Cryptography from One-Way Communication: On Completeness of Finite Channels

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, Chennai, India
shweta@iitm.ac.in

[2] Technion, Haifa, Israel
{yuvali,eyalk}@cs.technion.ac.il

[3] Tata Institute of Fundamental Research, Mumbai, India
varunnkv@gmail.com, vinodmp@tifr.res.in

[4] Indian Institute of Technology Bombay, Mumbai, India
mp@cse.iitb.ac.in

[5] IDC Herzliya, Herzliya, Israel
alon.rosen@idc.ac.il

Abstract. Garg et al. (Crypto 2015) initiated the study of cryptographic protocols over noisy channels in the non-interactive setting, namely when only one party speaks. A major question left open by this work is the completeness of *finite* channels, whose input and output alphabets do not grow with the desired level of security. In this work, we address this question by obtaining the following results:

1. **Completeness of Bit-ROT with Inverse Polynomial Error.** We show that bit-ROT (i.e., Randomized Oblivious Transfer channel, where each of the two messages is a single bit) can be used to realize general randomized functionalities with inverse polynomial error. Towards this, we provide a construction of string-ROT from bit-ROT with inverse polynomial error.
2. **No Finite Channel is Complete with Negligible Error.** To complement the above, we show that *no* finite channel can be used to realize string-ROT with negligible error, implying that the inverse polynomial error in the completeness of bit-ROT is inherent. This holds even with semi-honest parties and for computational security, and is contrasted with the (negligible-error) completeness of string-ROT shown by Garg et al.
3. **Characterization of Finite Channels Enabling Zero-Knowledge Proofs.** An important instance of secure computation is zero-knowledge proofs. Noisy channels can potentially be used to realize *truly non-interactive* zero-knowledge proofs, without trusted common randomness, and with non-transferability and deniability features that cannot be realized in the plain model. Garg et al. obtain such zero-knowledge proofs from the binary erasure channel (BEC)

© International Association for Cryptologic Research 2020

S. Moriai and H. Wang (Eds.): ASIACRYPT 2020, LNCS 12493, pp. 653–685, 2020.
https://doi.org/10.1007/978-3-030-64840-4_22

and the binary symmetric channel (BSC). We complete the picture by showing that in fact *any non-trivial channel* suffices.

1 Introduction

A noisy communication channel is a probabilistic function $\mathcal{C} : \mathcal{X} \rightarrow \mathcal{Y}$, mapping a sent symbol x to a received symbol y. Standard examples include the *binary symmetric channel* (BSC), which flips a bit $x \in \{0, 1\}$ with probability $0 < p < 1/2$, and the *binary erasure channel* (BEC), which erases x with probability p. A fundamental question in information-theoretic cryptography is – what cryptographic protocols can be constructed from noisy communication channels? This question has been studied extensively, with respect to various cryptographic tasks and a variety of channels, and has uncovered a rich landscape of structural relationships. Starting with the pioneering work of Wyner [30] who showed that the wiretap channel can be used for secure communication, many works studied the usefulness of noisy channels for additional cryptographic tasks (e.g., [5,6,14,23,25,28,29]). This culminated in a complete characterization of the channels on which oblivious transfer, and hence general secure two-party computation, can be based [12,13].

Most cryptographic constructions from noisy channels crucially require interaction. While this is not a barrier for some applications, there are several useful settings which are inherently non-interactive. A natural question that arises is what cryptographic tasks can be realized using only *one-way* noisy channels, namely by protocols over noisy channels in which only one party speaks. The question of realizing *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" [8,24].

A clean way to capture tasks that can potentially be realized using one-way noisy communication is via a *sender-receiver* functionality, which takes an input from a *sender* S and delivers a (possibly) randomized output to a *receiver* R. In more detail, such a sender-receiver functionality is a deterministic or randomized mapping $f : \mathcal{A} \rightarrow \mathcal{B}$ that takes an input $a \in \mathcal{A}$ from a sender S and delivers an output $b = f(a)$ to a receiver R. In the randomized case, the randomness is internal to the functionality; neither S nor R learn it or can influence its choice.

Useful Instances. Several important cryptographic tasks can be captured as sender-receiver functionalities. For instance, a foundational primitive in cryptography is non-interactive zero-knowledge (NIZK) [9,15], which is typically constructed in the common random string (CRS) model. NIZK proofs can be captured in the sender-receiver framework by 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. [17], secure implementation of this function over a one-way channel provides the first *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. Another example from [17] 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 and can only general a puzzle of the level of difficulty prescribed by the randomized algorithm that generates it. A third example of a useful sender-receiver functionality is randomized blind signatures, which can be used for applications such as e-cash [3,10,11]. Blind signatures are 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).[1] Another use-case for such randomized blind signatures is a 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. Applications notwithstanding, understanding the cryptographic power of noisy channels with one-way communication is a fundamental question from the theoretical standpoint.

Prior Work. A large body of theoretical and applied work studied how to leverage one-way communication to construct secure message transmission (see, e.g., [4,24] and references therein). More recently, Garg et al. [17] broadened the scope of this study to include more general cryptographic functionalities. Notably, they showed that one-way communication over the standard BEC or BSC channels suffices for realizing NIZK, or equivalently any *deterministic* sender-receiver functionality. Moreover, for general (possibly randomized) functionalities, a randomized *string-OT* channel or (string-ROT for short) is complete. A string-ROT channel takes a pair of random ℓ-bit strings from the sender and delivers only one of them, chosen at random by the channel, to the receiver. This completeness result was extended in [17] to other channels. However, in all of these general completeness results, the input and alphabet size of the channel grow (super-polynomially) with both the desired level of security and the complexity of the functionality being realized. On the negative side, it was shown in [17] that standard BEC/BSC channels are *not* complete. A major question that was left open is the existence of a complete *finite* channel, whose input and output alphabets do not grow with the security parameter or the complexity of the functionality. Furthermore, for the special case of deterministic functionalities (equivalently, NIZK), it was not known whether completeness holds for *all* non-trivial finite channels.

[1] In more detail, the sender can generate an anonymous $100 bill by letting the input be m =(Sender-name, 100) and the transmitted message be (m, id) for a random identifier *id* picked by the functionality. Consider the scenario where multiple $100 bills are sent to different receivers. The id is needed to prevent double spending. Anonymity comes from the fact that the sender doesn't learn *id*, so it cannot associate a particular $100 bill with the receiver to whom it was sent.

Next, we describe our framework in a bit more detail, followed by a summary of our results, which essentially settle the above mentioned questions.

Our Framework. Let $\mathcal{C}$ be a finite channel. We define a one-way secure computation protocol (OWSC) for a functionality f over channel $\mathcal{C}$ as a randomized encoder that maps the sender's input a into a sequence $\boldsymbol{x}$ of channel inputs, and a decoder that maps the sequence of receiver's channel outputs $\boldsymbol{y}$ into an output b. Given an error parameter ϵ, the protocol should satisfy the following security requirements: (i) given the sender's view, which consists of its input a and the message $\boldsymbol{x}$ 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 message $\boldsymbol{y}$ it received from the channel, can be simulated from $f(a)$. Note that (i) captures receiver security against a corrupt sender as well as correctness, while (ii) captures sender security against a corrupt receiver.

We will construct OWSC protocols for various functionalities over various finite channels. Of particular interest to us is the randomized ℓ-bit string-ROT channel discussed above, which we denote by $\mathcal{C}^{\ell}_{\mathsf{ROT}}$, and its finite instance $\mathcal{C}^{1}_{\mathsf{ROT}}$ that we refer to as the *bit-ROT* channel.

1.1 Our Results

We are ready to state our results:

1. **Completeness of Bit-ROT with Inverse Polynomial Error.** We show that bit-ROT is complete for randomized functionalities with *inverse polynomial* simulation error. Towards this, we provide a construction of string-ROT from bit-ROT with inverse polynomial error, and appeal to the completeness of string-ROT. This is captured by the following (formal statement in Theorem 7):

Theorem 1. *(Informal) The bit-ROT channel ($\mathcal{C}^{1}_{\mathsf{ROT}}$) is complete for one-way secure computation, with* inverse-polynomial error*. This holds for both semi-honest and malicious parties. The protocol establishing completeness can either be efficient in the circuit size, in which case it is computationally secure using any pseudorandom generator, or efficient in the branching program size, in which case is it information-theoretically secure.*

2. **No Finite Channel is Complete with Negligible Error.** To complement the above positive result, we show that *no* finite channel is complete for randomized functionalities with negligible error. This is contrasted with the completeness of string-ROT discussed above. In more detail, we prove the following theorem (formal statement in Theorem 9):

Theorem 2. *(Informal): No finite channel is complete for one-way secure computation, with negligible error, even with semi-honest parties and for computational security. More concretely, string-ROT cannot be implemented in this setting.*

3. **Every Non-trivial Finite Channel is Complete for Zero-Knowledge.** As discussed above, a particularly compelling use case for one-way communication over noisy channels is *truly non-interactive* zero-knowledge proofs, without a trusted common randomness setup and with desirable features such as non-transferability and deniability. The results of Garg et al. [17] obtain such NIZK proofs from the binary erasure channel (BEC) and the binary symmetric channel (BSC). This raises the question whether *all* non-trivial channels enable NIZK.

   We show that this is indeed the case if we define a "trivial" channel to be one that either does not enable communication at all, or is essentially equivalent to a noiseless channel, when used by malicious senders. In more detail, we prove the following theorem (see Sect. 5 for a formal statement):

Theorem 3. *(Informal): Given a language $L \in \mathrm{NP} \setminus \mathrm{BPP}$, a one-way secure computation protocol over channel $\mathcal{C}$ for zero-knowledge for L exists if and only if $\mathcal{C}$ is non-trivial.*

1.2 Our Techniques

In this section we provide an overview of our techniques.

Completeness of Bit-ROT with Inverse Polynomial Error. We show that bit-ROT is complete for randomized functionalities with inverse polynomial error. Towards this, we show, in Theorem 6, that (ℓ-bit) string-ROT can be realized with polynomially many invocations of bit-ROT channel with inverse-polynomial error. The OWSC protocol is efficient in ℓ and is secure even against malicious adversaries.

In more detail, we use *average case secret sharing*, which is a weak version of ramp secret sharing, where both the reconstruction and privacy conditions are to be satisfied for a random set of r players and t players respectively, where r and t are the reconstruction and privacy thresholds, respectively. Theorem 4 provides a construction of OWSC protocol for string-ROT using bit-ROT given an average case secret sharing schemes ($\mathsf{Avg\text{-}SSS}$) with sufficiently small gap parameter. The analysis of this theorem crucially uses the *anti-concentration bound* for Bernoulli sums for a small window around the mean. In Theorem 5, we construct efficient $\mathsf{Avg\text{-}SSS}$ for N players in which the gap between r and t is inverse polynomial in N and which have inverse polynomial privacy guarantee. The scheme we construct and its analysis build on techniques for secret sharing with binary shares that were recently introduced by Lin et al. [22] (for a different goal). Our result on efficient realization of string-ROT from bit-ROT directly follows from combining the above two results.

Impossibility of String-ROT from Finite Channel with Negligible Error. Next, we show that string-ROT cannot be constructed from bit-ROT with negligible error. We establish our result in two steps. Our first negative result in Theorem 8 shows that string-ROT cannot be realized with polynomially many invocations of bit-ROT channel while guaranteeing negligible error.

Our proof is inspired by [17]. In more detail, we use an isoperimetric inequality for Boolean hypercubes (Harper's theorem), to show the existence of strategies that can efficiently guess both input strings in any implementation of string-ROT with polynomially bounded number of bit-ROT invocations, which is a violation of the ROT security. The machine we describe for guessing the two input strings is computationally efficient, hence our impossibility result applies to computationally bounded semi-honest adversaries.

We then extend this result in Theorem 9 to show that no finite channel can be used to realize string-ROT using polynomially many invocations of the channel while guaranteeing negligible error. To show this, we model a channel as a function from the input of the channel and its internal randomness to the output of the channel. We then proceed to prove the impossibility in a manner similar to the impossibility for the bit-ROT channel.

Impossibility of Completeness of Finite Channels with Negligible Error. Theorem 9 shows that string-ROT cannot be realized over any finite channel efficiently (in terms of the number of channel invocations) and with negligible error, even in the computational setting. Since string-ROT is a simple functionality which has a small description in many functional representation classes, we obtain an impossibility result that strikes off the possibility of a complete channel with negligible error for most function representation classes of interest.

Characterization of Finite Channels Enabling Zero-Knowledge Proofs. It is a fundamental question to understand which channels enable ZK proofs. We give a complete characterization of all finite channels over which a OWSC protocol for zero-knowledge (proof of knowledge) functionality is possible. In fact, we show that the only channels which do not enable zero-knowledge proofs are "trivial" channels (a proof over a trivial channel translates to a proof over a plain one-way communication channel which is possible only for languages in BPP). Over any other finite channel, we build a statistical zero-knowledge proof of knowledge, which is unconditionally secure. Our result generalizes a result of [17], which gave OWSC zero-knowledge proof protocols over Binary Erasure Channels (BEC) and Binary Symmetric Channels (BSC) only. Extending this result to all non-trivial channels requires new ideas, exploiting a geometric view of channels.

2 Preliminaries

To begin, we define some notation that we will use throughout the paper.

Notation 1. *A member of a finite set $\mathcal{X}$ is represented by x and sampling an independent uniform sample from $\mathcal{X}$ is denoted by $x \xleftarrow{\$} \mathcal{X}$. A vector in $\mathcal{X}^n$ is represented by $\boldsymbol{x} \in \mathcal{X}^n$, whose coordinate $i \in [n]$ is represented by either x_i or $\boldsymbol{x}(i)$.*

For a vector $\boldsymbol{x} \in \mathcal{X}^n$ and a set $A \subseteq [n]$, the restriction of $\boldsymbol{x}$ to the set A, represented by $\boldsymbol{x}|_A$ is the vector with all the coordinates outside of A replaced by an erasure symbol $\perp$ which is not a member of $\mathcal{X}$. That is, $\boldsymbol{x}|_A(i) = \boldsymbol{x}(i)$ if $i \in A$ and $\boldsymbol{x}|_A(i) = \perp$ otherwise. Finally, $\Delta(\mu_0, \mu_1)$ denotes the total variation distance between distributions μ_0 and μ_1.

2.1 Sender-Receiver Functionalities and Channels

This work addresses 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. 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* and write $f(a; \rho)$ when we want to make the internal randomness of f explicit.

In order to realize f, we assume that S and R are given parallel access to a *channel* $\mathcal{C} : \mathcal{X} \to \mathcal{Y}$, which is a sender-receiver functionality that is typically much simpler than the target function f. We will typically view $\mathcal{C}$ as being *finite* whereas f will come from an infinite class of functions. We will be interested in the number of invocations of $\mathcal{C}$ required for realizing f with a given error ϵ (if possible at all).

We will be particularly interested in the following channel.

Definition 1 (ROT channel). *The ℓ-bit randomized string oblivious transfer channel (or ℓ-bit string-ROT for short), denoted by $\mathcal{C}^{\ell}_{\mathsf{ROT}}$, takes from S a pair of strings $\boldsymbol{a}_0, \boldsymbol{a}_1 \in \{0,1\}^{\ell}$, and delivers to R*

$$\mathcal{C}^{\ell}_{\mathsf{ROT}}(\boldsymbol{a}_0, \boldsymbol{a}_1) = \begin{cases} (\boldsymbol{a}_0, \perp) & \textit{w.p. } \frac{1}{2}, \\ (\perp, \boldsymbol{a}_1) & \textit{w.p. } \frac{1}{2}. \end{cases}$$

Finally, it is sometimes convenient to assume that a sender-receiver functionality f can additionally take a *public input* that is known to both parties. For instance, in a zero-knowledge proof such a public input can include the NP-statement, or in blind signatures it can include the receiver's public verification key (allowing f to check the validity of the secret key). All of our definitions and results can be easily extended to this more general setting.

