Unconditional Security using (Random) Anonymous Bulletin Board

Albert Yu[1], Hai H. Nguyen, Aniket Kate, Hemanta K. Maji ✉
Department of Computer Science, Purdue University, USA

Abstract

In a seminal work, Ishai et al. (FOCS–2006) studied the viability of designing unconditionally secure protocols for key agreement and secure multi-party computation (MPC) using an anonymous bulletin board (ABB) as a building block. While their results establish the feasibility of key agreement and honest-majority MPC in the ABB model, the optimality of protocols with respect to their round and communication complexity is not studied. This paper enriches this study of unconditional security in the ABB model in multiple ways.

- We present a key agreement protocol with a novel combinatorial insight to offer a 200% throughput over the (FOCS–2006) study; i.e., using the same number of messages, we can (almost) double the bit-length of the agreed key. We also prove the near optimality of our approach.
- We offer unconditionally secure protocols for the binary erasure channel and string oblivious transfer functionalities. Our BEC and (chosen message) random string oblivious transfer protocols employ one-shot access to an ABB and a single message from a sender and a receiver. We demonstrate the round optimality of our BEC constructions. Furthermore, we present a non-interactive (random) string oblivious transfer protocol.

Central to our technical contributions is the abstraction of a distributional variant of the random ABB functionality. Investigating the concrete efficiency of founding MPC from this primitive leads to fascinating new mathematical challenges in well-established MPC models, which will be of broader interest to the community.

2012 ACM Subject Classification Security and privacy → Cryptography

Keywords and phrases Anonymous Broadcast, MPC, Key-Agreement, Binary Erasure Channel, Oblivious Transfer, Security

1 Introduction

Securely realizing unconditionally secure cryptographic primitives is a topic of immense value and has a rich history. This work revisits a particularly surprising work by Ishai et al. [24] that analyzes the possibility of performing cryptography with unconditional security using an *anonymous bulletin board* (ABB). Ishai et al. establish unconditional security for prominent cryptographic tasks such as key agreement and honest-majority secure multiparty computation (MPC) based solely on access to an ABB that allows a sender to publish her message without revealing her identity. In particular, they demonstrate that ABB is sufficient to implement unconditionally secure point-to-point channels between two parties without making any other assumption. Ishai et al. then extend it to achieve MPC with unconditional security in the presence of an honest majority, diversifying the primitives that facilitate secure computation. Interestingly, they complement these constructions by showing the impossibility of unconditional secure computation using anonymous broadcast in the absence of an honest majority.

Since the publication of the paper by Ishai et al. in 2006, the field of anonymous communication has witnessed tremendous growth: the anonymous communication network

[1] Corresponding Author

Tor [16] serves more than two million unique users daily using an overlay network of several thousand nodes all over the Internet. As the use of blockchains brings users' financial dealing to the (public) Internet, there have been significant efforts towards introducing and improving anonymity over the Internet. Startups such as Nym [15] and xx.network [39, 40] are developing generic anonymous communication networks to break the link between users' identity and their transactions, and several blockchain projects have started incorporating anonymous communications, such as Tor and I2P, in their designs [23]. Academic literature on anonymous communication, as well as protocol implementations, have significantly expanded in the last two decades [1, 6, 13, 17, 28]. It is safe to say that ABBs are prevalent on the Internet today. Motivated by these real-world applications, our goal is to understand the efficacy and concrete efficiency of developing cryptography assuming access to such an ABB.

The utility of the ABB towards unconditional security is easy to illustrate using Ishai et al.'s [24] elegant key agreement protocol between Alice and Bob against an honest-but-curious adversary. Alice and Bob independently pick random integers (say r_A and r_B, respectively) and publish those to the anonymous broadcast channel. The agreed single secret bit is 1 if $r_A > r_B$ and 0 if $r_A < r_B$. If $r_A = r_B$, then Alice and Bob fail to establish the secret bit and rerun the protocol. Notice that Alice and Bob know their respective input and thus can compute the secret bit; however, eavesdroppers cannot distinguish r_A from r_B and have no information about the agreed bit. Moreover, the failure probability (using the birthday bound) depends on the size of the sample space of the integers.

This "indistinguishability property" can be abstracted as a multi-set. Conceptually, we observe that the use of ABB converts a vector (or key-value store) of user inputs to a multi-set. This brings us to the question: what if Alice and Bob send multiple (say m) messages each? Can we agree on more than m bits using this $2m$-sized multi-set? We answer this question affirmatively to demonstrate that Alice and Bob can indeed agree on close to $2m$ secret bits, which improves the throughput of the key agreement to 200%, as compared to Ishai et al. [24]. This work aims to determine the concrete communication and round complexity of key cryptographic functionalities based on anonymity. This investigation leads to both qualitative and quantitative research questions in this context.

