Secure Computation from One-Way Noisy Communication, or: Anti-Correlation via Anti-Concentration

Shweta Agrawal[1], Yuval Ishai[2], Eyal Kushilevitz[2], Varun Narayanan[3], Manoj Prabhakaran[4], Vinod Prabhakaran[3], and Alon Rosen[5]

[1] Indian Institute of Technology Madras, India,
shweta@iitm.ac.in,
[2] Technion, Israel,
yuvali@cs.technion.ac.il, eyalk@cs.technion.ac.il,
[3] Tata Institute of Fundamental Research, India,
varunnkv@gmail.com, vinodmp@tifr.res.in,
[4] Indian Institute of Technology Bombay, India,
mp@cse.iitb.ac.in,
[5] IDC Herzliya, Israel,
alon.rosen@idc.ac.il,

Abstract. Can a sender encode a pair of messages (m_0, m_1) jointly, and send their encoding over (say) a binary erasure channel, so that the receiver can decode exactly one of the two messages and the sender does not know which one?

Garg et al. (Crypto 2015) showed that this is information-theoretically impossible. We show how to circumvent this impossibility by assuming that the receiver is computationally bounded, settling for an inverse-polynomial security error (which is provably necessary), and relying on *ideal obfuscation*. Our solution creates a "computational anti-correlation" between the events of receiving m_0 and receiving m_1 by exploiting the *anti-concentration* of the binomial distribution.

The ideal obfuscation primitive in our construction can either be directly realized using (stateless) tamper-proof hardware, yielding an unconditional result, or heuristically instantiated in the plain model using existing indistinguishability obfuscation schemes.

As a corollary, we get similar feasibility results for *general secure computation* of sender-receiver functionalities by leveraging the completeness of the above "random oblivious transfer" functionality.

1 Introduction

Starting with the pioneering work of Wyner [57], who showed that the wiretap channel can be used for secure communication, a long line of work in cryptography studied the usefulness of noisy channels for general cryptographic tasks [13, 48, 12, 22, 55, 56, 51, 35]. A major landmark in this line of work is a full characterization of the "complete" channels on which oblivious transfer, and hence secure two-party computation, can be based [20, 21]. In a nutshell, almost all nontrivial noisy channels are complete in this sense.

However, most cryptographic constructions from noisy channels crucially require interaction, and while this is not always a barrier, there are applications in which interaction is inherently unidirectional. Indeed, secure communication in this setting was the topic of Wyner's work, and is a central theme in the big body of work on "physical layer security" [14, 50]. Given only one-way noisy communication, any functionality that can be securely realized can be expressed as a randomized mapping $f : \mathcal{A} \to \mathcal{B}$ that takes an input $a \in \mathcal{A}$ from a *sender* S and delivers an output $b = f(a)$ to a *receiver* R. Note that, here the randomness is internal to the functionality, and is neither known to nor can be influenced by the sender or the receiver. We will give examples for useful functionalities of this type in Section 1.3.

The goal is to realize such sender-receiver functionalities assuming that S and R are given access to a *channel* $\mathcal{C} : \mathcal{X} \to \mathcal{Y}$. Such channels are usually simpler than the target function f, and can be plausibly assumed to be available to the parties. Well-known examples of "simple" channels that correspond to naturally occurring processes are the *binary erasure channel (BEC)*, which erases each transmitted bit with some fixed probability $0 < p < 1$, and the *binary symmetric channel (BSC)* which flips each bit with probability $0 < p < 1/2$.

1.1 Complete Channels

The general study of secure computation from one-way noisy communication was initiated by Garg et al. [25], who showed that one-way communication over BEC or BSC suffices for realizing any *deterministic* sender-receiver functionality. This includes zero-knowledge proofs as a useful special case. For general, possibly randomized, functionalities, they showed that the following random string-OT functionality (ROT) described below (where a_0, a_1 are strings), is complete:

$$\mathcal{C}_{\mathsf{ROT}}(a_0, a_1) = \begin{cases} (a_0, \perp) & \text{w.p. } \frac{1}{2} \\ (\perp, a_1) & \text{w.p. } \frac{1}{2}, \end{cases}$$

This was recently extended to the case when a_0, a_1 are bits [2], albeit at the (necessary) cost of allowing an inverse polynomial, rather than negligible, error.

Note that in ROT the receiver must learn *exactly* one of the two messages but the sender should not be able to guess which one. This makes the secure realization of ROT highly non-trivial. Indeed, ROT appears to be significantly more powerful than BEC and BSC, and it is not clear how to realize it by a naturally occurring process. While BEC and BSC merely erase or flip bits of information *randomly and independently*, ROT induces a strong *anti-correlation* between events, namely the receipt of a_0 and the receipt of a_1.

Can the anti-correlation inherent in ROT be generated "out of thin air" by invoking simple channels such as BEC or BSC? This question was already addressed by Garg et al. [25], who showed that the simple noisy channels are indeed *not* complete. In fact, ROT cannot be securely realized from such channels even if one considers semi-honest parties (who do not deviate from the protocol) and allows a small constant security error.[6]

It is instructive to sketch the proof of this impossibility result. We consider the more general case of a *string* erasure channel (SEC) that erases each input string with probability p. The proof relies on a classical correlation inequality due to Harris and Kleitman [33, 43], asserting that for any two *monotone* Boolean functions $f_0, f_1 : \{0,1\}^n \to \{0,1\}$ and for any product distribution R over $\{0,1\}^n$, the events $f_0(R) = 1$ and $f_1(R) = 1$ are *not* anti-correlated. That is,

$$\Pr\left[f_0(R) = 1 \wedge f_1(R) = 1\right] \geq \Pr\left[f_0(R) = 1\right] \cdot \Pr\left[f_1(R) = 1\right].$$

Now, by the receiver's security requirement, even if we condition on a "typical" joint encoding $\boldsymbol{x}$ of (a_0, a_1) that the sender transmits over the SEC channel, the receiver's output should be distributed almost as prescribed by the ROT functionality. In particular, if p_i is the probability that the receiver can confidently decode a_i conditioned on $\boldsymbol{x}$ being sent, and E_i is the corresponding conditional event, then $p_0 \approx p_1 \approx 0.5$. Letting n denote the number of invocations of the SEC, $r \subseteq [n]$ represent the set of received symbols, and $f_i(r)$ indicate whether E_i occurs on received set r, the Harris-Kleitman inequality implies that $\Pr\left[E_0 \wedge E_1\right] \geq p_0 \cdot p_1 \approx 0.25$, contradicting the sender's security requirement.

The above impossibility result is purely information-theoretic and does not give rise to a constructive attack. In particular, the functions f_i are monotone because information is monotone: more received symbols mean more confidence. While there are examples for non-monotonicity of information in a computational setting, for instance in the context of generalized secret sharing [45], it is not clear that this has any relevance to the current setting. In fact, Garg et al. [25] showed an efficient attack that rules out computationally secure protocols with *negligible* security error. This leaves open the possibility of obtaining ROT from naturally-occurring channels with a small constant, or better yet *inverse-polynomial*, error.

1.2 Our Results

In this work, we show that the impossibility result for ROT from SEC and other simple channels *can* be circumvented, if one is willing to settle for security against a computationally bounded receiver and to allow for inverse-polynomial error. On the one hand, both of these relaxations are necessary in light of the above mentioned impossibility results but, on the other hand, we still find the positive result to be unexpected, even with these relaxations.

Our main result is cast in a generic model that assumes "ideal obfuscation," enabling the sender to give the receiver an *oracle access* to an obfuscated program. In this generic model, we can unconditionally obtain information-theoretic security by assuming that a malicious receiver is restricted to polynomially many

[6] The argument in [25] implicitly relies on the technical assumption that the ROT protocol is *Las Vegas*, in the sense that if the receiver does output a message a_b, then this message is correct; all existing protocols in this setting, including those presented in this work, satisfy this requirement.

queries to the program, but is otherwise computationally unbounded. Before discussing the question of instantiating the generic model, we state the main result.

Theorem 1 (Informal). *There is a one-way secure computation (OWSC) protocol for ROT over the binary erasure channel (BEC) as well as the binary symmetric channel (BSC) using ideal obfuscation, with inverse-polynomial statistical security error against a semi-honest sender and a query-bounded malicious receiver.*

Building on Theorem 1, we can leverage the completeness of ROT for sender-receiver functionalities [25] to obtain the following general completeness result:

Theorem 2 (Informal). *BEC and BSC are (each) complete for OWSC using ideal obfuscation, with inverse-polynomial statistical security against a semi-honest sender and a query-bounded malicious receiver.*

Instantiating ideal obfuscation. A direct way of implementing the ideal obfuscation in our construction is by sending (stateless) tamper-proof hardware to the receiver. To obtain a plain-model instantiation, a natural approach is to use *indistinguishability obfuscation* (iO)[6, 30] instead of ideal obfuscation. Following the first candidate construction of Garg et al. [24], iO has been studied extensively [26, 4, 1, 37, 19, 7, 46, 8, 15, 38, 27, 16, 54] and has been constructed from well-studied assumptions in the recent breakthrough work of Jain, Lin and Sahai [38]. Unfortunately, we were unable to prove that our protocols remain (computationally) secure when replacing ideal obfuscation by iO, and consider this to be a highly plausible conjecture. Since iO is "best possible" obfuscation [30], it follows that if *some* instantiation of ideal obfuscation in our protocols is secure then its instantiation with *any* iO scheme is secure. Concretely, we make the following conjecture.

Conjecture 1 (Informal). Replacing ideal obfuscation by any secure iO scheme in the protocol establishing Theorem 1 results in a OWSC protocol for ROT over BEC or BSC that has inverse-polynomial *computational* security against a semi-honest sender and a malicious receiver.

While there are strong negative results for instantiating ideal notions of obfuscation [6, 28], these results require at least one of the building blocks to be "contrived." They are not known to apply to any combination of a natural (unbroken) iO candidate and natural application. We believe that Conjecture 1 is qualitatively similar to the leap of faith one makes when heuristically instantiating natural protocols in generic models such as the Random Oracle Model [9] or the Generic Group Model [53]. Arguably, the leap of faith in our case is quite conservative because of the simple and "non-cryptographic" functions to which we apply ideal obfuscation. This should be contrasted with typical applications of obfuscation in cryptography, and also with heuristic iO candidates whose security needs to hold even for contrived pairs of equivalent circuits. See Section 1.5 for further discussion.

Assuming Conjecture 1, we can obtain a plain-model variant of Theorem 2 with security against a *malicious* sender by using OWSC for non-interactive zero knowledge to effectively emulate an honest sender behavior.

Theorem 3 (Informal). *Suppose* iO *exists and Conjecture 1 holds. Then, BEC and BSC are (each) complete for OWSC, with inverse-polynomial computational security against malicious sender and receiver.*

We leave open the question of eliminating Conjecture 1 or, better yet, basing the conclusion of Theorem 3 on a weaker or incomparable assumption to iO.

1.3 Why Base on One-Way Noisy Communication?

Several important cryptographic tasks can be captured as sender-receiver functionalities. A natural example, already given in [25] is that of randomly generating "puzzles" without giving any of the parties an advantage in solving them. For instance, the sender can transmit to a receiver a random Sudoku challenge, or a random image of a one-way function, while the receiver is guaranteed that the sender has no advantage in solving the puzzle. More generally, one could use secure realizations of sender-receiver functionalities to unidirectionally generate trusted parameters such as RSA moduli or common reference strings. Unlike the common interactive solutions to such problems, here we consider a setting that allows for completely non-interactive solutions.

Another example of a useful sender-receiver functionality is randomized *blind signatures*, which can be captured by a randomized function that takes a message and a signing key from the sender and delivers a signature on some randomized function of the message to the receiver (for instance by adding a random serial number to a given dollar amount). Randomized blind signatures are a fundamental building block for e-cash applications. They can also be used for non-interactive certified PKI generation, where an authority can issue to a user signed public keys, while only the users learn the corresponding secret keys.

Non-interactive zero-knowledge (NIZK), which is constructed in the common random string model, can also be implemented in the sender-receiver model, by modeling it as a deterministic function that takes an NP-statement and a witness from the sender and outputs the statement along with the output of the verification predicate to the receiver. As noted by Garg et al. [25], NIZK over a one-way noisy channel provides a truly non-interactive solution to zero knowledge proofs, where no trusted common randomness is available to the parties. Moreover, this solution can achieve useful properties of interactive zero-knowledge protocols such as non-transferability and deniability, which are impossible to achieve in the standard non-interactive setting.

While the above applications require security against a malicious sender, it is also meaningful (and non-trivial) to implement protocols that are secure against semi-honest senders. Such protocols can be generically compiled to be secure against malicious senders by invoking NIZK in the sender-receiver model. Note that NIZK by itself is not sufficient for realizing many non-trivial functionalities, including the ones mentioned above. For this, it is necessary (and sufficient) to have a secure realization of semi-honest ROT.