2.2 Secure Computation with One-Way Communication

A secure protocol for $f : \mathcal{A} \to \mathcal{B}$ over a channel $\mathcal{C}$ is formalized via the standard definitional framework of reductions in secure computation. Our default setting shall be that of *information-theoretic* security against *semi-honest* parties, with extensions to the setting of computational security and malicious parties. All our negative results in fact hold for the weakest setting of computational security against semi-honest parties. All our positive results hold for (either information-theoretic or computational) security against malicious parties.

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. Given an error parameter ϵ, the protocol should satisfy the following security requirements: (i) given the sender's view, which consists of an input a and the message $\boldsymbol{x}$ 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 message $\boldsymbol{y}$ it received from the channel, can be simulated from $f(a)$. Note that (i) captures receiver security against a corrupt sender as well as correctness, while (ii) captures sender security against a corrupt receiver. We formalize this below.

Definition 2 (One-way secure computation). *Given a randomized function $f : \mathcal{A} \to \mathcal{B}$ and a channel $\mathcal{C} : \mathcal{X} \to \mathcal{Y}$, a pair of randomized functions $\langle \mathsf{S}, \mathsf{R} \rangle$, where $\mathsf{S} : \mathcal{A} \to \mathcal{X}^N$ and $\mathsf{R} : \mathcal{Y}^N \to \mathcal{B}$ is said to be an (N, ϵ)* OWSC protocol for *f* over *$\mathcal{C}$ if there exists a simulator $\mathsf{Sim}_\mathsf{R} : \mathcal{B} \to \mathcal{Y}^N$, such that for all $a \in \mathcal{A}$,*

$$\varDelta\left((\mathsf{S}(a), f(a)), (\mathsf{S}(a), \mathsf{R}(\mathcal{C}(\mathsf{S}(a))))\right) \le \epsilon$$

$$\varDelta\left(\mathsf{Sim}_\mathsf{R}(f(a)), \mathcal{C}(\mathsf{S}(a))\right) \le \epsilon$$

OWSC for Malicious Parties. In this case, our security requirement coincides with UC security, but with simplifications implied by the communication model. Specifically, since a corrupt receiver has no input to the functionality nor any message in the protocol, UC security against a malicious receiver is the same as in the semi-honest setting. UC security against a malicious sender, on the other hand, requires that from any arbitrary strategy of the sender, 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 2, there exists a randomized simulator $\mathsf{Sim}_\mathsf{S} : \mathcal{X}^N \to \mathcal{A}$ such that for every $\boldsymbol{x} \in \mathcal{X}^N$,

$$\varDelta\left(f(\mathsf{Sim}_\mathsf{S}(\boldsymbol{x})), \mathsf{R}(\mathcal{C}(\boldsymbol{x}))\right) \le \epsilon.$$

In our (positive) results in this setting, we shall require the simulator to be computationally efficient as well.

OWSC with Computational Security. We can naturally relax the above definition of (statistical) (N, ϵ) OWSC to *computational* (N, T, ϵ) OWSC, for a distinguisher size bound T, by replacing each statistical distance bound $\varDelta(A, B) \le \epsilon$ by the condition that for all circuits C of size T, $|\Pr(C(A) = 1) - \Pr(C(B) = 1)| \le \epsilon$.

Complete Channels for OWSC. So far, we considered OWSC protocols for a concrete function f and with a concrete level of security ϵ. However, in a cryptographic context, one is typically interested in a single "universal" protocol that takes a description $\hat{f}$ of a function f and a security parameter λ as inputs and runs in polynomial time in its input length.

To meaningfully specify the goal of such a universal OWSC protocol, we need to fix a representation class $\mathcal{F}$ that defines an association between a bit-string $\hat{f}$ and the (deterministic or randomized) function f it represents. The representation classes $\mathcal{F}$ we will be interested in include *circuits* (capturing general polynomial-time computations) and *branching programs* (capturing logarithmic-space computations and logarithmic-depth circuits). The string-ROT channel $\mathcal{C}^{\ell}_{\mathsf{ROT}}$ can also be viewed as a degenerate function class $\mathcal{F}$ in which $\hat{f} = 1^{\ell}$ specifies the string length.

If a channel $\mathcal{C}$ enables a universal protocol for $\mathcal{F}$, we say that $\mathcal{C}$ is *OWSC-complete* for $\mathcal{F}$. 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 and computational security. We formalize this notion of completeness below.

Definition 3 (OWSC-complete channel). *Let $\mathcal{F}$ be a function representation class and $\mathcal{C}$ be a channel. We say that $\mathcal{C}$ is* OWSC-complete *for evaluating $\mathcal{F}$ with (statistical)* inverse-polynomial *error if for every positive integer c there is a polynomial-time protocol $\Pi = \langle \mathsf{S}, \mathsf{R} \rangle$ that, on common input $(1^{\lambda}, \hat{f})$, realizes (N, ϵ) OWSC of f over $\mathcal{C}$, where $\epsilon = \mathcal{O}(\frac{1}{\lambda^c})$ and $N = poly(\lambda, |\hat{f}|)$. We say that $\mathcal{C}$ is complete with* negligible *error if there is a single Π as above such that ϵ is negligible in λ. We similarly define the computational notions of completeness by requiring the above to hold with (N, T, ϵ) instead of (N, ϵ), for an arbitrary polynomial $T = T(\lambda)$.*

As discussed above, useful instances 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.

2.3 OWSC Zero-Knowledge Proof of Knowledge

For a language L in NP, let R_L denote a polynomial time computable relation such that $x \in L$ if and only if for some w of length polynomial in the length of x, we have $R_L(x, w) = 1$. In the classic problem of *zero-knowledge proof*, given a common input $x \in L$, a polynomial time prover who has access to a w such that $R_L(x, w) = 1$ wants to convince a polynomial time verifier that $x \in L$, without revealing any additional information about w. On the other hand, if $x \notin L$, even a computationally unbounded prover should not be able to make the verifier accept the proof, except with negligible probability.

While classically, the prover and the verifier are allowed to interact with each other, or in the case of Non-Interactive Zero-Knowledge (NIZK), are given a common random string generated by a trusted third party, in a ZK protocol in the OWSC model, a single string is transmitted from the prover to the receiver, over a channel $\mathcal{C}$, with no other trusted set up. We shall require information-theoretic security, with both soundness and zero-knowledge properties defined via simulation. As simulation-based soundness corresponds to a proof of knowledge (PoK), we shall refer to this primitive as $\mathsf{OWSC}/\mathcal{C}$ ZK-PoK.[2]

Definition 4 (OWSC Zero-knowledge Proof of Knowledge). *Given a channel $\mathcal{C}$, a pair of PPT algorithms* $(\mathsf{P}_{ZK}, \mathsf{V}_{ZK})$ *is a* $\mathsf{OWSC}/\mathcal{C}$ zero-knowledge proof of knowledge (ZK-PoK) *for an* NP *language L with an associated relation R_L if the following hold:*

Completeness. *There is a negligible function* negl*, such that $\forall x \in L$ and w such that $R_L(x, w) = 1$,*

$$\Pr\left[\mathsf{V}_{ZK}(1^\lambda, x, \mathcal{C}(\mathsf{P}_{ZK}(1^\lambda, x, w))) \neq 1\right] = \mathrm{negl}(\lambda)$$

(where the probability is over the randomness of P_{ZK} and V_{ZK} and that of the channel).

Soundness. *There exists a probabilistic polynomial time (PPT) extractor E such that, for all x and all collection of strings z_λ (collection indexed by λ)*

$$R_L\left(x, E(1^\lambda, x, z_\lambda)\right) = 0 \quad \Rightarrow \quad \Pr\left[\mathsf{V}_{ZK}(1^\lambda, x, \mathcal{C}(z_\lambda)) = 1\right] = \mathrm{negl}(\lambda).$$

Zero-Knowledge. *There exists a PPT simulator S such that, for all $x \in L$, and w such that $R_L(x, w) = 1$,*

$$\mathcal{C}(\mathsf{P}_{ZK}(1^\lambda, x, w)) \approx_{\mathrm{negl}(\lambda)} S(1^\lambda, x),$$

where $\approx$ represents computational indistinguishability.

In our construction we use the notion of oblivious zero-knowledge PCP, which was explicitly defined in [17]. In the problem of *oblivious zero-knowledge PCP*, a prover with access to $x \in L$ and w such that $R_L(x, w) = 1$ would like to publish a proof. The verifier's algorithm probes a constant number of random locations in the published proof and decides to accept or reject while guaranteeing correctness and soundness. The notion of oblivious zero-knowledge requires that the PCP is zero-knowledge when each bit in the proof is erased with finite probability.

Definition 5 (Oblivious ZK-PCP). *[17, Definition 1]* $(\mathsf{P}_{oZK}, \mathsf{V}_{oZK})$ *is a (c, ν)-oblivious ZK-PCP with knowledge soundness κ for an NP language L if, when λ is the security parameter, $\mathsf{P}_{oZK}, \mathsf{V}_{oZK}$ are probabilistic algorithms that run in polynomial time in λ and the length of the input x and satisfy the following conditions.*

Completeness. $\forall (x, w) \in R_L$ *when* $\pi \xleftarrow{\$} \mathsf{P}_{oZK}(x, w, \lambda)$, $\Pr(\mathsf{V}_{oZK}(x, \pi^*)) = 1$ *for all choices of π^* obtained by erasing arbitrary locations of π.*

[2] Indeed, an $\mathsf{OWSC}/\mathcal{C}$ ZK-PoK protocol is equivalent to an information-theoretic UC-secure protocol for the ZK functionality in the $\mathcal{C}$-hybrid model, with an additional requirement that the protocol involves a single invocation of $\mathcal{C}$ and no other communication.

c-Soundness. *There exists a PPT extractor E such that, for all x and purported proofs π', if $(x, E(x, \pi')) \notin R_L$ then*

$$\Pr(\mathsf{V}_{oZK}(x, g(\pi')) = 0) \geq \kappa,$$

where the probability is taken over the random choices of g, where g is any function that replaces all but c locations of π' with $\perp$ (and leaves the other locations untouched).

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

- *Sample $\pi \overset{\$}{\leftarrow} \mathsf{P}_{oZK}(\lambda, x, w)$, replace each bit in π with $\perp$ with probability $1 - \nu$ and output the resultant value.*
- *$S(x, \lambda)$.*

As described in [17], the following result is implied by a construction in [2]:

Proposition 1 *[17, Proposition 1]. For any constant $\nu \in (0, 1)$, there exists a $(3, \nu)$-oblivious ZK-PCP with a knowledge soundness $\kappa = 1 - \frac{1}{p(\lambda)}$, where $p(\lambda)$ is some polynomial in λ.*

3 String-ROT from Bit-ROT with Inverse Polynomial Error

In this section, we construct string-ROT from bit-ROT with inverse polynomial error, and apply this to show that bit-ROT is complete for general sender-receiver functionalities with inverse-polynomial error. Since the intuition was discussed in Sect. 1, we proceed directly with the construction.

3.1 Average Case Secret Sharing

An N player average case secret sharing scheme, for ℓ-bit secrets with reconstruction threshold r and privacy threshold t, consists of a sharing algorithm Share and a reconstruction algorithm Recst which guarantees that a random subset of t players learns nothing about the secret and that a random set of r players can reconstruct the secret with high probability. This is formalized by the next definition, where the following notation will be useful.

Notation 2. *For integers $1 \leq s \leq N$, we use the following families of subsets of $[N]$: $\mathcal{A}_s = \{A \subseteq [N] : |A| = s\}$, $\mathcal{A}_{\geq s} = \{A \subseteq [N] : |A| \geq s\}$, and $\mathcal{A}_{\leq s} = \{A \subseteq [N] : |A| \leq s\}$.*

Definition 6. *A $(\ell, N, t, r, \epsilon)$* average-case secret-sharing scheme *(*$\mathsf{Avg\text{-}SSS}$*, for short) is a pair of randomized algorithms $\langle \mathsf{Share}, \mathsf{Recst} \rangle$ such that,*

$$\mathsf{Share} : \{0, 1\}^\ell \times \mathcal{R} \to \{0, 1\}^N \quad \textit{and} \quad \mathsf{Recst} : \{0, 1, \perp\}^N \to \{0, 1\}^\ell,$$

where $\mathcal{R}$ is the private randomness, that satisfy the following properties.

Reconstruction Property: Recst *must be able to reconstruct any secret from a uniformly random set of r shares produced by* Share*, with at least* $1-\epsilon$ *probability. Formally, for all* $\boldsymbol{s} \in \{0,1\}^{\ell}$,

$$\Pr(\mathsf{Recst}(\mathsf{Share}(\boldsymbol{s})|_A) = \boldsymbol{s}) \geq 1-\epsilon,$$

where the probability is over the randomness used by Share *and the choice of* $A \overset{\$}{\leftarrow} \mathcal{A}_r$.

Privacy Property: *t random shares of every pair of secrets are ϵ-close to each other in statistical distance. Formally, for all* $\boldsymbol{s}, \boldsymbol{s}' \in \{0,1\}^{\ell}$*, and* $A \overset{\$}{\leftarrow} \mathcal{A}_t$,

$$\Delta\left((\mathsf{Share}(\boldsymbol{s})|_A), (\mathsf{Share}(\boldsymbol{s}')|_A)\right) \leq \epsilon.$$

We will typically be interested in $(\ell, N, t, r, \epsilon)$-$\mathsf{Avg\text{-}SSS}$ where ℓ, t, r, ϵ are functions of N and require $\mathsf{Share}, \mathsf{Recst}$ to be probabilistic algorithms with poly(N) complexity.

3.2 String-ROT from Bit-ROT and Average Case Secret Sharing

In this section, we show that an average case secret sharing scheme can be used to reduce string ROT to bit ROT. The following theorem demonstrates such a reduction.

Theorem 4. *For $\delta \in (0, \frac{1}{2})$ and for sufficiently large N, given a $(\ell, N, t, r, \epsilon)$-*$\mathsf{Avg\text{-}SSS}$*, with $t = \lfloor \frac{N}{2} \rfloor - N^{\delta}$, $r = \lceil \frac{N}{2} \rceil + N^{\delta}$ and $\epsilon = N^{\delta - \frac{1}{2}}$, there exists a secure (even against malicious parties) $(N, 4N^{\delta-\frac{1}{2}})$ OWSC protocol for $\mathcal{C}^{\ell}_{\mathsf{ROT}}$ over $\mathcal{C}^{1}_{\mathsf{ROT}}$. If the* $\mathsf{Avg\text{-}SSS}$ *scheme is efficient in N, then so is our protocol.*

Proof: Let $\langle \mathsf{Share}, \mathsf{Recst} \rangle$ be an $(\ell, N, t, r, \epsilon)$-$\mathsf{Avg\text{-}SSS}$. The protocol that realizes $\mathcal{C}^{\ell}_{\mathsf{ROT}}$ in the $\mathsf{OWSC}/\mathcal{C}^{1}_{\mathsf{ROT}}$ model proceeds as follows.

Let $(\boldsymbol{a}_0, \boldsymbol{a}_1) \in \{0,1\}^{\ell} \times \{0,1\}^{\ell}$ be the input to the $\mathcal{C}^{\ell}_{\mathsf{ROT}}$. Sender computes $\boldsymbol{x}_0 = \mathsf{Share}(\boldsymbol{a}_0)$ and $\boldsymbol{x}_1 = \mathsf{Share}(\boldsymbol{a}_1)$. For $i = 1, \ldots, N$, sender sends $(\boldsymbol{x}_0(i), \boldsymbol{x}_1(i))$ in the i-th invocation of the $\mathcal{C}^{1}_{\mathsf{ROT}}$ channel.