To this end, we establish connections of implementing functionalities using ABB in our context with various well-studied communication-limited MPC models (like *non-interactive correlation distillation* [34, 35], *secure non-interactive simulation/reduction* [2, 26], *one-way secure computation* [20], and *private simultaneous messages* [18]). Our problems translate into analytically tractable instances of these MPC models, which have generally been challenging to analyze. These connections lead us to several (near-optimal) protocol constructions. Our practically-motivated research objectives lead to fascinating research questions in these MPC models, potentially of interest to the broader cryptographic community.

1.1 Our Contributions

From the modeling perspective, this work assumes the existence of an anonymous broadcast, which we model as an Anonymous Bulletin Board (ABB) hybrid. There are four parties $\mathcal{A}, \mathcal{B}, \mathcal{C}$, and $\mathcal{D}$. The bulletin board ideal functionality, represented as $\mathsf{ABB}_{m_A, m_B, m_C}$, takes as input three multi-sets: (1) $A := \{a_1, \ldots, a_{m_A}\}$ from party $\mathcal{A}$, (2) $B := \{b_1, \ldots, b_{m_B}\}$ from $\mathcal{B}$, and (3) $C := \{c_1, \ldots, c_{m_C}\}$ from $\mathcal{C}$. Note that party $\mathcal{D}$ does not provide any input. The functionality outputs the multi-set $\Gamma = A \cup B \cup C := \{\gamma_1, \ldots, \gamma_{m_A+m_B+m_C}\}$ to all four parties.

▶ **Example 1** (Clarification on our ABB model). For illustrative purposes, consider $m_A =$

$m_B = m_C = 2$. Party $\mathcal{A}$ sends the vector (x_1, x_2) to the ABB, party $\mathcal{B}$ sends the vector (y_1, y_2) to the ABB, and party $\mathcal{C}$ sends the vector (z_1, z_2) to the ABB. Our ABB publishes the multi-set of all the received elements $\{x_1, x_2, y_1, y_2, z_1, z_2\}$. More concretely, interpret this multi-set represented as the sorted vector containing its elements (with multiplicities). In particular, this multi-set is different from the following alternative interpretations.

1. The set $\{\{x_1, y_1, z_1\}, \{x_2, y_2, z_2\}\}$. In this alternative version, messages with identical indices are *linked* to each other. If parties wish to link messages using our ABB, they need to explicitly encode the indices into their message, which causes a logarithmic increase in their length.
2. The set $\{\{x_1, x_2\}, \{y_1, y_2\}, \{z_1, z_2\}\}$. This alternative version links messages sent by the same party while hiding the identity of the party. Such linking is achieved using our ABB by encoding anonymized identities of the parties into their respective messages.

We refer to party $\mathcal{C}$ as the *helper* and party $\mathcal{D}$ as the *eavesdropper*. In the randomized version of bulletin board (rABB), the three multi-sets A, B, C are sampled according to some independent distributions P, Q, R, respectively. See Section 4 for a formal definition of ABB and its randomized version (rABB).

In addition to the bulletin board, parties also have public authenticated channels between them. In the ABB setting, we define each party's *communication complexity* as the number of bits that the party sends to the ABB plus the number of bits it sends to other parties through the public authenticated channels. For example, the communication complexity of party $\mathcal{A}$ is the sum of the following quantities.

- The bit length of A (the message that party $\mathcal{A}$ sends to the ABB)
- The bit length of the messages that party $\mathcal{A}$ sends to other parties ($\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$).

We define the communication complexity in the rABB setting in a similar manner. For example, the communication complexity of $\mathcal{A}$ is the sum of the following quantities.

- Bit length of A (the message that that party $\mathcal{A}$ receives from rABB)
- Bit length of the messages that party $\mathcal{A}$ sends to other parties ($\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$).

The sequel summarizes our contributions.

▶ **Result 1** (Key-agreement Protocol: Informal). *We present a* non-interactive *two-party key-agreement protocol using* $\mathsf{rABB}_{m,m,0}$ *with individual message length n that establishes (near-optimal) $2m$-bit keys with $(m \cdot n)$-bit communication complexity.*

Theorem 8 provides the formal statement for this result. Our construction is secure against a computationally unbounded eavesdropper $\mathcal{D}$. Our construction is straightforward to implement, and the key length (i.e., throughput) is near-optimal. Here, throughput is the ratio of the key length to the number of messages. The length n of the individual messages affects our algorithm's failure probability, the event where parties fail to agree on a key. Small messages would result in close-to-1 failure probability. Surprisingly, when n is larger than a particular threshold, it has essentially no impact on the key length. We also present a duplicate-recovery variant of the protocol in Result 1, which is suitable for other parameter regimes. Details on the duplicate-recovery variant can be found in Section 5.7.

▶ Remark 2 (Upper bound on our key length: additional comments). Our proof of the optimality of our key length considers a wide family of protocols. In these protocols, parties can interact over multiple rounds using the public authenticated channels after the rABB invocation. The parties $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ receive messages from arbitrary independent message distributions $P, Q,$

and R, respectively (not necessarily the uniform distribution). In our protocol, rABB delivers random independent messages to the parties. We prove this result using mutual information, entropy-based arguments, and the recent results of [29, 30].