Applications notwithstanding, understanding the cryptographic power of noisy channels with one-way communication is a fundamental question from the theoretical standpoint.

1.4 Technical Overview

To present the new idea underlying our constructions, we focus on a protocol for realizing ROT using a string erasure channel (SEC), with erasure probability $p = 0.5$. This can be extended to BEC and BSC as required by Theorem 1. To realize ROT, we want the symbols that the sender transmits over the SEC to partition the probability space into two events E_0 and E_1, such that $\Pr[E_0] \approx \Pr[E_1] \approx 0.5$, and in each event E_i the receiver can learn a_i but not a_{1-i}.

The protocol begins by having the sender transmit a random n-tuple $\boldsymbol{x} \in \varSigma^n$ over a large alphabet $\varSigma$ that makes the probability of predicting an erased symbol negligible. It sends $\boldsymbol{x}$ over the SEC. It then picks a small secret "test set" $S \subset [n]$ and sends to the receiver an obfuscated program $F = F_{S,\boldsymbol{x},\boldsymbol{a}}$ that expects the receiver to report all of the symbols it received from the channel. (When instantiating the ideal obfuscation, the sender needs to communicate the obfuscated program over a reliable channel; however, the latter can be implemented with constant rate over any standard noisy channel.) After checking that each unerased symbol reported by the receiver matches the corresponding symbol in $\boldsymbol{x}$, the program F counts how many symbols from the secret set S were reported; if this number is bigger than $|S|/2$ it outputs a_1, otherwise it outputs a_0.

Sender input: $\boldsymbol{a} = (a_0, a_1)$.
Sender: Sample random $\boldsymbol{x} \in \varSigma^n$ and send $\boldsymbol{x}$ over SEC.
Sender: Sample random $S \subset [n]$ of size $\sqrt{n}$ and send an obfuscation of $F = F_{S,\boldsymbol{x},\boldsymbol{a}}$ over reliable channel.
Receiver: Output $F(\boldsymbol{y})$, where $\boldsymbol{y}$ is the sequence of non-erased symbols.

Fig. 1: ROT from String Erasure Channel (SEC)

The erasures induced by the channel are independent of $\boldsymbol{x}$, and so whether the receiver outputs a_0 or a_1 is independent of the sender's view. Thus, the protocol is secure even against a computationally unbounded semi-honest sender.

For security against the receiver, we consider two cases. If the channel delivers a minority of the symbols from S, then an honest receiver can legitimately obtain a_0 from F, and even a dishonest receiver will need a super-polynomial number of calls to F to guess even one of the missing symbols.

On the other hand, what if the channel delivers a majority of the symbols from S, which occurs with probability ≈ 0.5? In this case, a dishonest receiver can obtain both messages by first acting honestly, legitimately obtaining a_1, and then invoking F again and obtaining a_0 by just "forgetting" some of the received symbols. The latter attack seems inherently impossible to defend against. How can we expect a receiver who obtained few symbols from S to prove its ignorance?

It turns out, however, that there is a surprisingly simple solution: F will not deliver a_0 when the total reported number of received symbols is significantly below $n/2$. In other words, F does not trust a receiver who claims to be too unlucky. Intuitively, the reason this simple approach works is that S is both small and secret. So without knowledge of S, for every symbol in S that the receiver tries to "forget" it needs to unwillingly forget a large number of additional received symbols. By choosing the size of S and the "unluckiness" threshold carefully, we can ensure that successfully mounting the above "forgetting" attack is computationally infeasible except for a bad event that occurs with inverse-polynomial probability.

The analysis however requires more care and crucially relies, in addition to standard Chernoff-style concentration inequalities, on a simple *anti-concentration* phenomenon: the binomial distribution with n trials is almost always $\Omega(n^{1/2})$-far from its mean. Metaphorically speaking, the events E_0 and E_1 that are separated by this anti-concentration can be viewed as "computational black holes" whose disjoint gravity zones cover almost the entire probability space.

In a bit more detail, for a transmitted $\boldsymbol{x} \in \Sigma^n$ and set $V \subseteq [n]$ indicating non-erased coordinates, let $\boldsymbol{x}|_V$ denote the vector $\boldsymbol{x}$ with all coordinates outside of V replaced by a special erasure symbol $\perp$. Set the "unluckiness" threshold to be $n/2 - n^{0.51}$ and the size of S to be $\sqrt{n}$. Define the function F as:

$$F_{S,\boldsymbol{x},\boldsymbol{a}}\left(\boldsymbol{y}|_V\right) = \begin{cases} (\perp,\perp) & \text{if } \ (\boldsymbol{y}|_V \neq \boldsymbol{x}|_V) \vee \left(|V| < n/2 - n^{0.51}\right), \\ (a_0,\perp) & \text{otherwise if } \ |V \cap S| < |S|/2, \\ (\perp, a_1) & \text{otherwise.} \end{cases}$$

where $\boldsymbol{y}|_V$ denotes a n-tuple of presumably received symbols.

An honest receiver, who always feeds $\boldsymbol{y}|_V = \boldsymbol{x}|_V$ to F, gets unlucky with negligible probability. This is because, over the random erasures of the SEC, $\Pr\left[|V| \geq n/2 - n^{0.51}\right] > 1 - \mathrm{negl}(n)$, and conditioned on this event, $|V \cap S|$ is symmetrically distributed around $|S|/2$. In particular, the output of F is almost equally likely to be a_0 as it is to be a_1.

A dishonest receiver, on the other may attempt to learn both a_0 and a_1 by feeding $\boldsymbol{y}|_U$ to F, where $U \neq V$ does not correspond to the set of non-erased coordinates. This is not a problem if $\boldsymbol{y}|_U \neq \boldsymbol{x}|_U$ as in such a case F will output $(\perp,\perp)$, but there is always a chance that the receiver can come up with $\boldsymbol{y}|_U = \boldsymbol{x}|_U$. Here we have two possible cases:

U **is not contained in** V**.** This case can be ruled out when $|\Sigma|$ is super-polynomially large, as it requires the receiver to correctly guess a randomly sampled x_i for $i \in U \setminus V$.

U **is a strict subset of** V**.** In this case, one cannot prevent the receiver from feeding an input $\boldsymbol{y}|_U = \boldsymbol{x}|_U$, as this merely amounts to erasing symbols from the received string $\boldsymbol{x}|_V$. Here, the only hope for the receiver to obtain both a_0 and a_1 is to be able to transition from the case $|V \cap S| \geq |S|/2$ to the case $|U \cap S| < |S|/2$. Note that, by anti-concentration, in this case $|V \cap S|$ is likely larger than $|S|/2$ by $\Omega(\sqrt{|S|})$ and, moreover, S is secret, hence the receiver cannot just find such U by only removing few elements of V in an exhaustive search. On the other hand, if the receiver tries to forget many symbols from the unknown S by just forgetting many symbols from V, it will hit the unlucky zone where F returns $(\perp, \perp)$.

To prevent attacks as in the first case, it is imperative that the obfuscation of F hide $\boldsymbol{x}$. Avoiding attacks as in the second case, on the other hand, requires the obfuscation to hide S. What type of obfuscation would be sufficient for hiding $\boldsymbol{x}$ and S? Ideal obfuscation limits the receiver to black-box access to F. Intuitively, this means that the receiver's attempts to mount the above attacks are restricted to random guesses, as $\boldsymbol{x}$ and S are information theoretically hidden.

1.5 Discussion

The unconditional result given by Theorem 1 (and subsequent theorems that build on it) captures the main contribution of this work. Our use of ideal obfuscation is technically equivalent to having a single, stateless, tamper-proof hardware token shipped from the sender to the receiver. In fact, unlike current candidates for cryptographic obfuscation, such an approach may be efficient enough to be implemented. Thus, our results can be cast as part of a long line of theory-oriented works on cryptography using tamper-proof hardware (see [40, 29, 32, 5], along many others).

From a complexity theoretic point of view, the ideal obfuscation primitive can be viewed as a (succinctly described) *oracle* generated by the sender, such that security holds unconditionally with respect to any *query-bounded* receiver that has access to this oracle. For instance, this is the model used in works on zero-knowledge PCP [42, 36, 47]. Alternatively, it can be seen as a second, "resettable" sender, analogously to the multi-prover proof model [10, 39, 31, 11].

An unusual aspect of our main feasibility result that separates it from almost all nontrivial applications of obfuscation in cryptography is that it is based on *ideal obfuscation alone*, without making any additional assumptions such as the existence of one-way functions (or alternatively $\mathsf{NP} \not\subseteq$ $\mathsf{io}\text{-}\mathsf{BPP}$ [44]). In particular, the functions we obfuscate are simple, explicit and "non-cryptographic."

We also note the analogy with the Random Oracle Model (ROM) methodology: there is a long tradition in cryptography of using a construction in an idealized "generic" model, such as the ROM [9], as a stepping stone towards heuristic plain-model realizations. The latter are obtained by using concrete hash functions as a substitute for the random oracle. For example, constructions of transparent SNARGs for NP follow this approach [49]. Our proposal is analogous: heuristically instantiate the ideal obfuscation by using any iO construction from the literature. There are strong negative results for instantiating ideal notions of obfuscation [6]. These are in a sense analogous to similar negative results for instantiating the ROM [18]. However, similarly to ROM instantiations, we do not see a reason why these negative results should apply to a combination of a *natural* application and a *natural* iO construction that was not designed with a counterexample in mind.

Finally, most solutions for natural cryptographic tasks that were initially cast in idealized models were later followed by plain-model constructions under simple and plausible cryptographic assumptions. We expect the current work to follow a similar path.

2 Preliminaries

Notation. We write $x \leftarrow \mathcal{X}$ to denote the process of freshly sampling a uniformly random element x from a finite set $\mathcal{X}$. We denote the i-th coordinate of a vector $\boldsymbol{x} \in \mathcal{X}^n$ by either x_i or $\boldsymbol{x}(i)$. For a vector $\boldsymbol{x} \in \mathcal{X}^n$ and set $A \subseteq [n]$, the restriction of $\boldsymbol{x}$ to A, denoted by $\left.\boldsymbol{x}\right|_A$, is the length n vector in $(\mathcal{X} \cup \{\perp\})^n$ with all the coordinates outside of A replaced by a special erasure symbol $\perp$. That is, $\left.\boldsymbol{x}\right|_A(i) = \boldsymbol{x}(i)$ if $i \in A$ and $\left.\boldsymbol{x}\right|_A(i) = \perp$ otherwise. The notation $\binom{[n]}{k}$ denotes the family of all subsets of $[n]$ with size k.

2.1 Sender-Receiver Functionalities and Channels

We study secure computation tasks that are made possible by one-way communication over a noisy channel. Such tasks can be captured by *sender-receiver* functionalities, that take an input from a *sender* S and deliver a (possibly) randomized output to a *receiver* R. In the randomized case, the randomness is picked by the functionality and is not revealed to the sender or the receiver. More precisely, a sender-receiver functionality is a randomized mapping $f : \mathcal{A} \to \mathcal{B}$ that takes an input $a \in \mathcal{A}$ from a sender S and delivers an output $b = f(a)$ to a receiver R. We will sometimes refer to f simply as a *function*.

In order to realize f, we assume that S and R are given parallel access to a *channel* $\mathcal{C} : \mathcal{X} \to \mathcal{Y}$. A channel is also a sender-receiver functionality but is usually much simpler than the target function f. We define three channels of interest below.

- **BSC.** $\mathcal{C}^p_{\mathsf{BSC}}$ denotes the *Binary Symmetric Channel* (BSC) with crossover probability p: i.e., for input $x \in \{0,1\}$, the output $\mathcal{C}^p_{\mathsf{BSC}}(x)$ is $1-x$ with probability p and is x otherwise.
- **SEC and BEC.** $\mathcal{C}^p_{\mathsf{SEC}}$ denotes the *String Erasure Channel* (SEC) which takes an input string of a fixed length and outputs $\perp$ with probability p and x otherwise. When the string length is 1, $\mathcal{C}^p_{\mathsf{SEC}}$ is called a *Binary Erasure Channel* (BEC), and denoted by $\mathcal{C}^p_{\mathsf{BEC}}$. When $p = \frac{1}{2}$, we may omit it from the notation.
- **ROT.** The *(String) Randomized Oblivious Transfer* channel $\mathcal{C}_{\mathsf{ROT}}$ takes as input a pair of fixed-length strings (x_0, x_1) and outputs $(x_0, \perp)$ or $(\perp, x_1)$ with probability $\frac{1}{2}$ each.

