$$R = \{(x, w) : W(x, w) = 1\}.$$

The associated NP-language $L_R = \{x : \exists w$ such that $W(x, w) = 1\}$. The witness set for an $x \in \{0,1\}^*$ is $R(x) = \{w : W(x, w) = 1\}$.

Consider the setting in which a prover wants to prove to a verifier the knowledge of a witness in $R(x)$, in zero-knowledge. The prover's algorithm $P_{\text{oZK}}(\lambda, x, w)$ takes as input the security parameter λ, a statement x and a witness $w \in R(x)$ and generates an n-bit long ($n = poly(\lambda, |x|)$) PCP proof π. The verifier V_{oZK} takes as input x and a partial proof π' and outputs 1 or 0, where π' is obtained from π by replacing some of the bits of π with $\perp$. We call the ZK-PCP *oblivious* because in our setting the verifier does not get to decide the subset of the bits of π that it receives (intuitively, those are determined by the "channel").

Definition 1 *[Oblivious ZK-PCP] We say that $(P_{\text{oZK}}, V_{\text{oZK}})$ is a (c, ν)-oblivious ZK-PCP with knowledge soundness κ for the relation R if:*

Perfect Completeness: $\forall x, w$, *such that* $(x, w) \in R$ *and* $\pi \leftarrow P_{\text{oZK}}(\lambda, x, w)$, *we have that* $V_{\text{oZK}}(x, \pi') = 1$ *over all choices of π' obtained by replacing π at arbitrary locations with $\perp$.*

c-Soundness (Proof of Knowledge): There exists a PPT extractor E such that, for all x and purported proofs π^*, if $E(x, \pi^*) \notin R(x)$ then

$$Pr[V_{\text{oZK}}(x, g(\pi^*)) = 0] \geq \kappa,$$

where the probability is taken over the random choices of g, and g is a random function that replaces $n - c$ random locations in π^* with $\perp$ (and leaves the other c locations untouched).

ν-**Zero-Knowledge:** *There exists a PPT simulator $\mathcal{S}$ such that, for all $x \in L_R$, the following distributions are statistically indistinguishable:*

- *Sample $\pi \leftarrow P_{oZK}(\lambda, x, w)$. Replace each bit of π with $\perp$ with probability $1 - \nu$ and output the resultant value.*
- $\mathcal{S}(\lambda, x)$.

The construction of oblivious ZK-PCP was implicit in [ISW03, Ajt10]. The following formal claim is implied by the construction of Ajtai [Ajt10].

Proposition 1 *For any constant $\nu \in (0, 1)$, there exists a $(3, \nu)$-oblivious ZK-PCP with a knowledge error $\kappa = 1 - \frac{1}{\xi(\lambda)}$, where $\xi(\lambda)$ is some polynomial in λ.*

We will sketch how the above proposition is directly implied by the construction of Ajtai. Ajtai [Ajt10] shows that given a circuit C, we can obtain a functionally-equivalent circuit C' of size $O(|C|m^4)$ (for appropriate parameter m) such that, even if the value of each wire of the circuit C' (obtained by evaluating C' on a certain input x) is independently revealed to the adversary with probability ν, then the input of the circuit (namely x) remains hidden (except with probability $e^{-\alpha m}|C|$, for some appropriate constant α).

Applying Ajtai's transformation to the verification circuit $V(x, \cdot)$, we obtain a functionally-equivalent circuit V' of size $O(|V|m^4)$ such that, even if each wire of the circuit (obtained by computing V' on a valid witness) is independently revealed to the adversary with a probability ν, the witness remains hidden (except with probability $e^{-\alpha m}|V|$). In our case, concatenation of the bit values that each wire of the circuit V' gets assigned gives the PCP proof string. The verification procedure V_{oZK} looks at bits at three random locations in the PCP proof. It always outputs 1 except if the three wires correspond to the input/output wires of a specific gate and are found to not satisfy the gate relationship. Furthermore, if the output wire of the gate is also the output of the entire circuit then it also checks that this value is 1. Completeness follows immediately. Soundness follows from the fact that exitance of an inconsistency will be caught with some noticeable probability. In fact, if there are no inconsistencies in the proof, then the PCP can be used to extract the witness used to generate the PCP. The zero-knowledge property, on the other hand, follows directly from the security of Ajtai's construction.

4   Different kinds of channels

In this work, we model a channel as an ideal functionality $\mathcal{C}$. This is done in order to capture the security properties of a channel in a clean way. A channel provides a (one-way) communication medium between a *sender* and a *receiver*. The sender can invoke the channel $\mathcal{C}$ on an input of its choice. The channel "based on its nature", processes the input and outputs the processed value to the receiver. The correctness and secrecy requirements of a channel can be specified by a two-party functionality, which takes an input from the sender, generates some internal randomness, and delivers an output to the receiver. Our formulation of channel functionalities, as well as the security definition of protocols that build on top of them, follow the standard UC framework [Can05]. All of our positive results hold with statistical security, and some of our negative results apply also to the case of computational security. We will consider the following types of channels.

Binary Erasure Channel.   The binary erasure channel (BEC) is perhaps the simplest non-trivial channel model considered in the literature. We denote this channel by $\mathcal{C}_{BEC}^{p}$. For this channel, the sender inputs a bit $x \in \{0, 1\}$ and the channel outputs (to the receiver) x with a probability p and $\perp$ with a probability $1 - p$.

Binary Symmetric Channel. The binary symmetric channel (BSC) denoted by $\mathcal{C}_{BSC}^p$ (for $p > \frac{1}{2}$) is a channel in which the sender inputs a bit $x \in \{0,1\}$ and the channel outputs (for the receiver) x with a probability p and $1 - x$ with a probability $1 - p$.

Generalized Erasure Channel. The generalized erasure channel (GEC) is a generalization of the BEC, where k strings are sent by the sender and some subset of them, determined by a probability distribution $\mathcal{D}$, is erased. We denote this channel by $\mathcal{C}_{GEC}^{k,\ell,\mathcal{D}}$. Formally, the functionality takes as input k strings $x_1, \ldots, x_k \in \{0,1\}^\ell$ from the sender. It samples a string $s \in \{0,1\}^k$ (which we call the *randomness of the channel*) according to the distribution $\mathcal{D}$. If $s_i = 1$ then set $y_i = x_i$ and, otherwise, $y_i = \bot$. The functionality outputs $y_1, \ldots, y_k$ to the receiver. We will consider the following special cases of the generalized erasure channel.

- *ℓ-Bit Random Oblivious Transfer.* The ℓ-bit random oblivious transfer channel (ℓ-ROT) denoted by $\mathcal{C}_{ROT}^\ell$ corresponds to the channel $\mathcal{C}_{GEC}^{2,\ell,\mathcal{D}_{2,OT}}$, where $\mathcal{D}_{2,OT}$ is the distribution that outputs a uniformly random value in $\{01, 10\}$. We also consider a p-biased ℓ-bit ROT channel denoted by $\mathcal{C}_{ROT}^{\ell,p}$ corresponds to the channel $\mathcal{C}_{GEC}^{2,\ell,\mathcal{D}_{2,p,OT}}$, where $\mathcal{D}_{2,p,OT}$ is the distribution that outputs 10 with probability p and 01 with a probability $1 - p$.

- *(k, ℓ, p)-Erasure Channel.* The (k, ℓ, p)-erasure channel corresponds to the channel $\mathcal{C}_{GEC}^{k,\ell,\mathcal{D}_{k,p}}$, where $\mathcal{D}_{k,p}$ is the distribution that outputs a k bit string s such that, for every $i \in [k]$, we have $s_i = 1$ with probability p and $s_i = 0$ with probability $1 - p$.

- *(k, ℓ)-Perfect Red-Blue Channel.* The (k, ℓ)-Perfect Red-Blue channel corresponds to the channel $\mathcal{C}_{GEC}^{k,\ell,\mathcal{D}_{k,RB}}$, where $\mathcal{D}_{k,RB}$ is any distribution such that each string in its output space (namely $\{0,1\}^k$) may be labeled either Red or Blue (or none) in a way that $\Pr[\text{Red} \cup \text{Blue}] = 1$, $\Pr[\text{Red}] = \Pr[\text{Blue}]$ and $\forall r \in \text{Red}$ and $\forall s \subseteq r$ we have that $s \notin \text{Blue}$ and, similarly, $\forall b \in \text{Blue}$ and $\forall c \subseteq b$ we have that $c \notin \text{Red}$.[3]

- *$(k, \ell, \mu, \nu, \eta)$-Statistical Red-Blue Channel.* The $(k, \ell, \mu, \nu, \eta)$-Statistical Red-Blue channel is a relaxed version of the Perfect Red-Blue Channel, that corresponds to the channel $\mathcal{C}_{GEC}^{k,\ell,\mathcal{D}_{k,\mu,\nu,\eta}}$, where $\mathcal{D}_{k,\mu,\nu,\eta}$ is any distribution whose output space can be labelled Red and Blue such that (i) $\Pr[\text{Red} \cup \text{Blue}] \geq 1 - \mu$, (ii) $|\Pr[\text{Red}] - \Pr[\text{Blue}]| \leq \nu$, (iii) $\Pr_{r \in \text{Red}}[\exists s \subseteq r$ such that $s \in \text{Blue}] \leq \eta$, and (iv) $\Pr_{b \in \text{Blue}}[\exists c \subseteq b$ such that $c \in \text{Red}] \leq \eta$.

- *(k, ℓ, b)-Perfect Bursty Channel.* This is an erasure channel where all b erasures appear in a "burst". Formally, the (k, ℓ, b)-Perfect bursty channel corresponds to the channel $\mathcal{C}_{GEC}^{k,\ell,\mathcal{D}_{k,b}}$, where $\mathcal{D}_{k,b}$ is the distribution that outputs a k bit string such that all the bits are set to 1 besides the bits in locations $x + 1, x + 2, \ldots, x + b$ where x is chosen uniformly from $\{0, \ldots, k - b\}$.

- *(k, ℓ, b, σ)-Noisy Bursty Channel.* This is an erasure channel where erasures still appear in a "burst" but their number b' is normally distributed around b. Formally, the (k, ℓ, b, σ)-noisy bursty channel corresponds to the channel $\mathcal{C}_{GEC}^{k,\ell,\mathcal{D}_{k,b,\sigma}}$ for typical $k \gg b$, where $\mathcal{D}_{k,b,\sigma}$ is the distribution that outputs a k bit string such that all the bits are set to 1 besides the bits in locations $x + 1, x + 2, \ldots, x + b'$ where b' is sampled from a gaussian and rounded to the closest non-negative integer $\leq k$ with mean b and standard deviation σ and then x is chosen uniformly from $\{0, \ldots, k - b'\}$.

---

[3]Here, again, we identify each $a \in \{0,1\}^k$ with a subset of $[k]$ in the natural way.

5 Classification of functionalities

Below we define the notion of one-way secure computation (OWSC) over a channel $\mathcal{C}$ (thought of as a non-reactive ideal functionality). We shall refer to such a OWSC scheme as $OWSC/\mathcal{C}$.

An $\mathsf{OWSC}^f/\mathcal{C}$ scheme for a function $f : X \to Y$ is a two-party protocol between Sender and Receiver and it follows the following format:

- Sender gets an input $x \in X$.

- Sender invokes the channel $\mathcal{C}$ (possibly multiple instances of the channel) with inputs of its choice. The channel, based on its nature, processes the input value and outputs it to the Receiver.

- Receiver carries out a local computation and outputs $f(x)$ or an error message.

Similarly, we can consider reactive functionality specified by a *stateful* function $f : \Sigma \times X \to \Sigma \times Y$. The Sender of a $\mathsf{OWSC}^f/\mathcal{C}$ scheme for a stateful function f obtains multiple inputs on the fly. On obtaining an input $x \in X$, Sender can invoke the channel $\mathcal{C}$ multiple times and in each execution the Receiver should either output y where $(\sigma', y) \leftarrow f(\sigma, x)$ (where $\sigma \in \Sigma$ is the current state and σ' is the state for the next execution) or an error message. The first execution of the protocol sets the state to ϵ.

The correctness and secrecy requirements of an OWSC scheme can be specified in terms of an ideal functionality. An $\mathsf{OWSC}^f/\mathcal{C}$ scheme for f is required to be a secure realization of the following function $\mathcal{F}_f$ in the $\mathcal{C}$-hybrid model.

- $\mathcal{F}_f$ accepts $x \in X$ from the Sender and outputs $f(x)$ to the receiver. If x is a special input `error`, then it outputs `error` to the Receiver.

We shall denote the security parameter by λ and require that the sender and the receiver in any scheme run in time polynomial in λ and the size of the function f. Further, for a scheme to be considered secure, we require that the simulation error be at most $2^{-\Omega(\lambda)}$.

Definition 2 (Completeness for deterministic functionalities) *A channel $\mathcal{C}$ is said to be* OWSC *complete for deterministic functionalities, if for every deterministic function $f : X \to Y$ there exists a $\mathsf{OWSC}^f/\mathcal{C}$ scheme that is a UC-secure realization of the functionality $\mathcal{F}_f$ in the $\mathcal{C}$-hybrid model.*

Definition 3 (Completeness for randomized functionalities) *A channel $\mathcal{C}$ is said to be* OWSC *complete for randomized functionalities, if for every randomized function $f : X \to Y$ there exists a $\mathsf{OWSC}^f/\mathcal{C}$ scheme that is a UC-secure realization of the functionality $\mathcal{F}_f$ in the $\mathcal{C}$-hybrid model.*

6 Reductions among channels

In this section, we study the relationships between different kinds of channels. Specifically:

- **Impossibility results for $\mathcal{C}_{ROT}$.** One of the key channels of interest to us is the random oblivious transfer channel. We start by establishing (in Section 6.1) that this channel cannot be securely realized out of the most basic channels such as $\mathcal{C}_{BEC}$ (in fact, from any $\mathcal{C}_{GEC}^{k,\ell,\mathcal{D}_{k,p}}$) and $\mathcal{C}_{BSC}$. In Section 9, we provide extensions of these results to the computational setting (but ruling out only protocols with negligible error rather than small noticeable error).

- **Positive results for $\mathcal{C}_{ROT}$.** We consider a variety of more structured channels, such as the Red-Blue channel and the bursty channel, and give constructions of random oblivious transfer channel from such channels (Section 6.2).

- **Self-transformations for $\mathcal{C}_{BEC}$ and $\mathcal{C}_{BSC}$.** We move back to the basic channels ($\mathcal{C}_{BEC}$ and $\mathcal{C}_{BSC}$) and study additional properties of them. Although both these channels do not imply $\mathcal{C}_{ROT}^1$, they are of a very different nature. We show (in Section 6.3) that erasure probabilities of the $\mathcal{C}_{BEC}$ can be easily manipulated but the flipping probability of $\mathcal{C}_{BSC}$ is harder to manipulate. In particular, we show that, given a $\mathcal{C}_{BEC}$, we can construct another $\mathcal{C}_{BEC}$ with amplified or diminished erasure probabilities. On the other hand, given a $\mathcal{C}_{BSC}$, we can only construct another $\mathcal{C}_{BSC}$ with amplified flipping probability. In fact, diminishing the flipping probability turns out to be is impossible.

We remark that all the impossibility results (in this section) are stated in terms of the simulation based notion but hold even for a weaker game-based security notion. These stronger impossibility results are implied by the proofs and are not spelled out explicitly.

6.1 Impossibility results for $\mathcal{C}_{ROT}$

In this subsection, we rule out the construction of $\mathcal{C}_{ROT}^1$ (random oblivious transfer) from the most basic channels such as $\mathcal{C}_{BEC}$ and $\mathcal{C}_{BSC}$. In particular, we show:

- $\mathcal{C}_{ROT}^{\ell'}$ (and, in fact, even biased-ROT) cannot be non-interactively securely realized from $\mathcal{C}_{GEC}^{k,\ell,\mathcal{D}_{k,p}}$.

- $\mathcal{C}_{BEC}^{p'}$ cannot be non-interactively securely realized from $\mathcal{C}_{BSC}^p$. It is easy to realize $\mathcal{C}_{BEC}^{\frac{1}{2}}$ from $\mathcal{C}_{ROT}^{\ell'}$. Hence, combining with the above result, we also conclude that $\mathcal{C}_{ROT}^{\ell'}$ cannot be non-interactively securely realized from $\mathcal{C}_{BSC}^p$.

The following theorem and its proof can be adapted to rule out even $\mathcal{C}_{ROT}^{\ell',q}$ for any constant q. We state the result and the proof in the simpler setting where $q = \frac{1}{2}$.

Theorem 1 $\exists \, \varepsilon \in (0,1)$ and $\ell' \in \mathbb{Z}^+$ such that $\forall k, \ell, p$, the channel $\mathcal{C}_{ROT}^{\ell'}$ cannot be ε-securely realized in the $\mathcal{C}_{GEC}^{k,\ell,\mathcal{D}_{k,p}}$ hybrid model even against semi-honest adversaries.

We start by giving some intuition for the case of binary erasure channel. The intuition extends to (k, ℓ, p)-erasure channels in a natural way. In any protocol for non-interactively realizing $\mathcal{C}_{ROT}^1$ the sender will need to encode both its inputs m_0, m_1 into its first message. Whether the receiver obtains m_0 or m_1 should depend solely on the random coins of the channel. In other words, erasure of certain bits (or more generally one combination from a list of possible choices) allows the receiver to obtain m_0 while erasure of another combination allows the receiver to learn m_1. The key issue is that a binary erasure channel erases each bit sent by the sender independently with a probability $1 - p$. Consider the scenario in which a receiver can obtain m_0 from the received bits. In this scenario, since each bit sent by the sender is treated independently we have that the receiver also obtains m_1 with a large enough probability, contradicting the security of the protocol. Arguing the last step formally is tricky and we rely on the Harris-Kleitman inequality for our argument. The full proof appears next.