The receiver gets $\boldsymbol{x}_0|_A$, $\boldsymbol{x}_1|_{[N]\setminus A}$, where A is a uniformly random subset of $[N]$. If $|A| \geq r$, it uniformly samples $A_0 \subseteq A$ such that $|A_0| = r$ and outputs $(\mathsf{Recst}(\boldsymbol{x}_0|_{A_0}), \perp)$, and if $|[N] \setminus A| \geq r$, it uniformly samples $A_1 \subseteq [N] \setminus A$ such that $|A_1| = r$ and outputs $(\perp, \mathsf{Recst}(\boldsymbol{x}_1|_{A_1}))$. If $|A| \in (t, r)$, R samples $\boldsymbol{a}_0, \boldsymbol{a}_1 \overset{\$}{\leftarrow} \{0,1\}^{\ell}$ and $i \overset{\$}{\leftarrow} \{0,1\}$ and outputs $(\boldsymbol{a}_0, \perp)$ if $i = 0$ and $(\perp, \boldsymbol{a}_1)$ if $i = 1$.

Complexity. The complexity of this reduction is N. If $\mathsf{Avg\text{-}SSS}$ is efficient, the protocol is efficient as well.

Security. We first show that the receiver's output is consistent with probability at least $1-3N^{\delta-\frac{1}{2}}$. That is, if the input to the sender is $(\boldsymbol{a}_0, \boldsymbol{a}_1)$, with probability $1-3N^{\delta-\frac{1}{2}}$, the receiver outputs either $(\perp, \boldsymbol{a}_1)$ or $(\boldsymbol{a}_0, \perp)$. To show this, we bound the probability of the event $|A| \in (t, r)$ using an anti-concentration bound on Bernoulli sums and then argue that conditioned on $|A| \notin (t, r)$, the receiver's output is consistent with probability $\geq 1-\epsilon$.

Claim 1. *Let X_i be i.i.d Bernoulli($\frac{1}{2}$) random variables for $i \in [N]$. Then, for all $\delta \in (0, 1/2)$,*

$$\Pr\left(\left|\sum_{i\in[N]} X_i - \left\lceil\frac{N}{2}\right\rceil\right| < N^{\delta}\right) \le 2N^{\delta-\frac{1}{2}}.$$

Proof: This follows from the fact that,

$$\forall k \in [N],\quad \Pr\left(\sum_{i\in[N]} X_i = k\right) \le \Pr\left(\sum_{i\in[N]} X_i = \lceil N/2\rceil\right) \le N^{-1/2}.$$

□

Denote the event $|A| \notin (t, r)$ by E. Since $r - t = 2N^{\delta}$, $\Pr(E) \ge 1 - 2N^{\delta-\frac{1}{2}}$ by the above claim. Conditioned on $|A| \ge r$, A is uniformly distributed in $\mathcal{A}_{\ge r}$. Hence, A_0 is uniformly distributed in $\mathcal{A}_r$. The receiver is correct if $\mathsf{Recst}(\mathsf{Share}(\boldsymbol{a}_0)|_{A_0}) = \boldsymbol{a}_0$. By the reconstruction property of $\langle \mathsf{Share}, \mathsf{Recst}\rangle$, for all $\boldsymbol{a}_0 \in \{0,1\}^{\ell}$, we have

$$\Pr(\mathsf{Recst}(\mathsf{Share}(\boldsymbol{a}_0)|_{A_0}) = \boldsymbol{a}_0) \ge 1 - \epsilon = 1 - N^{\delta-\frac{1}{2}},$$

where the probability is over the randomness used by Share and $A_0 \xleftarrow{\$} \mathcal{A}_r$. Similar bound applies for $\Pr(\mathsf{Recst}(\mathsf{Share}(\boldsymbol{a}_1)|_{A_1})$ conditioned on the event $|A| \le t$. From these observations, the probability that the receiver outputs $(\boldsymbol{a}_0, \perp)$ or $(\perp, \boldsymbol{a}_1)$ when the sender's input is $(\boldsymbol{a}_0, \boldsymbol{a}_1)$ can be lower bounded as,

$$\Pr(E) \cdot \Pr(\text{Receiver outputs } (\boldsymbol{a}_0, \perp) \text{ or } (\perp, \boldsymbol{a}_1)|E) \ge (1 - 2N^{\delta-\frac{1}{2}})(1 - N^{\delta-\frac{1}{2}}) \ge 1 - 3N^{\delta-\frac{1}{2}}.$$

Furthermore, when $|A| \notin (t, r)$, the events $|A| \ge r$ and $N - |A| \ge r$ are equiprobable. That is, the index on which the receiver outputs $\perp$ is decided entirely by the randomness in the channel. Hence, for all $\boldsymbol{a}_0, \boldsymbol{a}_1 \in \{0,1\}^{\ell}$,

$$\varDelta\left(\left(\boldsymbol{a}_0, \boldsymbol{a}_1, \mathsf{S}(\boldsymbol{a}_0, \boldsymbol{a}_1), \mathsf{R}(\mathcal{C}^1_{\mathsf{ROT}}(\mathsf{S}(\boldsymbol{a}_0, \boldsymbol{a}_1)))\right), \left(\boldsymbol{a}_0, \boldsymbol{a}_1, \mathsf{S}(\boldsymbol{a}_0, \boldsymbol{a}_1), \mathcal{C}^{\ell}_{\mathsf{ROT}}(\boldsymbol{a}_0, \boldsymbol{a}_1)\right)\right) \le 3N^{\delta-\frac{1}{2}}.$$

We now analyze security against the receiver. We claim that conditioned on the event $|A| \le t$, for any $\boldsymbol{a}_0, \boldsymbol{a}'_0, \boldsymbol{a}_1 \in \{0,1\}^{\ell}$, the view of the receiver when the input to the sender is $(\boldsymbol{a}_0, \boldsymbol{a}_1)$ is sufficiently close to its view when the sender's input is $(\boldsymbol{a}'_0, \boldsymbol{a}_1)$. Note that conditioned on $|A| \le t$, $|A|$ is a uniformly random set of size at most t. Our claim is that for all $\boldsymbol{a}_0, \boldsymbol{a}'_0 \in \{0,1\}^{\ell}$ and $A \xleftarrow{\$} \mathcal{A}_{\le t}$,

$$\varDelta\left(\mathsf{Share}(\boldsymbol{a}_0)|_A, \mathsf{Share}(\boldsymbol{a}'_0)|_A\right) \le \epsilon = N^{\delta-\frac{1}{2}}.$$

To show this, note that the output distributions of the following two experiments are the same for every $\boldsymbol{a} \in \{0,1\}^{\ell}$:

(1) Choose $0 \leq k \leq t$ with probability $\Pr_{S \xleftarrow{\$} \mathcal{A}_{\leq t}}(|S| = k)$. When $A \xleftarrow{\$} \mathcal{A}_t$, let B be a uniformly random subset of A of size k. Output $\mathsf{Share}(\boldsymbol{a})|_B$.

(2) $A \xleftarrow{\$} \mathcal{A}_{\leq t}$, output $\mathsf{Share}(\boldsymbol{a})|_A$.
Hence, the distribution $\mathsf{Share}(\boldsymbol{a}_0)|_A$ where $A \xleftarrow{\$} \mathcal{A}_{\leq t}$ can be generated by post-processing the distribution $\mathsf{Share}(\boldsymbol{a}_0)|_A$ where $A \xleftarrow{\$} \mathcal{A}_t$. The claim now follows from the privacy guarantee of Avg-SSS and the fact that statistical distance only decreases on post-processing.

On input $(\perp, \boldsymbol{a}_1)$ the simulator Sim_R proceeds as follows: Sample $\boldsymbol{a} \xleftarrow{\$} \{0,1\}^\ell$ and run the algorithm of the sender with input $(\boldsymbol{a}, \boldsymbol{a}_1)$, to generate $(\boldsymbol{x}_0, \boldsymbol{x}_1)$. Sample $A \xleftarrow{\$} \mathcal{A}_{\leq t}$ and output $(\boldsymbol{x}_0|_A, \boldsymbol{x}_1|_{[N]\setminus A})$. The case for $(\boldsymbol{a}_0, \perp)$ is symmetric.

That Sim_R satisfies sender's privacy follows from the following observations: (a) The event $|A| \notin (t, r)$ happens with probability at least $1 - 2N^{\delta - \frac{1}{2}}$. (b) $\boldsymbol{a}_0$ (resp. $\boldsymbol{a}_1$) is decoded correctly with probability $1 - N^{\delta - \frac{1}{2}}$ when $|A| \geq r$ (resp. $|A| \leq t$). Furthermore, conditioned on both these events, the receiver's view for input $(\boldsymbol{a}_0, \boldsymbol{a}_1)$ and for input $(\boldsymbol{a}'_0, \boldsymbol{a}_1)$ are at most $N^{\delta - \frac{1}{2}}$ far in statistical distance, for all $\boldsymbol{a}_0, \boldsymbol{a}'_0 \in \{0,1\}^\ell$. Hence,

$$\Delta\left(\mathsf{Sim}_\mathsf{R}(\mathcal{C}^\ell_{\mathsf{ROT}}(\boldsymbol{a}_0, \boldsymbol{a}_1)), \mathcal{C}^1_{\mathsf{ROT}}(\mathsf{S}(\boldsymbol{a}_0, \boldsymbol{a}_1))\right) \leq 4N^{\delta - \frac{1}{2}}$$

UC-Security Against Malicious Adversaries. For any $\boldsymbol{x} \in \{0,1\}^N$, simulator Sim_S works as follows. Sample $A_{\geq r} \xleftarrow{\$} \mathcal{A}_{\geq r}$ and $A_{\leq t} \xleftarrow{\$} \mathcal{A}_{\leq t}$ (this can be done efficiently by rejection sampling). Let $(\boldsymbol{b}_0, \perp) = \mathsf{R}(\boldsymbol{x}|_{A_{\geq r}})$ and $(\perp, \boldsymbol{b}_1) = \mathsf{R}(\boldsymbol{x}|_{A_{\leq t}})$. Sample $A \xleftarrow{\$} [N]$, if $|A| \in (t, r)$, output $(\boldsymbol{s}_0, \boldsymbol{s}_1)$, where $\boldsymbol{s}_0, \boldsymbol{s}_1 \xleftarrow{\$} \{0,1\}^\ell$, else output $(\boldsymbol{b}_0, \boldsymbol{b}_1)$.

We claim that distribution $\mathcal{C}^1_{\mathsf{ROT}}(\mathsf{Sim}_\mathsf{S}(\boldsymbol{x}))$ is identical to the output distribution of the receiver when a malicious sender sends $\boldsymbol{x}$. In the event that $|A| \in (t, r)$, the output of the receiver is distributed as if the input to the string-ROT were a pair of random strings. In the events $A \in \mathcal{A}_{\leq t}$ and $A \in \mathcal{A}_{\geq r}$, R outputs according to a random erasure from $\mathcal{A}_{\leq t}$ and $\mathcal{A}_{\geq r}$ respectively. This is indeed the distribution generated by the simulator and so this proves the theorem. □

Remark 1. The OWSC protocol is said to be Las-Vegas if it either aborts after returning $\perp$ or is correct conditioned on not aborting, *i.e.,* outputs $(\boldsymbol{a}_0, \perp)$ or $(\perp, \boldsymbol{a}_1)$ with equal probability. Suppose the Avg-SSS is Las-Vegas in the following sense. For every $A \in \mathcal{A}_r$, Recst either reconstructs the secret correctly or aborts after returning $\perp$. We can tweak the above OWSC protocol to output $\perp$ whenever $|A| \in (t, r)$ and to return whatever the Recst outputs when $|A| \geq r$ makes the OWSC protocol also Las-Vegas. This guarantees that in Theorem 4, if Avg-SSS is Las-Vegas, then OWSC protocol is also Las-Vegas. In the next section, we will construct an Avg-SSS scheme which is Las-Vegas.

3.3 Construction of Average Case Secret Sharing

In this section, we construct an average case secret sharing scheme. Our construction is similar to the construction of constant rate secret sharing schemes in [22]. The only difference is that the reconstruction and privacy properties are with respect to random corruptions, hence we are able to use randomized erasure correcting codes with better error parameters. Before we describe the construction, we provide the following definitions.

Definition 7. *A function* $\mathsf{Ext} : \{0,1\}^d \times \{0,1\}^n \to \{0,1\}^\ell$ *is a* (k,ϵ) *strong seeded extractor if for every random variable* X*, with alphabet* $\{0,1\}^n$ *and min-entropy* k*, when* $\boldsymbol{z} \xleftarrow{\$} \{0,1\}^d$ *and* $\boldsymbol{r} \xleftarrow{\$} \{0,1\}^\ell$*,*

$$\Delta\left((\mathsf{Ext}(\boldsymbol{z},X),\boldsymbol{z}),(\boldsymbol{r},\boldsymbol{z})\right) \le \epsilon.$$

A randomized map Ext^{-1} *is an inverter map of* Ext *if it maps* $\boldsymbol{z} \in \{0,1\}^d, \boldsymbol{s} \in \{0,1\}^\ell$ *to a sample from the uniform distribution over* $\{0,1\}^n$*, i.e.* U_n*, subject to* $(\mathsf{Ext}(\boldsymbol{z},U^n) = \boldsymbol{s})$*.*

The following lemma describes an improvement of Trevisan's extractor [27] due to Raz *et al.* [26]. The statement itself is from [22].

Lemma 1 *[22, Lemma 4]. There is an explicit linear* (k,ϵ) *strong seeded extractor* $\mathsf{Ext} : \{0,1\}^d \times \{0,1\}^n \to \{0,1\}^\ell$ *with* $d = \mathcal{O}(\log^3 n/\epsilon)$ *and* $\ell = k - \mathcal{O}(d)$*.*

The other component in our construction is an erasure correcting code. Since $\mathsf{Avg\text{-}SSS}$ allows for shared randomness between the sharing algorithm Share and the reconstruction algorithm Recst, we could use randomized erasure correcting codes.

Definition 8. *An* (n,k,r,ϵ)*-linear erasure correcting scheme* $(\mathsf{Enc},\mathsf{Dec})$ *consists of a linear encoder* $\mathsf{Enc} : \{0,1\}^k \to \{0,1\}^n$ *and a decoder* $\mathsf{Dec} : \{0,1\}^n \to \{0,1\}^k$ *such that, for all* $\boldsymbol{x} \in \{0,1\}^k$*,*

$$\Pr_{A \xleftarrow{\$} \mathcal{A}_r}\left(\mathsf{Dec}(\,\mathsf{Enc}(\boldsymbol{x})|_A) \neq \boldsymbol{x}\right) \;\le\; \epsilon.$$

Lemma 2. *For all* $k \le r \le n$*, there exist efficient* (n,k,r,ϵ)*-linear erasure correcting schemes with* $\epsilon = 2^{k-r}$*.*

A proof of the lemma is provided in the full version [1], where we will also argue that the erasure correcting code we construct is Las-Vegas *i.e.*, the decoder either aborts or correctly decodes the message. It can be verified that the $\mathsf{Avg\text{-}SSS}$ scheme we construct is Las-Vegas whenever the erasure correcting scheme is Las-Vegas.