For brevity, we shall write $\mathcal{C}(x_1, \ldots, x_m)$ to denote $(\mathcal{C}(x_1), \ldots, \mathcal{C}(x_m))$, i.e., the outcome of m independent invocations of a channel $\mathcal{C}$.

2.2 Secure Computation with One-Way Communication

A secure protocol for $f : \mathcal{A} \rightarrow \mathcal{B}$ over a channel $\mathcal{C}$ is formalized via the standard definitional framework of reductions in secure computation. Our definitions are in fact simpler because of the non-interactive setting. We start with the simplest case of defining *information-theoretic* security against *semi-honest* parties for a *finite* function f, ignoring computational complexity. We then describe extensions to malicious parties, computational security, and infinite families of functions.

OWSC protocols. A one-way secure computation protocol for f over $\mathcal{C}$ specifies a randomized encoder that maps the sender's input a into a sequence of channel inputs $\boldsymbol{x}$, and a decoder that maps the receiver's channel outputs $\boldsymbol{y}$ into an output b. Up to an error bound parameter ϵ, the protocol should satisfy the following security requirements: (i) given the sender's view, which consists of an input a and the messages $\boldsymbol{x}$ that it fed into the channel, the receiver's output should be distributed as $f(a)$, and (ii) the view of the receiver, namely the messages $\boldsymbol{y}$ it received from the channel, can be simulated from $f(a)$. Note that (i) captures receiver security against a semi-honest sender *as well as correctness*, while (ii) captures sender security against the receiver. Also note that since the receiver does not send messages, whether it is semi-honest or malicious does not make a difference. We formalize the above security requirements below, using $\varDelta$ to denote statistical distance.

Definition 1 (One-way secure computation: semi-honest sender). *Given a randomized function $f : \mathcal{A} \rightarrow \mathcal{B}$ and a channel $\mathcal{C} : \mathcal{X} \rightarrow \mathcal{Y}$, a pair of randomized functions $\langle \mathsf{S}, \mathsf{R} \rangle$, where $\mathsf{S} : \mathcal{A} \rightarrow \mathcal{X}^n$ and $\mathsf{R} : \mathcal{Y}^n \rightarrow \mathcal{B}$, is said to be an* ϵ-secure OWSC protocol for f over $\mathcal{C}$ *(with semi-honest sender) if there exists a simulator $\mathcal{S}_{\mathsf{R}} : \mathcal{B} \rightarrow \mathcal{Y}^n$, such that for all $a \in \mathcal{A}$, the following hold:*

$$\varDelta\left((\mathsf{S}(a), f(a))\ ,\ (\mathsf{S}(a), \mathsf{R}(\mathcal{C}(\mathsf{S}(a))))\right) \leq \epsilon \qquad \text{(Security against semi-honest sender)}$$

$$\varDelta\left(\mathcal{S}_{\mathsf{R}}(f(a))\ ,\ \mathcal{C}(\mathsf{S}(a))\right) \leq \epsilon \qquad \text{(Security against receiver)}$$

OWSC for malicious parties. In the case of a malicious sender, our security requirement coincides with the standard notion of universally composable (UC) security [17], but with simplifications implied by the communication model. The extra security requirement in this case is that for any strategy of the sender (for choosing $\boldsymbol{x}$), a simulator is able to extract a valid input. Formally, an OWSC protocol for f over $\mathcal{C}$ is *secure against malicious parties* if, in addition to the requirements in Definition 1, there exists a randomized simulator $\mathcal{S}_{\mathsf{S}} : \mathcal{X}^n \rightarrow \mathcal{A}$ such that for every $\boldsymbol{x} \in \mathcal{X}^n$,

$$\varDelta\left(f(\mathcal{S}_{\mathsf{S}}(\boldsymbol{x}))\ ,\ \mathsf{R}(\mathcal{C}(\boldsymbol{x}))\right) \leq \epsilon \qquad \text{(Security against malicious sender)}$$

Note that the first condition of Definition 1 is retained to imply correctness when the sender is honest, and the second condition implies security against malicious receiver as well.

OWSC with computational security. We can naturally relax the above definition of (statistical) ϵ-secure OWSC to a *computationally* (T, ϵ)*-secure* OWSC, for a distinguisher size bound T, by replacing each statistical distance bound $\Delta(A, B) \leq \epsilon$ by the condition that for all circuits C of size T, $|\Pr[C(A) = 1] - \Pr[C(B) = 1]| \leq \epsilon$.

Universal Protocols and Complete channels for OWSC. So far, we considered OWSC protocols for a concrete finite function f and with a concrete level of security. However, in a cryptographic context, one is often interested in a single "universal" protocol in which the sender and the receiver are given a circuit $\hat{f}$, representing a function f, and a security parameter 1^λ as common inputs (in addition to the sender being given an input a for f). More generally, one may consider any computational model – i.e., a representation of the function – instead of circuits (e.g., in the context of information-theoretic security, it will be useful to consider weaker representation models such as branching programs).

In a *polynomial time* universal protocol $\Pi = \langle \mathsf{S}, \mathsf{R} \rangle$, both S and R run in time polynomial in λ. Protocol Π is said to be a universal ϵ-secure (resp., (T, ϵ)-secure) OWSC protocol for $\mathcal{F}$ over $\mathcal{C}$, if for all $\hat{f} \in \mathcal{F}$ with $|\hat{f}| \leq \lambda$, the protocol obtained from Π by fixing the common inputs to $(\hat{f}, 1^\lambda)$ is an $\epsilon(\lambda)$-secure (resp., $(T(\lambda), \epsilon(\lambda))$-secure) OWSC for f over $\mathcal{C}$, where f denotes the function represented by $\hat{f}$.

While $\mathcal{F}$ above can be a narrow class of functions (e.g., string OTs), we shall be particularly interested in the case where it is a general computational model like circuits or branching programs. If a channel $\mathcal{C}$ enables such a universal protocol, we say that $\mathcal{C}$ is *OWSC-complete* for the corresponding computational model. We will distinguish between completeness with inverse-polynomial error and completeness with negligible error, depending on how fast the error vanishes with λ. We will also distinguish between completeness with statistical vs. computational security and between semi-honest vs. malicious senders.

Definition 2 (OWSC-complete channel). *For a computational model $\mathcal{F}$, we say that $\mathcal{C}$ is* OWSC-complete with inverse-polynomial statistical error *if, for every $c > 0$, there is a polynomial-time universal ϵ-secure OWSC protocol for $\mathcal{F}$ over $\mathcal{C}$, where $\epsilon(\lambda) = \mathcal{O}(\frac{1}{\lambda^c})$. We say that $\mathcal{C}$ is OWSC-complete with* negligible statistical *error if there exists a polynomial-time universal ϵ-secure OWSC protocol for $\mathcal{F}$ over $\mathcal{C}$ for some negligible function ϵ.*

We say that $\mathcal{C}$ is computational OWSC-complete with inverse-polynomial statistical error *(resp.,* negligible statistical error*) if, for every $c > 0$, there exists a polynomial-time universal OWSC protocol Π such that for every polynomial $T(\lambda)$, Π is a (T, ϵ)-secure OWSC protocol for $\mathcal{F}$ over $\mathcal{C}$, where $\epsilon(\lambda) = \mathcal{O}(\frac{1}{\lambda^c})$ (resp., ϵ is negligible).*

Completeness as defined above is said to be against malicious parties *if the definition of secure OWSC used is against malicious parties, with the simulator $\mathcal{S}_{\mathsf{S}}$ being polynomial time.*

As discussed above, useful instantiations of $\mathcal{F}$ include circuits, branching programs, and string-ROT. We will assume statistical security against semi-honest parties by default, and will explicitly indicate when security is computational or against malicious parties.

OWSC using ideal obfuscation. Our results, which are information-theoretic in nature, make use of obfuscation as an ideal primitive. An OWSC protocol for f over $\mathcal{C}$ *using ideal obfuscation* is defined similarly to the above except that, in addition to its inputs $\boldsymbol{x}$ for the channel $\mathcal{C}$, the sender specifies a function F (using, say, a circuit $\hat{F}$), to which the receiver is only given (bounded) oracle access. An *honest* receiver can make a single query q to F after observing the outputs $\boldsymbol{y}$ of $\mathcal{C}$, and then compute the output b based on $\boldsymbol{y}$ and $F(q)$. To define security, we extend the syntax of Definition 1 by adding a *query bound* parameter Q. The definition of ϵ-security against the receiver is modified to (Q, ϵ)-security as follows. The simulator $\mathcal{S}_{\mathsf{R}}$ is now an interactive algorithm that interacts with an arbitrary Q-bounded R^*. Given input b (output of f), $\mathcal{S}_{\mathsf{R}}$ first generates and sends to R^* a simulated channel output $\boldsymbol{y}$, and then provides a simulated response for each F-query made by R^*. We require that for *every Q-bounded* R^* and sender input $a \in \mathcal{A}$, the following holds:

$$\Delta\left(\left[\mathcal{S}_{\mathsf{R}}(f(a)) \leftrightarrow \mathsf{R}^*\right], \left[F \leftrightarrow \mathsf{R}^*(\mathcal{C}(x)) \mid (\hat{F}, x) \leftarrow \mathsf{S}(a)\right]\right) \leq \epsilon$$

(Security against a query-bounded receiver)

Here $[\mathcal{S}_{\mathsf{R}}(f(a)) \leftrightarrow \mathsf{R}^*]$ is the ideal-world transcript of the interaction of $\mathcal{S}_{\mathsf{R}}(f(a))$ with R^*, and $[F \leftrightarrow \mathsf{R}^*(\mathcal{C}(x))]$ denotes the real-world transcript of R^* interacting with the channel $\mathcal{C}$ and F, on sender input a. Note that in the latter F denotes the function corresponding to $\hat{F}$ generated by $\mathsf{S}(a)$. The completeness notions in Definition 2 are adapted to the ideal obfuscation setting by requiring that *for every polynomial query bound* $Q(\lambda)$, there is an appropriate ϵ such that Π is a universal (Q, ϵ)-secure OWSC protocol.

2.3 Probability Preliminaries

We state an anti-concentration bound for binomial distribution, which we crucially use in the analysis of all our constructions. The statement of the lemma is quoted verbatim from [52, Theorem 4.6].

Lemma 1 (Anti-concentration). *Let $0 < p < 1$, and $X = X_1 + \ldots + X_n$, where, for each $i \in [n]$, X_i is independently and identically distributed as* Bernoulli(p). *There exists $\Theta_p > 0$ depending only on p (where $\Theta_{\frac{1}{2}} = 1$), such that, for all $0 \leq k \leq n$, we have $\Pr[X = k] \leq \dfrac{\Theta_p}{\sqrt{n}}$.*

Following is a standard concentration inequality required for the analysis of our protocols.

Lemma 2 (Chernoff bound). *Let $0 < p < 1$, and $X_1, \ldots, X_n$ be random variables such that for each $i \in [n]$, X_i is independently and identically distributed as*

Bernoulli(p). *Further, let $X = X_1 + X_2 + \ldots + X_n$. When μ denotes the expected value of X, i.e., $\mu = \mathbb{E}(X) = p \cdot n$,*

$$\text{(i) } \Pr[X \geq (1+\delta)\mu] \leq e^{-\frac{\delta^2}{2+\delta}\mu} \text{ for all } \delta > 0,$$

$$\text{(ii) } \Pr[X \leq (1-\delta)\mu] \leq e^{-\mu\frac{\delta^2}{2}} \text{ for all } \delta \in (0,1).$$

In particular, for all $\eta \in \left(\frac{1}{2}, 1\right)$, for sufficiently large n,

$$\text{(iii) } \Pr[X \in [p(n-n^\eta), p(n+n^\eta)]] \geq 1 - 2e^{-\frac{1}{4p}n^{2\eta-1}} = 1 - \mathrm{negl}(n).$$

Proof: (iii) follows from applying (i) and (ii) by setting $\mu = p \cdot n$ and $\delta = \frac{n^{\eta-1}}{p}$. Note that $\delta \in (0,1)$ for sufficiently large n. □

3 ROT from SEC Using Ideal Obfuscation

In this section, we prove that ROT can be realized using a string erasure channel (with erasure probability $p = 0.5$), assuming ideal obfuscation, following the sketch discussed in Section 1.4. In more detail, we prove:

Theorem 4 (ROT from SEC using ideal obfuscation). *There exists an OWSC protocol for string-ROT over SEC using ideal obfuscation, with inverse-polynomial statistical security against a semi-honest sender and a query-bounded receiver.*

More concretely, for any constant $c > 0$, there exists an OWSC protocol which, for all $\lambda, t \in \mathbb{N}$, realizes t-bit string ROT with ϵ-security against a semi-honest sender and a polynomial query-bounded receiver, using n invocations of ℓ-bit SEC and an ideal obfuscation of a circuit $\hat{F}$, when $\epsilon = \mathcal{O}(\frac{1}{\lambda^c})$, $n = \mathcal{O}(\lambda^{8c})$, $\ell = \omega(\log \lambda)$, and $|\hat{F}| = \mathcal{O}(t + \lambda^{16c})$.