Proof of Theorem 1. For the sake of contradiction, lets start by fixing some $\varepsilon > 0$ and assuming that there exists a protocol $\pi = \langle S, R \rangle$ that ε-securely realizes $\mathcal{C}_{ROT}^{\ell'}$ in the $\mathcal{C}_{GEC}^{k,\ell,\mathcal{D}_{k,p}}$ hybrid model. More specifically, π proceeds as follows: S on input two strings $m_0, m_1 \in \{0,1\}^{\ell'}$ generates k strings $\boldsymbol{x} = (x_1, x_2, \ldots, x_k)$ which are provided as input to the functionality $\mathcal{C}_{GEC}^{k,\ell,\mathcal{D}_{k,p}}$. The functionality then outputs strings $\boldsymbol{y} = (y_1, y_2, \ldots, y_k)$ to the receiver R. R processes these values and outputs either $(m_0, \perp)$ or $(\perp, m_1)$. More formally, consider the experiment $\mathsf{EXPT}^{\langle S,R \rangle}(m_0, m_1)$ in Figure 2.

$$\textbf{EXPT}^{\langle S,R \rangle}(m_0, m_1)$$

1. $\boldsymbol{x} \xleftarrow{\$} S(m_0, m_1)$.

2. $\forall i \in [k]$, set $y_i = x_i$ with probability p and $\perp$ with probability $1-p$.

3. Set $z := R(\boldsymbol{y})$.

4. Output $(m_0, m_1, \boldsymbol{x}, \boldsymbol{y}, z)$.

Figure 2: Execution of the $\langle S, R \rangle$ protocol

Properties of the experiment. Let A be the event that $z = (m_0, \perp)$ and similarly let B be the event that $z = (\perp, m_1)$. Then, for the above experiment, we have:

Correctness: It implies that,
$$\Pr[A \cup B] \geq 1 - \varepsilon.$$
Let X' be the set of $\boldsymbol{x}$ such that $\Pr[A \cup B | \boldsymbol{x}] \geq 1 - \sqrt{\varepsilon}$. By a counting argument, $\Pr[\boldsymbol{x} \in X'] \geq 1 - \sqrt{\varepsilon}$. Otherwise, if $\Pr[\boldsymbol{x} \in X'] < 1 - \sqrt{\varepsilon}$, then $\Pr[A \cup B] < 1 \cdot (1 - \sqrt{\varepsilon}) + (1 - \sqrt{\varepsilon}) \cdot \sqrt{\varepsilon} = 1 - \varepsilon$, which is a contradiction.

Receiver privacy: It implies that,
$$|\Pr[A] - \Pr[B]| \leq \varepsilon.$$
The above condition is very weak and, in fact, receiver privacy implies something stronger: for a fixed $\boldsymbol{x}$ we have that A and B happen with roughly the same probability. More formally, receiver privacy implies that for large enough set X'' such that $\Pr[X''] \geq 1 - 2\sqrt{\varepsilon}$, we have that for every $\boldsymbol{x} \in X''$,

$$|\Pr[A|\boldsymbol{x}] - \Pr[B|\boldsymbol{x}]| \leq \sqrt{\varepsilon}.$$

The parameters in the stronger condition above are derived as follows. Assume that $\Pr[\boldsymbol{x} \in X''] < 1 - 2\sqrt{\varepsilon}$, then a cheating sender (with X'' hardcoded in it) can figure out whether A happened or B happened with probability $> (1 - 2\sqrt{\varepsilon}) \cdot \frac{1}{2} + 2\sqrt{\varepsilon} \cdot (\frac{1}{2} + \frac{\sqrt{\varepsilon}}{2}) = \frac{1}{2} + \varepsilon$, which is a contradiction.

Sender privacy: It implies that, no machine M can output both m_0, m_1 correctly using just $\boldsymbol{y}$,

$$\Pr[M(\boldsymbol{y}) = (m_0, m_1)] \leq \varepsilon .$$

Our proof will proceed by using correctness and receiver privacy to prove Lemma 2 which will then be used to reach a contradiction with sender privacy.

Lemma 2 *There exists a set X with $\Pr[\boldsymbol{x} \in X] \geq 1 - 3\sqrt{\varepsilon}$ such that:*

$$\Pr[A|\boldsymbol{x}] \geq \frac{1}{2} - \sqrt{\varepsilon},$$

$$\Pr[B|\boldsymbol{x}] \geq \frac{1}{2} - \sqrt{\varepsilon}.$$

9

PROOF: Consider the set $X = X' \cap X''$. By union bound, we have that $\Pr[\boldsymbol{x} \in X] \geq 1 - 3\sqrt{\varepsilon}$. Furthermore, for all $\boldsymbol{x} \in X$ we have $\Pr[A \cup B|\boldsymbol{x}] \geq 1 - \sqrt{\varepsilon}$ and $|\Pr[A|\boldsymbol{x}] - \Pr[B|\boldsymbol{x}]| \leq \sqrt{\varepsilon}$. This implies that $\forall \boldsymbol{x} \in X$, $\Pr[A|\boldsymbol{x}] \geq \frac{1}{2} - \sqrt{\varepsilon}$ and $\Pr[B|\boldsymbol{x}] \geq \frac{1}{2} - \sqrt{\varepsilon}$, as needed. $\qquad\square$

Contradicting sender privacy. To reach a contradiction, we will construct a machine M that, given $\boldsymbol{y}$, outputs (m_0, m_1) with probability greater than ε.

Some notation. We use $\boldsymbol{y}^r$ for any $r \in \{0,1\}^k$ to denote the vector constructed as follows: $\forall i \in [k]$, set $\boldsymbol{y}_i^r = \boldsymbol{y}_i$ if r_i is 1 and $\bot$ otherwise. Consider the following:

- $\boldsymbol{y} \in C$ if $\Pr[A|\boldsymbol{y}] \geq \frac{1}{4}$.

- $\boldsymbol{y} \in D$ if $\Pr[B|\boldsymbol{y}] \geq \frac{1}{4}$.

- $\boldsymbol{y} \in C'$ if there exists r such that $\Pr[A|\boldsymbol{y}^r] \geq \frac{1}{4}$.

- $\boldsymbol{y} \in D'$ if there exists r such that $\Pr[B|\boldsymbol{y}^r] \geq \frac{1}{4}$.

Now we make the following three observations:

1. If $\boldsymbol{y} \in C$ then $\boldsymbol{y} \in C'$ and, similarly, if $\boldsymbol{y} \in D$ then $\boldsymbol{y} \in D'$.

2. If $\boldsymbol{y} \in C'$, with respect to some r, then any $\boldsymbol{y}'$ such that $\boldsymbol{y}^r = \boldsymbol{y}'^r$ satisfies $\boldsymbol{y}' \in C'$. An analogous condition holds for the event D'.

3. $\Pr[A|\boldsymbol{x}] = \Pr[A|C] \cdot \Pr[C|\boldsymbol{x}] + \Pr[A|\neg C] \cdot \Pr[\neg C|\boldsymbol{x}] \leq \Pr[C|\boldsymbol{x}] + \Pr[A|\neg C]$ which implies that,

$$\Pr[C|\boldsymbol{x}] \geq \Pr[A|\boldsymbol{x}] - \frac{1}{4}, \tag{1}$$

and similarly,

$$\Pr[D|\boldsymbol{x}] \geq \Pr[B|\boldsymbol{x}] - \frac{1}{4}. \tag{2}$$

$$
\begin{aligned}
\Pr[C' \cap D'] &= \sum_{\boldsymbol{x}} \Pr[C' \cap D'|\boldsymbol{x}] \cdot \Pr[\boldsymbol{x}] \\
&\geq \sum_{\boldsymbol{x} \in X} \Pr[C' \cap D'|\boldsymbol{x}] \cdot \Pr[\boldsymbol{x}] \\
&\geq \sum_{\boldsymbol{x} \in X} \Pr[C'|\boldsymbol{x}] \cdot \Pr[D'|\boldsymbol{x}] \cdot \Pr[\boldsymbol{x}] \quad \text{(Observation 2 from above applied to Lemma 1.)} \\
&\geq \sum_{\boldsymbol{x} \in X} \Pr[C|\boldsymbol{x}] \cdot \Pr[D|\boldsymbol{x}] \cdot \Pr[\boldsymbol{x}] \quad (\Pr[C'|\boldsymbol{x}] \geq \Pr[C|\boldsymbol{x}] \ \& \ \Pr[D'|\boldsymbol{x}] \geq \Pr[D|\boldsymbol{x}].) \\
&\geq \sum_{\boldsymbol{x} \in X} \left(\Pr[A|\boldsymbol{x}] - \frac{1}{4}\right) \cdot \left(\Pr[B|\boldsymbol{x}] - \frac{1}{4}\right) \cdot \Pr[\boldsymbol{x}] \quad \text{(By Equations 1 and 2.)} \\
&\geq \left(\frac{1}{4} - \sqrt{\varepsilon}\right)^2 \cdot \sum_{\boldsymbol{x} \in X} \Pr[\boldsymbol{x}] \quad \text{(By Lemma 2.)} \\
&\geq \left(\frac{1}{4} - \sqrt{\varepsilon}\right)^2 \cdot (1 - 3\sqrt{\varepsilon}) \\
&= \frac{(1 - 4\sqrt{\varepsilon})^2 \cdot (1 - 3\sqrt{\varepsilon})}{16}
\end{aligned}
$$

Machine M. M waits for the case in which $\boldsymbol{y} \in C' \cap D'$. In this case, there exist r_1, r_2 such that $\Pr[A|\boldsymbol{y}^{r_1}] \geq \frac{1}{4}$ and $\Pr[B|\boldsymbol{y}^{r_2}] \geq \frac{1}{4}$. Machine M proceeds by executing R on inputs $\boldsymbol{y}^{r_1}$ and $\boldsymbol{y}^{r_2}$, obtaining both m_0 and m_1 with probability $\frac{1}{16}$.

Observe that for $\varepsilon = 0.0004$, we have that M succeeds in outputting both m_0 and m_1 with probability 0.003107875. Hence, protocol π does not have ε-sender security, leading to a contradiction.

Theorem 2 $\forall p \in (\frac{1}{2}, 1)$, $p' \in (0, 1)$ and protocol π, $\exists \varepsilon$ such that π does not ε-securely realize $\mathcal{C}^{p'}_{BEC}$ in the $\mathcal{C}^{p}_{BSC}$-hybrid model even against semi-honest adversaries.

We start by giving some intuition. Any protocol for non-interactively securely realizing $\mathcal{C}_{BEC}$ will need the sender to encode its input m into its first message. Whether the receiver obtains m or not should depend solely on the random coins of the channel. In other words when certain bits (or, more generally, one combination from a list of possible choices) is flipped then the receiver loses all information about m while flipping another combination allows the receiver to learn m completely. Consider a sequence of hybrid strings between a pair of strings on which the receiver outputs m and $\perp$ respectively. Among the hybrid strings there must exist two strings that differ in exactly one bit but are such that the receiver's output on the two differs completely. At this point, we argue that a change of just one bit cannot affect the receiver's best guess about the sent bit very dramatically, contradicting the security of the protocol. The key technical challenge of the proof lies in proving that this happens with a noticeable probability. The full proof appears next.

Proof of Theorem 2. For the sake of contradiction, fix $p \in (\frac{1}{2}, 1)$ and $p' \in (0, 1)$ and assume that there exists a protocol $\pi = \langle S, R \rangle$ that ε-securely realizes $\mathcal{C}^{p'}_{BEC}$ in the $\mathcal{C}^{p}_{BSC}$ hybrid model, for an appropriate ε (to be decided later). More specifically, S on input a bit $m \in \{0, 1\}$ generates k bits $\boldsymbol{x} = (x_1, x_2, \ldots, x_k)$ which are provided as input to the functionality $\mathcal{C}^{p}_{BSC}$. The functionality outputs bits $\boldsymbol{y} = (y_1, y_2, \ldots, y_k)$ to the receiver R, that processes these values and outputs $z \in \{0, 1, \perp\}$. More formally, consider the experiment $\mathsf{EXPT}^{\langle S, R \rangle}(m)$ in Figure 3.

$$\mathsf{EXPT}^{\langle S, R \rangle}(m)$$

1. $\boldsymbol{x} \xleftarrow{\$} S(m)$.

2. $\forall i \in [k]$, set $y_i = x_i$ with probability p and $y_i = 1 - x_i$ with probability $1 - p$.

3. Set $z := R(\boldsymbol{y})$.

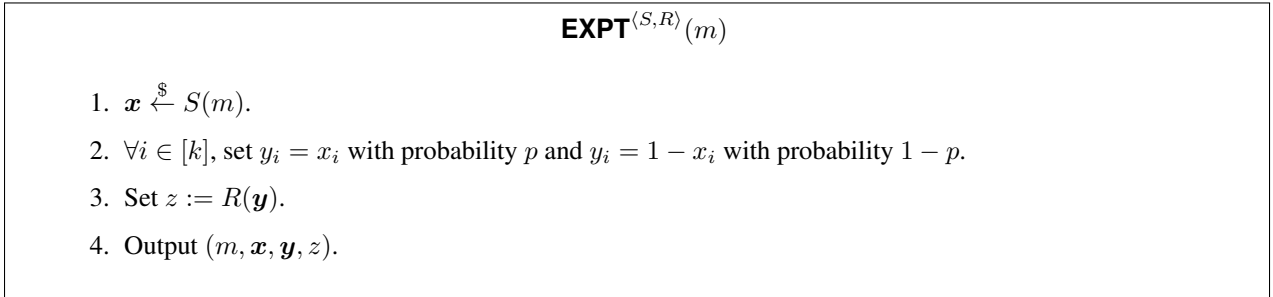4. Output $(m, \boldsymbol{x}, \boldsymbol{y}, z)$.

Figure 3: Execution of the $\langle S, R \rangle$ protocol

Without loss of generality, assume that the receiver R is deterministic at the cost of increasing the error by a constant factor. If the receiver R of the protocol $\langle S, R \rangle$ that ε-securely realizes $\mathcal{C}^{p'}_{BEC}$ is not deterministic then we use $\langle S, R \rangle$ and construct a protocol $\langle S', R' \rangle$ that ε-securely realizes $\mathcal{C}^{p'}_{BEC}$ and furthermore has a deterministic receiver. $\langle S', R' \rangle$ is essentially the same as $\langle S, R \rangle$ except how R' works. R' on input $\boldsymbol{y}$ outputs the most likely (breaking ties arbitrarily) output generated by R on input $\boldsymbol{y}$.

We start by observing that for a large fraction of the inputs $\boldsymbol{y}$, the receiver R must output the same value with high probability. More specifically, let B be the set for which this is not true. For this set, either (1) $\Pr[R(\boldsymbol{y}) = 1 - m] \geq \alpha$, for some small constant α (say $\alpha = \frac{1}{100}$). Then, we claim that $\Pr[B] \leq \frac{\varepsilon}{\alpha}$, as otherwise correctness is violated with probability at least ε. Or (2) $\Pr[R(\boldsymbol{y}) = m] > \alpha$, $\Pr[R(\boldsymbol{y}) = \perp] > \alpha$

11

and $\Pr[R(\boldsymbol{y}) \in \{m, \perp\}] > 1 - \alpha$. Therefore, again, we must have that $\Pr[B] \leq \frac{\varepsilon}{\alpha}$ otherwise contradicting sender privacy.

Properties of the experiment. Let A be the event that $z = m$ and, similarly, let B be the event that $z = \perp$. Then for the above experiment we have:

Correctness: It implies that,
$$\Pr[A \cup B] \geq 1 - \varepsilon.$$

Let X' be a set such that $\forall \boldsymbol{x} \in X'$ we have that $\Pr[A \cup B | \boldsymbol{x}] \geq 1 - \sqrt{\varepsilon}$. By a counting argument (argument follows) it follows that $\Pr[\boldsymbol{x} \in X'] \geq 1 - \sqrt{\varepsilon}$. Lets start by assuming that $\Pr[\boldsymbol{x} \in X'] < 1 - \sqrt{\varepsilon}$. Then we have that $\Pr[A \cup B] < 1 \cdot (1 - \sqrt{\varepsilon}) + (1 - \sqrt{\varepsilon}) \cdot \sqrt{\varepsilon} = 1 - \varepsilon$, which is a contradiction.

Receiver privacy: Next given the set X', using receiver privacy we claim that there exists a set $X \subseteq X'$ such that for all $\boldsymbol{x} \in X$ we have
$$\Pr[A | \boldsymbol{x}] \geq p' - \sqrt{\varepsilon},$$
$$\Pr[B | \boldsymbol{x}] \geq 1 - p' - \sqrt{\varepsilon},$$
and
$$\Pr[X] \geq 1 - 3\sqrt{\varepsilon}.$$

The parameters in the above condition above are derived as follows. Lets assume that $\Pr[\boldsymbol{x} \in X] < 1 - 3\sqrt{\varepsilon}$, then we have that a cheating sender (with X hardcoded in it) can figure out whether A happened with (ignoring the case when $\boldsymbol{x} \notin X'$, that is probability $\sqrt{\varepsilon}$) probability $> (1 - 2\sqrt{\varepsilon}) \cdot p' + 2\sqrt{\varepsilon} \cdot (p' + \frac{\sqrt{\varepsilon}}{2}) = p' + \varepsilon$, which is a contradiction.

Towards Contradiction. Define a set Bad of vectors. Roughly speaking, a vector $\boldsymbol{b} \in$ Bad if

- $R(\boldsymbol{b}) \notin \{m, \perp\}$, or

- $\Pr[R(\boldsymbol{b}) = 1 - m | \boldsymbol{y} = \boldsymbol{b}] > \frac{1}{2}$, or

- $\Pr[\exists\, c \in \{0, 1\} \text{ such that } m = c \land R(\boldsymbol{b}) = \perp | \boldsymbol{y} = \boldsymbol{b}] \geq \frac{1-p}{2p}$.