Theorem 5. *For parameters* $t < n < n+d < r < N$ *and* ℓ, ϵ*, let* $\mathsf{Ext} : \{0,1\}^d \times \{0,1\}^n \to \{0,1\}^\ell$ *be a linear* $(n-t,\epsilon)$ *strong seeded extractor with inverter map* Ext^{-1}*. Let* $(\mathsf{Enc},\mathsf{Dec})$ *be a* $(N, n+d, r, \epsilon)$*-randomized linear erasure correcting code. Then,* $\langle \mathsf{Share}, \mathsf{Recst} \rangle$*, described below, is a* $(\ell, N, t, r, 8\epsilon)$*-*$\mathsf{Avg\text{-}SSS}$*:*

$$\mathsf{Share}(\boldsymbol{s}) = \mathsf{Enc}(\boldsymbol{z}||\mathsf{Ext}^{-1}(\boldsymbol{z},\boldsymbol{s})), \textit{where } \boldsymbol{z} \xleftarrow{\$} \{0,1\}^d,$$
$$\mathsf{Recst}(\boldsymbol{v}|_A) = \mathsf{Ext}(\boldsymbol{z}||\boldsymbol{x}), \textit{where } \boldsymbol{z}||\boldsymbol{x} = \mathsf{Dec}(\boldsymbol{v}|_A)$$

where $\boldsymbol{s} \in \{0,1\}^\ell$ *and* $A \subset [N]$*, when* $(\cdot||\cdot)$ *is the concatenation operator.*

Proof: We show that the scheme satisfies the reconstruction and privacy properties.

Reconstruction. By the performance guarantee of the error correcting code, for any $\boldsymbol{v} \in \{0,1\}^{n+d}$,

$$\Pr_{A \xleftarrow{\$} \mathcal{A}_r} (\mathsf{Dec}(\mathsf{Enc}(\boldsymbol{v})|_A) = \boldsymbol{v}) \geq 1 - \epsilon.$$

Hence, $\mathsf{Recst}(\boldsymbol{v}|_A) = \boldsymbol{s}$, for a random A, with probability $1 - \epsilon$.

Privacy. We use the following result from [22]:

Lemma 3 *[22, Lemma 13]. Let $\mathsf{Ext} : \{0,1\}^d \times \{0,1\}^n \to \{0,1\}^\ell$ be a linear (k, ϵ) strong extractor. Let $f_A : \{0,1\}^{n+d} \to \{0,1\}^t$ be an affine function with $t \leq n - k$. For any $\boldsymbol{s}, \boldsymbol{s}' \in \{0,1\}^\ell$, when $(Z, X) = (U_d, U_n)|(\mathsf{Ext}(U_d, U_n) = \boldsymbol{s})$ and $(Z', X') = (U_d, U_n)|(\mathsf{Ext}(U_d, U_n) = \boldsymbol{s}')$, we have*

$$\Delta\left(f_A(Z, X), f_A(Z', X')\right) \leq 8\epsilon.$$

Enc is a linear function and for any $A \subseteq [N]$ the restriction operator $(\cdot)|_A$ is a projection. Hence, for any $\boldsymbol{s} \in \{0,1\}^\ell$ and $A \subseteq [N]$ such that $|A| = t$, $\mathsf{Share}(\boldsymbol{s})|_A$ is an affine map with range $\{0,1\}^t$ applied to $(U_d, U_n)|(\mathsf{Ext}(U_d, U_n) = \boldsymbol{s})$. Ext used in the theorem is a $(n - t, \epsilon)$ extractor, hence the privacy follows directly from the above lemma. □

For any N and $\delta \in (0, 1/2)$, Lemma 1 guarantees an explicit linear $(N^\delta, \frac{1}{8N})$ strong seeded extractor $\mathsf{Ext} : \{0,1\}^d \times \{0,1\}^{\frac{N}{2}} \to \{0,1\}^\ell$ with $d = \mathcal{O}(\log^3 N)$ and $\ell = N^\delta - \mathcal{O}(\log^3 N)$. Furthermore, Lemma 2 guarantees a (N, k, r, ϵ)-linear erasure correcting code for $k = \frac{N}{2} + d$, $r = \frac{N}{2} + N^\delta$ and $\epsilon = \frac{1}{8N}$ (in fact, the lemma gives much better maximum error probability guarantees, but we would not need this). Note that both Ext^{-1} and $(\mathsf{Enc}, \mathsf{Dec})$ are efficient. Using this extractor and the erasure correcting scheme in Theorem 5, we obtain the following corollary.

Corollary 1. *For large enough N and $\delta \in (0, \frac{1}{2})$, when $\ell = \frac{N^\delta}{2}, t = \frac{N}{2} - N^\delta, r = \frac{N}{2} + N^\delta$ and $\epsilon = \frac{1}{N}$, there exists an efficient $(\ell, N, t, r, \epsilon)$-$\mathsf{Avg\text{-}SSS}$.*

Given such a $\mathsf{Avg\text{-}SSS}$, we appeal to the Theorem 4 to get the following theorem.

Theorem 6. *For $\delta \in (0, \frac{1}{2})$, there exists an efficient protocol that realizes (N, ϵ) secure OWSC for $\mathcal{C}^\ell_{\mathsf{ROT}}$ over $\mathcal{C}^1_{\mathsf{ROT}}$, with $\epsilon = \mathcal{O}(N^{\delta - \frac{1}{2}})$, and $\ell = \frac{N^\delta}{2}$. In particular, bit-ROT is complete for string-ROT with inverse-polynomial error.*

3.4 General Completeness of Bit-ROT with Inverse Polynomial Error

In the previous section, we showed that bit-ROT is complete for string-ROT with inverse-polynomial error. Garg *et al.* [17] (Theorem 11) showed that string-ROT is complete for arbitrary *finite* functionalities even for the case of malicious parties, where the (statistical) error is negligible in the ROT string length ℓ. Combined with our reduction from string-ROT to bit-ROT, this gives a similar completeness result for bit-ROT with inverse-polynomial error. Below we extend this to functions represented by branching programs and circuits, where in the latter case we need to settle for computational security using any (black-box) pseudorandom generator. Thus, assuming the existence of a one-way function, bit-ROT is complete with inverse-polynomial computational error for any polynomial-time computable functionality.

Theorem 7 (Bit-ROT is complete with inverse-polynomial error). *The bit-ROT channel $\mathcal{C}^1_{\mathsf{ROT}}$ is OWSC-complete, with* inverse-polynomial error*, for evaluating* circuits *with computational security against malicious parties, assuming a (black-box) pseudorandom generator. Moreover, replacing circuits by* branching programs*, the same holds* unconditionally *with inverse-polynomial* statistical *error.*

Proof: We start by addressing the simpler case of semi-honest parties. In this case, the computational variant follows by combining the reduction from string-ROT to bit-ROT with Yao's garbled circuit construction [31] in the following way. Given a randomized sender-receiver functionality $f(a; r)$, define a deterministic (two-way) functionality 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 f' with uniformly random choices of r_1, r_2, we get a computationally secure reduction of f to (chosen-input) string-OT where the receiver's inputs are random. Replacing the random choices of the receiver by the use of a string-ROT channel, we get a computational OWSC protocol for f over string-ROT using any (black-box) PRG. Finally, applying the reduction from string-ROT to bit-ROT with a suitable choice of parameters, we get the inverse-polynomial completeness result for circuits with semi-honest parties. A similar result for branching programs with statistical (and unconditional) security can be obtained using information-theoretic analogues of garbled circuits [16,18,20].

To obtain similar protocols for malicious parties, we appeal to a result of [19], which obtains an analogue of Yao's protocol with security against *malicious* parties by only making a black-box use of a pseudorandom generator along with parallel calls to a string-OT oracle.[3] (This result too has an unconditional version for the case of branching programs.) Unlike Yao's protocol, the protocol from [19] encodes the receiver's input before feeding it into the parallel OTs. However, this encoding has the property that a random receiver input is mapped to random OT choice bits. Thus, the same reduction as before applies. □

The unconditional part of Theorem 7 implies polynomial-time statistically-secure protocols (with inverse-polynomial error) for the complexity classes $\mathbf{NC}^1$ and **Logspace**. This is a vast generalization of the positive result for $\mathcal{C}^\ell_{\mathsf{ROT}}$. In the result for general circuits, the use of a pseudorandom generator is inherent given the current state of the art on constant-round secure computation.

[3] Note that the conceptually simpler approach of applying NIZK proofs is not applicable here, since in the setting of secure computation over noisy channels there is no public transcript to which such a proof can apply.

```
$\langle \mathsf{S}, \mathsf{R} \rangle (\boldsymbol{a}_0, \boldsymbol{a}_1)$
1. $(\boldsymbol{x}_0, \boldsymbol{x}_1) = \mathsf{S}(\boldsymbol{a}_0, \boldsymbol{a}_1)$.
2. Sample $\boldsymbol{s} \xleftarrow{\$} \{0,1\}^N$ and let $(\boldsymbol{y}_0, \boldsymbol{y}_1) = f^N_{\mathcal{C}^1_{\mathsf{ROT}}}((\boldsymbol{x}_0, \boldsymbol{x}_1), \boldsymbol{s})$.
3. $(\boldsymbol{b}_0, \boldsymbol{b}_1) = \mathsf{R}(\boldsymbol{y}_0, \boldsymbol{y}_1)$.
4. Output $((\boldsymbol{a}_0, \boldsymbol{a}_1), (\boldsymbol{x}_0, \boldsymbol{x}_1), (\boldsymbol{y}_0, \boldsymbol{y}_1), (\boldsymbol{b}_0, \boldsymbol{b}_1))$.
```

Fig. 1. Execution of a protocol $\langle \mathsf{S}, \mathsf{R} \rangle$ for OWSC of $\mathcal{C}^\ell_{\mathsf{ROT}}$ over $\mathcal{C}^1_{\mathsf{ROT}}$ channel. Here $\boldsymbol{a}_0, \boldsymbol{a}_1$ are the ℓ-bit input strings for $\mathcal{C}^\ell_{\mathsf{ROT}}$, the N-bit strings $\boldsymbol{x}_0, \boldsymbol{x}_1$ are the inputs for the N invocations of the $\mathcal{C}^1_{\mathsf{ROT}}$ channel, $\boldsymbol{y}_0, \boldsymbol{y}_1$ are the outputs of these N invocations, and $\boldsymbol{b}_0, \boldsymbol{b}_1$ are the outputs of $\mathcal{C}^\ell_{\mathsf{ROT}}$.

4 Impossibility of String-ROT from Bit-ROT with Negligible Error

In this section we show that string-ROT with negligible error is impossible to achieve from bit-ROT. Moreover, this holds even against a computationally bounded semi-honest adversary.

Theorem 8. *For sufficiently large N and $\ell \geq 2 \log N$, an $(N, \frac{1}{N^2})$ OWSC protocol for $\mathcal{C}^\ell_{\mathsf{ROT}}$ over $\mathcal{C}^1_{\mathsf{ROT}}$ is impossible even against semi-honest parties. In fact, the same holds even if one settles for OWSC with computational security. That is, there exists a polynomial $T = T(N)$ such that there is no computational $(N, T, \frac{1}{N^2})$ OWSC protocol for $\mathcal{C}^\ell_{\mathsf{ROT}}$ over $\mathcal{C}^1_{\mathsf{ROT}}$.*

Proof: $\mathcal{C}^1_{\mathsf{ROT}}$ may be equivalently described as a randomized function $f_{\mathcal{C}^1_{\mathsf{ROT}}}$ from the input of the channel and the internal randomness of the channel to the output of the channel. Formally, For $(x_0, x_1) \in \{0,1\} \times \{0,1\}$, and $s \in \{0,1\}$,

$$f_{\mathcal{C}^1_{\mathsf{ROT}}}((x_0, x_1), s) = \begin{cases} (x_0, \perp) & \text{if } s = 0, \\ (\perp, x_1) & \text{if } s = 1. \end{cases}$$

Observe that for all $(x_0, x_1) \in \{0,1\} \times \{0,1\}$, the following distributions are identical: (1) $\mathcal{C}^1_{\mathsf{ROT}}(x_0, x_1)$ and (2) Sample $s \xleftarrow{\$} \{0,1\}$ and output $f_{\mathcal{C}^1_{\mathsf{ROT}}}((x_0, x_1), s)$. Similarly, N invocations of $\mathcal{C}^1_{\mathsf{ROT}}$ are equivalent to the randomized function $f^N_{\mathcal{C}^1_{\mathsf{ROT}}}$ which on input $(\boldsymbol{x}_0, \boldsymbol{x}_1) \in \{0,1\}^N \times \{0,1\}^N$, samples $\boldsymbol{s} \xleftarrow{\$} \{0,1\}^N$ and outputs $(\boldsymbol{y}_0, \boldsymbol{y}_1)$, where $(\boldsymbol{y}_0(i), \boldsymbol{y}_1(i)) = f_{\mathcal{C}^1_{\mathsf{ROT}}}((\boldsymbol{x}_0(i), \boldsymbol{x}_1(i)), \boldsymbol{s}(i))$.

Suppose $\langle \mathsf{S}, \mathsf{R} \rangle$ is a $(N, \frac{1}{N^2})$ OWSC protocol for $\mathcal{C}^\ell_{\mathsf{ROT}}$ over $\mathcal{C}^1_{\mathsf{ROT}}$ channel. The joint distribution generated by this protocol for an input (pair of strings) $(\boldsymbol{a}_0, \boldsymbol{a}_1) \in \{0,1\}^\ell \times \{0,1\}^\ell$ is described in Fig. 1. The receiver's algorithm R can be assumed to be deterministic w.l.o.g. since we may fix the randomness in the decoder incurring only a constant hit to the $\epsilon = \frac{1}{N^2}$ parameter. This is because, for most values of $(\boldsymbol{y}_0, \boldsymbol{y}_1)$, R should decode one of the indices with low probability of error and should be almost entirely unsure of the other index. Refer to the full version [1] for a formal proof.

```
$\mathsf{M}(\boldsymbol{y}_0, \boldsymbol{y}_1)$
1. Compute $(\boldsymbol{b}_0, \boldsymbol{b}_1) = \mathsf{R}(\boldsymbol{y}_0, \boldsymbol{y}_1)$ (suppose $(\boldsymbol{b}_0, \boldsymbol{b}_1) = (\hat{\boldsymbol{a}}_0, \perp)$ w.l.o.g).
2. Compute $(\hat{\boldsymbol{y}}_0, \hat{\boldsymbol{y}}_1)$ as follows: Sample $j \xleftarrow{\$} [N]$. For $i \in \{0,1\}$ and $k \in [N] \setminus \{j\}$,
   set $\hat{\boldsymbol{y}}_i(k) = \boldsymbol{y}_i(k)$. If $\boldsymbol{y}_i(j) = \perp$, sample $\hat{\boldsymbol{y}}_i(j) \xleftarrow{\$} \{0,1\}$, and if $\boldsymbol{y}_i(j) \neq \perp$ then
   $\hat{\boldsymbol{y}}_i(j) = \perp$.
3. Compute $(\hat{\boldsymbol{b}}_0, \hat{\boldsymbol{b}}_1) = \mathsf{R}(\hat{\boldsymbol{y}}_0, \hat{\boldsymbol{y}}_1)$.
4. If $(\hat{\boldsymbol{b}}_0, \hat{\boldsymbol{b}}_1) = (\perp, \hat{\boldsymbol{a}}_1)$, then output $(\hat{\boldsymbol{a}}_0, \hat{\boldsymbol{a}}_1)$; else, abort.
```

Fig. 2. Execution of the machine M

In the sequel, for brevity, we would represent the tuples $(\boldsymbol{a}_0, \boldsymbol{a}_1), (\boldsymbol{x}_0, \boldsymbol{x}_1)$, $(\boldsymbol{y}_0, \boldsymbol{y}_1)$ and $(\boldsymbol{b}_0, \boldsymbol{b}_1)$ also by $\boldsymbol{a}, \boldsymbol{x}, \boldsymbol{y}$ and $\boldsymbol{b}$, respectively, whenever this does not cause confusion. For $(\boldsymbol{a}_0, \boldsymbol{a}_1) \in \{0,1\}^\ell \times \{0,1\}^\ell$, consider the joint distribution $\langle \mathsf{S}, R\rangle(\boldsymbol{a}_0, \boldsymbol{a}_1)$ described in Fig. 1. We now make some claims about this distribution.