Proof: An OWSC protocol $\langle \mathsf{S}, \mathsf{R} \rangle$ for t-bit string ROT over ℓ-bit SEC is provided in Figure 2. The proof follows the argument sketched in the technical overview (See Section 1.4). We will use the following lemmas to prove the theorem; they are formally proved in the full version of this work [3] using the anti-concentration bound (Lemma 1) and Chernoff bound (Lemma 2).

Lemma 3. *Let $\eta > \frac{1}{2}$, and U, V be arbitrary subsets of $[n]$ such that $|U|, |V| \in [\frac{n-n^\eta}{2}, \frac{n+n^\eta}{2}]$ and $V \subseteq U$. For all $\delta \in (\eta - \frac{1}{2}, \frac{1}{2})$, and for sufficiently large n,*

$$\Pr_{S \leftarrow \binom{[n]}{\sqrt{n}}}\left[|S \cap V| \leq \frac{\sqrt{n}}{2} \middle| |S \cap U| \geq \frac{\sqrt{n}}{2} + n^\delta\right] \leq e^{-\frac{n^\delta}{4}+6}.$$

Lemma 4. *Let $k \in [\frac{n-n^\eta}{2}, \frac{n+n^\eta}{2}]$ and $0 < \delta < \min(\frac{1}{4}, 1-\eta)$. For sufficiently large n such that $\frac{\sqrt{n}}{2}$ is an integer, for any $S \subset [n]$ with $|S| = \sqrt{n}$,*

$$\Pr_{U \leftarrow \binom{[n]}{k}}\left[|S \cap U| \in \left[\frac{\sqrt{n}}{2} - n^\delta, \frac{\sqrt{n}}{2} + n^\delta\right]\right] \leq 2n^{\delta - \frac{1}{4}}e^3.$$

Correctness. For any $\boldsymbol{x} = (x_1, \ldots, x_n)$ such that $x_i \in \{0,1\}^\ell$, the output of $\mathcal{C}_{\mathsf{SEC}}$ on input $\boldsymbol{x}$ is $\mathcal{C}_{\mathsf{SEC}}(\boldsymbol{x}) = \boldsymbol{x}|_U$, where U is a uniformly random subset of $[n]$. Hence, when $|S| = \sqrt{n}$ is an odd number, by symmetry, the event $|U \cap S| \leq \frac{|S|}{2} = \frac{\sqrt{n}}{2}$ occurs with probability $\frac{1}{2}$. By Lemma 2, with all but negligible probability, $|U| \geq \frac{n}{2} - n^{0.51}$. Hence, by a union bound, $F_{S,\boldsymbol{x},a_0,a_1}(\boldsymbol{x}|_U) = (a_0, \perp)$ with probability $\frac{1}{2} - \text{negl}(n)$ and $F_{S,\boldsymbol{x},a_0,a_1}(\boldsymbol{x}|_U) = (\perp, a_1)$ with probability $\frac{1}{2} - \text{negl}(n)$. This proves the correctness of the protocol.

Security. Next, we argue that the protocol presented in Figure 2 achieves sender and receiver privacy. To argue receiver privacy against (even a computationally unbounded) semi-honest sender, we need to show that for all (a_0, a_1), it holds that:

$$\Delta\left(\left(\mathsf{S}(a_0,a_1), \mathcal{C}_{\mathsf{ROT}}(a_0,a_1)\right), \left(\mathsf{S}(a_0,a_1), \mathsf{R}(\mathcal{C}_{\mathsf{SEC}}(\mathsf{S}(a_0,a_1)))\right)\right) \leq \text{negl}(n)$$

Note that the erasures induced by the string erasure channel are independent of the input to the channel. Hence, as we already observed, for any $\boldsymbol{x}$ sent by the sender, the receiver R obtains $\boldsymbol{x}|_U$, where U is a uniformly random subset of $[n]$, independent of $\boldsymbol{x}$ (as well as single query access to $F_{S,\boldsymbol{x},a_0,a_1}$). By definition of F, the output of an honest receiver, viz. $F_{S,\boldsymbol{x},a_0,a_1}(\boldsymbol{x}|_U)$, is only a function of the size of the sets U and $U \cap S$. Thus, whether the receiver outputs $(a_0, \perp)$ or $(\perp, a_1)$ is independent of the view of the sender. Receiver privacy now follows from the fact that the receiver is correct with negligible error.

To argue sender privacy, we need to construct a simulator $\mathcal{S}_{\mathsf{R}} : \mathcal{B} \to \mathcal{Y}^n$ as an interactive algorithm that interacts with an arbitrary Q-bounded R^*. In the sequel, for ease of presentation, for $a_0, a_1 \in \{0,1\}^t$, we will denote $(\perp, a_1)$ by $(1, a_1)$ and $(a_0, \perp)$ by $(0, a_0)$ (i.e., we will use the format (index revealed, message at the revealed index)). Given input (b, a_b) for a random bit b, $\mathcal{S}_{\mathsf{R}}$ first generates and sends to R^* a simulated channel output $\boldsymbol{y}$, and then provides a simulated response for each F-query made by R^*.

Simulator $\mathcal{S}_{\mathsf{R}}(b, a_b)$:

1. Sample $S \leftarrow \binom{[n]}{\sqrt{n}}$.
2. Let $\boldsymbol{x} = (x_1, \ldots, x_n)$, where $x_i \leftarrow \{0,1\}^\ell$ for $i \in [n]$.
3. Sample $U \leftarrow 2^{[n]}$ conditioned on
   (a) $|U \cap S| \geq \frac{\sqrt{n}}{2}$ if $b = 0$,
   (b) $|U \cap S| < \frac{\sqrt{n}}{2}$ if $b = 1$.
4. Output $\boldsymbol{x}|_U$ to R^*.

Next, the simulator answers Q queries by R^* to $F_{S,\boldsymbol{x},a_0,a_1}$ as follows: Upon query $\boldsymbol{y}|_V$, if $\left(|V| \geq \frac{n}{2} - n^{0.51}\right) \wedge \left(\boldsymbol{y}|_V = \boldsymbol{x}|_V\right)$ it outputs (b, a_b). If not, it outputs $\perp$.

We will argue that the statistical distance between the simulated transcript resulting from the interaction of $\mathcal{S}_{\mathsf{R}}(b, a_b)$ with R^* and the real view of R^* on sender input (a_0, a_1) is at most $O(n^{-\frac{1}{8}})$. The distribution on $\boldsymbol{x}|_U$ received by

R^* is identical when it interacts with S or with the simulator $\mathcal{S}_{\mathsf{R}}$. It remains to argue that R^* cannot make a query which $\mathcal{S}_{\mathsf{R}}(b, a_b)$ cannot simulate (except with probability $O(n^{\frac{-1}{8}})$).

First, we argue that,

$$\Pr\left[|U| \in \left[\frac{n}{2} - n^{0.51}, \frac{n}{2} + n^{0.51}\right] \text{ and } |U \cap S| \notin \left[\frac{\sqrt{n}}{2}, \frac{\sqrt{n}}{2} + n^{\frac{1}{8}}\right]\right] \geq 1 - O(n^{\frac{-1}{8}}). \tag{1}$$

To see this, observe that by Lemma 2, with all but negligible probability, $|U| \in \left[\frac{n}{2} - n^{0.51}, \frac{n}{2} + n^{0.51}\right]$. Conditioned on this event, by Lemma 4, probability with which $|U \cap S| \in [\frac{\sqrt{n}}{2} - n^{\frac{1}{8}}, \frac{\sqrt{n}}{2} + n^{\frac{1}{8}}]$ is $O(n^{-\frac{1}{8}})$.

Now we show that in the above event, the simulator answers any query by R^* as in the real world, except with negligible probability. To see this, note that the simulator has access to a_b, and the only cases in which it *cannot* answer correctly is when R^* makes a query to $\hat{F}$ whose output is $(1-b, a_{1-b})$. We argue that this does not happen, except with negligible probability. Consider the following cases:

Case 1: $|U \cap S| < \frac{\sqrt{n}}{2}$. R^* is given $\boldsymbol{x}|_U$ where $F_{S,\boldsymbol{x},a_0,a_1}(\boldsymbol{x}|_U) = (\perp, a_1)$. To recover a_0, R^* must output $(\boldsymbol{y}|_V)$ such that $|V \cap S| \geq \frac{\sqrt{n}}{2}$ and $\boldsymbol{y}|_V = \boldsymbol{x}|_V$. However, since $\forall i \in [n]$, x_i is uniform in $\{0,1\}^\ell$, the probability of guessing even a single string x_i is negligible. Thus in this case, R^* succeeds with probability at most $2^{-\ell}$, which is negligible.

Case 2: $|U \cap S| \geq \frac{\sqrt{n}}{2} + n^{\frac{1}{8}}$. R^* is given $\boldsymbol{x}|_U$ s.t. $F_{S,\boldsymbol{x},a_0,a_1}(\boldsymbol{x}|_U) = (a_0, \perp)$. To recover the other output a_0, R^* must output $(\boldsymbol{y}|_V)$ such that $|V \cap S| < \frac{\sqrt{n}}{2}$ and $\boldsymbol{y}|_V = \boldsymbol{x}|_V$. As before, for any $i \notin U$, it can guess x_i correctly only with negligible probability. By Lemma 3, when $|U| \leq \frac{n}{2} + n^{0.51}$ (this happens with overwhelming probability by Lemma 2), for all $V \subseteq U$ such that $|U| \geq \frac{n}{2} - n^{0.51}$, the probability that $|V \cap S| < \frac{\sqrt{n}}{2}$ is negligible. Thus in this case also, R^* succeeds in coming up with a query that makes $F_{S,\boldsymbol{x},a_0,a_1}$ output $(1-b, a_{1-b})$ with at most negligible probability.

Thus, by taking a union bound, we can conclude that the simulator can answer the queries of a poly(λ)-bounded R^* except with negligible probability.

Finally, we show the bound on the circuit $|\hat{F}|$ in the theorem statement. Each position of the input $\boldsymbol{y}$ is encoded using $\ell + 1$ bits, with say, the first bit used as a flag denoting if it is $\perp$. Then a circuit of size $\mathcal{O}(n^2)$ on the n flag bits suffices for computing the two threshold conditions on $|V|$ and $|V \cap S|$ used in F, and a circuit of size $\mathcal{O}(n\ell)$ suffices to compute the equality condition $\boldsymbol{x}|_V = \boldsymbol{y}|_V$. The output is encoded, say, as (b, a_b) for $b \in \{0,1\}$ with an additional flag to indicate if it is $(\perp, \perp)$. Each of these $t+2$ output bits can be computed as a function of two bits from a_0 and a_1 and the three condition bits computed above. So overall $\hat{F}$ is of size $\mathcal{O}(t + n^2 + n\ell)$). The theorem now follows by setting $n = \lambda^{8c}$. This concludes the proof. □

ROT from String Erasure Channel

Function F

Parameters: $S \subseteq [n]$ with $|S| = \sqrt{n}$, $\boldsymbol{x} = (x_1, \ldots, x_n)$, where $x_i \in \{0,1\}^\ell$ for $i \in [n]$, $a_0, a_1 \in \{0,1\}^t$.
Input: $\boldsymbol{y} = (y_1, \ldots, y_n)$, where, for all $i \in [n]$, $y_i \in \{0,1\}^\ell$ or $y_i = \perp$. We write $\boldsymbol{y}|_V$ to indicate that $V = \{i | y_i \neq \perp\}$.

$$F_{S,\boldsymbol{x}}\left(\boldsymbol{y}|_V, a_0, a_1\right)$$
$$= \begin{cases} (\perp, a_1), \text{ if } \left(|V| \geq \frac{n}{2} - n^{0.51}\right) \wedge \left(\boldsymbol{y}|_V = \boldsymbol{x}|_V\right) \wedge \left(|V \cap S| < \frac{\sqrt{n}}{2}\right), \\ (a_0, \perp), \text{ if } \left(|V| \geq \frac{n}{2} - n^{0.51}\right) \wedge \left(\boldsymbol{y}|_V = \boldsymbol{x}|_V\right) \wedge \left(|V \cap S| \geq \frac{\sqrt{n}}{2}\right), \\ (\perp, \perp), \text{ otherwise.} \end{cases} \tag{2}$$

We write $F_{S,\boldsymbol{x},a_0,a_1}(\cdot)$ to denote $F_{S,\boldsymbol{x}}(\cdot, a_0, a_1)$.