Observe that all we are left to argue is that that $\Pr[\mathsf{Bad}] > \frac{6p\varepsilon}{1-p}$. Because if $\Pr[\mathsf{Bad}] > \frac{6p\varepsilon}{1-p}$, then at least one of the above three conditions happens with probability at least $\frac{2p\varepsilon}{1-p}$ contradicting correctness, receiver guarantee or receiver guarantee, respectively.

Probability of Bad. Lets restrict ourselves to the case in which $\boldsymbol{x} \in X$. This happens with probability at least $1 - 3\sqrt{\varepsilon}$. Consider any two vectors $\boldsymbol{y}$ and $\boldsymbol{y}'$ such that $R(\boldsymbol{y}) = m$ and $R(\boldsymbol{y}') = \perp$. Now consider $k + 1$ hybrid-vectors $\boldsymbol{h}_0, \boldsymbol{h}_1, \ldots, \boldsymbol{h}_k$ where $\boldsymbol{h}_i = (y_1', y_2', \ldots, y_i', y_{i+1}, \ldots, y_k)$. Note that $\boldsymbol{h}_0 = \boldsymbol{y}$ and $\boldsymbol{h}_k = \boldsymbol{y}'$. Note that $R(\boldsymbol{h}_0) = m$ and $R(\boldsymbol{h}_k) = \perp$. We will show that at least one of these hybrid-vectors is in Bad.

If R on any one of these hybrid-vectors outputs a value not in $\{m, \perp\}$ then that vector is clearly in Bad. On the other hand, if R on each of these hybrid-vectors outputs a value in $\{m, \perp\}$ then that implies a switch at some hybrid-vector. In other words $\exists j \in \{0, 1 \ldots, k-1\}$ such that $R(\boldsymbol{h}_j) = m$ and $R(\boldsymbol{h}_{j+1}) = \perp$. Namely, the output of R was switched when just one of the input bits to R was flipped. Next, we argue that R's best guess about the sent bit cannot have changed substantially with this one bit flip. In other words, we will show that either $\boldsymbol{h}_j$ or $\boldsymbol{h}_{j+1}$ is in Bad.

If $\Pr[R(\boldsymbol{h}_j) = 1 - m | \boldsymbol{y} = \boldsymbol{h}_j] > \frac{1}{2}$ then we are done. Otherwise, we have that $\boldsymbol{h}_{j+1}$ differs from $\boldsymbol{h}_j$ in only one bit therefore we can conclude that R's guess about the sent bit can decrease at most by a factor of $\frac{1-p}{p}$ (recall that $p \in (\frac{1}{2}, 1)$). In other words, $\Pr[R(\boldsymbol{h}_j) = m | \boldsymbol{y} = \boldsymbol{h}_{j+1}] \geq \frac{1}{2} \cdot \frac{1-p}{p}$ where $(m, \boldsymbol{x}, \boldsymbol{y}, z) \leftarrow \mathsf{EXPT}^{\langle S,R \rangle}(m)$. However, $R(\boldsymbol{h}_{j+1}) = \bot$. In other words R outputs $\bot$ even though it can guess the sent value correctly with a probability at least $\frac{1-p}{2p}$. In summary, we claim that for every two vectors $\boldsymbol{y}$ and $\boldsymbol{y}'$ such that $R(\boldsymbol{y}) = m$ and $R(\boldsymbol{y}') = \bot$ there exists a hybrid vector (as defined above) that is in Bad.

$\mathsf{EXPT}'^{\langle S,R \rangle}(m)$

1. $\boldsymbol{x} \xleftarrow{\$} S(m)$.

2. $\forall i \in [k]$, set $y_i = x_i$ with probability p and $y_i = 1 - x_i$ with probability $1 - p$.

3. $\forall i \in [k]$, set $y_i' = x_i$ with probability p and $y_i' = 1 - x_i$ with probability $1 - p$.

4. Choose j randomly in $[k]$ and let $w_i = y_i$ for $i < j$ and $w_i = y_i'$ for $i \geq j$. Finally set $z := R(\boldsymbol{w})$.

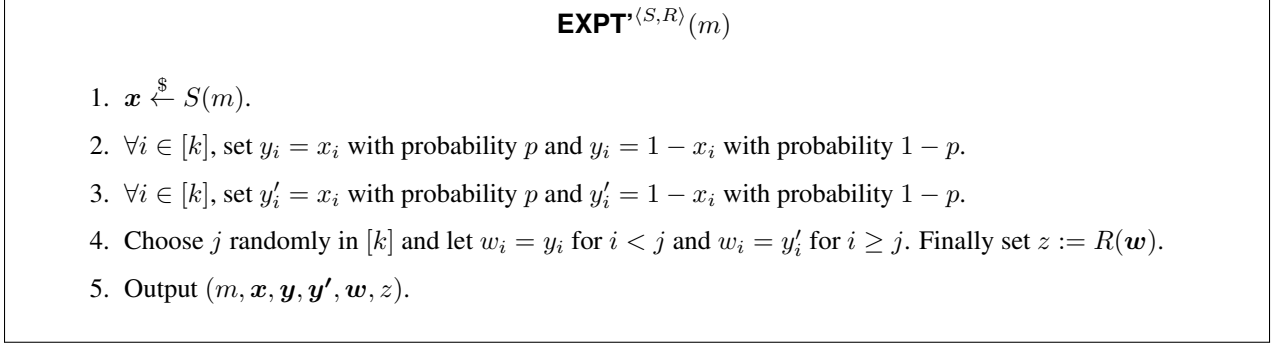5. Output $(m, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{y}', \boldsymbol{w}, z)$.

Figure 4: Modified Execution of the $\langle S, R \rangle$ protocol

Now we are left to argue that R receives the elements in Bad with noticeable probability. We argue this by considering a modified experiment $\mathsf{EXPT}'^{\langle S,R \rangle}(m)$. We start by noting that, in this experiment, the distribution of each of the outputs $\boldsymbol{y}, \boldsymbol{y}', \boldsymbol{w}$ individually is identical to the distribution of the $\boldsymbol{y}$ in the output of $\mathsf{EXPT}^{\langle S,R \rangle}(m)$. This, along with conditions proved earlier, implies that $\Pr[R(\boldsymbol{y}) = m \bigwedge R(\boldsymbol{y}') = \bot] \geq (1 - 3\sqrt{\varepsilon}) \cdot (1 - p' - \sqrt{\varepsilon}) \cdot (p' - \sqrt{\varepsilon})$, where $(m, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{y}', \boldsymbol{w}, z) \leftarrow \mathsf{EXPT}'^{\langle S,R \rangle}(m)$.

Finally, observe that in the experiment $\mathsf{EXPT}'^{\langle S,R \rangle}(m)$, given that $R(\boldsymbol{y}) = m$ and $R(\boldsymbol{y}') = \bot$, there exists a hybrid vector (as defined above) such that it is in Bad. Furthermore, $\boldsymbol{w}$ takes this value with probability $\frac{1}{k}$. In other words, $\boldsymbol{w} \in \mathsf{Bad}$ with probability at least $(1 - 3\sqrt{\varepsilon}) \cdot (1 - p' - \sqrt{\varepsilon}) \cdot (p' - \sqrt{\varepsilon}) \cdot \frac{1}{k}$. Since the distribution of $\boldsymbol{w}$ in the output of $\mathsf{EXPT}'^{\langle S,R \rangle}(m)$ is identical to the distribution of $\boldsymbol{y}$ in the output of $\mathsf{EXPT}^{\langle S,R \rangle}(m)$, we can conclude that $\boldsymbol{y}$ in the output of $\mathsf{EXPT}^{\langle S,R \rangle}(m)$ is in Bad with probability at least $(1 - 3\sqrt{\varepsilon}) \cdot (1 - p' - \sqrt{\varepsilon}) \cdot (p' - \sqrt{\varepsilon}) \cdot \frac{1}{k} \geq \frac{1}{2} \cdot \frac{1-p'}{2} \cdot \frac{p'}{2} \cdot \frac{1}{k} > \frac{6p\varepsilon}{1-p}$ (when ε is at most $\frac{1}{36}$, $\left(\frac{1-p'}{2}\right)^2$, $\left(\frac{p'}{2}\right)^2$ and $\frac{p'(1-p')(1-p)}{48pk}$). This is a contradiction when we set $\varepsilon < \min\left\{\left(\frac{1-p'}{2}\right)^2, \left(\frac{p'}{2}\right)^2, \frac{p'(1-p')(1-p)}{48pk}\right\}$.

6.2 Positive constructions for $\mathcal{C}_{ROT}$

We start by presenting a construction of a random oblivious transfer channel in Red-Blue channel hybrid model. Our construction provides a solution for any arbitrary Red-Blue channel and is inefficient. Furthermore, such a channel in its generality is not very natural. Therefore, we study natural examples of Red-Blue channels (and their approximate variants) and attempt at more efficient solutions.

We start by considering the basic setting of an arbitrary Red-Blue Channel and prove that it is sufficient to realize a random oblivious transfer channel.

Theorem 3 $\mathcal{C}_{ROT}^{\ell}$ *can be* $\max\{\mu, \nu, \eta\}$*-UC-securely realized (even against malicious adversaries) in the* $(k, \ell', \mu, \nu, \eta)$*-Red-Blue Channel hybrid model where* $\ell' = \ell \cdot 2^k$.

Proof of Theorem 3. We start by giving our construction of $\mathcal{C}_{ROT}^{\ell}$ in the $(k, \ell', \mu, \nu, \eta)$-Red-Blue Channel hybrid model. Recall that for a red blue channel $\mathcal{C}_{GEC}^{k,\ell,\mathcal{D}_{k,\mu,\nu,\eta}}$ the distribution $\mathcal{D}_{k,\mu,\nu,\eta}$ is such that its output space (namely $\{0,1\}^k$) can be partitioned into two events Red and Blue such that $\Pr[\mathsf{Red} \cup \mathsf{Blue}] \geq 1 - \mu, |\Pr[\mathsf{Red}] - \Pr[\mathsf{Blue}]| \leq \nu$ and $\Pr_{r \in \mathsf{Red}}[\exists s \subseteq r$ such that $s \in \mathsf{Blue}] \leq \eta$ and $\Pr_{b \in \mathsf{Blue}}[\exists c \subseteq b$ such that $c \in \mathsf{Red}] \leq \eta$.

<div style="border:1px solid black; padding:10px;">

$\Pi = \langle S, R \rangle$ **protocol with sender input** m_0, m_1

1. Let $r_1, r_2 \ldots, r_n$ be the elements of Red in lexicographic order. For each $i \in [n]$, for each $j \in [k]$ let $\alpha_{i,j}$ be randomly chosen strings in $\{0,1\}^{\ell}$ subject to the constraints: $\bigoplus_{j \in [k]} \alpha_{i,j} = m_0$ and $\forall j \in [k]$ such that $r_{i,j} = 0$ (the jth bit of r_i is zero) we have $\alpha_{i,j} = 0^{\ell}$.

2. Similarly let $b_1, b_2 \ldots b_{n'}$ be the elements of Blue in lexicographic order and then for each $i \in [n], j \in [k]$ let $\beta_{i,j}$ be randomly chosen strings in $\{0,1\}^{\ell}$ subject to the constraint that $\bigoplus_{j \in [k]} \beta_{i,j} = m_1$ and $\forall j \in [k]$ such that $b_{i,j} = 0$ (the jth bit of b_i is zero) we have $\beta_{i,j} = 0^{\ell}$.

3. For each $j \in [k]$ let $s_j = \alpha_{1,j} || \alpha_{2,j} \ldots \alpha_{n,j} || \beta_{1,j} || \beta_{2,j} \ldots \beta_{n',j}$.

4. The sender sends $s_1, s_2 \ldots s_k$ invoking the Red-Blue Channel.

5. Let $t \in \{0,1\}^k$ be a string such that for each $j \in [k]$, $t_j = 1$ (t_j being the jth bit of j) if and only if the receiver obtained s_j.

6. If $t = r_i \in \mathsf{Red}$ then the receiver obtains $\alpha_{i,j}$ for each $j \in [k]$ either from s_j if $r_{i,j} = 1$ and by setting it to 0^{ℓ} otherwise. Finally it computes $m_0 = \bigoplus_{j \in [k]} \alpha_{i,j}$ and outputs $(m_0, \perp)$. On the other hand, if $t \in \mathsf{Blue}$ then similarly compute m_1 and output $(\perp, m_1)$. If $t \notin \mathsf{Red} \cup \mathsf{Blue}$ then output $\perp$.
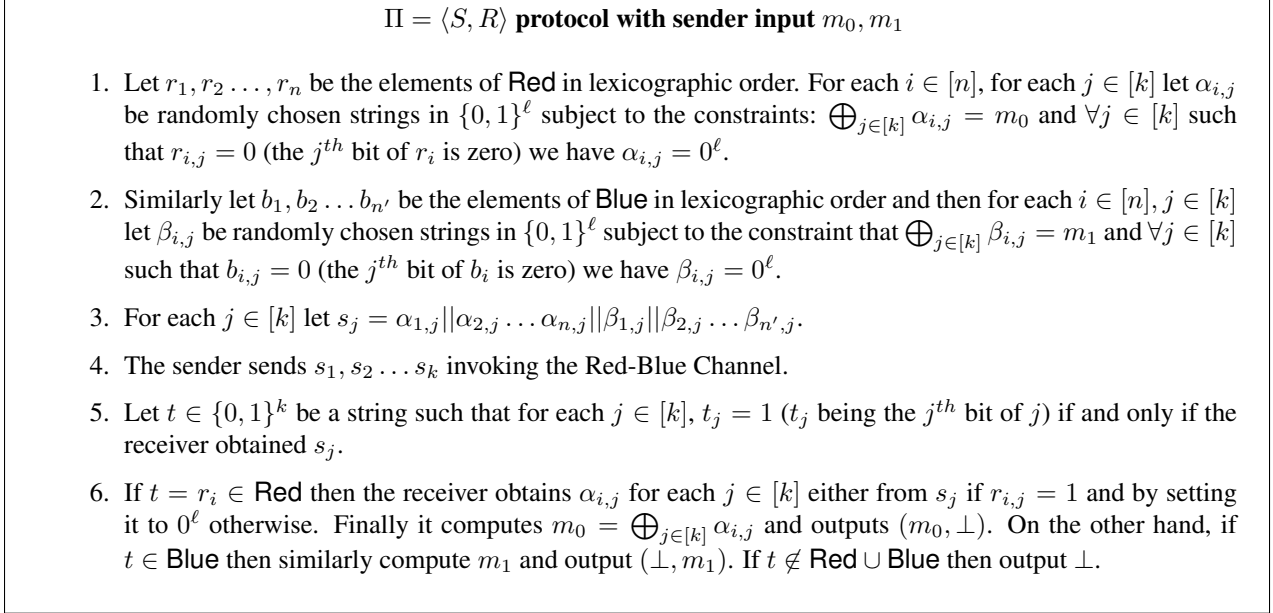
</div>

Figure 5: $\mathcal{C}_{ROT}^{\ell}$ in the $(k, \ell', \mu, \nu, \eta)$-Red-Blue Channel hybrid model

---

Correctness. Using the fact that $\Pr[\mathsf{Red} \cup \mathsf{Blue}] \geq 1 - \mu$, it follows that the receiver always (except with probability μ) outputs either $(m_0, \perp)$ or $(\perp, m_1)$. This proves correctness of our protocol.

Simulating Cheating Receiver. The simulator for a cheating receiver proceeds as follows. Obtain that value that needs to be forced upon the cheating receiver from the ideal functionality. If the value provided by the ideal functionality is $(m_0, \perp)$ then proceed as follows. Using rejection sampling, sample a string $t \in \mathsf{Red}$ from $\mathcal{D}_{k,\mu,\nu,\eta}$. Now generate the messages generated by an honest sender on input (m_0, m_1) for a random value of m_1. The output generated consists of k strings. For each $i \in [k]$, replace the ith string with $\perp$ if $t_i = 0$. Provide the generated string for the receiver. It is straightforward to see that this simulation generates a perfectly indistinguishable transcript. We skip the full proof.

Simulating Cheating Sender. Now we provide the simulator for the cheating sender. From the sender's message s we can obtain both m_0 and m_1, which can then be passed on to the ideal functionality. The problem with this strategy is that a malicious sender could generate s maliciously and in doing so embed different messages causing the receiver to output different outputs depending on the random coins of the channel. However note this is easy to handle by having the simulator sample two random strings t, t' from $\mathcal{D}_{k,\mu,\nu,\eta}$ such that $t \in \mathsf{Red}$ and $t' \in \mathsf{Blue}$. It could then obtain the messages m_0 (using the string t) and m_1 (using the string t') which it could then forward to the ideal functionality. This ensures that the output distributions in the real world and the ideal world are identical. Note that for the case of perfect Red-Blue Channel, we have that $\mu = \nu = \eta = 0$, and hence $\mathcal{C}_{ROT}^{\ell}$ can be perfectly-UC-securely realized in the (k, ℓ')-Perfect Red-Blue Channel hybrid model where $\ell' = \ell \cdot 2^k$.

Efficient construction for ROT. We will start by considering the case of perfect bursty channel and show that it can be used to realize ROT. Recall that a (k, ℓ, b)-perfect bursty channel corresponds to the channel $\mathcal{C}_{GEC}^{k,\ell,\mathcal{D}_{k,b}}$, where $\mathcal{D}_{k,b}$ is the distribution that outputs a k bit string such that all the bits are set to 1 besides the "burst" of bits in locations $x+1, x+2, \ldots, x+b$ which are set to 0, where x is chosen uniformly from $\{0, \ldots, k-b\}$. In this setting we claim that:

Theorem 4 $\mathcal{C}_{ROT}^{\ell}$ *can be UC-securely realized (even against malicious adversaries) in the (k, ℓ, b)-perfect bursty channel hybrid model when $b > \frac{k}{2}$ or when b is odd.*

PROOF: We start by giving the intuition. The key idea is to use Shamir's secret sharing (with shares of length ℓ) and secret share the first string in the first half and the second string in the second half (with some appropriate threshold). Both when $b > \frac{k}{2}$ or when b is odd we will have an asymmetry in terms of the deletion pattern. If more terms from the first half are erased then the first string is deleted and, on the other hand, if more terms from the second half get erased then the second string is deleted. If k is odd then our construction will only give a biased-ROT but this bias can be corrected using the transformation from Section 8. Similarly, we note that in our construction we do not need the distribution over where the burst happens to be uniform. Our protocol can be very easily modified so that this restriction is not crucial. This would however only give biased ROT protocols and this bias will need to be corrected using the transformation from Section 8.