Lemma 4. *There exists a set* $X \subseteq \{0,1\}^N \times \{0,1\}^N$ *such that* $\Pr(\boldsymbol{x} \in X) \geq 1 - \frac{2}{N}$ *and for all* $\boldsymbol{x} \in X$*,*

$$\Pr(\boldsymbol{b}_0 = \perp | \boldsymbol{x}) \geq \frac{1}{2} - \frac{1}{N} \quad \textit{and} \quad \Pr(\boldsymbol{b}_1 = \perp | \boldsymbol{x}) \geq \frac{1}{2} - \frac{1}{N}.$$

The lemma is a consequence of computational $\frac{1}{N^2}$-security against sender. Intuitively, the sender can guess the index of the message output by the receiver with substantial probability if $\Pr(\boldsymbol{x} \in X) < 1 - \frac{2}{N}$. Refer to the full version [1] for a formal proof.

We now design a machine M that guesses both $\boldsymbol{a}_0$ and $\boldsymbol{a}_1$ from $(\boldsymbol{y}_0, \boldsymbol{y}_1)$ with substantial probability, contradicting sender's privacy. On receiving $\boldsymbol{y}$, machine M uses the receiver's strategy $\mathsf{R}(\boldsymbol{y})$ to decode one of the messages, say $\boldsymbol{a}_i$, where i is either 1 or 0. It then computes $\boldsymbol{a}_{1-i}$ by 'guessing' a random neighbor of $\boldsymbol{y}$, say $\hat{\boldsymbol{y}}$ and computing $\mathsf{R}(\hat{\boldsymbol{y}})$. We would show that with substantial probability, $\mathsf{R}(\hat{\boldsymbol{y}})$ yields $\boldsymbol{a}_{1-i}$, breaking sender's privacy property. M is formally described in Fig. 2.

Analysis of M: We show that M outputs $(\boldsymbol{a}_0, \boldsymbol{a}_1)$ with substantial probability. We would analyze the output of the machine M for a fixed $\boldsymbol{x} \in X$, where X is as guaranteed by Lemma 4. Define function $f_{\boldsymbol{x}} : \{0,1\}^N \to \{0,1\}$ such that when $\boldsymbol{y} = f^N_{\mathcal{C}^1_{\mathsf{ROT}}}(\boldsymbol{x}, \boldsymbol{s})$, $f_{\boldsymbol{x}}(\boldsymbol{s}) = 1$ if $\mathsf{R}(\boldsymbol{y}) = (\boldsymbol{b}_0, \boldsymbol{b}_1)$ such that $\boldsymbol{b}_0 = \perp$ and 0 otherwise. We next observe a property of $f_{\boldsymbol{x}}$ which is a consequence of an isoperimetric inequality on Boolean hypercubes (Harper's Lemma). For binary strings $\boldsymbol{u}, \boldsymbol{v} \in \{0,1\}^n$, denote the Hamming distance between them by $|\boldsymbol{u} - \boldsymbol{v}|$.

Lemma 5. *For any function* $f : \{0,1\}^n \to \{0,1\}$*, if* $\Pr_{\boldsymbol{v} \xleftarrow{\$} \{0,1\}^n} (f(\boldsymbol{v}) = i) \geq \frac{1}{2}(1 - \frac{1}{\sqrt{n}})$ *for each* $i \in \{0,1\}$*, then* $\Pr_{\boldsymbol{v} \xleftarrow{\$} \{0,1\}^n} (\exists \tilde{\boldsymbol{v}} : |\boldsymbol{v} - \tilde{\boldsymbol{v}}| = 1 \textit{ and } f(\tilde{\boldsymbol{v}}) = 1 - f(\boldsymbol{v})) \geq \Omega(\frac{1}{\sqrt{n}})$.

In words, the lemma says that if f is a 2-coloring of the Boolean hypercube, where the colors are (almost) balanced, then a significant fraction of the nodes of the hypercube, have a neighbor of a different color.

By Harper's Lemma, Hamming balls have the smallest vertex boundary amongst all sets of the same probability. W.l.o.g, the probability of $f(\boldsymbol{v}) = 1$ is at most $\frac{1}{2}$ and at least $\frac{1}{2}(1 - \frac{1}{\sqrt{n}})$ and $\Pr_{\boldsymbol{v} \xleftarrow{\$} \{0,1\}^n}(|\boldsymbol{v} - \mathbf{0}| = \lfloor \frac{n}{2} \rfloor) \geq \frac{1}{2\sqrt{n}}$, where $\mathbf{0}$ is the all zero string. Hence the Hamming ball centered at $\mathbf{0}$ with probability at most $\frac{1}{2}$ and at least $\frac{1}{2}(1 - \frac{1}{\sqrt{n}})$ has strings with $\lfloor \frac{n}{2} \rfloor$ or $\lfloor \frac{n}{2} \rfloor - 1$ number of 1's in its boundary. Consequently, the size of this boundary is $\Omega(\frac{1}{\sqrt{n}})$.

For any $\boldsymbol{x} \in \{0,1\}^N \times \{0,1\}^N$, the input to M is $\boldsymbol{y} = f^N_{\mathcal{C}^1_{\mathsf{ROT}}}(\boldsymbol{x}, \boldsymbol{s})$, where $\boldsymbol{s} \xleftarrow{\$} \{0,1\}^N$. The process of generating $\hat{\boldsymbol{y}}$ in $M(\boldsymbol{y})$ is equivalent to the following process. Compute $(\hat{\boldsymbol{x}}_0, \hat{\boldsymbol{x}}_1)$ and $\hat{\boldsymbol{s}}$ as follows: Sample $j \leftarrow [N]$, set $\hat{\boldsymbol{s}}(j) = 1 - \boldsymbol{s}(j)$ and $(\hat{\boldsymbol{x}}_0(j), \hat{\boldsymbol{x}}_1(j)) \xleftarrow{\$} \{0,1\} \times \{0,1\}$. For all $k \neq j$, set $\hat{\boldsymbol{s}}(k) = \boldsymbol{s}(k)$ and $(\hat{\boldsymbol{x}}_0(k), \hat{\boldsymbol{x}}_1(k)) = (\boldsymbol{x}_0(k), \boldsymbol{x}_1(k))$. Compute $\hat{\boldsymbol{y}} = f^N_{\mathcal{C}^1_{\mathsf{ROT}}}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{s}})$. We make the following observations about the above process.

(i.) $\hat{\boldsymbol{s}}$ is uniformly distributed over $\{0,1\}^N$ and $|\boldsymbol{s} - \hat{\boldsymbol{s}}| = 1$.
(ii.) $\hat{\boldsymbol{y}} = f^N_{\mathcal{C}^1_{\mathsf{ROT}}}(\boldsymbol{x}, \hat{\boldsymbol{s}})$ with probability $\frac{1}{2}$.
(iii.) For any $\boldsymbol{x} \in X$, $\Pr(f_{\boldsymbol{x}}(\boldsymbol{s}) = 1 - f_{\boldsymbol{x}}(\hat{\boldsymbol{s}})) \geq \Omega(\frac{1}{N\sqrt{N}})$.

(i) follows from $\boldsymbol{s}$ being uniform in $\{0,1\}^N$ and $\hat{\boldsymbol{s}}$ being obtained by flipping the value of a random coordinate of $\boldsymbol{s}$. (ii) can be verified easily from the process description. When $\boldsymbol{x} \in X$ and $\boldsymbol{s} \xleftarrow{\$} \{0,1\}^N$, $\Pr(f_{\boldsymbol{x}}(\boldsymbol{s}) = i) \geq \frac{1}{2}(1 - \frac{1}{\sqrt{N}})$ for $i \in \{0,1\}$, by Lemma 4. Hence, by Harper's Lemma,

$$\Pr\left(\exists \tilde{\boldsymbol{s}} : |\boldsymbol{s} - \tilde{\boldsymbol{s}}| = 1 \text{ and } f_{\boldsymbol{x}}(\tilde{\boldsymbol{s}}) = 1 - f_{\boldsymbol{x}}(\boldsymbol{s})\right) \geq \Omega(\frac{1}{\sqrt{N}}).$$

Conditioned on the event that such a $\tilde{\boldsymbol{s}}$ exists, $\hat{\boldsymbol{s}} = \tilde{\boldsymbol{s}}$ with probability at least $\frac{1}{N}$. This proves (iii).

$(\boldsymbol{b}_0, \boldsymbol{b}_1)$ is said to be correct if it is either $(\boldsymbol{a}_0, \perp)$ or $(\perp, \boldsymbol{a}_1)$. Let E_1 be the event '$\boldsymbol{b} = \mathsf{R}\left(f^N_{\mathcal{C}^1_{\mathsf{ROT}}}(\boldsymbol{x}, \boldsymbol{s})\right)$ is correct'. Since $\boldsymbol{s}$ is uniform in $\{0,1\}^N$, by the correctness property, E_1 happens with probability $1 - \frac{1}{N^2}$. Let E_2 be the event '$\boldsymbol{b} = \mathsf{R}(f^N_{\mathcal{C}^1_{\mathsf{ROT}}}(\boldsymbol{x}, \hat{\boldsymbol{s}})$ is correct'. By (i), $\hat{\boldsymbol{s}}$ is also uniform in $\{0,1\}^N$, hence E_2 happens with probability $1 - \frac{1}{N^2}$. From (ii) and (iii) we conclude that, when $\boldsymbol{x} \in X$, $M(\boldsymbol{y})$ outputs $(\hat{\boldsymbol{a}}_0, \hat{\boldsymbol{a}}_1)$ (instead of aborting) with probability $\Omega(\frac{1}{N\sqrt{N}})$. Since $\boldsymbol{x} \in X$ happens with probability $(1 - \frac{2}{N})$, we may conclude that with probability at least $(1 - \frac{2}{N})\Omega(\frac{1}{N\sqrt{N}})$, the following event E_3 occurs: $\hat{\boldsymbol{y}} = f^N_{\mathcal{C}^1_{\mathsf{ROT}}}(\boldsymbol{x}, \hat{\boldsymbol{s}})$ and M outputs $(\hat{\boldsymbol{a}}_0, \hat{\boldsymbol{a}}_1)$. In the event $E_1 \cap E_2 \cap E_3$, the machine M guesses the input correctly and outputs $(\boldsymbol{a}_0, \boldsymbol{a}_1)$. By a union bound, $E_1 \cap E_2 \cap E_3$ happens with probability $(1 - \frac{2}{N})\Omega(\frac{1}{N\sqrt{N}}) - \frac{2}{N^2}$. Hence, M predicts $(\boldsymbol{a}_0, \boldsymbol{a}_1)$ with probability $\Omega(\frac{1}{N\sqrt{N}})$. This is a contradiction since, when $\ell = 2\log N$ and the protocol is $\frac{1}{N^2}$-secure, the adversary can succeed in guessing both inputs with at most $2^{-2\log N} + \frac{1}{N^2} = \frac{2}{N^2}$ probability. This proves the theorem. □

4.1 Extending Impossibility to All Finite Channels

In this section we show that the negative result from the previous section applies not only to bit-ROT but, in fact, to all finite channels. W.l.o.g we consider channels with rational conditional probability matrices. We begin by modeling an arbitrary finite channel as a randomized function.

Definition 9. *Consider a channel $\mathcal{C} : \mathcal{X} \to \mathcal{Y}$ with rational conditional distribution matrix. We define the* states of $\mathcal{C}$ *as a finite set* $\mathcal{C}.\mathsf{states}$ *and the* channel function $f_{\mathcal{C}} : \mathcal{X} \times \mathcal{C}.\mathsf{states} \to \mathcal{Y}$*, such that for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$,*

$$\Pr(\mathcal{C}(x) = y) = \Pr_{s \overset{\$}{\leftarrow} \mathcal{C}.\mathsf{states}} (f(x, s) = y).$$

We emphasize that our channels are all memoryless, and that "states" in this context should be interpreted as the internal randomness of the channel used in each invocation (uniform distribution over the set $\mathcal{C}.\mathsf{states}$).

The existence of $\mathcal{C}.\mathsf{states}$ and $f_{\mathcal{C}}$ is proved in the full version [1]. For the convenience of modeling we have defined $f_{\mathcal{C}}$ in such a way that the state is chosen uniformly at random from $\mathcal{C}.\mathsf{states}$. Given the above definition, for a fixed input $x \in \mathcal{X}$, the channel $\mathcal{C}$ essentially samples a state uniformly from $\mathcal{C}.\mathsf{states}$ and deterministically maps x to the output y. This model motivates our next observation about multiple uses of the channel.

For a finite N, let $\boldsymbol{x} = (x_1, \ldots, x_N) \in \mathcal{X}^N$ and let $\boldsymbol{y} = (y_1, \ldots, y_N) \in \mathcal{Y}^N$ be the output of N independent uses of $\mathcal{C}$ with input $\boldsymbol{x}$. Then the distribution $(\boldsymbol{x}, \boldsymbol{y})$ can be thought to be generated by the following equivalent process: Sample $\boldsymbol{s} = (s_1, \ldots, s_N) \leftarrow (\mathcal{C}.\mathsf{states})^N$ and for $i = 1, \ldots, N$, compute $y_i = f_{\mathcal{C}}(x_i, s_i)$.

Before we state the next lemma, we set up some notation for generalizing distance between strings over finite alphabets. For $\boldsymbol{x}, \tilde{\boldsymbol{x}} \in \mathcal{X}^n$, $|\boldsymbol{x} - \tilde{\boldsymbol{x}}| = 1$ if they differ in exactly one of the n coordinates, *i.e.,* there exists $i \in [n]$ such that $x_i \neq \tilde{x}_i$ and $x_j = \tilde{x}_j$ for all $j \neq i$. The following lemma is an extension of the isoperimetric bound in Lemma 5 that we used for proving Theorem 8. The lemma is formally proved in the full version [1].

Lemma 6. *Let $\mathcal{X}$ be a finite set such that $|\mathcal{X}| = 2^k$ for some k. For any function $f : \mathcal{X}^n \to \{0, 1\}$, if $\Pr_{\boldsymbol{x} \overset{\$}{\leftarrow} \mathcal{X}^n}(f(\boldsymbol{x}) = i) \geq \frac{1}{2} - \frac{1}{\sqrt{k \cdot n}}$, for each $i \in \{0, 1\}$, then*

$$\Pr_{x \overset{\$}{\leftarrow} \mathcal{X}^n} (\exists \tilde{\boldsymbol{x}} : |\boldsymbol{x} - \tilde{\boldsymbol{x}}| = 1 \textit{ and } f(\tilde{\boldsymbol{x}}) = 1 - f(\boldsymbol{x})) \geq \Omega\left(\frac{1}{\sqrt{k \cdot n}}\right).$$

We are now ready to state the generalization of Theorem 8.