Sender $\mathsf{S}(a_0, a_1)$
(inputs $a_0, a_1 \in \{0,1\}^t$)

1. Sample $S \leftarrow \binom{[n]}{\sqrt{n}}$
2. Let $\boldsymbol{x} = (x_1, \ldots, x_n)$, where each $x_i \leftarrow \{0,1\}^\ell$.
3. Output $(\hat{F}, \boldsymbol{x})$ where $\hat{F}$ is a circuit for $F_{S,\boldsymbol{x},a_0,a_1}$.

$\boldsymbol{x}$ sent over $\mathcal{C}_{\mathsf{SEC}}$

$\xrightarrow{\hspace{2cm}}$

$\hat{F}$ given as oracle to R

Receiver R

(with 1-query access to F)

1. Receive $\mathcal{C}_{\mathsf{SEC}}(\boldsymbol{x}) = \boldsymbol{x}|_U$ where $U \leftarrow 2^{[n]}$.
2. Output $F\left(\boldsymbol{x}|_U\right)$.

Fig. 2: The OWSC protocol $\langle \mathsf{S}, \mathsf{R} \rangle$ for realizing ROT over the string erasure channel assuming ideal obfuscation.

4 Completeness of BEC and BSC Using Ideal Obfuscation

In this section, we show that the binary erasure channel and the binary symmetric channel are (each) complete, assuming ideal obfuscation. In Section 4.1, we construct the string erasure channel from the binary erasure channel and from the binary symmetric channel. We then appeal to a composition theorem 5 to argue that BEC/BSC can be used to construct ROT. Finally, in Section 4.2 we discuss completeness of BEC/BSC for general sender-receiver functionalities.

4.1 String Erasure Channel from BEC/BSC

In this section, we provide constructions of string erasure channel from binary erasure channel and from binary symmetric channel using ideal obfuscation. [7]

[7] We remark that OWSC of SEC over BEC with inverse polynomial statistical security exists without using ideal obfuscation. Such a protocol can be obtained following the ideas in [2], where an OWSC protocol was constructed for string-ROT over bit-

We first define a quantity that will be used in the construction and analysis of the following protocols. Let $0 < p < 1$, and $X_1, \ldots, X_n$ be random variables such that for each $i \in [n]$, X_i is independently and identically distributed as Bernoulli(p). Further, let $X = X_1 + X_2 + \ldots + X_n$. Define

$$\mathsf{Centre}(p, n) = \max \left\{ t \in [n] : \Pr\left[X < t\right] \leq \frac{1}{2} \right\}.$$

Claim 1. *For $\Theta_p > 0$ that depends only on p (as described in Lemma 1),*

$$\Pr\left[X \leq \mathsf{Centre}(p, n)\right] \in \left(\frac{1}{2}, \frac{1}{2} + \frac{\Theta_p}{\sqrt{n}} \right].$$

Proof: $\Pr\left[X = \mathsf{Centre}(p, n)\right] \leq \frac{\Theta_p}{\sqrt{n}}$ by the anti-concentration bound in Lemma 1. Claim follows from this and the definition of $\mathsf{Centre}(p, n)$. □

We now proceed to formally state and prove the first main result in this section.

Lemma 5 (SEC from BEC using ideal obfuscation). *There exists an OWSC protocol for SEC over BEC using ideal obfuscation, with inverse-polynomial statistical security against a semi-honest sender and a query-bounded receiver.*

More concretely, for all $p \in (0, 1)$ and $c > 0$, there exists an OWSC protocol which, for all $\lambda, \ell \in \mathbb{N}$, realizes ℓ-bit SEC with ϵ-security against a semi-honest sender and a polynomial query-bounded receiver, using n invocations of the BEC with erasure probability p and an ideal obfuscation of a circuit $\hat{F}$, when $\epsilon = \mathcal{O}(\frac{1}{\lambda^c})$, $n = \mathcal{O}(\lambda^{4c})$, and $|\hat{F}| = \mathcal{O}(\ell \cdot \lambda^{8c})$.

Proof: The OWSC protocol $\langle \mathsf{S}, \mathsf{R} \rangle$ for an ℓ-bit SEC over BEC with erasure probability $p \in (0, 1)$ is provided in Figure 3. We argue correctness and security below.

Correctness. Since $\mathcal{C}^p_{\mathsf{BEC}}$ erases each bit in $\boldsymbol{x}$ with probability p independently, the number of non-erasures $|U|$ is distributed according to Binomial$(n, 1 - p)$. Hence, by Claim 1, the probability with which receiver reports an erasure is

$$\Pr\left[|U| \leq \mathsf{Centre}(1 - p, n)\right] \in \left(\frac{1}{2}, \frac{1}{2} + \frac{\Theta_{1-p}}{\sqrt{n}} \right).$$

Thus, the input string a is output with probability $\frac{1}{2}$ (with inverse polynomial bias), which proves correctness of SEC.

Security. We first prove the statistical security against a computationally unbounded semi-honest sender by arguing that for all $a \in \{0, 1\}^\ell$

$$\Delta\left((\mathsf{S}(a), \mathcal{C}_{\mathsf{SEC}}(a)), (\mathsf{S}(a), \mathsf{R}(\mathcal{C}^p_{\mathsf{BEC}}(\mathsf{S}(a)))) \right) \leq \frac{\Theta_{1-p}}{\sqrt{n}}.$$

ROT with inverse polynomial statistical security. We do not explore the possibility of building such an OWSC protocol for SEC over BSC. Instead, we stick to constructions using ideal obfuscation since our next step towards realizing OWSC of ROT over BEC/BEC, i.e. of constructing OWSC of ROT over SEC, anyway uses ideal obfuscation.

String Erasure Channel from Binary Erasure Channel

Function F

- *Parameters:* $\boldsymbol{x} \in \{0,1\}^n$ and $a \in \{0,1\}^\ell$.
- *Input:* $\boldsymbol{y}|_V$ such that $V \subseteq [n]$ and $y_i \in \{0,1\}^n$ for all $i \in V$.

$$F_{\boldsymbol{x}}(\boldsymbol{y}|_V, a) = \begin{cases} a, & \text{if } (\boldsymbol{y}|_V = \boldsymbol{x}|_V) \wedge (|V| > \mathsf{Centre}(1-p,n)), \\ \perp, & \text{otherwise.} \end{cases}$$

We write $F_{\boldsymbol{x},a}(\cdot)$ to denote $F_{\boldsymbol{x}}(\cdot, a)$.

Sender $\mathsf{S}(a \in \{0,1\}^\ell)$		Receiver R
1. $\boldsymbol{x} \leftarrow \{0,1\}^n$. 2. Output $(\hat{F}, \boldsymbol{x})$ where $\hat{F}$ is a circuit for $F_{\boldsymbol{x},a}$.	$\boldsymbol{x}$ sent over $\mathcal{C}^p_{\mathsf{BEC}}$ $\xrightarrow{\hspace{2em}}$ $\hat{F}$ given as oracle to R	(with 1-query access to F) 1. Receive $\boldsymbol{x}|_U = \mathcal{C}^p_{\mathsf{BEC}}(\boldsymbol{x})$. 2. Output $F(\boldsymbol{x}|_U)$.

Fig. 3: Protocol $\langle \mathsf{S}, \mathsf{R} \rangle$ for realizing ℓ-bit string-Erasure Channel using n invocations of a binary erasure channel with erasure probability $p \in (0,1)$.

The erasure pattern over n uses of the channel is independent of the sender's input $\boldsymbol{x}$. Consequently, whether the receiver outputs a or $\perp$ is independent of the view of the sender. The bound on the statistical distance now follows from the correctness of the protocol.

To argue security against the receiver, we need to construct a simulator $\mathcal{S}_{\mathsf{R}} : \mathcal{B} \to \mathcal{Y}^n$ as an interactive algorithm that interacts with an arbitrary poly(n)-bounded R^*. Given input $a \in \{0,1\}^\ell \cup \{\perp\}$, $\mathcal{S}_{\mathsf{R}}$ first generates and sends to R^* a simulated channel output $\boldsymbol{y}$, and then provides a simulated response for each $\hat{F}$-query made by R^*.

Simulator $\mathcal{S}_{\mathsf{R}}(a)$: Simulator constructs $\boldsymbol{y}$ as follows:

1. Sample $\boldsymbol{x} \leftarrow \{0,1\}^n$
2. Sample erasure pattern $[n] \setminus U$ (as generated on n independent uses of $\mathcal{C}^p_{\mathsf{BEC}}$) under the conditioning $|U| > \mathsf{Centre}(1-p,n)$ if $a \neq \perp$ and under the conditioning $|U| \leq \mathsf{Centre}(1-p,n)$ if $a = \perp$.
3. Output $\boldsymbol{x}|_U$ to R^*.

For Q queries by R^* to F, the simulator replies to a query $\boldsymbol{y}|_V$ follows:

- *Case 1:* If $|U| > \mathsf{Centre}(1-p,n)$, simulator outputs $F_{\boldsymbol{x},a}(\boldsymbol{y}|_V)$ as it has access to x, a, and U.
- *Case 2:* If $|U| \leq \mathsf{Centre}(1-p,n)$, simulator simply outputs $\perp$.

Since the distribution on $\boldsymbol{x}|_U$ received by R^* is identical when it interacts with S or with the simulator $\mathcal{S}_{\mathsf{R}}$, it is sufficient to argue that R^* cannot make any query which $\mathcal{S}_{\mathsf{R}}$ cannot correctly respond to, except with probability $O(n^{\frac{-1}{4}})$. In case 1, when $|U| > \mathsf{Centre}(1-p,n)$, the simulator/predictor can honestly compute $F_{\boldsymbol{x},a}(\boldsymbol{y}|_V)$ and the query is answered correctly. In case 2, the simulator/predictor fails if R^* makes a query $\boldsymbol{y}|_V$ such that $F_{\boldsymbol{x},a}(\boldsymbol{y}|_V) = a$. Define the set

$$\mathsf{Bad} = \left\{U : |U| \in \left[\mathsf{Centre}(1-p,n) - n^{\delta}, \mathsf{Centre}(1-p,n)\right)\right\}.$$

Since $[n] \setminus U$ is the erasure pattern during n independent uses of $\mathcal{C}^p_{\mathsf{BEC}}$, $|U|$ is distributed according to the Binomial$(n, 1-p)$ distribution independent of $\boldsymbol{x}$. Hence, for all $\boldsymbol{x} \in \{0,1\}^n$, by applying the anti-concentration bound in Lemma 1 together with a union bound,

$$\Pr[\mathsf{Bad}] = \Pr_U\left[|U| \in \left[\mathsf{Centre}(1-p,n) - n^{\delta}, \mathsf{Centre}(1-p,n)\right)\right] \leq \frac{\Theta_{1-p}}{\sqrt{n}} \cdot n^{\delta}.$$

We will show that, except under the event Bad (which happens with probability at most $\Theta_{1-p} \cdot n^{-\frac{1}{4}}$, when $\delta = \frac{1}{4}$), R^* outputs a query $\boldsymbol{y}|_V$ such that $F_{\boldsymbol{x},a}(\boldsymbol{y}|_V) = a$ with negligible probability. Taking a union bound over poly(n) queries, we achieve the desired security condition.

It suffices to show that for all $a \in \{0,1\}^{\ell}$ and computationally unbounded algorithms Adv that take $\boldsymbol{x}|_U$ as input,

$$\Pr_{\boldsymbol{x}\leftarrow\{0,1\}^n, U}\left[F_{\boldsymbol{x}}(\boldsymbol{y}|_V, a) \neq \bot | \neg\mathsf{Bad}, \boldsymbol{y}|_V = \mathsf{Adv}(\boldsymbol{x}|_U), F_{\boldsymbol{x}}(\boldsymbol{y}|_V, a) = \bot\right] = \text{negl}(n). \tag{3}$$

The event '$\neg\mathsf{Bad}$ and $F_{\boldsymbol{x}}(\boldsymbol{y}|_V, a) = \bot$' is the same as '$|U| \leq \mathsf{Centre}(1-p,n) - n^{\delta}$'. Hence,

$$\begin{aligned}
&\Pr_{\boldsymbol{x}\leftarrow\{0,1\}^n, U}\left[F_{\boldsymbol{x}}(\boldsymbol{y}|_V, a) \neq \bot | \neg\mathsf{Bad}, \boldsymbol{y}|_V = \mathsf{Adv}(\boldsymbol{x}|_U), F_{\boldsymbol{x}}(\boldsymbol{y}|_V, a) = \bot\right] \\
&\quad \leq \Pr_{\boldsymbol{x}\leftarrow\{0,1\}^n, U}\Big[|V \setminus U| \geq n^{\delta} \text{ and } \boldsymbol{y}|_{V\setminus U} = \boldsymbol{x}|_{V\setminus U}\Big| \\
&\qquad\qquad |U| \leq \mathsf{Centre}(1-p,n) - n^{\delta}, \boldsymbol{y}|_V = \mathsf{Adv}(\boldsymbol{x}|_U)\Big] \\
&\quad \leq \Pr_{x_i\leftarrow\{0,1\}, \forall i\in[n^{\delta}]}\left[y_i = x_i, \forall i \in [n^{\delta}]\right] = 2^{-n^{\delta}}.
\end{aligned}$$

The function F can be realized using $\ell + 1$ Boolean circuits (to compute each bit of the output encoded with one extra bit to report $\bot$). When the input is appropriately encoded, the Boolean circuits need to compute a thresholding function on n-bit inputs (quadratic blow-up), and equality check for $\mathcal{O}(n)$-bit inputs (linear blow-up). Hence, the size of $\hat{F}$ is $\mathcal{O}(\ell \cdot n^2)$. The lemma now follows by setting $n = \lambda^4 c$. This concludes the proof. □

We would like to remark that the above construction can also be used to realize string erasure channel with erasure probability $\frac{1}{2}$ from another string erasure channel (possibly of different string length) with arbitrary probability of erasure (ℓ'-bit $\mathcal{C}^p_{\mathsf{SEC}}$ for $0 < p < 1$). We can then put this result together with the result in Theorem 4 to show that ROT can be realized from general SEC (See Section 4.2).