Next we give the construction for the case when b is odd. We assume, for simplicity, that k is even and $t = \frac{k}{2}$. The construction for the setting when k is odd or when b is not necessarily odd but $k > b/2$ are identical except that the parameters should be adjusted appropriately.
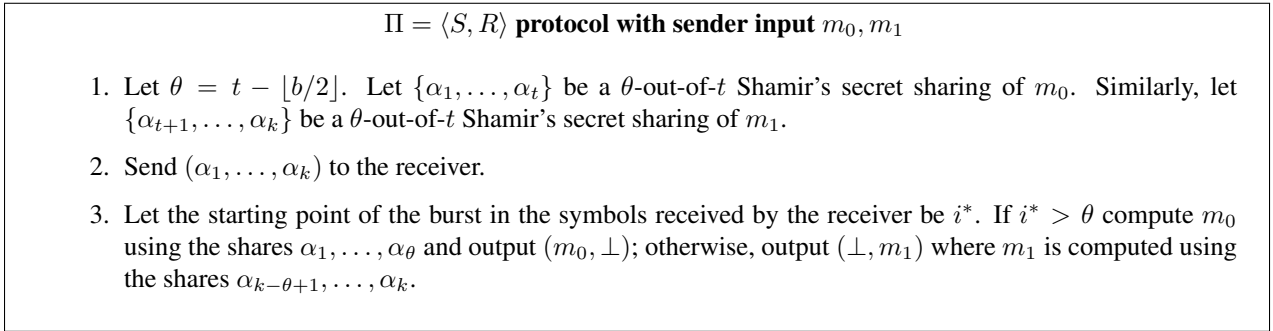

$\Pi = \langle S, R \rangle$ **protocol with sender input** m_0, m_1

1. Let $\theta = t - \lfloor b/2 \rfloor$. Let $\{\alpha_1, \ldots, \alpha_t\}$ be a θ-out-of-t Shamir's secret sharing of m_0. Similarly, let $\{\alpha_{t+1}, \ldots, \alpha_k\}$ be a θ-out-of-t Shamir's secret sharing of m_1.

2. Send $(\alpha_1, \ldots, \alpha_k)$ to the receiver.

3. Let the starting point of the burst in the symbols received by the receiver be i^*. If $i^* > \theta$ compute m_0 using the shares $\alpha_1, \ldots, \alpha_\theta$ and output $(m_0, \perp)$; otherwise, output $(\perp, m_1)$ where m_1 is computed using the shares $\alpha_{k-\theta+1}, \ldots, \alpha_k$.

Figure 6: $\mathcal{C}_{ROT}^{\ell}$ in the (k, ℓ, b)-perfect bursty channel hybrid model, for odd b

The construction appears in Figure 6. Since b is odd, either in the first half or in the second half at least $\lceil b/2 \rceil$ of the strings are erased and hence that value remains hidden. On the other hand, in the other half the value can always be computed since at most $\lfloor b/2 \rfloor$ strings are deleted. The proof is identical to the case of Red-Blue Channel (proved earlier) and is therefore omitted. $\square$

Channel with Imprecise Burst. Finally, we consider a bursty erasure channel where the size of burst is not precisely known but comes from roughly a discrete gaussian distribution. Recall that (k, ℓ, b, σ)-noisy bursty channel corresponds to the channel $\mathcal{C}_{GEC}^{k,\ell,\mathcal{D}_{k,b,\sigma}}$, where $\mathcal{D}_{k,b,\sigma}$ is the distribution that outputs a k bit string such that all the bits are set to 1 besides the bits in locations $x+1, x+2, \ldots, x+b'$ where b' is sampled from a gaussian and rounded to the closest non-negative integer $\leq k$ with mean b and standard deviation σ and then x is chosen uniformly from $\{0, \ldots, k-b'\}$.

Theorem 5 $\mathcal{C}_{ROT}^{\ell}$ can be $\frac{(1-\alpha)b}{k-(1+\alpha)b} + \frac{\sigma^2}{\alpha^2 b^2}$-UC-securely realized in the (k, ℓ, b, σ)-noisy bursty channel hybrid model for any constant $\alpha \in (0, 1)$.

PROOF: We use the same construction as in Figure 6 except the threshold parameter θ of the Shamir secret sharing. We set it up in a way so that it is possible to obtain m_0 if less than $(1-\alpha)b/2$ symbols are erased from the first half. Similarly secret sharing is done for the second half. By Chebyshev's inequality, the probability that the size of the burst, b', lies outside the range $\{(1-\alpha)b, \ldots, (1+\alpha)b\}$ is at most $\frac{\sigma^2}{\alpha^2 b^2}$ (if b' is too big the receiver may not learn any value, while if b' is too small it may learn both values). Assuming this does not happen, then the receiver gets only one of the sent values as long as the burst does not happen "in the middle" (i.e., $(1-\alpha)b/2$ symbols are erased from each half). The probability that the burst happens in the middle is at most $\frac{(1-\alpha)b}{k-(1+\alpha)b}$. □

6.3 Self-transformations for $\mathcal{C}_{BEC}$ and $\mathcal{C}_{BSC}$

In this subsection, we show that any erasure channel can be used to construct a binary erasure channel with any desired erasure probability. On the other hand, the case of BSC is very different. The probability of correct transmission in a BSC channel can be reduced but cannot be increased. Formally,

Theorem 6 $\forall \, \mathcal{C}_{GEC}^{k,\ell,\mathcal{D}}$ such that $\mathcal{D}$ is not a constant distribution, $\exists \, p$ such that $\mathcal{C}_{BEC}^{p}$ can be (perfectly) UC-securely realized (even against malicious adversaries) in the $\mathcal{C}_{GEC}^{k,\ell,\mathcal{D}}$-hybrid model.

Proof of Theorem 6. This construction is very straightforward. For every channel $\mathcal{C}_{GEC}^{k,\ell,\mathcal{D}}$ we claim that there exists $i \in [k]$ such that $\Pr[r_i = 1 | r \leftarrow \mathcal{D}]$ is bounded away from both 0 and 1. This follows from the fact that $\mathcal{D}$ is not a constant distribution. Let the probability that this bit is 1 be p. It is easy to implement a binary erasure channel by embedding the bit that we want to transmit in the ith string sent to $\mathcal{C}_{GEC}^{k,\ell,\mathcal{D}}$. As argued above this string (and hence the desired bit) will be received with probability p. The security of the constructed BEC channel follows in a straight forward manner from the security of the underlying $\mathcal{C}_{GEC}$ channel and the full proof for the same is therefore skipped.

Theorem 7 $\forall p, p' \in (0, 1)$ and $\epsilon > 1$, $\exists p'' \in [p', \epsilon p']$, such that $\mathcal{C}_{BEC}^{p''}$ can be (perfectly) UC-securely realized (even against malicious adversaries) in the $\mathcal{C}_{BEC}^{p}$-hybrid model.

Proof of Theorem 7. We will start by giving protocols for boosting and diminishing the probabilities with which the BEC channel transmits. We will then use these transformations repeated to obtain a BEC with the desired transmission probability.

Boosting. The protocol for boosting the transmission probability of BEC is provided in Figure 7. It is easy to see that the protocol transmits the bit with probability $1 - (1-p_1)(1-p_2)$. Also, arguing security against cheating receivers is straight forward. However, proving security against cheating senders requires some care. More specifically, given an message of the sender for the ideal functionalities $\mathcal{C}_{BEC}^{p_1}$ and $\mathcal{C}_{BEC}^{p_2}$, our simulation we will need to generate a message for the functionality $\mathcal{C}_{BEC}^{1-(1-p_1)(1-p_2)}$. If the cheating sender sends 00 then the simulator sends 0 to $\mathcal{C}_{BEC}^{1-(1-p_1)(1-p_2)}$. Similarly if the cheating sender sends 11 then the simulator sends 1 to $\mathcal{C}_{BEC}^{1-(1-p_1)(1-p_2)}$. The complication arises in simulation when the cheating sender sends different bits. In particular if it sends the bits 01 or 10. In this case our simulator simulates by sending the first bit with probability $\frac{p_1}{p_1+p_2-p_1 p_2}$ and the second one with probability $\frac{p_2-p_1 p_2}{p_1+p_2-p_1 p_2}$. This will ensure that the output distributions in the real and the ideal world are identical.
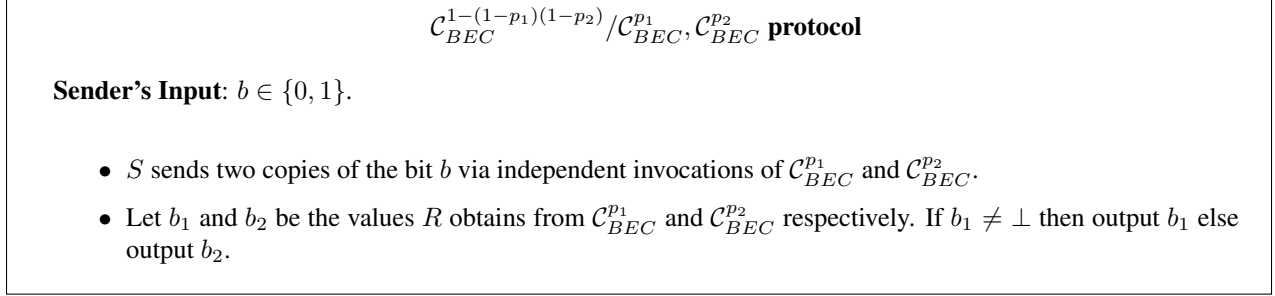

$$\mathcal{C}_{BEC}^{1-(1-p_1)(1-p_2)}/\mathcal{C}_{BEC}^{p_1}, \mathcal{C}_{BEC}^{p_2} \text{ protocol}$$

Sender's Input: $b \in \{0, 1\}$.

- S sends two copies of the bit b via independent invocations of $\mathcal{C}_{BEC}^{p_1}$ and $\mathcal{C}_{BEC}^{p_2}$.
- Let b_1 and b_2 be the values R obtains from $\mathcal{C}_{BEC}^{p_1}$ and $\mathcal{C}_{BEC}^{p_2}$ respectively. If $b_1 \neq \perp$ then output b_1 else output b_2.

Figure 7: Boosting transmission probability in Binary Erasure Channel

Diminishing. The protocol for diminishing the transmission probability of BEC is provided in Figure 8. The basic idea is that the bit sent by the sender remains completely hidden from the receiver as long as at least one of the bits sent by the receiver is not obtained. We skip a formal argument.

$$\mathcal{C}_{BEC}^{p_1 p_2}/\mathcal{C}_{BEC}^{p_1}, \mathcal{C}_{BEC}^{p_2} \text{ protocol}$$

Sender's Input: $b \in \{0, 1\}$.

- S samples a uniformly random bit a and sends a and $a \oplus b$ via independent invocations of $\mathcal{C}_{BEC}^{p_1}$ and $\mathcal{C}_{BEC}^{p_2}$ respectively.
- Let b_1 and b_2 be the values R obtains from $\mathcal{C}_{BEC}^{p_1}$ and $\mathcal{C}_{BEC}^{p_2}$ respectively. If $b_1 \neq \perp$ and $b_2 \neq \perp$ then output $b_1 \oplus b_2$ else output $\perp$.
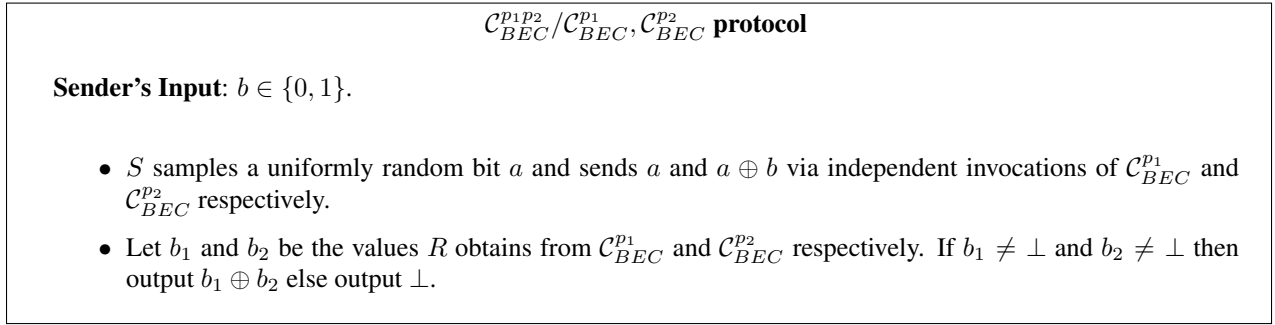
Figure 8: Diminishing transmission probability in Binary Erasure Channel

Putting things together. Now we will describe how to use the above transformations to obtain a $\mathcal{C}_{BEC}^{p''}$ from $\mathcal{C}_{BEC}^{p}$ for some p'' such that $p'' \in [p', \epsilon p']$. First we will proceed by repeatedly using the boosting transformation and boost the transmission probability of the channel to be just above the desired value of p' and then we will diminish it slowly to the desired level. Let r be the smallest positive integer such that $p_0 = 1 - (1 - p)^r \geq p'$. (Note that if $p' < p$ then r will be 1 and $p_0 = p$.) We note that constant positive integer r exists because values $(1 - p)^r$ tends to 0 as r becomes large. Similarly, repeated applying the boosting transforation we obtained a BEC with transmission probability f such that $f > \frac{1}{\epsilon}$.

Now step by step we need to diminish the probability p_0 and bring it closer to the value p'. In particular, let t be a positive integer such that $p_0 \cdot f^{t+1} \leq p' \leq p_0 \cdot f^t$. Exitance of such a t is guaranteed by the fact that f^t tends to 0 as t becomes large. Finally note that $p_0 \cdot f^t \in [p', \epsilon p']$.

Theorem 8 $\forall p \in (\frac{1}{2}, 1)$ and $t \in \mathbb{Z}^+$, the channel $\mathcal{C}_{BSC}^{p'}$ can be (perfectly) UC-securely realized (even against malicious adversaries) in the $\mathcal{C}_{BSC}^{p}$-hybrid model where $p' = \frac{1}{2} + 2^{t-1} \left(p - \frac{1}{2} \right)^t$.

Proof of Theorem 8. We will start by giving a protocol for diminishing the probability with which the BSC channel transmits correctly. We will then use this transformation repeated to obtain a BSC channel with the desired transmission probability.

The protocol for diminishing the transmission probability of BSC is provided in Figure 9. Roughly speaking the sender proceeds by generating two bits such that their exclusive or corresponds to its input. It then sends the two bits to the receiver via two separate invocations of BSC. The receiver outputs the

exclusive or of the received bits. The key idea is that received bit matches the senders input if both bits are transmitted as such or if both bits are flipped. This happens with probability $p_1 p_2 + (1 - p_1)(1 - p_2)$ which evaluates to $\frac{1}{2} + 2\left(p_1 - \frac{1}{2}\right)\left(p_2 - \frac{1}{2}\right)$. Security follows immediately and we skip the formal argument.
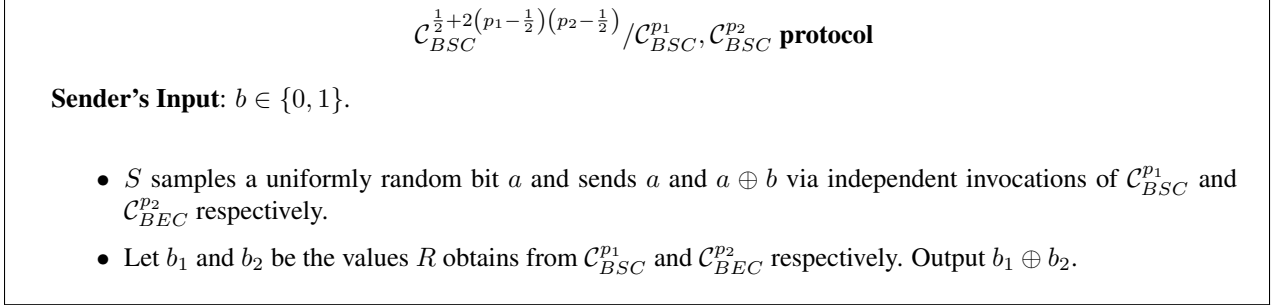

$\mathcal{C}_{BSC}^{\frac{1}{2} + 2\left(p_1 - \frac{1}{2}\right)\left(p_2 - \frac{1}{2}\right)} / \mathcal{C}_{BSC}^{p_1}, \mathcal{C}_{BSC}^{p_2}$ **protocol**

Sender's Input: $b \in \{0, 1\}$.

- S samples a uniformly random bit a and sends a and $a \oplus b$ via independent invocations of $\mathcal{C}_{BSC}^{p_1}$ and $\mathcal{C}_{BEC}^{p_2}$ respectively.
- Let b_1 and b_2 be the values R obtains from $\mathcal{C}_{BSC}^{p_1}$ and $\mathcal{C}_{BEC}^{p_2}$ respectively. Output $b_1 \oplus b_2$.

Figure 9: Diminishing transmission probability in Binary Symmetric Channel

Now we can repeated use the above transformations to obtain a $\mathcal{C}_{BSC}^{p'}$ from $\mathcal{C}_{BSC}^{p}$ for $p' = \frac{1}{2} + 2^{t-1}\left(\frac{1}{2} - p\right)^t$ for any non-negative integer t.