Theorem 9. *Let $\mathcal{C}$ be a finite channel. For sufficiently large N and $\ell \geq 2 \log N$, an $(N, \frac{1}{N^2})$ OWSC protocol for $\mathcal{C}_{\mathsf{ROT}}^{\ell}$ over $\mathcal{C}$ is impossible even against semi-honest parties. In fact, the same holds even if one settles for computational security.*

Proof: We proceed in the same way we showed the impossibility in Theorem 8. To prove a contradiction, suppose $\langle \mathsf{S}, \mathsf{R} \rangle$ is a $(N, \frac{1}{N^2})$ OWSC protocol for $\mathcal{C}_{\mathsf{ROT}}^{\ell}$

```
$\langle \mathsf{S}, \mathsf{R} \rangle(\boldsymbol{a}_0, \boldsymbol{a}_1)$
1. $\boldsymbol{x} \overset{\$}{\leftarrow} S(\boldsymbol{a}_0, \boldsymbol{a}_1)$.
2. Sample $\boldsymbol{r} \overset{\$}{\leftarrow} (\mathcal{C}.\mathsf{states})^N$.
3. Compute $\boldsymbol{y}$ where $y_i = f_{\mathcal{C}}(x_i, r_i)$.
4. $(\boldsymbol{b}_0, \boldsymbol{b}_1) = \mathsf{R}(\boldsymbol{y})$.
5. Output $((\boldsymbol{a}_0, \boldsymbol{a}_1), \boldsymbol{x}, \boldsymbol{y}, (\boldsymbol{b}_0, \boldsymbol{b}_1))$.
```

Fig. 3. Execution of a protocol $\langle \mathsf{S}, \mathsf{R} \rangle$ for OWSC of $\mathcal{C}^{\ell}_{\mathsf{ROT}}$ over channel $\mathcal{C} : \mathcal{X} \to \mathcal{Y}$. Here $\boldsymbol{a}_0, \boldsymbol{a}_1$ are the ℓ-bit input strings for $\mathcal{C}^{\ell}_{\mathsf{ROT}}$, the N-bit strings $\boldsymbol{x}_0, \boldsymbol{x}_1$ are the inputs for the N invocations of $\mathcal{C}$, $\boldsymbol{y}_0, \boldsymbol{y}_1$ are the outputs of these N invocations, and $\boldsymbol{b}_0, \boldsymbol{b}_1$ are the outputs of $\mathcal{C}^{\ell}_{\mathsf{ROT}}$.

```
$\mathsf{M}(\boldsymbol{y})$
1. Compute $(\boldsymbol{b}_0, \boldsymbol{b}_1) = \mathsf{R}(\boldsymbol{y})$.
2. Sample $i \overset{\$}{\leftarrow} [N], x \overset{\$}{\leftarrow} \mathcal{X}, r \overset{\$}{\leftarrow} \mathcal{C}.\mathsf{states}$.
3. Compute $\tilde{\boldsymbol{y}}$, where $\tilde{y}_i = f_{\mathcal{C}}(x, r)$ and $\tilde{y}_j = y_j$ for all $j \neq i$.
4. Compute $(\tilde{\boldsymbol{b}}_0, \tilde{\boldsymbol{b}}_1) = \mathsf{R}(\tilde{\boldsymbol{y}})$.
5. If $(\boldsymbol{b}_1, \tilde{\boldsymbol{b}}_0) = (\perp, \perp)$, output $(\boldsymbol{b}_0, \tilde{\boldsymbol{b}}_1)$ and if $(\boldsymbol{b}_0, \tilde{\boldsymbol{b}}_1) = (\perp, \perp)$, output $(\tilde{\boldsymbol{b}}_0, \boldsymbol{b}_1)$; else,
   abort.
```

Fig. 4. Execution of the machine M

over $\mathcal{C}$. The joint distribution, generated by the protocol for input $(\boldsymbol{a}_0, \boldsymbol{a}_1) \in \{0,1\}^{\ell} \times \{0,1\}^{\ell}$, is described in Fig. 3. We would use a machine M similar to the one used in the proof of Theorem 8 to guess both $\boldsymbol{a}_0$ and $\boldsymbol{a}_1$ from the received $\boldsymbol{y}$ with substantial probability, contradicting sender's privacy. The machine is described in Fig. 4. Intuitively, M tries to obtain one string from $\boldsymbol{y}$ (due to correctness of the ROT protocol) and the other string, by changing one item of $\boldsymbol{y}$, and hoping to get into a case where the receiver outputs the other string.

Analysis of M. We show that M outputs $(\boldsymbol{a}_0, \boldsymbol{a}_1)$ with substantial probability. As observed in Lemma 4, since the protocol is $\frac{1}{N^2}$-secure, due to the receiver's privacy property, there exists a set $X \subseteq \mathcal{X}^N$ such that $\Pr(\boldsymbol{x} \in X) \geq 1 - \frac{2}{N}$ and for all $\boldsymbol{x} \in X$,

$$P(\boldsymbol{b}_0 = \perp | \boldsymbol{x}) \geq \frac{1}{2} - \frac{1}{N} \quad \text{and} \quad P(\boldsymbol{b}_1 = \perp | \boldsymbol{x}) \geq \frac{1}{2} - \frac{1}{N}.$$

Fix an $\boldsymbol{x} \in X$. Recall that for a fixed $\boldsymbol{x} \in \mathcal{X}^N$, the output $\boldsymbol{y}$ of the channel is a deterministic function of the state of the channel $\boldsymbol{r}$, *i.e.*, $\boldsymbol{y} = f^N_{\mathcal{C}}(\boldsymbol{x}, \boldsymbol{r})$. Here $f^N_{\mathcal{C}}(\boldsymbol{x}, \boldsymbol{r})$ outputs $\boldsymbol{y}$ such that $y_i = f_{\mathcal{C}}(x_i, r_i)$. Define function $f_{\boldsymbol{x}} : (\mathcal{C}.\mathsf{states})^N \to \{0,1\}$ as follows: for $\boldsymbol{r} \in (\mathcal{C}.\mathsf{states})^N$, when $f^N_{\mathcal{C}}(\boldsymbol{x}, \boldsymbol{r}) = \boldsymbol{y}$ and $(\boldsymbol{b}_0, \boldsymbol{b}_1) = \mathsf{R}(\boldsymbol{y})$, then $f_{\boldsymbol{x}}(\boldsymbol{r}) = 0$ if $\boldsymbol{b}_0 = \perp$ and $f_{\boldsymbol{x}}(\boldsymbol{r}) = 1$ otherwise. Hence, for all $\boldsymbol{x} \in X$, function $f_{\boldsymbol{x}}$ is such that $\Pr_{\boldsymbol{r} \overset{\$}{\leftarrow} (\mathcal{C}.\mathsf{states})^N}(f(\boldsymbol{x}) = i) \geq \frac{1}{2} - \frac{1}{N}$ for $i = 0, 1$. When $\frac{1}{N^2} \leq \frac{1}{k \cdot N}$, invoking Lemma 6,

$$\Pr_{\boldsymbol{r} \xleftarrow{\$} (\mathcal{C}.\mathsf{states})^N} (\exists \tilde{\boldsymbol{r}} : |\boldsymbol{r} - \tilde{\boldsymbol{r}}| = 1 \text{ and } f_{\boldsymbol{x}}(\boldsymbol{r}) = 1 - f_{\boldsymbol{x}}(\tilde{\boldsymbol{r}})) \geq \Omega\left(\frac{1}{\sqrt{k \cdot n}}\right).$$

Note that $\boldsymbol{y}$ is generated by $\boldsymbol{x}$ and a random state $\boldsymbol{r} \leftarrow (\mathcal{C}.\mathsf{states})^N$ (see Fig. 3). On input $\boldsymbol{y}$, machine M can be equivalently thought to be computing $\tilde{\boldsymbol{y}}$ as $f_{\mathcal{C}}^N(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{r}})$, where $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{r}}$ can be described as follows: Choose a random coordinate $i \xleftarrow{\$} [N]$ (see Fig. 4) and $\tilde{\boldsymbol{x}}$ is computed as $\tilde{x}_i \xleftarrow{\$} \mathcal{X}$ and $\tilde{x}_j = x_j$ for $j \neq i$ and $\tilde{\boldsymbol{r}}$ is computed as $\tilde{r}_i \xleftarrow{\$} \mathcal{C}.\mathsf{states}$ and $\tilde{r}_j = r_j$ for $j \neq i$. We make the following simple observations.

(i). $\tilde{\boldsymbol{r}}$ is distributed uniformly in $(\mathcal{C}.\mathsf{states})^N$ and $|\boldsymbol{r} - \tilde{\boldsymbol{r}}| = 1$.
(ii). $\Pr(\tilde{\boldsymbol{x}} = \boldsymbol{x}) = \frac{1}{|\mathcal{X}|}$.
(iii). With probability $\Omega(\frac{1}{N\sqrt{N}})$, we have $f_{\boldsymbol{x}}(\tilde{\boldsymbol{r}}) = 1 - f_{\boldsymbol{x}}(\boldsymbol{r})$.

Here, (i) and (ii) are clear from the process. For any $\boldsymbol{s} \in \{0,1\}^N$ such that $|\boldsymbol{r} - \boldsymbol{s}| = 1$, $\tilde{\boldsymbol{r}} = \boldsymbol{s}$ with probability $\frac{1}{N \cdot |\mathcal{C}.\mathsf{states}|} = \frac{1}{2^k \cdot N}$. Hence, when $\boldsymbol{x} \in X$ and $\boldsymbol{r} \xleftarrow{\$} \{0,1\}^N$, the probability of the event '$f_{\boldsymbol{x}}(\boldsymbol{r}) = 1 - f_{\boldsymbol{x}}(\tilde{\boldsymbol{r}})$' is at least $\frac{1}{2^k \cdot N} \cdot \Omega(\frac{1}{\sqrt{k \cdot N}}) = \Omega(\frac{1}{N\sqrt{N}})$.

We are now ready to show that M outputs $\boldsymbol{a}_0, \boldsymbol{a}_1$ with substantial probability. Let E_1 be the event '$\tilde{\boldsymbol{x}} = \boldsymbol{x}$ and $f_{\boldsymbol{x}}(\tilde{\boldsymbol{r}}) = 1 - f_{\boldsymbol{x}}(\boldsymbol{r})$'. We have already established that conditioned on any $\boldsymbol{x} \in X$, the event E_1 occurs with probability $\Omega(\frac{1}{N\sqrt{N}})$. Since $\Pr(\boldsymbol{x} \in X) \geq 1 - \frac{2}{N}$, the probability of E_1 is at least $(1 - \frac{2}{N}) \cdot \Omega(\frac{1}{N\sqrt{N}})$. Let E_2 be the event '$\mathsf{R}(f_{\mathcal{C}}^N(\boldsymbol{x}, \boldsymbol{r}))$ is correct' and E_3 be the event '$\mathsf{R}(f_{\mathcal{C}}^N(\boldsymbol{x}, \tilde{\boldsymbol{r}}))$ is correct'. Since $\boldsymbol{r}$ and $\tilde{\boldsymbol{r}}$ are uniformly distributed in $\{0,1\}^N$, by the correctness of the protocol, E_2 and E_3 occur with probability at least $1 - \frac{1}{N^2}$. In the event $E_1 \cap E_2 \cap E_3$, the machine M guesses the input correctly and outputs $(\boldsymbol{a}_0, \boldsymbol{a}_1)$. By a union bound, $E_1 \cap E_2 \cap E_3$ happens with probability $(1 - \frac{2}{N})\Omega(\frac{1}{N\sqrt{N}}) - 2\frac{1}{N^2}$. Hence, M predicts $(\boldsymbol{a}_0, \boldsymbol{a}_1)$ with probability $\Omega(\frac{1}{N\sqrt{N}})$. Note that this is a contradiction since, when $\ell = 2 \log N$, such a machine should not exist when the protocol is $\frac{1}{N^2}$-secure. This proves the theorem. □

5 Zero-Knowledge Proofs from Any Non-trivial Channel

In this section, we characterize finite channels that allow OWSC of zero-knowledge proofs of knowledge. Our result states that *zero-knowledge proofs of knowledge* (ZK PoK) can be realized with OWSC over a channel if and only if the channel is non-trivial. A trivial channel is one which is *essentially equivalent* (as formalized below) to a noiseless channel, when used by actively corrupt senders.

Theorem 10 (Informal). *Given a language* $L \in \mathrm{NP} \backslash \mathrm{BPP}$*, an* $\mathsf{OWSC}/\mathcal{C}$ *zero-knowledge protocol for* L *exists if and only if* $\mathcal{C}$ *is non-trivial.*

Previously, this result was known only for two special channels, namely, BEC and BSC [17]. To extend it to all non-trivial channels, we need to take a closer look at the properties of abstract channels. To understand what a non-trivial channel is, it is helpful to geometrically model a channel as we do below.

Redundant Inputs, Core and Trivial Channels. Given a channel $\mathcal{C} : \mathcal{X} \to \mathcal{Y}$, for each input $\alpha \in \mathcal{X}$, define a $|\mathcal{Y}|$-dimensional vector $\boldsymbol{\mu}_\alpha$ with coordinates indexed by elements of $\mathcal{Y}$, such that $\boldsymbol{\mu}_\alpha(\beta) = \Pr(\mathcal{C}(\alpha) = \beta)$ for each $\beta \in \mathcal{Y}$. We define the convex polytope $R_\mathcal{C}$ associated with $\mathcal{C}$ as the convex hull of the vectors $\{\boldsymbol{\mu}_\alpha | \alpha \in \mathcal{X}\}$.

Any $\alpha \in \mathcal{X}$ such that $\boldsymbol{\mu}_\alpha$ is a convex combination of $\{\boldsymbol{\mu}_{\alpha'} | \alpha' \in \mathcal{X} \setminus \{\alpha\}\}$ is a *redundant* input, because a sender could perfectly simulate the use of α with a linear combination of other inputs, without being detected (and possibly obtaining more information about the output at the receiver's end). Geometrically, a redundant input corresponds to a point in the interior of (possibly a face of) $R_\mathcal{C}$ (or multiple inputs that share the same vertex of the polytope). Consider a new channel $\widehat{\mathcal{C}}$ without any redundant inputs, obtained by restricting $\mathcal{C}$ to a subset of inputs, one for each vertex of the convex hull. $\widehat{\mathcal{C}}$ is called the *core* of $\mathcal{C}$.[4]

We note that $\mathcal{C} : \mathcal{X} \to \mathcal{Y}$ can be securely realized over $\widehat{\mathcal{C}} : \widehat{\mathcal{X}} \to \mathcal{Y}$, with security (in fact, UC security) against *active* adversaries. In this protocol, when the sender is given an input $\alpha \in \mathcal{X} \setminus \widehat{\mathcal{X}}$, it samples an input α' from $\widehat{\mathcal{X}}$ according to a distribution that results in the same channel output distribution as produced by α (this is always possible since $R_\mathcal{C}$ is the same as $R_{\widehat{\mathcal{C}}}$). Correctness (when both parties are honest) and security against a corrupt receiver are immediate from the fact that the output distribution is correct; security against a corrupt sender follows from the fact that its only action in the protocol – sending an input to $\widehat{\mathcal{C}}$– can be carried out as it is in the ideal world involving $\mathcal{C}$, with the same effect. This means that there is a secure OWSC protocol over $\mathcal{C}$ only if such a protocol exists over $\widehat{\mathcal{C}}$. In turn, since $\widehat{\mathcal{C}}$ has no redundant inputs, it suffices to characterize which channels among those without redundant inputs, admit ZK proofs.

A channel without any redundant inputs is *trivial* if the output distributions for each of its input symbols are disjoint from each other. Such a channel corresponds to a noiseless channel, as the receiver always learns exactly the symbol that was input to the channel. Over a noiseless channel, zero-knowledge proofs exist only for languages in BPP.