Using a similar construction we can realize string erasure channel from binary symmetric channel using ideal obfuscation. Formally, we prove the following lemma:

Lemma 6 (SEC from BSC using ideal obfuscation). *For $p \in (0, \frac{1}{2})$, there exists an OWSC protocol for SEC over BSC with crossover probability p using ideal obfuscation, with inverse-polynomial statistical security against a semi-honest sender and a query-bounded receiver.*

More concretely, for all $p \in (0, \frac{1}{2})$ and $c > 0$, there exists an OWSC protocol which, for all $\lambda, \ell \in \mathbb{N}$, realizes ℓ-bit SEC with ϵ-security against a semi-honest sender and a polynomial query-bounded receiver, using n invocations of the BSC with crossover probability p and an ideal obfuscation of a circuit $\hat{F}$, when $\epsilon = \mathcal{O}(\frac{1}{\lambda^c})$, $n = \mathcal{O}(\lambda^{4c})$, and $|\hat{F}| = \mathcal{O}(\ell \cdot \lambda^{8c})$.

Proof: The OWSC protocol $\langle \mathsf{R}, \mathsf{S} \rangle$ for SEC over BSC is provided in Figure 4. We argue correctness and security below.

Correctness. Since $\mathcal{C}^p_{\mathsf{BSC}}$ flips each bit in $\boldsymbol{x}$ with probability p independently, $|\boldsymbol{x} \oplus \boldsymbol{y}|$ is distributed according to Binomial(n, p). Hence, by Claim 1,

$$\Pr\left[|\boldsymbol{x} \oplus \boldsymbol{y}| \leq \mathsf{Centre}(p, n)\right] \in \left(\frac{1}{2}, \frac{1}{2} + \Theta_p \cdot n^{-\frac{1}{2}}\right).$$

Thus, the input string a is output with probability $\frac{1}{2}$ (with inverse polynomial bias), which proves correctness of SEC.

Security. We first argue statistical security against a computationally unbounded semi-honest sender by showing that for all $a \in \{0,1\}^\ell$

$$\Delta\left((\mathsf{S}(a), \mathcal{C}_{\mathsf{SEC}}(a)), (\mathsf{S}(a), \mathsf{R}(\mathcal{C}^p_{\mathsf{BSC}}(\mathsf{S}(a)))\right) \leq \Theta_p \cdot n^{-\frac{1}{2}}.$$

Observe that the noise added by the BSC is independent of the sender's input $\boldsymbol{x}$. Consequently, whether the receiver outputs a or $\perp$ is independent of the view of the sender. The bound on the statistical distance now follows from the correctness of the protocol.

To argue security against the receiver, we need to construct a simulator $\mathcal{S}_{\mathsf{R}} : \mathcal{B} \to \mathcal{Y}^n$ as an interactive algorithm that interacts with an arbitrary poly(n)-bounded R^*. Given input $a \in \{0,1\}^\ell \cup \{\perp\}$, $\mathcal{S}_{\mathsf{R}}$ first generates and sends to R^* a simulated channel output $\boldsymbol{y}$, and then provides a simulated response for each F-query made by R^*.

Simulator $\mathcal{S}_{\mathsf{R}}(a)$: Simulator constructs $\boldsymbol{y}$ as follows:

1. Sample $\boldsymbol{x} \leftarrow \{0,1\}^n$.

String Erasure Channel from Binary Symmetric Channel

Function F

- *Parameters:* $\boldsymbol{x} \in \{0,1\}^n$ and $a \in \{0,1\}^\ell$.
- *Input:* $\boldsymbol{y} \in \{0,1\}^n$.

$$F_{\boldsymbol{x}}(\boldsymbol{y}, a) = \begin{cases} a, & \text{if } |\boldsymbol{x} \oplus \boldsymbol{y}| \le \mathsf{Centre}(p,n), \\ \bot, & \text{otherwise.} \end{cases}$$

We write $F_{\boldsymbol{x},a}(\cdot)$ to denote $F_{\boldsymbol{x}}(\cdot, a)$.

Sender $\mathsf{S}(a \in \{0,1\}^\ell)$

1. $\boldsymbol{x} \leftarrow \{0,1\}^n$.
2. Output $(\hat{F}, \boldsymbol{x})$ where $\hat{F}$ is a circuit for $F_{\boldsymbol{x},a}$.
   ($\boldsymbol{x}$ will be sent to R over $\mathcal{C}^p_{\mathsf{BSC}}$, and $\hat{F}$ will be used as the oracle for R, below.)

Receiver R with 1 query oracle access to F

1. Receive $\boldsymbol{y} = \mathcal{C}^p_{\mathsf{BSC}}(\boldsymbol{x})$.
2. Output $F(\boldsymbol{y})$.

Fig. 4: The protocol $\langle \mathsf{S}, \mathsf{R} \rangle$ for realizing ℓ-bit String-Erasure Channel using n invocations of a binary symmetric channel with crossover probability p.

2. Sample $\boldsymbol{y} = \mathcal{C}^p_{\mathsf{BSC}}(\boldsymbol{x})$ conditioned on $|\boldsymbol{x} \oplus \boldsymbol{y}| \le \mathsf{Centre}(p,n)$ if $a \neq \bot$ and $|\boldsymbol{x} \oplus \boldsymbol{y}| > \mathsf{Centre}(p,n)$ if $a = \bot$.
3. Output $\boldsymbol{y}$ to R^*.

For Q queries by R^* to $\hat{F}$, the simulator replies to a query $\hat{\boldsymbol{y}}$ follows:

- *Case 1:* If $|\boldsymbol{x} \oplus \boldsymbol{y}| \le \mathsf{Centre}(p,n)$, simulator outputs $F_{\boldsymbol{x},a}(\hat{\boldsymbol{y}})$ as it has access to x and a.
- *Case 2:* If $|\boldsymbol{x} \oplus \boldsymbol{y}| > \mathsf{Centre}(p,n)$, simulator simply outputs $\bot$.

Since the distribution on $\boldsymbol{x}|_U$ received by R^* is identical when it interacts with S or with the simulator $\mathcal{S}_{\mathsf{R}}$, it is sufficient to argue that R^* cannot make any query which $\mathcal{S}_{\mathsf{R}}$ cannot correctly respond to (except with probability $O(n^{\frac{-1}{4}})$). In case 1, when $|U| > \mathsf{Centre}(1-p,n)$, the simulator/predictor can honestly compute $F_{\boldsymbol{x},a}(\boldsymbol{y}|_V)$ and the query is answered correctly. In case 2, the simulator/predictor fails if R^* makes a query $\boldsymbol{y}|_V$ such that $F_{\boldsymbol{x},a}(\boldsymbol{y}|_V) = a$. Define the set

$$\mathsf{Bad} = \left\{ (\boldsymbol{x}, \boldsymbol{y}) \in \{0,1\}^{2n} : |\boldsymbol{x} \oplus \boldsymbol{y}| \in \left(\mathsf{Centre}(p,n), \mathsf{Centre}(p,n) + n^\delta \right] \right\}.$$

In the sequel, we will denote $\mathsf{Centre}(p,n)$ by t. When $\boldsymbol{x} \leftarrow \{0,1\}^n$ and $\boldsymbol{y} = \mathcal{C}^p_{\mathsf{BSC}}(\boldsymbol{x})$, $|\boldsymbol{x} \oplus \boldsymbol{y}|$ is the number of bits noise added by $\mathcal{C}^p_{\mathsf{BSC}}$. Hence, it is distributed according to the Binomial(n,p) distribution. By applying the anti-concentration bound in Lemma 1 together with a union bound, we get

$$\Pr_{(\boldsymbol{x}\leftarrow\{0,1\}^n, \boldsymbol{y}=\mathcal{C}^p_{\mathsf{BSC}}(\boldsymbol{x})}[\mathsf{Bad}] = \Pr_{\boldsymbol{x}\leftarrow\{0,1\}^n, \boldsymbol{y}=\mathcal{C}^p_{\mathsf{BSC}}(\boldsymbol{x})}\left[|\boldsymbol{x}\oplus\boldsymbol{y}| \in \left(t, t+n^\delta\right]\right] \leq \Theta_p \cdot n^{\delta-\frac{1}{2}}.$$

We will show that, except under the event Bad (which happens with probability at most $\Theta_{1-p}\cdot n^{-\frac{1}{4}}$, when $\delta = \frac{1}{4}$), R^* outputs a query $\boldsymbol{y}|_V$ such that $F_{\boldsymbol{x},a}(\boldsymbol{y}|_V) = a$ with negligible probability. Taking a union bound over poly(n) queries, we achieve the desired security condition.

It suffices to show that for all $a \in \{0,1\}^\ell$ and computationally unbounded algorithms Adv that take $\boldsymbol{y}$ as input,

$$\Pr_{\boldsymbol{x}\leftarrow\{0,1\}^n, \boldsymbol{y}=\mathcal{C}^p_{\mathsf{BSC}}(\boldsymbol{x})}\left[F_{\boldsymbol{x},a}(\hat{\boldsymbol{y}}) \neq \perp | \neg\mathsf{Bad}, F_{\boldsymbol{x},a}(\hat{\boldsymbol{y}}) = \perp, \hat{\boldsymbol{y}} = \mathsf{Adv}(\boldsymbol{y})\right] = \operatorname{negl}(n). \quad (4)$$

The event '$\neg\mathsf{Bad}$ and $F_{\boldsymbol{x},a}(\hat{\boldsymbol{y}}) = \perp$' is the same as '$|\boldsymbol{x}\oplus\boldsymbol{y}| \geq \mathsf{Centre}(p,n) + n^\delta$'. We complete the argument by appealing to the following claim.