Theorem 9 $\forall\, p, p' \in (\frac{1}{2}, 1), p' > p$ and protocol π, $\exists \varepsilon$ such that π does not ε-securely realize $\mathcal{C}_{BSC}^{p'}$ in the $\mathcal{C}_{BSC}^{p}$-hybrid model even against semi-honest adversaries.

Proof of Theorem 9. The proof is very similar to the proof of Theorem 2. We start by giving some intuition. Any protocol for non-interactively securely realizing $\mathcal{C}_{BSC}^{p'}$ will need the sender to encode its input m into its first message. Whether the receiver obtains m or $1 - m$ should depend solely on the random coins of the channel. In other words when certain bits (or more generally one combination from a list of possible choices) is flipped then the receiver outputs m while on the other hand flipping of another combination makes the receiver output $1 - m$. Consider a sequence of hybrid strings between a pair of strings on which the receiver outputs m and $1 - m$ respectively. Among these hybrid strings there must exist two strings such that they differ in exactly one bit but are such that the receiver's output on the two are different. At this point we argue that change of just one bit cannot affect the receiver's best guess about the received bit very dramatically contradicting security of the protocol. The key technical challenge of the proof then lies in proving that this happens with a noticeable probability.

For the sake of contradiction lets start by fixing $p, p' \in (\frac{1}{2}, 1)$ for $p' > p$ and assuming that there exists a protocol $\pi = \langle S, R \rangle$ that ε-securely realizes $\mathcal{C}_{BSC}^{p'}$ in the $\mathcal{C}_{BSC}^{p}$ hybrid model for appropriate ε, to be decided later. More specifically, S on input a bit $m \in \{0, 1\}$ generates k bits $\boldsymbol{x} = (x_1, x_2 \ldots x_k)$ which are provided as input to the functionality $\mathcal{C}_{BSC}^{p}$. The functionality outputs bits $\boldsymbol{y} = (y_1, y_2 \ldots y_k)$ to the receiver R. R processes these values and outputs $z \in \{0, 1\}$. More formally, consider the experiment $\mathsf{EXPT}^{\langle S,R \rangle}(m)$ in Figure 10.

Without loss of generality we assume that the receiver of the protocol is deterministic at the cost of increasing the error by a constant factor. If the receiver R of the protocol $\langle S, R \rangle$ that ε-securely realizes $\mathcal{C}_{BSC}^{p'}$ is not deterministic then we use $\langle S, R \rangle$ and construct a protocol $\langle S', R' \rangle$ that ε-securely realizes $\mathcal{C}_{BSC}^{p'}$ and furthermore has a deterministic receiver. $\langle S', R' \rangle$ is essentially the same as $\langle S, R \rangle$ except how R' works. R' on input $\boldsymbol{y}$ outputs the best guess (breaking ties arbitrarily) about the bit sent by S'.

We argue that that R and R' output the same value for a large fraction of inputs $\boldsymbol{y}$ with probability $1 - \alpha$. Let C be the set of choices of $\boldsymbol{y}$ for which this is not true. We claim that $\Pr[C] \leq \frac{\varepsilon}{\alpha}$ for some large constant α. Because if this were not the case then we will have that sender privacy is violated with probability $> \varepsilon$ in the sense that the receiver outputs a value different from what its best guess is.

$$\textbf{EXPT}^{\langle S,R \rangle}(m)$$

1. $\boldsymbol{x} \xleftarrow{\$} S(m)$.

2. $\forall i \in [k]$, set $y_i = x_i$ with probability p and $y_i = 1 - x_i$ with probability $1 - p$.

3. Set $z := R(\boldsymbol{y})$.

4. Output $(m, \boldsymbol{x}, \boldsymbol{y}, z)$.

Figure 10: Execution of the $\langle S, R \rangle$ protocol

Properties about the experiment. Let A be the event that $z = m$ and similarly let B be the event that $z = 1 - m$. Then for the above experiment wee have:

Correctness: It implies that,
$$\Pr[A \cup B] \geq 1 - \varepsilon.$$
Let X' be a set such that $\forall \boldsymbol{x} \in X'$ we have that $\Pr[A \cup B | \boldsymbol{x}] \geq 1 - \sqrt{\varepsilon}$. By a counting argument (argument follows) it follows that $\Pr[\boldsymbol{x} \in X'] \geq 1 - \sqrt{\varepsilon}$. Lets start by assuming that $\Pr[\boldsymbol{x} \in X'] < 1 - \sqrt{\varepsilon}$. Then we have that $\Pr[A \cup B] < 1 \cdot (1 - \sqrt{\varepsilon}) + (1 - \sqrt{\varepsilon}) \cdot \sqrt{\varepsilon} = 1 - \varepsilon$, which is a contradiction.

Receiver privacy: Next given the set X', using receiver privacy we claim that there exists a set $X \subseteq X'$ such that for all $\boldsymbol{x} \in X$ we have
$$\Pr[A|\boldsymbol{x}] \geq p' - \sqrt{\varepsilon},$$
$$\Pr[B|\boldsymbol{x}] \geq 1 - p' - \sqrt{\varepsilon},$$
and
$$\Pr[X] \geq 1 - 3\sqrt{\varepsilon}.$$
The parameters in the above condition above are derived as follows. Lets assume that $\Pr[\boldsymbol{x} \in X] < 1 - 3\sqrt{\varepsilon}$, then we have that a cheating sender (with X hardcoded in it) can figure out whether A happened with (ignoring the case when $\boldsymbol{x} \notin X'$, that is probability $\sqrt{\varepsilon}$) probability $> (1 - 2\sqrt{\varepsilon}) \cdot p' + 2\sqrt{\varepsilon} \cdot (p' + \frac{\sqrt{\varepsilon}}{2}) = p' + \varepsilon$, which is a contradiction.

Towards Contradiction. Now we define a set Bad of vectors. Roughly speaking a vector $\boldsymbol{b} \in \mathsf{Bad}$ if

- $R(\boldsymbol{b}) \notin \{m, 1 - m\}$, or

- $\Pr[R(\boldsymbol{b}) = m | \boldsymbol{y} = \boldsymbol{b}] < p' - \sqrt{\varepsilon}$, or

- $\Pr[R(\boldsymbol{b}) = 1 - m | \boldsymbol{y} = \boldsymbol{b}] > 1 - p' + \sqrt{\varepsilon}$.

Observe that all we are left to argue is that that $\Pr[\mathsf{Bad}] > 3\sqrt{\varepsilon}$. Because if $\Pr[\mathsf{Bad}] > 3\sqrt{\varepsilon}$ then we can claim that at least one of the above three conditions happens with probability at least $\sqrt{\varepsilon}$ contradicting correctness, receiver guarantee and receiver guarantee respectively.

19

Probability of Bad. Lets restrict ourselves to the case in which $x \in X$. This happens with probability at least $1 - 3\sqrt{\varepsilon}$. Now consider any two vectors $\boldsymbol{y}$ and $\boldsymbol{y'}$ such that $R(\boldsymbol{y}) = m$ and $R(\boldsymbol{y'}) = \perp$. Now consider k hybrid-vectors $\boldsymbol{h}_0, \boldsymbol{h}_1 \ldots \boldsymbol{h}_k$ where $\boldsymbol{h}_i = (y'_1, y'_2 \ldots, y'_i, y_{i+1}, \ldots y_k)$. Note that $\boldsymbol{h}_0 = \boldsymbol{y}$ and $\boldsymbol{h}_k = \boldsymbol{y'}$. Note that $R(\boldsymbol{h}_0) \in m$ and $R(\boldsymbol{h}_k) = \perp$. We will show that at least one of these hybrid-vectors is in Bad.

If R on any one of these hybrid-vectors outputs a value not in $\{0, 1\}$ then that vector is clearly in Bad. On the other hand if R on each of these hybrid-vectors outputs a value in $\{m, 1\}$ then that implies a switch at some hybrid-vector. In other words $\exists j \in \{0, 1 \ldots k - 1\}$ such that $R(\boldsymbol{h}_j) = m$ and $R(\boldsymbol{h}_{j+1}) = \perp$. Now observe that the output of R was switched when just one of bits of the input to R was flipped. Next we will argue that R's best guess about the sent bit cannot have changed substantially with this one bit flip. In other words we will show that either $\boldsymbol{h}_j$ or $\boldsymbol{h}_{j+1}$ is in Bad.

If $\Pr[R(\boldsymbol{h}_j) = m | \boldsymbol{y} = \boldsymbol{h}_j] < p' - \sqrt{\varepsilon}$ then we are done. Otherwise we have that $\boldsymbol{h}_{j+1}$ differs from $\boldsymbol{h}_j$ in only one bit therefore we can conclude that the R's guess about the sent bit can decrease at most by a factor of $\frac{1-p}{p}$ (recall that $p \in (\frac{1}{2}, 1)$). In other words $\Pr[R(\boldsymbol{h}_j) = m | \boldsymbol{y} = \boldsymbol{h}_{j+1}] \geq (p' - \sqrt{\varepsilon}) \cdot \frac{1-p}{p} = \frac{p' - \sqrt{\varepsilon}}{p} - p' + \sqrt{\varepsilon} > 1 - p' + \sqrt{\varepsilon}$ when $p < p' - \sqrt{\varepsilon}$ where $(m, \boldsymbol{x}, \boldsymbol{y}, z) \leftarrow \mathsf{EXPT}^{\langle S,R \rangle}(m)$. However, $R(\boldsymbol{h}_{j+1}) = \perp$. In other words R outputs $1 - m$ even though it can guess the sent value correctly with a probability at least $1 - p' + \sqrt{\varepsilon}$. In summary we claim that for every two vectors $\boldsymbol{y}$ and $\boldsymbol{y'}$ such that $R(\boldsymbol{y}) = m$ and $R(\boldsymbol{y'}) = \perp$ there exists a hybrid vector (as defined above) that is in Bad.

Now we are left to argue that R receives the elements in Bad with noticeable probability. We argue this by considering a modified experiment $\mathsf{EXPT}'^{\langle S,R \rangle}(m)$. We start by noting that in this experiment

---

EXPT'$^{\langle S,R \rangle}(m)$

1. $\boldsymbol{x} \xleftarrow{\$} S(m)$.

2. $\forall i \in [k]$, set $y_i = x_i$ with probability p and $y_i = 1 - x_i$ with probability $1 - p$.

3. $\forall i \in [k]$, set $y'_i = x_i$ with probability p and $y'_i = 1 - x_i$ with probability $1 - p$.

4. Choose j randomly in $[k]$ and let $w_i = y_i$ for $i < j$ and $w_i = y'_i$ for $i \geq j$. Finally set $z := R(\boldsymbol{w})$.

5. Output $(m, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{y'}, \boldsymbol{w}, z)$.

Figure 11: Modified Execution of the $\langle S, R \rangle$ protocol

$\mathsf{EXPT}'^{\langle S,R \rangle}(m)$ the distribution of each of the outputs $\boldsymbol{y}, \boldsymbol{y'}, \boldsymbol{w}$ individually is identical to the distribution of the $\boldsymbol{y}$ in the output of $\mathsf{EXPT}^{\langle S,R \rangle}(m)$. This along with conditions prover earlier, implies that the probability $\Pr[R(\boldsymbol{y}) = m \bigwedge R(\boldsymbol{y'}) = 1 - m] \geq (1 - 3\sqrt{\varepsilon}) \cdot (1 - p' - \sqrt{\varepsilon}) \cdot (p' - \sqrt{\varepsilon})$ where $(m, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{y'}, \boldsymbol{w}, z) \leftarrow \mathsf{EXPT}'^{\langle S,R \rangle}(m)$.

Finally observe that in the experiment $\mathsf{EXPT}'^{\langle S,R \rangle}(m)$ given that $R(\boldsymbol{y}) = m$ and $R(\boldsymbol{y'}) = 1 - m$ we have that there exists a hybrid vector (as defined above) such that it is in Bad. Furthermore $\boldsymbol{w}$ takes this value with probability $\frac{1}{k}$. In other words $\boldsymbol{w} \in$ Bad with probability at least $(1 - 3\sqrt{\varepsilon}) \cdot (1 - p' - \sqrt{\varepsilon}) \cdot (p' - \sqrt{\varepsilon}) \cdot \frac{1}{k}$. Since the distribution of $\boldsymbol{w}$ in the output of $\mathsf{EXPT}'^{\langle S,R \rangle}(m)$ is identical to the distribution of $\boldsymbol{y}$ in the output of $\mathsf{EXPT}^{\langle S,R \rangle}(m)$, therefore we can conclude that $\boldsymbol{y}$ in the output of $\mathsf{EXPT}^{\langle S,R \rangle}(m)$ is in Bad with probability at least $(1 - 3\sqrt{\varepsilon}) \cdot (1 - p' - \sqrt{\varepsilon}) \cdot (p' - \sqrt{\varepsilon}) \cdot \frac{1}{k} \geq \frac{1}{2} \cdot \frac{1-p'}{2} \cdot \frac{p'}{2} \cdot \frac{1}{k} > 3\sqrt{\varepsilon}$ (when $\varepsilon \leq \frac{1}{36}, \left(\frac{1-p'}{2}\right)^2, \left(\frac{p'}{2}\right)^2$ and $\frac{(p'(1-p'))^2}{24^2 k}$). This is a contradiction when we set $\varepsilon < \min\left\{\left(\frac{1-p'}{2}\right)^2, \left(\frac{p'}{2}\right)^2, \frac{(p'(1-p'))^2}{24^2 k}\right\}$.

7   OWSC scheme for Deterministic Functionalities

$\mathsf{OWSC}^f/\mathcal{C}$ is a meaningful notion only for those deterministic functions f such that given a value y identifying if there exists an input x such that $y = f(x)$ is non-trivial (cannot be done in efficiently). This, in particular, rules out all functions with polynomial sized input domains. Furthermore, this notion is useful only in the setting of malicious adversaries because it is trivial to realize this notion in the setting of semi-honest adversaries.

We start by noting that a $\mathsf{OWSC}^f/\mathcal{C}$ scheme, for any deterministic function f, can be realized by using a $\mathsf{OWSC}^{\mathsf{zk}}/\mathcal{C}$ scheme for the zero-knowledge functionality. This can be achieved simply by having the sender send the output to the receiver and along with it prove in zero-knowledge, knowledge of an input x for which $f(x)$ yields the provided output. Here we implicitly assume that besides the channel $\mathcal{C}$ the sender also has access to an error free channel which can be implemented using $\mathcal{C}$ itself (with a negligible error). Formally,

Theorem 10 *For every deterministic function f, there exists a $\mathsf{OWSC}^f/\mathcal{C}$ scheme that is a UC-secure realization (even against malicious adversaries) of the functionality $\mathcal{F}_f$ in the $\mathcal{C}$-hybrid model where $\mathcal{C} \in \{\mathcal{C}_{GEC}^{k,\ell,\mathcal{D}}, \mathcal{C}_{BSC}^p\}$.*

As already mentioned, proving the above theorem reduces to the task of realizing a $\mathsf{OWSC}^{\mathsf{zk}}/\mathcal{C}$ scheme. In our construction, we will make use of oblivious ZK-PCPs (see Definition 1).

Lemma 3 *There exists a $\mathsf{OWSC}^{\mathsf{zk}}/\mathcal{C}$ scheme that is a UC-secure realization (even against malicious adversaries) of the zero-knowledge functionality in the $\mathcal{C}$-hybrid model where $\mathcal{C} \in \{\mathcal{C}_{GEC}^{k,\ell,\mathcal{D}}, \mathcal{C}_{BSC}^p\}$.*

We start by giving some intuition. The key idea is to use an erasure channel or a binary symmetric channel to send over multiple instances of independently chosen ZK-PCPs and observe the statistical gap that can be created only if valid proofs were sent. However, a number of difficulties arise in realizing this intuition, particularly in our construction from BSC. Below, we provide our construction from erasure channels. The more involved construction from binary symmetric channel is given next.

Erasure Channels.   We start by considering the case of binary erasure channels with error probability $\frac{1}{2}$; i.e., when $\mathcal{C} = \mathcal{C}_{BEC}^{\frac{1}{2}}$. It follows from Theorem 6 and Theorem 7 that any $\mathcal{C}_{GEC}^{k,\ell,\mathcal{D}}$ can be used to realize $\mathcal{C}_{BEC}^{\frac{1}{2}}$.[4] We give the protocol in Figure 12.

Completeness.   For every $i \in [k]$, using Chernoff bound, we have that:

$$\Pr\left[\Upsilon(\pi_i') \leq \frac{n}{4}\right] \leq e^{-\frac{n}{16}},$$

where $\Upsilon(\pi_i')$ denotes the number of occurrences of $\perp$ in π_i'.

Hence, except with negligible probability for each $i \in [k]$, R receives at least c. Given this the completeness of the protocol follows from the completeness of the oblivious ZK-PCP.

---

[4]Theorem 7 only guarantees a channel $\mathcal{C}_{BEC}^{p'}$ with p' close enough to p. We will use the value $\frac{1}{2}$ for concreteness but any value close enough to $\frac{1}{2}$, say in the range $\frac{1}{2}$ to $\frac{51}{100}$, will suffice as well.

$$\mathbf{OWSC}^{\mathsf{zk}}/\mathcal{C}^p_{BEC} \text{ protocol for language } L$$

Common Input: $x \in \{0,1\}^\lambda$.

Auxiliary Input for prover P: w such that $(x,w) \in R_L$.

Parameters: Let $(P_{\mathsf{oZK}}, V_{\mathsf{oZK}})$ be any (c,ν)-oblivious ZK-PCP system (with $c \leq \frac{n}{4}$ and $\nu \geq \frac{3}{4}$) with knowledge soundness κ. Let $\ell = \frac{\lambda}{\kappa}$.