Our main goal then, is to show that if a channel $\mathcal{C}$ without redundant inputs is non-trivial, then every language in NP has an $\mathsf{OWSC}/\mathcal{C}$ zero-knowledge protocol. We start by providing some intuition about how we achieve this.

5.1 Intuition Behind the Construction

The ZK protocol involves sending many independently generated copies of an Oblivious ZK-PCP over the channel, after encoding it appropriately; the verifier

[4] The notions of redundancy and core were defined more generally in [21], in the context of 2-party functionalities where both parties have inputs and outputs. Here we present simpler definitions that suffice for the case of channels.

tests the proof using a carefully designed scheme before accepting it. The encoding and testing are designed to ensure, on one hand, erasure of a large fraction of the bits in the proofs (to guarantee zero-knowledge) and, on the other hand, delivery of sufficiently many bits so that the verifier can detect if the transmitted proof is incorrect (for soundness). At a high-level, the transmission and testing of the proof takes place over three "layers": (i) an inner-most *binary channel layer* at the bottom, (ii) an *erasure layer* over it, and (iii) an outer *PCP layer*.

The inner-most and outer-most layers are used to ensure soundness while the middle and outer-most layers work in tandem to obtain the zero-knowledge property.

Binary-Input Channel Layer. A given channel $\mathcal{C}$ (without redundant inputs) may have an arbitrary number of inputs, which may provide the prover with room for cheating in the protocol. The binary-input channel layer involves a mechanism to enforce that the prover (mostly) uses only a prescribed pair of distinct input symbols α_0 and α_1. We require that over several uses of the channel, if the sender uses a different symbol significantly often, then the receiver can detect this from the empirical distribution of the output symbols it received. This requires that the sender cannot simulate the effect of sending a combination of these two symbols by using a combination of some other symbols. Using the geometric interpretation of the channel, this corresponds to the requirement that the line segment connecting the two vertices $\boldsymbol{\mu}_{\alpha_0}$ and $\boldsymbol{\mu}_{\alpha_1}$ of the polytope $R_{\mathcal{C}}$ actually form an *edge* of the polytope. However, for the erasure layer (described below) to work we require that the output distributions of α_0 and α_1 have intersecting supports. In Lemma 7, we show that in any non-trivial channel $\mathcal{C}$ (without redundant inputs), there indeed exist α_0, α_1 which satisfy both these requirements simultaneously. Then, in Lemma 8, we show that there is a statistical test—whose parameters are determined by the geometry of the polytope $R_{\mathcal{C}}$—that can distinguish between a sender who sends a long sequence of these two symbols from a sender who uses other symbols in a significant fraction of positions.

Erasure Layer. We can obtain a non-zero probability of *perfect erasure* by encoding 0 as the pair (α_0, α_1) and 1 as the pair (α_1, α_0), to be transmitted over two independent uses of the channel $\mathcal{C}$. Since there is some symbol β such that both $q_0 := \Pr(\mathcal{C}(\alpha_0) = \beta) > 0$ and $q_1 := \Pr(\mathcal{C}(\alpha_1) = \beta) > 0$, the probability of the receiver obtaining (β, β) is the same positive value $q_0 q_1$, whether 0 or 1 is sent as above.[5] Hence, one can interpret the view of the receiver as obtained by post-processing the output of a BEC with erasure probability $q_0 q_1$, so that the erasure symbol is mapped to the outcome (β, β).

At the receiver's end, we use a maximum likelihood decoding, that always outputs a bit (rather than allowing an erasure symbol as well); if the likelihood of a received pair of symbols is the same for 0 and 1, it is decoded as a uniformly random bit. Note that if the sender sends a pair (α_0, α_0) or (α_1, α_1), then the decoding strategy will have the same effect as when the sender sends the encoding

[5] This is essentially identical to the Von Neumann extractor trick.

of a random bit – namely, it will be decoded to a uniformly random bit. Thus, the net effect of these two layers is that the prover communicates with the verifier using bits sent via a BSC, except for a few positions where the sender may arbitrarily control the channel characteristics. While the receiver's view includes more information than the output of the BSC, it can be entirely simulated from the output of a BEC.

PCP Layer. At the outer-most layer, our proof resembles the OWSC/BSC ZK protocol of [17], but is in fact somewhat simpler.[6] Here, the prover simply sends several independently generated copies of an Oblivious ZK-PCP (routed through the inner layers discussed above). As we noted above, the view of the receiver is obtained by post-processing the output of a BEC; hence, by choosing the parameters of the ZK-PCP appropriately, we can ensure that the receiver's view can be statistically simulated.

Ensuring soundness requires more work. The receiver, after obtaining the bits decoded from the inner layers (provided that no deviation was detected at the inner-most layer), can try to execute the PCP verification on each proof. However, it cannot reject the proof on encountering a single proof that fails the verification, because, even if the prover is honest, the channel can introduce errors in the received bits. As such, the verifier should be prepared to tolerate a certain probability of error. One may expect that if the proof was originally incorrect, then the probability of error would increase. However, this intuition is imprecise: it is plausible that a wrong proof can match or even surpass some honest proofs in the probability of passing the PCP verification.

To deal with this, we note that it is not necessary to carry out the original PCP verification test on the received bits, but rather one should design a statistical test that separates all correct proofs from incorrect proofs, *as received through the inner layers*. We show that for any predicate used by the original PCP verifier, there is an *error-score* one can assign to the bits decoded from the BSC, so that the *expected* error-score of the decoded bits is lower when they originally satisfy the PCP verifier's predicate. The verifier accepts or rejects the proof by computing the empirical average of the score across all repetitions of the proof, and thresholding it appropriately.

We remark that our scoring scheme and its analysis are more direct, and perhaps simpler, compared to the one in [17]. An additional subtlety that arises in our case is that there can be a few positions where the inner layers do not constitute the BSC that we try to enforce. Nevertheless, the above approach remains robust to such deviations, by ensuring that the scores come from a suitably bounded range.

[6] In [17], an encoding scheme was used to argue that with some probability, the bits sent through the BSC are "erased." But this encoding turns out to be redundant, as a BSC implicitly guarantees erasure: Concretely, a BSC with error probability p can be simulated by post-processing a BEC with erasure probability $2p$. The post-processing corresponds to decoding the erasure symbol as a uniformly random bit.

5.2 Properties of Non-trivial Channels

The following lemma shows that if $\mathcal{C}$ is non-trivial and without redundant inputs, there is a pair of input symbols α_0, α_1 with properties that we can use to enforce binary-input channel layer in Lemma 8 and to realize erasure channel layer in Lemma 9. Proofs of these lemmas are provided in the full version [1] (Fig. 5).

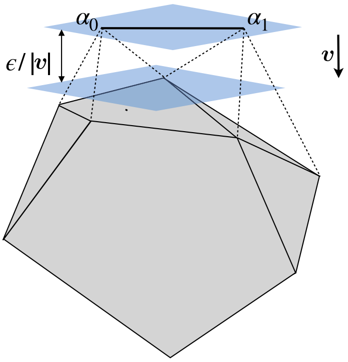

Fig. 5. Illustration of condition (ii) in Lemma 7. The polytope $R_{\mathcal{C}}$ is illustrated here. Since $\mathcal{C}$ has no redundant symbols, there is a bijection between vertices of $R_{\mathcal{C}}$ and the input symbols of the channel. The edge between $\boldsymbol{\mu}_{\alpha_0}$ and $\boldsymbol{\mu}_{\alpha_1}$ is highlighted. The solid part is the convex hull of the vertices other than $\boldsymbol{\mu}_{\alpha_0}$ and $\boldsymbol{\mu}_{\alpha_1}$. By the separating hyperplane theorem [7], there exists a vector $\boldsymbol{v} \in [-1,1]^{\mathcal{Y}}$ and $\epsilon > 0$ as illustrated. In Lemma 8, the existence of $\boldsymbol{v}, \epsilon$ is used to devise the statistical test that enforces the binary input channel layer. That $\boldsymbol{\mu}_{\alpha_0}$ and $\boldsymbol{\mu}_{\alpha_1}$ have intersecting support is used in realizing the erasure layer.

Lemma 7. *If $\mathcal{C} : \mathcal{X} \to \mathcal{Y}$ without redundant inputs is non-trivial, then there exist distinct symbols $\alpha_0, \alpha_1 \in \mathcal{X}$, $\boldsymbol{v} \in [-1,1]^{\mathcal{Y}}$ and $\epsilon > 0$ with the following properties:*

(i) $\exists y \in \mathcal{Y}$ such that $\boldsymbol{\mu}_{\alpha_0}(y), \boldsymbol{\mu}_{\alpha_1}(y) > 0$.
(ii) $\langle \boldsymbol{\mu}_{\alpha_0}, \boldsymbol{v} \rangle = \langle \boldsymbol{\mu}_{\alpha_1}, \boldsymbol{v} \rangle$, and for all $\alpha \in \mathcal{X} \setminus \{\alpha_0, \alpha_1\}$, $\langle \boldsymbol{\mu}_{\alpha}, \boldsymbol{v} \rangle - \langle \boldsymbol{\mu}_{\alpha_0}, \boldsymbol{v} \rangle \geq \epsilon$.

In the next lemma, we show that, over several uses of $\mathcal{C}$, a sender who uses only α_0, α_1 described in the previous lemma, can be distinguished from one that uses other symbols (different than α_0, α_1) significantly often, using the empirical distribution of the output symbols. Let histogram of a vector $\boldsymbol{y} \in \mathcal{Y}^m$ be defined as $\mathsf{hist}_{\boldsymbol{y}}(\beta) = \frac{1}{m}|\{i \in [m] : y_i = \beta\}|$ for all $\beta \in \mathcal{Y}$. The following function is a statistical test that achieves this: $f_m(\boldsymbol{y}) = \langle \mathsf{hist}_{\boldsymbol{y}}, \boldsymbol{v} \rangle - \langle \boldsymbol{\mu}_{\alpha_0}, \boldsymbol{v} \rangle$.

Lemma 8. *If a channel $\mathcal{C}$ without redundant inputs is non-trivial, then there exist $\alpha_0, \alpha_1 \in \mathcal{X}, \epsilon > 0$ and functions $f_m : \mathcal{Y}^m \to \mathbb{R}$, for $m \in \mathbb{N}$, such that, for all $\lambda > 0$, when $\boldsymbol{x} \in \mathcal{X}^m, t = |\{i \in [m] : x_i \notin \{\alpha_0, \alpha_1\}\}|$ and $\boldsymbol{y} = \mathcal{C}(\boldsymbol{x})$,*

$\langle \mathsf{Enc}, \mathsf{Dec} \rangle(a)$

For channel $\mathcal{C} : \mathcal{X} \to \mathcal{Y}$, choose $\alpha_0, \alpha_1 \in \mathcal{X}$ that satisfy the conditions in Lemma 7. When $a \in \{0, 1\}$,

1. $\mathsf{Enc}(a) = (x_0, x_1)$ where $x_0 = \alpha_a$ and $x_1 = \alpha_{1-a}$.
2. $(y_0, y_1) = \mathcal{C}(x_0, x_1)$.
3. $\mathsf{Dec}(y_0, y_1) = \begin{cases} b \text{ if } \Pr\left[\mathcal{C}(\alpha_b, \alpha_{1-b}) = (y_0, y_1)\right] > \Pr\left[\mathcal{C}(\alpha_{1-b}, \alpha_b) = (y_0, y_1)\right], \\ 0 \text{ (resp. 1) w. p. } \frac{1}{2} \text{ if } \Pr\left[\mathcal{C}(\alpha_0, \alpha_1) = (y_0, y_1)\right] = \Pr\left[\mathcal{C}(\alpha_1, \alpha_0) = (y_0, y_1)\right]. \end{cases}$

Fig. 6. Realizing BSC using a channel $\mathcal{C} : \mathcal{X} \to \mathcal{Y}$. Here, a is the input bit to BSC channel and b is its output. The messages are encoded using symbols $\alpha_0, \alpha_1 \in \mathcal{X}$ that satisfy the conditions in Lemma 7.

$$\Pr\left(f_m(\boldsymbol{y}) \geq \sqrt{\frac{\lambda}{m}} \cdot \epsilon \,\middle|\, t = 0 \right) \leq 2e^{-\frac{\lambda \cdot \epsilon^2}{2}} \quad \textit{and}$$

$$\Pr\left(f_m(\boldsymbol{y}) \leq \sqrt{\frac{\lambda}{m}} \cdot \epsilon \,\middle|\, t \geq 2\sqrt{m \cdot \lambda} \right) \leq 2e^{-\frac{\lambda \cdot \epsilon^2}{2}}.$$

The following lemma analyzes the coding scheme in Fig. 6 that realizes erasure layer using α_0, α_1 described in Lemma 7. The fidelity of the scheme is a consequence of $\boldsymbol{\mu}_{\alpha_0}$ and $\boldsymbol{\mu}_{\alpha_1}$ being distinct. As we already observed, receiving (β, β) in this scheme is effectively the same as receiving an erasure. The lemma shows that since $\boldsymbol{\mu}_{\alpha_0}, \boldsymbol{\mu}_{\alpha_1}$ having intersecting supports, erasure happens with non-zero probability. The lemma also formalizes the observation that sending invalid encodings (α_i, α_i) for $i \in \{0, 1\}$ is effectively the same as sending the valid encoding of a random bit.

Lemma 9. *The scheme* $\langle \mathsf{Enc}, \mathsf{Dec} \rangle$ *in Fig. 6 satisfies the following properties:*

(i). $\Pr\left[\mathsf{Dec}\left(\mathsf{Enc}(a)\right) = a\right] = p > \frac{1}{2}$ *for* $a \in \{0, 1\}$*;*
(ii). $\Pr\left[\mathsf{Dec}\left(\mathcal{C}(\alpha_i, \alpha_i)\right) = 0\right] = \frac{1}{2}$ *for* $i = 0, 1$*;*
(iii). Let $\perp$ *be the event that the receiver gets* (β, β) *as output, where* β *is in the support of* $\boldsymbol{\mu}_{\alpha_0}$ *and* $\boldsymbol{\mu}_{\alpha_1}$*. Then* $\Pr(\perp | \mathsf{Enc}(a)) = \rho > 0$*, for all* $a \in \{0, 1\}$*.*

The Binary Symmetric Channel (BSC), with parameter p, is defined as $\mathsf{BSC}^p : \{0, 1\} \to \{0, 1\}$ such that for $b \in \{0, 1\}$, $\Pr(\mathsf{BSC}^p(b) = b) = p$. Consider the scenario where a configuration $\boldsymbol{x} \in \{0, 1\}^k$ is sent through BSC^p amongst which $S \subset \{0, 1\}^k$ is the set of acceptable configurations. The following lemma assigns scores $\{\gamma^S_{\boldsymbol{y}}\}_{\boldsymbol{y} \in \{0,1\}^k}$ to the received configurations in such a way that the expected score is 0 when an acceptable configuration $\boldsymbol{x} \in S$ is sent and the expected score is a strictly positive constant ϕ^S when an unacceptable configuration $\boldsymbol{x} \notin S$ in sent.