Claim 2. *For any computationally unbounded algorithm A, for sufficiently large values of n,*

$$\Pr_{\boldsymbol{x}\leftarrow\{0,1\}^n, \boldsymbol{y}=\mathcal{C}^p_{\mathsf{BSC}}(\boldsymbol{x})}\left[F_{\boldsymbol{x}}(\hat{\boldsymbol{y}}, a) \neq \perp \middle| |\boldsymbol{x}\oplus\boldsymbol{y}| \geq \mathsf{Centre}(p,n) + n^\delta, \hat{\boldsymbol{y}} \leftarrow A(\boldsymbol{y})\right]$$
$$\leq 3e^{-\frac{(1-2p)^2}{4}n^\delta}.$$

Proof: Let $t = \mathsf{Centre}(p,n)$ and $V = \{i \in [n] : \hat{y}_i \oplus y_i = 1\}$. For $\boldsymbol{x} \leftarrow \{0,1\}^n, \boldsymbol{y} = \mathcal{C}^p_{\mathsf{BSC}}(\boldsymbol{x})$, and $\hat{\boldsymbol{y}} \leftarrow A(\boldsymbol{y})$,

$$\begin{aligned}
&\Pr\left[F_{\boldsymbol{x},a}(\hat{\boldsymbol{y}}) \neq \perp \middle| |\boldsymbol{x}\oplus\boldsymbol{y}| \geq t + n^\delta\right] \\
&\quad= \Pr\left[|\boldsymbol{x}\oplus\hat{\boldsymbol{y}}| \leq t \middle| |\boldsymbol{x}\oplus\boldsymbol{y}| \geq t + n^\delta\right] \\
&\quad= \Pr\left[|(\boldsymbol{x}\oplus\boldsymbol{y})\oplus(\boldsymbol{y}\oplus\hat{\boldsymbol{y}})| \leq t \middle| |\boldsymbol{x}\oplus\boldsymbol{y}| \geq t + n^\delta\right] \\
&\quad\leq \Pr\left[\sum_{i\in V}(x_i \oplus y_i) - \left(|V| - \sum_{i\in V}(x_i\oplus y_i)\right) \geq n^\delta \middle| |\boldsymbol{x}\oplus\boldsymbol{y}| \geq t + n^\delta\right] \\
&\quad= \Pr\left[\sum_{i\in V}(x_i\oplus y_i) \geq \frac{|V| + n^\delta}{2} \middle| |\boldsymbol{x}\oplus\boldsymbol{y}| \geq t + n^\delta\right].
\end{aligned}$$

Since $\boldsymbol{x}$ is uniformly distributed, $\boldsymbol{x}\oplus\boldsymbol{y}$ is independent of $\boldsymbol{y}$ and, therefore, independent of $(\boldsymbol{y}, \hat{\boldsymbol{y}}, V)$. Conditioned on V (and suppressing this conditioning in the steps below), we have, for all $V \subseteq [n]$,

$$\Pr\left[\sum_{i\in V}(x_i\oplus y_i) \geq \frac{|V| + n^\delta}{2}, |\boldsymbol{x}\oplus\boldsymbol{y}| \geq t + n^\delta\right] \leq \Pr\left[\sum_{i\in V}(x_i\oplus y_i) \geq \frac{|V| + n^\delta}{2}\right],$$

where $x_i \oplus y_i$, $i \in V$, are independent and identically distributed with distribution Bernoulli(p). This probability is clearly zero if $|V| < n^\delta$. For $|V| \geq n^\delta$, by the Chernoff bound in Lemma 2,

$$\begin{aligned}\Pr\left[\sum_{i \in V}(x_i \oplus y_i) \geq \frac{|V| + n^\delta}{2}\right] &\leq \Pr\left[\sum_{i \in V}(x_i \oplus y_i) \geq \frac{|V|}{2}\right] \\ &= \Pr\left[\sum_{i \in V}(x_i \oplus y_i) \geq \left(1 + \left(\frac{1}{2p} - 1\right)\right) p \cdot |V|\right] \\ &\leq e^{-\frac{\left(\frac{1}{2p}-1\right)^2}{\frac{1}{2p}+1} p \cdot |V|} \leq e^{-\frac{(1-2p)^2}{4} n^\delta}.\end{aligned}$$

Moreover, since $|\boldsymbol{x} \oplus \boldsymbol{y}|$ is Binomial(n, p), we have $\Pr[|\boldsymbol{x} \oplus \boldsymbol{y}| < \mathsf{Centre}(p, n)] < \frac{1}{2}$, which along with the anti-concentration bound in Lemma 1, gives

$$\Pr\left[|\boldsymbol{x} \oplus \boldsymbol{y}| \geq t + n^\delta\right] \geq \frac{1}{2} - \frac{\Theta_p}{\sqrt{n}} \cdot (1 + n^\delta) \geq \frac{1}{3},$$

for sufficiently large n since $\delta < \frac{1}{2}$. This proves the claim. □

The function F can be realized using $\ell + 1$ Boolean circuits (to compute each bit of the output encoded with one extra bit to report $\perp$). When the input is appropriately encoded, the Boolean circuits need to compute a XOR and thresholding function on n-bit input (quadratic blow-up). Hence, the size of $\hat{F}$ is $\mathcal{O}(\ell \cdot n^2)$. The lemma now follows by setting $n = \lambda^4 c$. This concludes the proof. □

4.2 Completeness of BEC/BSC Using Ideal Obfuscation

We can put together the results in Section 4.1 (that the string erasure channel (SEC) can be constructed using the binary erasure and binary symmetric channels, using ideal obfuscation) with the result from Section 3 (that ROT can be constructed using SEC, using ideal obfuscation), to obtain the following.

Theorem 5 (ROT from BEC or BSC using ideal obfuscation). *There exists an OWSC protocol $\Pi_{\text{ROT}}^{\text{BEC}}$ (respectively, $\Pi_{\text{ROT}}^{\text{BSC}}$) for ROT over BEC (respectively, BSC) using ideal obfuscation, with inverse-polynomial statistical security against a semi-honest sender and a polynomial query-bounded receiver.*

Proof: We shall compose the OWSC protocol for ROT over SEC from Theorem 4 with the protocol from Lemma 5 (respectively, from Lemma 6). For this, we need to argue that OWSC protocols compose. The security definition of OWSC (Definition 1) could be seen as a specialization of the UC security notion, to the one-way communication setting, and a semi-honest sender, in a $(\mathcal{C}, \mathcal{B})$-hybrid model, where $\mathcal{C}$ is the channel, and $\mathcal{B}$ is a functionality that takes a circuit from the sender and provides the receiver with black-box access to it (for a bounded number of queries). To see this is indeed the case, note that when the sender is (passively) corrupt, a simulator for passive-security should merely forward the sender's input a to the functionality, resulting in the receiver obtaining $f(a)$; hence the environment's views in the ideal and real executions (in addition to a, which is universally quantified over) are simply $(\mathsf{S}(a), \mathsf{R}(\mathcal{C}(\mathsf{S}(a))))$ and $(\mathsf{S}(a), f(a))$.

When the receiver is (possibly actively) corrupt, its view includes an output from the channel $\mathcal{C}$ and its interaction with the oracle $\mathcal{B}$; the security definition for OWSC in this case is the same as for UC security, by treating the receiver as the environment (the input a is part of the corrupt receiver's view in the OWSC definition, due to the universal quantifier over a).

Before we can apply composition, note that we have a mixed corruption model with *fixed roles*. That is, the party playing the sender in all of the protocols or functionalities is the same (i.e., corrupting one corrupts all), and similarly for the receiver. Hence we have only two non-trivial corruption scenarios: all the senders are passively corrupt, or all the receiver's are actively corrupt. In either case, the protocol for ROT from SEC, as well as the protocol for SEC from BEC (or BSC) satisfies the corresponding security guarantee. We note that in a corruption scenario, if UC or passive security holds for each protocol instance, then, it holds for the composed protocol for the same corruption scenario (this is implicit in the proof of composition theorems for static adversaries, which fixes a corruption scenario and derives a simulator for the composed protocol from individual simulators for the constituent protocols).

Finally, note that in the composed secure protocol, there are several instances of $\mathcal{B}$ invoked by the sender (and each one accessed a bounded number of times by the receiver). These multiple instances, with programs, say $F_1, \cdots, F_n$ can be replaced by a single instance of $\mathcal{B}$ to which the sender inputs a combined program F such that $F(i,x) = F_i(x)$. Thus we obtain an OWSC protocol using ideal obfuscation for ROT from either BEC or BSC. □

We are now ready to show that the binary erasure channel and the binary symmetric channel are complete, using ideal obfuscation. To generalize the above construction to arbitrary functionalities, we rely on a previous result by Garg et al. [25], which showed that ROT is complete for arbitrary finite functionalities even for the case of malicious parties, with statistical security. Combined with our reductions from ROT to BSC and BEC, we get a similar completeness result for BEC/BSC with inverse-polynomial error.

In more detail, we claim that:

Theorem 6 (Completeness of BEC/BSC using ideal obfuscation: semi-honest sender). *BEC and BSC are (each) complete for OWSC using ideal obfuscation, with inverse-polynomial statistical security against a semi-honest sender and a polynomial query-bounded receiver.*

Proof: [Proof sketch] Analogously to [2], let us first consider the setting of semi-honest parties. In this case, we may combine the reduction from ROT to BEC/BSC with Yao's garbled circuits [58] as follows. Given a randomized sender receiver functionality $F(a;r)$, define a deterministic (two-way) functionality $\tilde{F}$ that takes (a, r_1) from the sender and r_2 from the receiver, and outputs

$F(a; r_1 \oplus r_2)$ to the receiver. Using Yao's protocol to securely evaluate $\tilde{F}$ with uniformly random choices of r_1, r_2, we get a secure reduction of F to OT where the receiver's inputs are random. We may now replace the random choices of the receiver by leveraging a ROT channel, and then apply the reduction from ROT to BEC/BSC.

The above compiler makes use of Yao's garbled circuits, which assume the existence of one way functions. In the setting of ideal obfuscation, we may obtain an unconditional result as follows. First, note that for the case of branching programs, we may use information theoretic garbled circuits [41, 23, 34]. For the case of circuits, we use a result of Goyal et al. [32] which implies unconditionally secure garbled circuits from ideal obfuscation. In more detail, [32] show how to obtain unconditionally secure computation from hardware tokens. Our setting requires only a degenerate "single-use" version of the construction of Goyal et al., that replaces symmetric encryption with a one-time pad. □

5 OWSC in the Plain Model and Against Malicious Adversaries

In this section, we address the question of implementing our protocols in the plain model. We also show how to augment a plain model OWSC protocol to be secure against active corruption (of the sender, as the receiver is always passive), using a NIZK proof.

5.1 OWSC in the Plain Model

Recall that an OWSC protocol Π using ideal obfuscation uses oracle access to a function F (specified as a circuit $\hat{F}$). We denote by $\Pi[\mathcal{O}]$ the protocol in the plain model that is obtained by communicating $\mathcal{O}(\hat{F})$ instead of providing the oracle. Here, for the purpose of error-free communication, we use an error correcting (or erasure correcting, resp.) code to encode $\mathcal{O}(\hat{F})$ before sending it over BSC (resp., BEC).

As discussed earlier, given the statistical nature of the functions used in the protocols $\Pi^{\text{BEC}}_{\text{ROT}}$ and $\Pi^{\text{BSC}}_{\text{ROT}}$, it is conceivable that there *exists* an obfuscation scheme $\mathcal{O}$ such that the protocols $\Pi^{\text{BEC}}_{\text{ROT}}$ and $\Pi^{\text{BSC}}_{\text{ROT}}$ can be converted to secure protocols in the plain model by using this obfuscation scheme to replace the ideal obfuscation scheme. We state this as a conjecture below.[8]

Conjecture 2. There exists an obfuscation scheme $\mathcal{O}$ such that $\Pi^{\text{BEC}}_{\text{ROT}}[\mathcal{O}]$ and $\Pi^{\text{BSC}}_{\text{ROT}}[\mathcal{O}]$ are OWSC protocols (in the plain model) for ROT, over BEC and BSC respectively, with inverse-polynomial security against a semi-honest sender and a computationally bounded receiver.

[8] We remark that a more general conjecture about obfuscation of a generalized notion of "evasive" functions is plausible, and would in turn imply Conjecture 2. As such a generalization is somewhat tangential to the focus of this work, we do not present this formalization here.

Interestingly, if *any* such scheme as conjectured above exists, then an indistinguishability obfuscation (iO) scheme can be used in its place. More formally, we have the following theorem. Its proof follows standard ideas and is deferred to the full version.

Theorem 7. *Suppose Conjecture 2 holds, with an obfuscation scheme $\mathcal{O}$. Further, suppose there is an iO scheme $i\mathcal{O}$ for all polynomial sized circuits. Let* $\mathrm{pad}(\hat{F})$ *be a padded version of the circuit $\hat{F}$ which is of the same size as $\mathcal{O}(\hat{F})$. Then $\Pi^{\mathrm{BEC}}_{\mathrm{ROT}}[i\mathcal{O} \circ \mathrm{pad}]$ and $\Pi^{\mathrm{BSC}}_{\mathrm{ROT}}[i\mathcal{O} \circ \mathrm{pad}]$ are OWSC protocols (in the plain model) for ROT, over BEC and BSC respectively, with inverse-polynomial security against a semi-honest sender and a computationally bounded receiver.*

5.2 Security against Malicious Sender

In this section, we argue that BEC and BSC are (each) complete even against malicious adversaries in the plain model, assuming Conjecture 2. The key observation here is that UC-secure OWSC protocols for NIZK exist over BEC as well as over BSC, as shown by Garg et al. [25, Lemma 3]. We show that such a NIZK can be used to turn the ROT protocols $\Pi^{\mathrm{BEC}}_{\mathrm{ROT}}[\mathcal{O}]$ and $\Pi^{\mathrm{BSC}}_{\mathrm{ROT}}[\mathcal{O}]$ to be secure against malicious senders. We then appeal to another result of Garg et al. [25] which shows that for general (possibly randomized) functionalities, the ROT channel is complete.