- P samples proofs $\pi_1, \ldots, \pi_\ell$ from $P_{\mathsf{oZK}}(\lambda, x, w)$ and sends $(\pi_1, \ldots, \pi_\ell)$ to V via the erasure channel $\mathcal{C}^p_{BEC}$.

- V receives $\pi'_1, \ldots, \pi'_\ell$ and for all $i \in [\ell]$ checks if $V_{\mathsf{oZK}}(\pi'_i)$. It outputs **accept** if all the checks pass and **reject** otherwise.

Figure 12: Realizing zero-knowledge from Binary Erasure Channel

Soundness. We will construct an extractor E', that extracts valid witnesses from any cheating prover P^* that makes the honest verifier accept with non-negligible probability. We will first describe our extractor E' and then argue that it indeed works (with overwhelming probability).

Our extractor E' proceeds as follows. Let $(\pi_1, \pi_2, \ldots, \pi_\ell)$ be the proofs generated by the cheating prover P^*. For every $i \in [\ell]$, E' obtains $y_i = E(x, \pi_i)$. If $\exists i^* \in [\ell]$ such that $y_{i^*} \in R(x)$ then output y_{i^*} (breaking ties arbitrarily). If no such i^* exists then output $\perp$.

Note that since our extractor E' failed to extract witness out of π_i for any $i \in [\ell]$ we have (by soundness of the ZK-PCP) that $\Pr[V_{\mathsf{oZK}}(x, \pi'_i) = 0] \geq \kappa$, for every $i \in [\ell]$, where the probability is taken over the random choices of obtaining π'_i from π_i. Hence, if E' outputs $\perp$ then the verifier must also always reject, except with probability at most $\leq (1-\kappa)^\ell$, which is negligible for $\ell = \frac{\lambda}{\kappa}$.

Zero-Knowledge. We need to construct a simulator $\mathcal{S}'$ for our protocol. This construction follows immediately from the ν-zero-knowledge property of the oblivious ZK-PCP.

Proof for the BSC case. We now provide the proof for the more involved case of binary symmetric channel.

Binary Symmetric Channel. Next we consider the case of binary symmetric channel, i.e. when $\mathcal{C} = \mathcal{C}^p_{BSC}$. We give the protocol in Figure 13. We start by giving the intuition. The key idea is to send over multiple independently chosen ZK-PCPs over the BSC channel and observe the statistical gap that can be observed only if valid proofs were sent.

Notation, Observations and Intuition. We start by making some observations. We start by observing that we can use BSC to get a sort of a partial erasure channel. In particular if we repeat and send a bit 2τ times over BSC and the receiver takes majority of the received bits then this suffices as a sort of erroneous binary erasure channel. We will choose the parameter τ to be such that the probability that a sent value is correctly received is $\sum_{0 \leq i < \tau} \binom{2\tau}{i} p^{n-i} \cdot (1-p)^i \geq \sqrt[3]{\frac{2}{3}}$. We argue this by using Chernoff bound. Let X be the random variable denoting the number of bits among the 2τ bits transmitted that get flipped when each bit is flipped with a probability $1-p$. Now note that expected number of flips is $2(1-p)\tau$. Now by a Chernoff bound, $\Pr[X \geq \tau] = \Pr[X \geq 2(1-p)\tau(1 + \frac{(2p-1)}{2(1-p)})] \leq e^{-\frac{(2p-1)^2\tau}{6(1-p)}}$ (note that $p > \frac{1}{2}$). Hence the desired probability is at least $1 - e^{-\frac{(2p-1)^2\tau}{6(1-p)}} > \sqrt[3]{\frac{2}{3}}$ when $\tau = \left\lceil -\frac{6(1-p)}{(2p-1)^2} \ln\left(1 - \sqrt[3]{\frac{2}{3}}\right) \right\rceil$.

<div style="border:1px solid black; padding:10px;">

$$\mathbf{OWSC^{zk}}/\mathcal{C}_{BSC}^p \text{ protocol}$$

Common Input: $x \in \{0,1\}^\lambda$.

Auxiliary Input for P: w such that $(x,w) \in R_L$.

Parameters: Let $(P_{\mathsf{oZK}}, V_{\mathsf{oZK}})$ be any $(3,\alpha)$-oblivious ZK-PCP system with knowledge soundness κ. Let $\ell = \left(\frac{\lambda \cdot n}{\kappa}\right)^2$, $\tau = \left\lceil -\frac{6(1-p)}{(2p-1)^2} \ln\left(1 - \sqrt[3]{\frac{2}{3}}\right) \right\rceil$ (n is the length of the ZK-PCP proof) and $\alpha = \binom{2\tau}{\tau}(p(1-p))^\tau$. Let U and S be the sets of all possible configurations/accepting states as explained in the text. Let $p_{i,j}$ be the probability that a value i goes to a value j where $i,j \in U$. Let γ_j for $j \in S$ be constants such that $1 = \sum_{i,j \in S} p_{i,j}\gamma_j$. Also $\Upsilon(\cdot)$ outputs the number of occurrences of $\perp$ in the input string as explained in the text.

- P samples proofs $\pi_1, \ldots \pi_\ell$ from $P_{\mathsf{oZK}}(\lambda, x, w)$ and sends $(\pi_1, \ldots \pi_\ell)$ to V via $\mathcal{C}_{BSC}^p$ repeating each bit 2τ times.

- V receives $\pi'_1, \ldots \pi'_\ell$ for all $i \in [\ell]$ (where each bit is obtained by taking majority over the 2τ copies of the a sent bit and using $\perp$ for a tie). If $\Upsilon(\pi'_1, \ldots \pi'_\ell) \geq \alpha\ell \cdot n + \sqrt{\alpha\ell n\lambda}$ then abort everything and output reject.

  Otherwise, for each $i \in [\ell]$ consider 3 random bits in π'_i and check if $V_{\mathsf{oZK}}(\lambda, x, \pi'_i) = 1$. Set $t_i = \gamma_j$ if this is the case and the received configuration is j. Output accept if $\sum_i t_i \geq \ell - \sqrt{\ell\lambda}$ and reject otherwise.

</div>

Figure 13: Realizing zero-knowledge from Binary Symmetric Channel

Next note that the PCP verifier looks at a tuple of 3 bits and decides whether to accept or reject. This allows for 8 combinations out of which only 4 are acceptable. Let U be the set of all state configurations and let S be the set of possible configurations that are accepting. Let $p_{i,j}$ denote the probability of going from state i to j such that both $i,j \in U$. Furthermore let γ_j for each $j \in S$ be constants such that $1 = \sum_{i,j \in S} p_{i,j}\gamma_j$. Since we have that for each $i \in S$, $p_{i,i} > \frac{2}{3}$ (obtained as $\left(\sqrt[3]{\frac{2}{3}}\right)^3$) we can conclude that for every $i \in S$, $\gamma_i < \frac{3}{2}$. Depending on the PCP we could have a different accepting configurations for each gate and hence the constant γ_is could also be defined per gate. We will give the construction assuming only one kind of accepting configuration. It extends to the general setting in a natural way.

Intuitively in our construction γ_i's help achieve normalization. In particular consider the case in which the sender sends an accepting configuration γ_i for $i \in S$ and the receiver obtains γ_j for some $j \in U$. In our construction if $j \in S$ then the receiver will output γ_j and if $j \notin S$ then it will just output 0. This has two affects. First if $i \in S$ then the expected value the receiver outputs is 1 regardless of what the sent value was. On the other hand if $i \notin S$ then the expected value the receiver outputs is $< 1/2$. This follows from the fact that if $i \notin S$ then the received value is in S with probability less that $\frac{1}{3}$ and each γ_j is less that $\frac{3}{2}$.

We give the protocol in Figure 13. Let $\Upsilon(\cdot)$ be a function that takes a string over alphabets $\{0,1,\perp\}$ as input and outputs the number of occurrences of $\perp$ in the string.

Completeness. Using Chernoff bound we have that:

$$\Pr\left[\Upsilon(\pi'_1 || \ldots \pi'_\ell) \geq \alpha\ell n \left(1 + \frac{\lambda^{1/2}}{\sqrt{\alpha\ell n}}\right)\right] \leq e^{-\frac{\lambda}{3}}$$

$$\Pr\left[\sum_i t_i \leq \ell \left(1 - \frac{\lambda^{1/2}}{\sqrt{\ell}}\right)\right] \leq e^{-\frac{\lambda}{2}}$$

Given these two facts the completeness of the protocol follows directly.

Soundness. Let α' be the least probability that a bit is erased when the input is not all 0 or all 1. In particular $\alpha' = \binom{2\tau}{\tau-1} p^{\tau+1}(1-p)^{\tau-1}$. Note that $\alpha' > \alpha$. Furthermore lets consider the setting in which the adversary sends more that $c_1\sqrt{\alpha\ell n\lambda}$ symbols in a way that they are not honestly generated. We will argue that in this case the adversary almost always gets caught. The expected number of $\perp$s that will be received in this case are $> \alpha\ell n + c_1(\alpha' - \alpha)\sqrt{\alpha\ell n\lambda}$. Then by a Chenroff bound we (set $c_1 = \frac{c_2}{(\alpha-\alpha')}$) have that:

$$\Pr\left[\Upsilon(\pi_1'||\dots\pi_\ell') \le \alpha\ell n\left(1 + \frac{\lambda^{1/2}}{\sqrt{\alpha\ell n}}\right)\right] = \Pr\left[\Upsilon(\pi_1'||\dots\pi_\ell') \le (\alpha\ell n + c_2\sqrt{\alpha\ell n\lambda})(1 - \frac{(c_2-1)\sqrt{\lambda}}{(\sqrt{\alpha\ell n} + c_2\sqrt{\lambda})})\right]$$

$$\le e^{-\frac{(c_2-1)^2\lambda}{2(\sqrt{\alpha\ell n}+c_2\sqrt{\lambda})^2}\cdot(\alpha\ell n + c_2\sqrt{\alpha\ell n\lambda})}$$

$$= e^{-\frac{(c_2-1)^2\lambda\sqrt{\alpha\ell n}}{2(\sqrt{\alpha\ell n}+c_2\sqrt{\lambda})}}$$

$$< e^{-\frac{(c_2-1)^2\sqrt{\lambda}}{2c_2}}$$

$$(3)$$

Hence, we have that at most $c_2\sqrt{\alpha\ell n\lambda}$ bits were tampered with. This implies that for at least $\ell - c_2\sqrt{\alpha\ell n\lambda} = \ell - c_2\ell^{3/4} > \frac{\ell}{2}$ (as $\ell^{1/4} = \omega(\sqrt{n\lambda})$) of the proofs we have that none of the bits are tampered with. In this case the expected value of $\sum_i t_i$ is at most $\ell - \frac{\kappa\ell}{4}$ (κ of the $\ell/2$ proof are such that the t_i values for them is less than $1/2$). We will now argue that in this case the value $\sum_i t_i$ will be more than $\ell - \sqrt{\ell\lambda}$ only with a negligible probability.

$$\Pr[\sum_i t_i \ge \ell - \sqrt{\ell\lambda} = \ell(1 - \frac{\kappa}{4})(1 + \frac{\frac{\kappa\ell}{4} - \sqrt{\ell\lambda}}{\ell(1 - \frac{\kappa}{4})})] \le e^{-\frac{(\frac{\kappa\ell}{4} - \sqrt{\ell\lambda})^2}{\ell(1-\frac{\kappa}{4})}} \le e^{-\frac{\kappa^2\ell}{64}},$$

which is negligible for our choice of parameters.

We have just argued that if the prover is not caught then it must be the case that a large number of the proofs sent by the prover are without inconsistency. Hence they can in fact be used to extract the witness. The extraction procedure and the argument that it works is identical to the argument for the binary erasure case.

Zero-Knowledge. We need to construct a simulator $\mathcal{S}'$ for our protocol. This construction follows immediately from the the α-zero-knowledge property of the oblivious ZK-PCP.

8 $\mathcal{C}_{ROT}^\ell$ is **OWSC** complete for randomized functionalities

In this section, we describe an OWSC scheme for any randomized function in the $\mathcal{C}_{ROT}$-hybrid model that uses only a *single* round of random OTs and no additional interaction. The functionalities considered here provide output to only one party. This result follows directly from [IPS08, Appendix B] and we include the construction and proof in Appendix A for completeness (much of the text have been taken verbatim from [IPS08, Appendix B]). More efficient alternatives have been considered by [IKO+11a] however we consider the simplest feasibility result for our setting.

One technical difference in our setting compared to [IPS08] is in the underlying primitive from which the protocols are constructed. While the protocol in [IPS08] uses a regular 1-out-of-N OT protocol, in our case we only have access to a 1-out-of-2 ROT protocol and need to convert it to a 1-out-of-N ROT protocol. (Recall that the choice about which 1-out-of-N strings the receiver obtains is made by the channel in the

ROT protocol.) This however can be done easily using standard techniques and a sketch of the construction has been provided in Appendix A.3.

Theorem 11 *For every randomized function f, $\exists \ell$ and a $\mathsf{OWSC}^f / \mathcal{C}_{ROT}^\ell$ scheme that is a UC-secure realization (even against malicious adversaries) of the functionality $\mathcal{F}_f$ in the $\mathcal{C}_{ROT}^\ell$-hybrid model.*

ϵ-**secure variant.** We can also use the ϵ-UC realization of ROT (based on noisy bursty channel as in Theorem 5) in order to obtain a $\epsilon \cdot r$-UC realization of OWSC^f where r is the number of ROT calls made inside our construction. r for our construction is a fixed polynomial in the security parameter λ, independent of the size of the function being computed.

Construction using biased-ROT. The above theorem is stated just for the case of $\mathcal{C}_{ROT}^\ell$-hybrid model. However we note that the same construction continues to work in the $\mathcal{C}_{ROT}^{\ell,p}$-hybrid model, for any constant $p \in (0, 1)$, with one small change. When using the $\mathcal{C}_{ROT}^{\ell,p}$ channel, the input provided by the channel for the function evaluation will be biased. This issue can be resolved by using security parameter λ number of independent bits from the channel to obtain each bit for the functionality being evaluated. More specifically, each input bit for the functionality is obtained by taking the exclusive or of λ independent input bits. By the XOR Lemma, we claim that the obtained bits will be close to uniform.

Furthermore, when using the $\mathcal{C}_{ROT}^{\ell,p}$-hybrid model, the construction itself does not depend on the precise value of the constant p. Hence, our construction is robust in the sense that it remains secure even if the adversary gets to specify the value of p (within some bounded range).

9 Extending impossibilities for constructing ROT from BEC and BSC to the Computational Setting

In this section we will show that even with computation assumptions it is impossible to construct ROT using BEC and BSC.

9.1 Impossibility of ROT from BEC

Theorem 12 $\forall k, \ell, p$ (k, ℓ are poly in λ and p is a constant), and any negligible in λ function ε, $\mathcal{C}_{ROT}^1$ cannot be ε-securely realized in the $\mathcal{C}_{GEC}^{k,\ell,\mathcal{D}_{k,p}}$ hybrid model even against semi-honest computationally bounded adversaries.

PROOF: For the sake of contradiction lets start by fixing a negligible ε and assuming that there exists a protocol $\pi = \langle S, R \rangle$ that ε-securely realizes $\mathcal{C}_{ROT}^1$ in the $\mathcal{C}_{GEC}^{k,\ell,\mathcal{D}_{k,p}}$ hybrid model. More specifically π proceeds as: S on input two bits $m_0, m_1 \in \{0, 1\}$ generates k strings $\boldsymbol{x} = (x_1, x_2 \ldots x_k)$ which are provided as input to the functionality $\mathcal{C}_{GEC}^{k,\ell,\mathcal{D}_{k,p}}$. The functionality then outputs strings $\boldsymbol{y} = (y_1, y_2 \ldots y_k)$ to the receiver R. R processes these values and outputs either $(m_0, \bot)$ or $(\bot, m_1)$. More formally, consider the experiment $\mathsf{EXPT}^{\langle S,R \rangle}(m_0, m_1)$ in Figure 14.

Let A be the event that $z = (m_0, \bot)$ and similarly let B be the event that $z = (\bot, m_1)$. For the sake of contradiction lets assume that the protocol is indeed secure. Then correctness implies that except with negligible probability either A or B happens. Also receiver privacy implies that with overwhelming probability over choices of $\boldsymbol{x}$ sent by the sender we have that A and B happen with roughly the same probability. To reach a contradiction we will construct a machine M that can output both m_0, m_1 correctly using just $\boldsymbol{y}$ with probability noticeably better than $\frac{1}{2}$.

<div style="border:1px solid black">

EXPT$^{\langle S,R \rangle}(m_0, m_1)$

1. $\boldsymbol{x} \xleftarrow{\$} S(m_0, m_1)$.

2. $\forall i \in [k]$, set $y_i = x_i$ with probability p and $\bot$ with probability $1 - p$. Set $s_i = 1$ if $y_i = x_i$ and 0 otherwise.

3. Set $z := R(\boldsymbol{y})$.

4. Output $(m_0, m_1, \boldsymbol{x}, \boldsymbol{y}, s, z)$.

</div>

Figure 14: Execution of the $\langle S, R \rangle$ protocol

Our machine M proceeds as follows. M proceeds by following the honest receiver strategy to obtain one of the values m_0 or m_1. Let S denote the set of symbols (the set corresponding to the string s) received by the receiver, and m_b be the ROT message output by the receiver. The receiver tries to recover m_{1-b} by considering a random subset of S of size $|S| - 1$ and for that subset $S' \subset S$ it runs the honest receiver. If the receiver guess for S' makes receiver output m_{1-b}, then M uses this values and otherwise setting it to a random value in $\{0, 1\}$. M finally outputs (m_0, m_1).

Now we need to argue that M outputs the correct values (m_0, m_1) with a probability noticeably better than $\frac{1}{2}$. In particular observe that it suffices to show that (1) with noticeable probability there exists a set S' such that the honest receiver outputs m_{1-b}, (2) the output value m_{1-b} is correct. We will argue these two properties using the following lemma (the lemma itself is proved later).