Lemma 10. *For $k \in \mathbb{N}$, let $U = \{0,1\}^k$ and $S \subseteq U$. For $\boldsymbol{x}, \boldsymbol{y} \in U$, define $p_{\boldsymbol{xy}} = \Pr(\mathsf{BSC}^p(\boldsymbol{x}) = \boldsymbol{y})$. There exists $\phi^S > 0$ and $\{\gamma^S_{\boldsymbol{y}}\}_{\boldsymbol{y} \in U} \in [-1,1]$ such that*

$$\sum_{\boldsymbol{y} \in U} p_{\boldsymbol{xy}} \gamma^S_{\boldsymbol{y}} = 0, \forall \boldsymbol{x} \in S \quad \textit{and} \quad \sum_{\boldsymbol{y} \in U} p_{\boldsymbol{xy}} \gamma^S_{\boldsymbol{y}} = \phi^S, \forall \boldsymbol{x} \notin S.$$

Proof: Consider the matrix $M \in \mathbb{R}^{U \times U}$ such that $M_{\boldsymbol{xy}} = p_{\boldsymbol{xy}}$. By the definition of BSC^p, when $|\boldsymbol{x} - \boldsymbol{y}|$ denotes the Hamming distance between $\boldsymbol{x}, \boldsymbol{y} \in U$, $p_{\boldsymbol{xy}} = (1-p)^{|\boldsymbol{x}-\boldsymbol{y}|} \cdot p^{k-|\boldsymbol{x}-\boldsymbol{y}|}$. It can be verified that, when $\otimes$ denotes the tensor operation,

$$M = H^{\otimes k}, \text{ where } H = \begin{bmatrix} p & 1-p \\ 1-p & p \end{bmatrix}.$$

Since H is invertible and tensor operation preserves non-singularity, M is an invertible matrix. The existence of $\phi^S > 0$ and $\{\gamma^S_{\boldsymbol{y}}\}_{\boldsymbol{y} \in U} \in [-1,1]$ follows directly from the invertibility of M. □

5.3 Construction and Analysis

The scheme $\langle \mathsf{P}_{ZK}, \mathsf{V}_{ZK} \rangle$ is given in Fig. 7. We now formally prove that this is a zero-knowledge proof of knowledge with negligible completeness and soundness error.

We first comment on the strategy of a malicious prover who encodes bits as (α_i, α_i) for $i = 0, 1$. Notice that the statistical test of thresholding $f_{2n \cdot \ell}(\boldsymbol{y})$ is insensitive to such a malicious strategy. But, by statement (ii) in Lemma 9, a bit that is encoded as (α_i, α_i) is decoded as 0 (resp. 1) with probability $\frac{1}{2}$. Hence, with respect to decoding, such a malicious strategy is effectively the same as encoding a random bit honestly using Enc. Consequently, every malicious prover strategy (including ones that encode bits incorrectly using (α_i, α_i)) can be thought of as a randomized strategy over a sub-class of strategies in which each bit is encoded as (α, α'), where $\alpha \neq \alpha'$. Hence, in the sequel, we analyze soundness only with respect to this class of strategies.

The proof proceeds by bounding the number of bad proofs a malicious sender can send without getting rejected by the tests performed by the verifier. We define B_{encoding} as the set of bad proofs in which at least one bit is encoded using symbols outside the set $\{\alpha_0, \alpha_1\}$. Also, define $B_{\text{incorrect}}$ as the set of proofs in which each bit is correctly encoded using Enc, but the proof itself is invalid. This is formalized as the proofs from which the extractor E for $\langle \mathsf{P}_{oZK}, \mathsf{V}_{oZK} \rangle$ cannot extract a valid witness. We would argue soundness by showing that if the sizes of B_{encoding} and $B_{\text{incorrect}}$ are substantial, then V_{ZK} rejects with all but negligible probability. Furthermore, completeness follows from the tests accepting an honest prover with all but negligible probability. These are established in the following claims; see the full version [1] for formal proofs. Formally, B_{encoding} and $B_{\text{incorrect}}$ are defined as follows.

$$B_{\text{encoding}} = \{i \in [n] : \exists (j,k) \in [\ell] \times \{0,1\} \text{ s.t. } x^{i,j}_k \notin \{\alpha_0, \alpha_1\}\},$$
$$B_{\text{incorrect}} = \{i \in [n] : i \notin B_{\text{encoding}} \text{ and } R_L(x, E(\pi_i, x)) = 0\}.$$

$\langle \mathsf{P}_{ZK}, \mathsf{V}_{ZK} \rangle$

Common input to prover and verifier $x \in L$.
Auxiliary input to prover w such that $R_L(x, w) = 1$.
For a non-trivial channel $\mathcal{C}$, without redundant symbols, consider symbols $\alpha_0, \alpha_1 \in \mathcal{X}$, functions f_m, for $m \in \mathbb{N}$, and $\epsilon > 0$ as described in Lemma 8. Let $\langle \mathsf{Enc}, \mathsf{Dec} \rangle$ be the encoding scheme described in Figure 6 w.r.t. α_0, α_1. Let p and ρ be as described in Lemma 9 for this encoding scheme. For $S \subset \{0,1\}^3$, consider $\gamma^S_{\boldsymbol{y}}$, for each $\boldsymbol{y} \in \{0,1\}^3$, and ϕ^S, from Lemma 10 with respect to BSC^p. Define $\phi = \min_{S \subset \{0,1\}^3} \phi^S$. For security parameter λ, let $(\mathsf{P}_{oZK}, \mathsf{V}_{oZK})$ be a $(3, 1-\rho)$-ZK-PCP with knowledge soundness κ. Finally, when $\ell = \mathrm{poly}(\lambda, |x|)$ is the length of proof output by P_{oZK}, let $n = \left(\frac{\ell\lambda}{\kappa}\right)^2$.

1. P_{ZK} samples $\pi_1, \ldots, \pi_n \xleftarrow{\$} \mathsf{P}_{oZK}(x, w, \lambda)$. For all $i \in [n], j \in [\ell]$, let the j^{th} bit in the proof π_i be $b_{i,j}$, then encode $b_{i,j}$ using Enc to obtain $(x_0^{i,j}, x_1^{i,j})$.
2. For all $i \in [n], j \in [\ell]$, let $\left(y_0^{i,j}, y_1^{i,j}\right) = \mathcal{C}\left(x_0^{i,j}, x_1^{i,j}\right)$. Let $\boldsymbol{y}$ be the vector $\left(y_k^{i,j}\right)_{i \in [n], j \in [\ell], k \in \{0,1\}}$.
3. If $f_{2n\cdot\ell}(\boldsymbol{y}) \geq \sqrt{\frac{\lambda}{2n\ell}}$, then V_{ZK} aborts and rejects the proof. Otherwise, V_{ZK} decodes $\pi_1, \ldots, \pi_n$ as $\hat{\pi}_1, \ldots, \hat{\pi}_n$ such that, for $i \in [n]$ and $j \in [\ell]$, the bit $b_{i,j}$ is decoded as $\hat{b}_{i,j} = \mathsf{Dec}\left(y_0^{i,j}, y_1^{i,j}\right)$. For each $k \in [n]$, choose 3 random indices $a_1, a_2, a_3 \in [\ell]$ of $\hat{\pi}_k$. If S is the set of accepting configurations for the indices (a_1, a_2, a_3) w.r.t. $\mathsf{V}_{oZK}(x, \cdot)$, set $s_k = \gamma^S_{\hat{\boldsymbol{b}}_k}$, where $\hat{\boldsymbol{b}}_k = (\hat{b}_{k,a_1}, \hat{b}_{k,a_2}, \hat{b}_{k,a_3})$. If $\frac{1}{n}\sum_{k \in n} s_k < \frac{\kappa\cdot\phi}{12}$, then V_{ZK} accepts, else it rejects.

Fig. 7. Description of $\mathsf{OWSC}/\mathcal{C}$ ZKPoK scheme for a non-trivial channel $\mathcal{C}$ without redundant input symbols.

Claim 2. *If $B_{encoding}$ is empty, then the probability with which $f_{2n\cdot\ell}(\boldsymbol{y}) \geq \sqrt{\frac{\lambda}{2n\ell}}$ is negligible in λ. If $|B_{encoding}| \geq \frac{n\kappa\phi}{6}$, then the probability with which $f_{2n\cdot\ell}(\boldsymbol{y}) < \sqrt{\frac{\lambda}{2n\ell}}$ is negligible in λ.*

Claim 3. *If $B_{encoding} = B_{incorrect} = \emptyset$, then $\frac{1}{n}\sum_{k=1}^{n} s_k \geq \frac{\kappa\cdot\phi}{12}$ with probability at most $2e^{-\frac{1}{2}\left(\frac{\ell\lambda\cdot\phi}{12}\right)^2}$. If $|B_{encoding}| \leq n\kappa\phi$ and $|B_{incorrect}| \geq \frac{n}{3}$, then $\frac{1}{n}\sum_{k=1}^{n} s_k < \frac{\kappa\cdot\phi}{12}$ with probability at most $2e^{-\frac{1}{2}\left(\frac{\ell\lambda\cdot\phi}{12}\right)^2}$.*

Below, we argue that $\langle \mathsf{P}_{ZK}, \mathsf{V}_{ZK} \rangle$ is a zero-knowledge proof using these claims.

Completeness. The above claims directly imply that if $\pi_1, \ldots, \pi_n$ are valid proofs which are correctly encoded, then V_{ZK} accepts with all but negligible probability.

Soundness. We build an extractor E' from E (the extractor for $\langle \mathsf{P}_{oZK}, \mathsf{V}_{oZK} \rangle$) as follows. For each $i \in [n]$, extractor E' tries to extract a proof π_i^* from the encoding of the purported proof π_i. Rejecting each purported proof π_i that is incorrectly encoded, *i.e.*, $i \in B_{\text{encoding}}$. If for some i, we have

$R_L(x, E(\pi_i^*, x)) = 1$, output $E(\pi_i^*, x)$; else, output $\perp$. Clearly, E' aborts only if $B_{\text{encoding}} \cup B_{\text{incorrect}} = [n]$. But the above claims imply that V_{ZK} rejects with all but negligible probability, whenever $|B_{\text{encoding}} \cup B_{\text{incorrect}}| \geq \frac{2n}{3}$.

Zero-Knowledge. By Lemma 9, Enc induces an erasure ($\perp$ in the lemma) with probability $\rho > 0$. Recall that the proof uses a $(3, 1-\rho)$-ZK-PCP $\langle \mathsf{P}_{oZK}, \mathsf{V}_{oZK} \rangle$. Let S be a simulator for this ZK-PCP. The construction of simulator S' for $\langle \mathsf{P}_{ZK}, \mathsf{V}_{ZK} \rangle$, using the simulator S is quite straightforward: S' runs n independent executions of $S(x, \lambda)$ to get $\pi_1^*, \ldots, \pi_n^*$. It is easy to see that if S produced a perfect simulation of the ZK-PCP, then S' would also produce a perfect simulation of the verifier's view in the ZK proof. Since the simulation by S incurs a negligible error, so does the simulation by S'.

Acknowledgements. We thank the anonymous Asiacrypt 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. 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).

References

1. Agrawal, S., Ishai, Y., Kushilevitz, E., Narayanan, V., Prabhakaran, M., Prabhakaran, V., Rosen, A.: Cryptography from one-way communication: on completeness of finite channels. In: Cryptology ePrint Archive (2020)
2. Ajtai, M.: Oblivious rams without cryptogrpahic assumptions. In: STOC 2010, pp. 181–190 (2010)
3. Bellare, M., et al.: iKP - a family of secure electronic payment protocols. In: USENIX Workshop on Electronic Commerce (1995)
4. Bellare, M., Tessaro, S., Vardy, A.: Semantic security for the wiretap channel. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 294–311. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_18
5. Bennett, C.H., Brassard, G., Crepeau, C., Maurer, U.M.: Generalized privacy amplification. IEEE Trans. Inf. Theor. **41**(6), 1915–1923 (1995)
6. Bennett, C.H., Brassard, G., Robert, J.-M.: Privacy amplification by public discussion. SIAM J. Comput. **17**(2), 210–229 (1988)
7. Bertsimas, D., Tsitsiklis, J.N.: Introduction to Linear Optimization. Athena Scientific, Nashua (1997)
8. Bloch, M., Barros, J.: Physical-Layer Security: from Information Theory to Security Engineering. Cambridge University Press, Cambridge (2011)

9. Blum, M., Feldman, P., Micali, S.: Proving security against chosen ciphertext attacks. In: Goldwasser, S. (ed.) CRYPTO 1988. LNCS, vol. 403, pp. 256–268. Springer, New York (1990). https://doi.org/10.1007/0-387-34799-2_20
10. Chaum, D.: Blind signatures for untraceable payments. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds.) Advances in Cryptology, pp. 199–203. Springer, Boston, MA (1983). https://doi.org/10.1007/978-1-4757-0602-4_18
11. Chaum, D.: Online cash checks. In: Quisquater, J.-J., Vandewalle, J. (eds.) EUROCRYPT 1989. LNCS, vol. 434, pp. 288–293. Springer, Heidelberg (1990). https://doi.org/10.1007/3-540-46885-4_30
12. Crepeau, C., Kilian, J.: Achieving oblivious transfer using weakened security assumptions. In: FOCS, pp. 42–52 (1988)
13. Crépeau, C., Morozov, K., Wolf, S.: Efficient unconditional oblivious transfer from almost any noisy channel. In: Blundo, C., Cimato, S. (eds.) SCN 2004. LNCS, vol. 3352, pp. 47–59. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30598-9_4
14. Damgård, I., Kilian, J., Salvail, L.: On the (Im)possibility of basing oblivious transfer and bit commitment on weakened security assumptions. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 56–73. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_5
15. Feige, U., Lapidot, D., Shamir, A.: Multiple non-interactive zero knowledge proofs based on a single random string. In: FOCS, vol. 1, pp. 308–317, October 1990
16. Feige, U., Kilian, J., Naor, M.: A minimal model for secure computation (extended abstract). In: STOC, pp. 554–563 (1994)
17. Garg, S., Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Cryptography with one-way communication. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9216, pp. 191–208. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48000-7_10
18. Ishai, Y., Kushilevitz, E.: Private simultaneous messages protocols with applications. In: ISTCS 1997, pp. 174–184. IEEE Computer Society (1997)
19. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Prabhakaran, M., Sahai, A.: Efficient non-interactive secure computation. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 406–425. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20465-4_23
20. Kilian, J.: Founding cryptography on oblivious transfer. In: STOC, pp. 20–31 (1988)
21. Kraschewski, D., Maji, H.K., Prabhakaran, M., Sahai, A.: A full characterization of completeness for two-party randomized function evaluation. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 659–676. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55220-5_36
22. Lin, F., Cheraghchi, M., Guruswami, V., Safavi-Naini, R., Wang, H.: Secret sharing with binary shares. In: ITCS, pp. 53:1–53:20 (2019)
23. Maurer, U.M.: Perfect cryptographic security from partially independent channels. In: STOC 1991, pp. 561–571 (1991)
24. Poor, H.V., Schaefer, R.F.: Wireless physical layer security. Proc. Natl. Acad. Sci. **114**(1), 19–26 (2017)
25. Ranellucci, S., Tapp, A., Winkler, S., Wullschleger, J.: On the efficiency of bit commitment reductions. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 520–537. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25385-0_28
26. Raz, R., Reingold, O., Vadhan, S.: Extracting all the randomness and reducing the error in trevisan's extractors. J. Comput. Syst. Sci. **65**, 97–128 (2002)

27. Trevisan, L.: Extractors and pseudorandom generators. J. ACM **48**(4), 860–879 (2001)
28. Winter, A., Nascimento, A.C.A., Imai, H.: Commitment capacity of discrete memoryless channels. In: Paterson, K.G. (ed.) Cryptography and Coding 2003. LNCS, vol. 2898, pp. 35–51. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-40974-8_4
29. Wullschleger, J.: Oblivious transfer from weak noisy channels. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 332–349. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00457-5_20
30. Wyner, A.D.: The wire-tap channel. Bell Syst. Tech. J. **54**(8), 1355–1387 (1975)
31. Yao, A.C.C.: How to generate and exchange secrets (extended abstract). In: FOCS 1986, pp. 162–167 (1986)