To obtain security against malicious senders, we need to ensure that the receiver's output is of the form $(a_0, \perp)$ with probability $\frac{1}{2}$ and $(\perp, a_1)$ otherwise (except for a small inverse polynomial error). The strings (a_0, a_1) may be probabilistic, but should be extracted by a simulator. For this, we show that it is enough for the sender to additionally provide a NIZK proof of the fact that the program communicated is indeed an obfuscation $\mathcal{O}(\hat{F})$ of a valid function $\hat{F}$ as specified by the protocol. Recall that in the original protocol, the receiver is supposed to feed the message it received over the channel (BEC or BSC) to the obfuscated program and output whatever the program outputs. In the modified ROT protocol, if the verification of the NIZK proof fails, or if the program outputs an error, then the receiver outputs $(a, \perp)$ or $(\perp, a)$ (for some fixed a) with probability $\frac{1}{2}$ each.

We briefly sketch why this modification yields a OWSC for ROT that is secure against a malicious sender (we defer further details to the full version [3]). If the NIZK proof fails or if the program outputs an error, the protocol corresponds to an ideal ROT execution in which the sender sends (a, a) as its input. We need to analyze the behavior of the protocol when this does not happen. Note that the program $\hat{F}$ contains a string $\boldsymbol{x}$ that the sender is supposed to send over the channel, but a malicious sender may send a different string $\boldsymbol{x}'$. If $\boldsymbol{x}'$ differs from $\boldsymbol{x}$ in a lot of positions, then with all but negligible probability the program outputs an error, captured by the above case. On the other hand, if $\boldsymbol{x}'$ agrees with $\boldsymbol{x}$ in most places, then *conditioned on the program not outputting an error*, it can be shown that the output continues to be of the form $(a_0, \perp)$ or $(\perp, a_1)$

with almost equal probabilities, as in the original analysis. A formal analysis of this modification is provided in the full version of this work [3].

It remains to argue that BEC and BSC are complete, even in the plain model, assuming Conjecture 2. Recall that in Section 4.2, we argued that BEC and BSC are complete for OWSC assuming ideal obfuscation, by composing OWSC protocols over ROT for general sender-receiver functionalities with OWSC protocols over BEC/BSC for ROT using ideal obfuscation. The argument for the plain model remains the same, except that we now use the ROT protocols in the plain model. Using standard garbled circuits based on one way functions in the compiler described by Theorem 6, we obtain:

Theorem 8 (Completeness of BEC/BSC against malicious adversary). *Suppose Conjecture 2 holds and one-way functions exist. Then BEC and BSC are (each) complete for OWSC with inverse-polynomial security against a malicious sender and a computationally bounded receiver.*

Acknowledgements

We thank the anonymous Crypto reviewers for their careful reading and many helpful comments. This Research was supported by Ministry of Science and Technology, Israel and Department of Science and Technology, Government of India, and in part by the International Centre for Theoretical Sciences (ICTS) during a visit for participating in the program-Foundational Aspects of Blockchain Technology (ICTS/Prog-fabt2020/01). In addition, S. Agrawal was supported by the DST "Swarnajayanti" fellowship, and Indo-French CEFIPRA project; Y. Ishai was supported by ERC Project NTSC (742754), NSF-BSF grant 2015782, ISF grant 2774/20, and BSF grant 2018393; E. Kushilevitz was supported by ISF grant 2774/20, BSF grant 2018393, and NSF-BSF grant 2015782; V. Prabhakaran was supported by the Science & Engineering Research Board, India through project MTR/2020/000308. V. Narayanan and V. Prabhakaran were supported by the Department of Atomic Energy, Government of India, under project no. RTI4001, DAE OM No. 1303/4/2019/R&D-II/DAE/1969 dated 7.2.2020; M. Prabhakaran was supported by the Dept. of Science and Technology, India via the Ramanujan Fellowship; A. Rosen was supported in part by ISF grant No. 1399/17 and Project PROMETHEUS (Grant 780701). This work was conducted in part when the first and second author were visiting the Simons Institute for Theory of Computing.

References

1. Shweta Agrawal. Indistinguishability obfuscation without multilinear maps: New techniques for bootstrapping and instantiation. In *EUROCRYPT*. Springer, 2019.
2. Shweta Agrawal, Yuval Ishai, Eyal Kushilevitz, Varun Narayanan, Manoj Prabhakaran, Vinod Prabhakaran, and Alon Rosen. Cryptography from one-way communication: On completeness of finite channels. In *Asiacrypt*, 2020.

3. Shweta Agrawal, Yuval Ishai, Eyal Kushilevitz, Varun Narayanan, Manoj Prabhakaran, Vinod Prabhakaran, and Alon Rosen. Secure computation from one-way noisy communication, or: Anti-correlation via anti-concentration. eprint, 2021.
4. Prabhanjan Ananth, Aayush Jain, and Amit Sahai. Indistinguishability obfuscation without multilinear maps: iO from LWE, bilinear maps, and weak pseudorandomness. In *CRYPTO*. Springer, 2019.
5. Benny Applebaum. Bootstrapping obfuscators via fast pseudorandom functions. In *ASIACRYPT*, pages 162–172. Springer, 2014.
6. Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. *J. ACM*, 59(2):6:1–6:48, 2012.
7. James Bartusek, Jiaxin Guan, Fermi Ma, and Mark Zhandry. Return of GGH15: provable security against zeroizing attacks. In *TCC*, pages 544–574. Springer, 2018.
8. James Bartusek, Yuval Ishai, Aayush Jain, Fermi Ma, Amit Sahai, and Mark Zhandry. Affine determinant programs: A framework for obfuscation and witness encryption. In *ITCS*, volume 151, pages 82:1–82:39, 2020.
9. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *CCS*, pages 62–73, 1993.
10. Michael Ben-Or, Shafi Goldwasser, Joe Kilian, and Avi Wigderson. Multi-prover interactive proofs: How to remove intractability assumptions. In *STOC*, pages 113–131. ACM, 1988.
11. Eli Ben-Sasson, Alessandro Chiesa, Michael A. Forbes, Ariel Gabizon, Michael Riabzev, and Nicholas Spooner. Zero knowledge protocols from succinct constraint detection. In *TCC*, pages 172–206. Springer, 2017.
12. C. H. Bennett, G. Brassard, C. Crepeau, and U. M. Maurer. Generalized privacy amplification. *IEEE Trans. Inf. Theor.*, 41(6):1915–1923, 1995.
13. Charles H. Bennett, Gilles Brassard, and Jean-Marc Robert. Privacy amplification by public discussion. *SIAM J. Comput.*, 17(2):210–229, 1988.
14. Matthieu Bloch and Joao Barros. *Physical-layer security: from information theory to security engineering*. Cambridge University Press, 2011.
15. Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Candidate iO from homomorphic encryption schemes. In *EUROCRYPT*, pages 79–109. Springer, 2020.
16. Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Factoring and pairings are not necessary for iO: Circular-secure LWE suffices. *IACR Cryptol. ePrint Arch.*, 2020.
17. Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. Cryptology ePrint Archive, Report 2000/067, 2005. Extended abstract in FOCS 2001.
18. Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, 2004.
19. Yilei Chen, Vinod Vaikuntanathan, and Hoeteck Wee. GGH15 beyond permutation branching programs: Proofs, attacks, and candidates. In *CRYPTO*, pages 577–607. Springer, 2018.
20. C. Crepeau and J. Kilian. Achieving oblivious transfer using weakened security assumptions. In *FOCS*, pages 42–52, 1988.
21. Claude Crépeau, Kirill Morozov, and Stefan Wolf. Efficient unconditional oblivious transfer from almost any noisy channel. In *Security in Communication Networks*, pages 47–59, 2005.

22. Ivan Damgård, Joe Kilian, and Louis Salvail. On the (im)possibility of basing oblivious transfer and bit commitment on weakened security assumptions. In *EUROCRYPT*, pages 56–73. Springer, 1999.
23. Uriel Feige, Joe Kilian, and Moni Naor. A minimal model for secure computation (extended abstract). In *STOC*, pages 554–563, 1994.
24. Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. *SIAM Journal on Computing*, 45:882–929, 01 2016.
25. Sanjam Garg, Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography with one-way communication. In *CRYPTO*, pages 191–208, 2015.
26. Sanjam Garg, Eric Miles, Pratyay Mukherjee, Amit Sahai, Akshayaram Srinivasan, and Mark Zhandry. Secure obfuscation in a weak multilinear map model. In *TCC*, pages 241–268. Springer, 2016.
27. Romain Gay and Rafael Pass. Indistinguishability obfuscation from circular security. In *STOC '21*, pages 736–749, 2021.
28. Shafi Goldwasser and Yael Tauman Kalai. On the impossibility of obfuscation with auxiliary input. In *FOCS*, pages 553–562, 2005.
29. Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. One-time programs. In *CRYPTO*, pages 39–56. Springer, 2008.
30. Shafi Goldwasser and Guy N. Rothblum. On best-possible obfuscation. In *TCC*, pages 194–213, 2007.
31. Vipul Goyal, Yuval Ishai, Mohammad Mahmoody, and Amit Sahai. Interactive locking, zero-knowledge pcps, and unconditional cryptography. In *CRYPTO*, pages 173–190, 2010.
32. Vipul Goyal, Yuval Ishai, Amit Sahai, Ramarathnam Venkatesan, and Akshay Wadia. Founding cryptography on tamper-proof hardware tokens. In *TCC*. Springer, 2010.
33. T. E. Harris. A lower bound for the critical probability in a certain percolation process. *Mathematical Proceedings of the Cambridge Philosophical Society*, 56(1), 1960.
34. Yuval Ishai and Eyal Kushilevitz. Private simultaneous messages protocols with applications. In *ISTCS 1997*, pages 174–184. IEEE Computer Society, 1997.
35. 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. Springer, 2011.
36. Yuval Ishai, Mohammad Mahmoody, and Amit Sahai. On efficient zero-knowledge pcps. In *TCC*, 2012.
37. Aayush Jain, Huijia Lin, and Amit Sahai. Simplifying constructions and assumptions for $i\mathcal{O}$. Cryptology ePrint Archive, Report 2019/1252, 2019. https://eprint.iacr.org/2019/1252.
38. Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In *STOC '21*, pages 60–73, 2021.
39. Yael Tauman Kalai and Ran Raz. Interactive PCP. In *ICALP*, pages 536–547, 2008.
40. Jonathan Katz. Universally composable multi-party computation using tamper-proof hardware. In *EUROCRYPT*, pages 115–128, 2007.
41. Joe Kilian. Founding cryptography on oblivious transfer. In *STOC*, pages 20–31, 1988.
42. Joe Kilian, Erez Petrank, and Gábor Tardos. Probabilistically checkable proofs with zero knowledge. In *STOC*, pages 496–505. ACM, 1997.

43. Daniel J. Kleitman. Families of non-disjoint subsets. *Journal of Combinatorial Theory*, 1(1):153 – 155, 1966.
44. Ilan Komargodski, Tal Moran, Moni Naor, Rafael Pass, Alon Rosen, and Eylon Yogev. One-way functions and (im)perfect obfuscation. In *FOCS 2014*, pages 374–383, 2014.
45. Ilan Komargodski, Moni Naor, and Eylon Yogev. Secret-sharing for NP. *J. Cryptology*, 30(2):444–469, 2017.
46. Fermi Ma and Mark Zhandry. The MMap strikes back: obfuscation and new multilinear maps immune to CLT13 zeroizing attacks. In *TCC*, pages 513–543. Springer, 2018.
47. Mohammad Mahmoody and David Xiao. Languages with efficient zero-knowledge PCPs are in SZK. In Amit Sahai, editor, *TCC*, pages 297–314. Springer, 2013.
48. Ueli M. Maurer. Perfect cryptographic security from partially independent channels. In *STOC*, pages 561–571, 1991.
49. Silvio Micali. Computationally sound proofs. *SIAM J. Comput.*, 30(4):1253–1298, 2000.
50. H. Vincent Poor and Rafael F. Schaefer. Wireless physical layer security. *Proceedings of the National Academy of Sciences*, 114(1):19–26, 2017.
51. Samuel Ranellucci, Alain Tapp, Severin Winkler, and Jürg Wullschleger. On the efficiency of bit commitment reductions. In *ASIACRYPT*, pages 520–537, 2011.
52. R. Sedgewick and P. Flajolet. *An Introduction to the Analysis of Algorithms*. Pearson Education, 2013.
53. Victor Shoup. Lower bounds for discrete logarithms and related problems. In *EUROCRYPT*, pages 256–266, 1997.
54. Hoeteck Wee and Daniel Wichs. Candidate obfuscation via oblivious LWE sampling. In *Eurocrypt*, 2021.
55. Andreas Winter, Anderson C. A. Nascimento, and Hideki Imai. Commitment capacity of discrete memoryless channels. In *Cryptography and Coding*, pages 35–51, 2003.
56. Jürg Wullschleger. Oblivious transfer from weak noisy channels. In *TCC*, pages 332–349. Springer, 2009.
57. A. D. Wyner. The wire-tap channel. *Bell System Technical Journal*, 54(8):1355–1387, 1975.
58. Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *FOCS*, pages 162–167, 1986.