Lemma 4 *For any balanced function $f : \{0,1\}^n \to \{0,1\}$ (i.e. $\Pr_x[f(x) = 1] = \Pr[f(x) = 0] = \frac{1}{2}$), $\Pr_x[\exists\, x' \mid \delta(x, x') = 1 \,\wedge\, f(x') = 1 - f(x)] \geq \Omega(1/\sqrt{n})$, where $\delta(x, x')$ denotes the hamming distance between x and x'. (The distribution on x is as follows. Each bit of x is set to 1 with probability p and 0 otherwise.)*

Argument that an S' exists. Note that with overwhelming probability over the choice of $\boldsymbol{x}$ we have that the A and B happen with roughly the same probability. Let $f_{\boldsymbol{x}}$ be the function that on input $s \in \{0,1\}^k$ outputs 1 if deletion of entries in $\boldsymbol{x}$ based on s leads to event A and 0 otherwise. Note that this function $f_{\boldsymbol{x}}$ is balanced. Hence we have that with probability at least $\Omega(1/\sqrt{k})$ an s' such that $\delta(s, s') = 1$ exists.

Argument that the value is correct. We will now argue that the value m_{1-b} output by M is correct except with negligible probability. For the sake of contradiction lets start by assuming that this is not the case. In that scenario, we will show that the scheme has non-negligible correctness error. Consider the following two experiments:

1. Pick a random $s \in \{0,1\}^k$ where each bit is set to 1 with probability p and 0 otherwise.

2. Pick a random $s \in \{0,1\}^k$ as above and then replace a random bit in s that is 1 to 0.

Then the probability of each s happening in the two experiments are polynomially related. This implies that if the probability of getting an incorrect answer in the second experiment is non-negligible then the same must be the case in the first experiment. This contradicts the correctness of our scheme. □

Proof of Lemma 5. Lets start by arguing for the case when the distribution of x is uniform, i.e. $p = \frac{1}{2}$. Let S be a set such that $x \in S$ if we have that $f(x) = 0$. Then by the isoperimetric inequality (or Harper's Theorem) we have that among all sets S of a given size s, the one that has the smallest boundary (namely, the smallest number of points that have neighbors outside the set) is the Hamming ball. In particular, if S contains half of the points in $0, 1^n$, then its boundary is of size at least $\binom{n}{n/2}$ which is at least a $\frac{1}{\Omega(\sqrt{n})}$ fraction of all the points. This implies the claim.

Next we generalize the argument to the setting where p is an arbitrary fixed constant in $(0, 1)$. In particular, we can define $Q_n = \{0, 1\}^n$ (the n-cube) and identify each element $x \in Q_n$ with the corresponding subset of $[n]$; i.e., $\{i \mid x_i = 1\}$. Then the probability measure $\Pr$ on Q_n is:

$$\Pr(x) = \prod_{i \in x} p \prod_{i \notin a} (1 - p) .$$

This gives a weighted cube and we can use a weighted version of the Harper's Theorem and get an argument just like for the case of $p = 1/2$ above. We will argue for the case when $p > 1/2$. The other case is symmetric. In particular, as pointed out in [Liu, Pg. 18] we have that even for this weighted cube, the set S of a fixed weight with the smallest boundary is a Hamming Ball. Let S be a set such that $x \in S$ if we have that $f(x) = 0$. Then the set S with weight $1/2$ and the smallest boundary is the Hamming Ball of radius at least cn for some constant c. This follows from the fact that for any radius $o(pn)$ we have that using Hoeffding's inequality the weight of the Hamming ball is negligible [Wik13]. Therefore the boundary of S is of size at least $\binom{n}{cn}$ and these points are sampled with a probability at least $\frac{1}{\Omega(\sqrt{n})}$.

9.2 Impossibility of ROT from BSC

Theorem 13 $\forall k, \ell, p$ (*k, ℓ are poly in λ and p is a constant), and any negligible in λ function ε, $\mathcal{C}^1_{ROT}$ cannot be ε-securely realized in the $\mathcal{C}^p_{BSC}$ hybrid model even against semi-honest computationally bounded adversaries.*

PROOF: The proof is analogous to the proof of Theorem 12 and we just sketch the differences. Note BSC channel with correct transmission probability p can be thought of as a channel that transmits the bit correctly with a probability p' and replaces it with a random value with probability $1 - p'$ for an appropriate p'. With this setting in mind similar to proof of Theorem 12, we can also construct a machine M that outputs both m_0, m_1 as follows.

Let S denote the set of symbols correctly transmitted by the channel to the receiver, and m_b be the ROT message output by the receiver. Note that the receiver is unaware of the set S itself. The receiver tries to recover m_{1-b} by considering a uniform position of the received string, inverting it and evaluating the honest receiver on the obtained string. If the receiver obtains an output m_{1-b}, then M uses this values and otherwise sets it to a random value in $\{0, 1\}$. M finally outputs (m_0, m_1).

Just like in the proof of Theorem 12 we need to show that this procedure indeed finds the correct value m_{1-b} with noticeable probability. The argument for this is identical to the argument presented in the proof of Theorem 12. $\square$

References

[Ajt10]   Miklós Ajtai. Oblivious RAMs without cryptogrpahic assumptions. In Leonard J. Schulman, editor, *42nd Annual ACM Symposium on Theory of Computing*, pages 181–190, Cambridge, Massachusetts, USA, June 5–8, 2010. ACM Press.

[BBCM95] Charles H. Bennett, Gilles Brassard, Claude Crepeau, and Ueli M. Maurer. Generalized privacy amplification. *Information Theory, IEEE Transactions on*, 41(6):1915 –1923, Nov 1995.

[BBR88] Charles H. Bennett, Gilles Brassard, and Jean-Marc Robert. Privacy amplification by public discussion. *SIAM Journal on Computing*, 17(2):210–229, 1988.

[BCR86] Gilles Brassard, Claude Crépeau, and Jean-Marc Robert. Information theoretic reductions among disclosure problems. In *FOCS*, pages 168–173, 1986.

[BFM90] Manuel Blum, Paul Feldman, and Silvio Micali. Proving security against chosen cyphertext attacks. In Shafi Goldwasser, editor, *Advances in Cryptology – CRYPTO'88*, volume 403 of *Lecture Notes in Computer Science*, pages 256–268, Santa Barbara, CA, USA, August 21–25, 1990. Springer, Berlin, Germany.

[BGW88] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proc. 20th STOC*, pages 1–10. ACM, 1988.

[BP04] Boaz Barak and Rafael Pass. On the possibility of one-message weak zero-knowledge. In Moni Naor, editor, *TCC 2004: 1st Theory of Cryptography Conference*, volume 2951 of *Lecture Notes in Computer Science*, pages 121–132, Cambridge, MA, USA, February 19–21, 2004. Springer, Berlin, Germany.

[BTV12] Mihir Bellare, Stefano Tessaro, and Alexander Vardy. Semantic security for the wiretap channel. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 294–311, Santa Barbara, CA, USA, August 19–23, 2012. Springer, Berlin, Germany.

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. Electronic Colloquium on Computational Complexity (ECCC) TR01-016, 2001. Previous version "A unified framework for analyzing security of protocols" availabe at the ECCC archive TR01-016. Extended abstract in FOCS 2001.

[Can05] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. Cryptology ePrint Archive, Report 2000/067, 2005. Revised version of [Can01].

[CK88] Claude Crépeau and Joe Kilian. Achieving oblivious transfer using weakened security assumptions (extended abstract). In *FOCS*, pages 42–52, 1988.

[CMW04] Claude Crépeau, Kirill Morozov, and Stefan Wolf. Efficient unconditional oblivious transfer from almost any noisy channel. In Carlo Blundo and Stelvio Cimato, editors, *SCN 04: 4th International Conference on Security in Communication Networks*, volume 3352 of *Lecture Notes in Computer Science*, pages 47–59, Amalfi, Italy, September 8–10, 2004. Springer, Berlin, Germany.

[DFMS04] Ivan Damgård, Serge Fehr, Kirill Morozov, and Louis Salvail. Unfair noisy channels and oblivious transfer. In Moni Naor, editor, *TCC 2004: 1st Theory of Cryptography Conference*, volume 2951 of *Lecture Notes in Computer Science*, pages 355–373, Cambridge, MA, USA, February 19–21, 2004. Springer, Berlin, Germany.

[DKS99] Ivan Damgård, Joe Kilian, and Louis Salvail. On the (im)possibility of basing oblivious transfer and bit commitment on weakened security assumptions. In Jacques Stern, editor, *Advances in*

Cryptology – EUROCRYPT'99, volume 1592 of *Lecture Notes in Computer Science*, pages 56–73, Prague, Czech Republic, May 2–6, 1999. Springer, Berlin, Germany.

[FLS99]   Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple noninteractive zero knowledge proofs under general assumptions. *SIAM J. Comput.*, 29(1):1–28, 1999.

[GMW87]   Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play ANY mental game. In ACM, editor, *Proc. 19th STOC*, pages 218–229. ACM, 1987. See [Gol04, Chap. 7] for more details.

[Gol04]   Oded Goldreich. *Foundations of Cryptography: Basic Applications*. Cambridge University Press, 2004.

[Har60]   Theodore E. Harris. A lower bound for the critical probability in a certain percolation process. *Proc. Cambridge Phil. Soc.*, 56:13–20, 1960.

[IKO+11a] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, Manoj Prabhakaran, and Amit Sahai. Efficient non-interactive secure computation. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 406–425, Tallinn, Estonia, May 15–19, 2011. Springer, Berlin, Germany.

[IKO+11b] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, Manoj Prabhakaran, Amit Sahai, and Jürg Wullschleger. Constant-rate oblivious transfer from noisy channels. In *CRYPTO*, pages 667–684, 2011.

[IPS08]   Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. Founding cryptography on oblivious transfer - efficiently. In *CRYPTO*, pages 572–591, 2008.

[ISW03]   Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 463–481, Santa Barbara, CA, USA, August 17–21, 2003. Springer, Berlin, Germany.

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

[Kle66]   Daniel J. Kleitman. Families of non-disjoint subsets. *J. Combin. Theory*, 1:153–155, 1966.

[KM01]   Valeri Korjik and Kirill Morozov. Generalized oblivious transfer protocols based on noisy channels. In *MMM-ACNS*, pages 219–229, 2001.

[Liu]   Henry Liu. M400 msci project - discrete isoperimetric inequalities.

[Mau91]   Ueli M. Maurer. Perfect cryptographic security from partially independent channels. In *STOC*, pages 561–571, 1991.

[Mau02]   Ueli M. Maurer. Secure multi-party computation made simple. In Stelvio Cimato, Clemente Galdi, and Giuseppe Persiano, editors, *SCN*, volume 2576 of *Lecture Notes in Computer Science*, pages 14–28. Springer, 2002.

[Pas03]   Rafael Pass. On deniability in the common reference string and random oracle model. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 316–337, Santa Barbara, CA, USA, August 17–21, 2003. Springer, Berlin, Germany.

[RTWW11] Samuel Ranellucci, Alain Tapp, Severin Winkler, and Jürg Wullschleger. On the efficiency of bit commitment reductions. In *ASIACRYPT*, pages 520–537, 2011.

[SW02] Douglas Stebila and Stefan Wolf. Efficient oblivious transfer from any non-trivial binary-symmetric channel. In *Information Theory, 2002. Proceedings. 2002 IEEE International Symposium on*, page 293, 2002.

[Wik13] Wikipedia. Binomial distribution, 2013. [Online; accessed 17-Oct-2013].

[WNI03] Andreas Winter, Anderson C. A. Nascimento, and Hideki Imai. Commitment capacity of discrete memoryless channels. In *In: Cryptography and Coding. LNCS*, pages 35–51. Springer-Verlag, 2003.

[Wul07] Jürg Wullschleger. Oblivious-transfer amplification. In Moni Naor, editor, *Advances in Cryptology – EUROCRYPT 2007*, volume 4515 of *Lecture Notes in Computer Science*, pages 555–572, Barcelona, Spain, May 20–24, 2007. Springer, Berlin, Germany.

[Wul09] Jürg Wullschleger. Oblivious transfer from weak noisy channels. In Omer Reingold, editor, *TCC 2009: 6th Theory of Cryptography Conference*, volume 5444 of *Lecture Notes in Computer Science*, pages 332–349. Springer, Berlin, Germany, March 15–17, 2009.

[Wyn75] Aaron D. Wyner. The wire-tap channel. *Bell Syst. Tech. J.*, 54(8):1334–1387, 1975.

A   Proof of Theorem 11

First, we shall describe a protocol that achieves security against covert-adversaries, but with a deterrence probability that can be easily reduced to negligible by choosing parameters appropriately. We describe the protocol in two parts. First we give a scheme $\mathsf{OWSC}^f/\mathcal{F}_{\mathsf{cov\text{-}cROT}}$ scheme that is a UC-secure realization of the functionality $\mathcal{F}_f$ in the $\mathcal{F}_{\mathsf{cov\text{-}cROT}}$-hybrid model. Subsequently we will give a scheme for securely realizing $\mathcal{F}_{\mathsf{cov\text{-}cROT}}$ in the in the $\mathcal{C}_{ROT}^\ell$-hybrid model for appropriate choice of ℓ. This is very similar to [IPS08] except that our protocol is in the ROT hybrid model as opposed to the protocol of [IPS08] which was in the OT hybrid model.

A.1   Reducing Covert-adversary OWSC to Covert-adversary Certified-ROT

We start by defining the functionality $\mathcal{F}_{\mathsf{cov\text{-}cROT}}$, for "covert-adversary certified-ROT."

Covert-adversary Certified-ROT Functionality.   Parameters of $\mathcal{F}_{\mathsf{cov\text{-}cROT}}$ include a function C, the number of pairs of strings that are being transferred, m, the length of these strings, and a "deterrence probability" ϵ.

1. $\mathcal{F}_{\mathsf{cov\text{-}cROT}}$ takes from the sender input $\Gamma = ((s_0^1, s_1^1), \ldots, (s_0^m, s_1^m); w)$ that is m pairs of strings and a "witness" w. The receiver has no input and hence the functionality does not take any input from the receiver .

2. If the sender is corrupt, it allows the sender to send a command cheat also. In this case, with probability ϵ, $\mathcal{F}_{\mathsf{cov\text{-}cROT}}$ will produce the message corrupted as output to both the parties and terminates, and with probability $1 - \epsilon$, will allow the sender to specify an output for the receiver.

3. If the sender does not include the cheat command in the input, then the receiver gets $(s_{c_1}^1, \ldots, s_{c_m}^m; C(\Gamma))$ where $\forall i \in [m], c_i \xleftarrow{\$} \{0, 1\}$.

30

Given a decomposable randomized encoding h for a function g, it is fairly straight forward to use a simple generalization of Yao's protocol to get a OWSC scheme for g in the certified-ROT-hybrid model. In the $\mathcal{F}_{\text{cov-cROT}}$-hybrid model, this protocol is a secure realization of covert-adversary OWSC of g.

Let g's input consist of two parts: A's input a and the random input (provided by the channel in our case) b. Recall that a decomposable randomized encoding of g can be written as $h(x, r) = \{h_i(x_i, r)\}_{i=1}^{|x|}$, where x is the input to g. Since $x = (a, b)$, we will rewrite this as $h(a, b, r) = \{h_i^A(a_i, r)\}_{i=1}^{|a|} \circ \{h_i^B(b_i, r)\}_{i=1}^{|b|}$.

The function C associated with $\mathcal{F}_{\text{cov-cROT}}$ is defined as

$$C(\{(s_0^i, s_1^i)\}_{i=1}^{|b|}; (a, r)) = (\{h_i^A(a_i, r)\}_{i=1}^{|a|}; R(\{(s_0^i, s_1^i)\}_{i=1}^{|b|}, r)),$$

where R is a predicate which checks that $\{(s_0^i, s_1^i)\}_{i=1}^{|b|} = \{(h_i^B(0, r), h_i^B(1, r))\}_{i=1}^{|b|}$.

1. A picks a random string r (for the randomized encoding of g) and prepares the following input for $\mathcal{F}_{\text{cov-cROT}}$ (with the associated function C described above): $(\{(h_i^B(0, r), h_i^B(1, r))\}_{i=1}^{|b|}; (a, r))$.

2. B obtains $(\{h_i^B(b_i, r)\}_{i=1}^{|b|}; \{h_i^A(a_i, r)\}_{i=1}^{|a|}; z)$ where for each i, b_i is a random bit chosen by the channel. B aborts if $z \neq 1$.

3. B computes $g(a, b)$ from $h(a, b, r) = \{h_i^A(a_i, r)\}_{i=1}^{|a|} \circ \{h_i^B(b_i, r)\}_{i=1}^{|b|}$, and outputs it.

Proof of Security. If A is corrupt, the simulation is straight forward: the simulator obtains her inputs to $\mathcal{F}_{\text{cov-cROT}}$, computes C and checks if the predicate R evaluates to 1 on this input. If so, it sends a to $\mathcal{F}_{\text{cov-}g}$. If A sends a cheat command to $\mathcal{F}_{\text{cov-cROT}}$, then the simulator also sends a cheat command to $\mathcal{F}_{\text{cov-}g}$. If $\mathcal{F}_{\text{cov-}g}$ responds, then, A sets the outputs of $\mathcal{F}_{\text{cov-cROT}}$, which are received by the simulator, who uses it to carry out the rest of the protocol of B; the simulator will then instruct $\mathcal{F}_{\text{cov-}g}$ to output whatever this simulated B outputs. It is easily seen that this is a perfect simulation if $\mathcal{F}_{\text{cov-}g}$ has the same deterrence probability as $\mathcal{F}_{\text{cov-cROT}}$.

If B is corrupt, then also there is a simple simulation, which depends on the privacy property of the randomized encoding. The simulator obtains the output for B from $\mathcal{F}_{\text{cov-}g}$. It then constructs a random encoding consistent with this output value. This is used to prepare a simulated output from $\mathcal{F}_{\text{cov-cROT}}$. (Note that $\mathcal{F}_{\text{cov-cROT}}$ does not allow B to send a cheat message.)

A.2 Reducing Covert-adversary Certified-ROT to ROT

In this section we give a OWSC protocol $\mathsf{cROT}^{\mathsf{ROT}}$ in the ROT-hybrid model, which achieves the "covert-adversary" Certified-ROT functionality $\mathcal{F}_{\text{cov-cROT}}$. Our protocol is built by compiling an MPC protocol, η (involving more than two parties, with a certain level of information theoretic security against passive corruption) into a two-party protocol in the ROT-hybrid model.

Protocol η. First we describe the requisite properties of the protocol η.

- **Participants:** There are 2 *input clients*, q *servers* and $2Lm + 1$ *output clients* for some $L > 1$. (Here q will be a constant. m is the number of pairs of strings that the sender wants to send in the certified ROT functionality provided by the compiled protocol. L can be set to 2 to get a deterrence value of $1/2$, or if a higher deterrence value is needed, a higher constant.) We denote the 2 input clients by I_0 and I_1; we denote the $2Lm + 1$ output clients by $Z_{\ell 0}^i$ and $Z_{\ell 1}^i$, (for $i = 1, \ldots, m$ and $\ell = 1, \ldots, L$) and Z^0.

- **Functionality:** We define the following functionality $\mathcal{H}$. Let x_0 and x_1 denote the inputs of I_0 and I_1. $\mathcal{H}$ parses $x_0 \oplus x_1$ as m pairs of strings $((s_0^1, s_1^1), \ldots, (s_0^m, s_1^m))$ and a "witness" w.

  Z^0 is given the function $C((s_0^1, s_1^1), \ldots, (s_0^m, s_1^m); w)$. For each i, $Z_{\ell 0}^i$ and $Z_{\ell 1}^i$ ($\ell = 1, \ldots, L$) receive random strings $z_{\ell 0}^i$ and $z_{\ell 1}^i$ subject to the constraint that $\bigoplus_\ell z_{\ell r_\ell}^i = s_{\bigoplus_\ell r_\ell}^i$ for all $r \in \{0, 1\}^L$. That is, $z_{\ell b}^i$ are random such that $\bigoplus_\ell z_{\ell 0}^i = s_0^i$ and $z_{\ell 1}^i = z_{\ell 0}^i \oplus s_0^i \oplus s_1^i$.

- **Security:** η must be t_0-private, for some $t_0 \geq 2$. More precisely, it securely realizes the functionality $\mathcal{H}$ against passive (honest-but-curious), adversaries who can corrupt up to t_0 servers, and any number of input and output clients. The security is perfect.

- **Structure of the protocol:** We will require that the input clients talk only to the servers and that output clients only receive messages and never send messages.

- **Complexity:** We require the communication complexity of the protocol to be linear in the circuit size of C.

Standard MPC protocols from the literature can be easily adapted to obtain a protocol η that fits the above requirements. So, for instance, let $q = 8$ and use the BGW protocol [BGW88] (or a simpler protocol due to Maurer [Mau02]). Note that [IPS08] allowed the protocol to be in the OT-hybrid model and could use the GMW protocol [GMW87] as well. However we only have access to ROT and hence we cannot use that. Furthermore we set $q = 8$ so that it is a power of 2([IPS08] set q as 5).

Protocol ϕ. Similar to η we also need a (simpler) protocol for an "equality check," with a similar protocol structure and security guarantee. ϕ has 4 input clients and one output client, and q servers without input or output. ϕ (stand-alone) securely realizes the following functionality $\mathcal{E}$, against $t_0 \geq 2$ passive server corruptions. The security is perfect.

Let the input clients be I_0, I_1, I_0' and I_1', with inputs x_0, x_1, x_0' and x_1', respectively. Then $\mathcal{E}$ outputs 1 to the output client if and only if $x_0 \oplus x_1 = x_0' \oplus x_1'$.

Protocol $\mathsf{cROT}^{\mathsf{ROT}}$. The certified-ROT protocol $\mathsf{cROT}^{\mathsf{ROT}}$ proceeds as follows in the ROT-hybrid model. Let κ be a statistical security parameter. (Later we will set κ to be a constant, independent of the final security parameter, m and the final circuit size.)

- *Run "MPC in the head":* The sender prepares κ total views of the execution of the protocol η and $\binom{\kappa}{2}$ total views of the execution of the protocol ϕ. We will refer to the κ executions of η as η_j ($j = 1, \ldots, \kappa$) and the $\binom{\kappa}{2}$ executions of ϕ as $\phi_{jj'}$ (for $1 \leq j < j' \leq \kappa$). The servers are distinct in all these executions (thus there are $q(\kappa + \binom{\kappa}{2})$ servers in all), but the input and output clients in these different executions are identified as follows. There are 2κ input clients I_{j0} and I_{j1} with inputs x_{j0} and x_{j1} respectively for $j = 1, \ldots, \kappa$; η_j has (I_{j0}, I_{j1}) as its two input clients; $\phi_{jj'}$ has $(I_{j0}, I_{j1}, I_{j'0}, I_{j'1})$ as its four clients. There are $Lm + 1$ clients $Z_{\ell 0}^i$ and $Z_{\ell 1}^i$, (for $i = 1, \ldots, m$ and $\ell = 1, \ldots, L$) and Z^0, which serve as the output clients in *all* κ instances of η. Further Z^0 will serve as the output client in all the $\binom{\kappa}{2}$ instances of ϕ.

  In these executions the inputs (x_{j0}, x_{j1}) are set independently for each j as a random additive sharing of the input to $\mathsf{cROT}^{\mathsf{ROT}}$, Γ; i.e., $x_{j0} \oplus x_{j1} = \Gamma$ for each j.

- *Cut and choose:* Using a single round of multiple (1-out-of-N) ROTs,[5] where N is always a power of 2, the sender and the receiver do the following:

---

[5]We only have 1-out-of-2 ROT at our disposal however it is easy to construct a 1-out-of-N ROT from 1-out-of-2 ROT extending a construction of [BCR86]. For completeness we describe the modification in Section A.3.

- For each j, the sender sends views of the pair of input clients (I_{j0}, I_{j1}) via a 1-out-of-2 ROT channel, and the receiver gets one of the views at random.

- In each of the κ executions of η and each of the $\binom{\kappa}{2}$ executions of ϕ, the sender makes *two* lists of the q server views, and sends each list via a 1-out-of-q ROT channel. From each list, the receiver gets at random the view of one of the servers.

- For each $i = 1, \ldots, m$, the sender sends the views of $(Z_{\ell 0}^i, Z_{\ell 1}^i)$ for $\ell = 1, \ldots, L$ through L 1-out-of-2 ROT channels. The receiver gets the views $Z_{\ell r_\ell}^i$ for a random $r \in \{0,1\}^L$. It then sets $c_i = \bigoplus_\ell r_\ell$.

In addition, the sender sends the view of the Z^0 directly (i.e., by invoking the ROT on input (Z^0, Z^0)).

- The receiver checks for consistency in the input it received:

1. *The input clients:* Views of all the exposed input clients are locally correct, i.e., each input client's view is according to its program given its initial input and random tapes (In particular each of them feeds the same input to the instance of η as well as to the $\kappa - 1$ instances of ϕ that it participates in.)

2. *The servers and the "edges":*
   - The views of the exposed servers are locally correct (given the incoming messages and the random tapes).
   - The views of the exposed servers are consistent with the incoming messages reported in the views of the exposed output clients and the outgoing messages implicit in the views of the exposed input clients.
   - The views of the exposed servers are consistent with each other (in particular, if the same server was exposed twice, the two views are identical).

3. *The output clients:*
   - In all executions $\phi_{jj'}$, $(1 \le j < j' \le \kappa)$, Z^0 outputs 1. Also, in all executions η_j, $(1 \le j \le \kappa)$, Z^0 produces the same output (say γ).
   - The views of (i.e., outputs produced by) all the exposed output clients (including Z^0) are correct given the incoming messages.
   - Let $z_{j\ell b}^i$ denote the output of the output client $Z_{\ell b}^i$ in the η_j. Then, for each i, $\bigoplus z_{j\ell r_\ell}^i$ evaluates to the same value (say $\tilde{s}^i$) for all j.

If all the verifications succeed, then the receiver outputs $(\tilde{s}^i, \ldots, \tilde{s}^i; \gamma)$. Else it aborts the protocol and outputs abort.

Lemma 5 *Given a protocol η satisfying the conditions above, $\mathsf{cROT}^{\mathsf{ROT}}$ defined above is a UC-secure realization of the certified ROT functionality $\mathcal{F}_{\mathsf{cov\text{-}cROT}}$. The simulation is perfect.*

PROOF OVERVIEW: The interesting cases are when exactly one of the sender or the receiver is corrupted. *Corrupt Receiver.* In this case, the security easily follows from the privacy of the protocol η. Consider a simulator in the ideal world interacting with $\mathcal{F}_{\mathsf{cov\text{-}cROT}}$ as the receiver, and simulating the protocol to the corrupt receiver. Note that a receiver gets to see only the views of some of the servers and some of the clients. The simulator obtains the outputs for the receiver from $\mathcal{F}_{\mathsf{cov\text{-}cROT}}$. Observe that the views that the receiver obtains in each execution of η or ϕ are of up to 2 servers, input clients (with only one share of an additive sharing of Γ) and some output clients. By the security guarantee on η and ϕ, this view can be

perfectly simulated given just the outputs that these output clients receive. Since these outputs are available to the simulator, it can carry out a perfect simulation.

Corrupt Sender. This case is the more interesting one.

Consider the graph on the parties in the protocol with an edge between two parties who can exchange a message in the protocol. (That is, there are edges between the input clients and the servers, among the q servers, and between the servers and the output clients.)

Let δ be the minimum probability of detecting an "internally" inconsistent execution of ϕ or of η. Note that $\delta \geq 1/q^2$.

The simulator obtains the entire collection of views that the sender submits as inputs to the ROT executions. It examines these views and first prepares an "input consistency graph" as follows: For each j ($j = 1, \ldots, \kappa$), such that both I_{j0} and I_{j1} are locally consistent, add a node to the graph. For each pair (j, j') such that the entire execution of $\phi_{jj'}$ is correct (given the randomness of the servers), add an edge between the corresponding nodes (if present) in the graph. Note that for any connected component in this graph, there is a unique input value Γ such that for all j in the connected component, $x_{j0} \oplus x_{j1} = \Gamma$ and the input clients of η_j use this input.

Now the simulator proceeds as follows:

- The simulator sends the cheat command to $\mathcal{F}_{\text{cov-cROT}}$ if any of the following conditions hold.

  1. The input consistency graph has no connected component of more than $\kappa/2$ nodes.

  2. η_j was internally consistent only for $\kappa/2$ or fewer values of j.

  3. For some i, for all ℓ ($\ell = 1, \ldots, L$), output client $Z^i_{\ell 0}$ or $Z^i_{\ell 1}$ was locally incorrect.

  Then, with probability ϵ, $\mathcal{F}_{\text{cov-cROT}}$ will send corrupted to both parties; in this case the simulator aborts the simulated protocol. With probability $1 - \epsilon$, $\mathcal{F}_{\text{cov-cROT}}$ will allow the simulator to cheat: the simulator samples random coins for the receiver and calculates the probability p that the simulator would have aborted the protocol for this randomness. We shall see that $p > \epsilon$. Then with probability $(p - \epsilon)/(1 - \epsilon)$ the simulator will abort the simulated protocol (so that total probability of the simulated protocol being aborted is exactly p) and send corrupted to $\mathcal{F}_{\text{cov-cROT}}$; with probability $(1 - p)/(1 - \epsilon)$ it will continue the simulation conditioned on the receiver not aborting, derive the output that the receiver obtains, and send this to $\mathcal{F}_{\text{cov-cROT}}$ as the output for the receiver.

- If the above conditions do not hold (and so the simulator does not send cheat to $\mathcal{F}_{\text{cov-cROT}}$), then

  1. The simulator can derive an input Γ, which is the input defined by the majority of the nodes in the input consistency graph.

  2. Also, since more than $\kappa/2$ executions of η were internally consistent, there is some j such that η_j was internally consistent and used inputs x_{j0} and x_{j1} such that $x_{j0} \oplus x_{j1} = \Gamma$.

  3. Further, for each i, for at least one ℓ (ℓ can depend on i), both $Z^i_{\ell 0}$ and $Z^i_{\ell 1}$ were locally correct (for all κ execution of η).

  In this case the simulator proceeds to give a perfect simulation as follows. Note that the only information the sender learns is whether the receiver aborts the protocol or not. The simulator carries out all the checks like the receiver, except for the last step. For the last check, the receiver gets, for each i, $r \in \{0, 1\}^L$ and sets $c_i = \bigoplus_\ell r_\ell$. But the simulator (who does not know the randomness of the channel and hence c_i), simply picks a random string r. However, this is equivalent to picking r' where the ℓ-th bit of r is flipped. This is because both the views $Z^i_{\ell 0}$ and $Z^i_{\ell 1}$ are locally correct. Thus the simulated protocol is a perfect simulation of the real protocol so far.

If the simulated protocol does not abort, then the simulator sends Γ to $\mathcal{F}_{\text{cov-cROT}}$. Otherwise it sends corrupted to $\mathcal{F}_{\text{cov-cROT}}$. By the correctness of η_j for some j, we know that the output of the receiver in the real protocol is perfectly simulated by the output $\mathcal{F}_{\text{cov-cROT}}$ delivers in the ideal world, on input Γ.

To complete the argument we need to argue that the probability p of the real protocol aborting in the three cases where the simulator would send cheat to $\mathcal{F}_{\text{cov-cROT}}$ is indeep at least ϵ. We consider the three cases below.

1. If the input consistency graph has no more than $\kappa/2$ nodes in a single connected component, then there must be either $\Omega(\kappa)$ missing nodes (i.e., j for which the node was not added to the graph), or $\Omega(\kappa^2)$ missing edges (i.e., edges (j, j') that were not added to the graph).

   - For each missing node j, one of the two input clients I_{j0} and I_{j1} is locally incorrect (sending different inputs to executions of η and ϕ). If there are d such missing nodes, the probability of the real protocol aborting is at least $1 - 2^{-d}$, because for each j there is an independent probability of at least half of exposing an incorrect view.

   - For each missing edge (j, j') (between nodes which are present in the input consistency graph), the probability of the protocol aborting is δ if $\phi_{jj'}$ is internally inconsistent, or is 1, if Z^0 is either locally incorrect or produces an output 0 in $\phi_{jj'}$. If there are d such missing edges, the probability of the protocol aborting is at least $1 - (1 - \delta)^{-d}$ (as these events are independent of each other).

   In any of these cases, the real protocol execution would abort with probability at least $p_1 = 1 - (1 - \delta)^{\Omega(\kappa)}$.

2. If η_j was internally consistent only for $\kappa/2$ or fewer values of j, then the protocol will abort with probability at least $p_2 = 1 - (1 - \delta)^{\kappa/2}$.

3. If for some i, for all ℓ ($\ell = 1, \ldots, L$), at least one of the output clients $Z^i_{\ell 0}$ and $Z^i_{\ell 1}$ was locally incorrect, then the protocol would abort with probability at least $p_3 = 1 - 2^{-(L-1)}$.

To ensure that these abort probabilities are at least ϵ, we set $\epsilon := \min(p_1, p_2, p_3)$. Note that with $L = 2$, and a large enough κ, we can get $\epsilon = \frac{1}{2}$. By choosing $L = \Omega(\kappa)$, we get $\epsilon = 1 - 2^{-\Omega(\kappa)}$. $\triangleleft$

A.3 1-out-of-N ROT from 1-out-of-2 ROT

[BCR86] provide a transformation of 1-out-of-N OT (where N is a power of 2) from 1-out-of-2 OT. In our setting we need a similar transformation for the setting of random OT. The reduction of [BCR86, Section 2.2] works almost directly for us and we can use it to guarantee that the receiver obtains exactly among the N sent strings. However this protocol does not guarantee that the received string is uniform among the sent strings. We will next sketch a construction for realizing 1-out-of-N ROT from 1-out-of-2 ROT and hint at the proof. We will only sketch the construction for the setting of bit-ROT (that is when the sent messages are actually bits), but the same protocol generalizes to strings in a natural manner.

Suppose the sender's input bits are $b_1, b_2 \ldots b_N$ and the goal of the ROT protocol is to enable the receiver to obtain exactly one (uniform among the N bits) of these bits. The sender generates a complete binary tree with N leaves, sampling:

1. a random bit r_e corresponding to each edge e of the tree (call them edge mask bits), and

2. a random bit r_i corresponding to every leaf $i \in [N]$ in the tree.

The sender generates generates $x_i = r_i \oplus b_i$ for each $i \in [N]$ and sends it to the receiver. Additionally it for each node v let r_{v_l} and r_{v_r} be edge masks for the left and the right edges going out of that node. For every node v in the tree, the sender uses one execution of 1-out-of-2 ROT to transfer either r_{v_l} or r_{v_r} to the receiver.

Observe that the receiver will always get all the edge mask bits from the root to one of the leaves (lets say the kth leaf). The receiver at this point, xors these edge masks together with x_k and outputs the resulting value as b_k.

It is easy to see that k remains hidden from the sender. Arguing that the receiver only obtains a single value among $b_1, \ldots, b_2$ requires some work and we will hint at the argument. Notice that if the receiver gets to see the left edge mask bit of a node then all bits in the right sub-tree are perfectly hidden because the edge mask for the right edge is hidden. Note however that the receiver still obtains the edge masks for some of the edges in the right subtree which are correlated with the x_i values. But still even with these correlations we can argue that enough edge masks will be hidden in order to hide every leaf of the subtree. We refer the reader to [BCR86] for a complete argument (where the same argument is given for a linear structure instead of a tree structure).