In our protocol, parties have access to a *single* ABB *or* rABB that they call once. Many other protocols (such as [9], [14], and some protocols in [24]) either require additional assumptions, such as on the synchrony of the system model or require multiple independent instances of ABB to be implemented. We emphasize that this is qualitatively different from our protocol setting, and a direct comparison of the communication costs of these protocols against ours results in an inaccurate representation of both their protocols and ours. Therefore, we focus our concrete communication cost analysis on comparison with the state-of-the-art protocol in this setting, which is [5]. The result of this concrete communication cost comparison is presented in Figure 1. For typical values of k such as 128, [5] requires roughly 2.9× our communication cost.

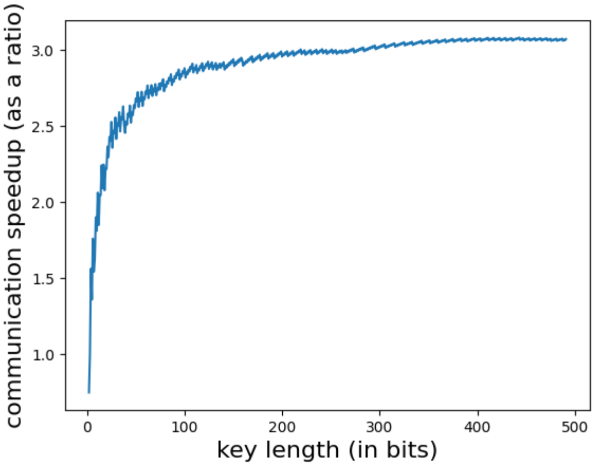


Figure 1 Plot of the ratio of communication required by [5] over our protocol's communication needed to achieve various expected values of key-length k. For example, to generate a 128-bit key (on average), [5] requires roughly 2.9× our communication cost.

▶ **Result 2** (Binary Erasure Channel: Informal)**.** *We present a one-round (round-optimal) protocol for binary erasure channel from $\mathcal{B}$ to $\mathcal{A}$ utilizing the* rABB *(with a helper) and a public authenticated channel from $\mathcal{B}$ (the sender) to $\mathcal{A}$ (the receiver).*

Theorem 16 provides the formal statement for this result. In the honest majority setting, [24] showed that BEC could be realized using general MPC construction techniques. In this work, we present protocols for BEC that is straightforward to implement and determine their concrete round and communication complexity. Our protocols achieve unconditional security against a semi-honest adversary. It uses only one round of communication between the parties (sender $\mathcal{B}$ sends one message to the receiver $\mathcal{A}$ in our protocol). When the erasure probability is (close to) a rational number e/d, where $\gcd(e, d) = 1$, our protocol's communication complexity is proportional to the denominator d.

To complement this result, we show that BEC is impossible in rABB-hybrid without additional communication, proving the round optimality of our protocol. We employ the techniques developed recently in the SNIS/SNIR literature [2, 26] to prove this impossibility.

Determining the optimal communication complexity (or tightly estimating it) remains an open problem.

▶ Remark 3 (A subtlety). We prove the round optimality of our result when the messages are sampled from *uniform distributions* P, Q, R (see Theorem 18 for details). Proving the optimality for *arbitrary independent distributions* P, Q, R remains open. Recall P, Q, R represent the distribution of the messages sent by rABB to the parties $\mathcal{A}, \mathcal{B}$, and $\mathcal{C}$, respectively. Analyzing this distributional variant of rABB motivates new research directions in interaction-limited MPC models, like SNIS and SNIR. This problem is challenging even when P, Q, R are flat distributions over a sparse subset of the message space. Typically, these models (like NICD [34, 35], SNIS [26], SNIR [2], OWSC [20], and PSM [18]) have strong hardness-of-computation results. However, for our application scenarios, there are non-trivial and practically useful construction as well.

In the context of implementing oblivious transfers, Ishai et al. [24] proved the impossibility of realizing oblivious transfer (OT) using ABB when honest parties are not in the majority. This implies that it is impossible to realize oblivious transfers (as well as their randomized versions) in the ABB-hybrid without the helper party $\mathcal{C}$. We construct oblivious transfer protocols that achieve a few different functionality variants – a step towards diversifying setups for oblivious transfers.

▶ **Result 3.** *We present a one-round (round-optimal) protocol for establishing (chosen message) random string oblivious transfer (*cmROT^ℓ*) from sender* $\mathcal{B}$ *to receiver* $\mathcal{A}$ *with the helper* $\mathcal{C}$*.*

The cmROT^ℓ functionality takes as input two ℓ-bit messages x_0 and x_1 from the sender and delivers the tuple (b, x_b) to the receiver, where the bit b is chosen uniformly at random. Theorem 19 provides the formal statement of this result. This protocol achieves unconditional security against semi-honest adversaries. In our protocol, the sender $\mathcal{B}$ sends a message to $\mathcal{A}$ using a private authenticated channel. The protocol is similar to the BEC protocol and interpreted as an implementation of two BEC with correlated inputs. The round optimality of this construction is a consequence of the fact that one can use (chosen message) random string OT to implement an erasure channel.[2] We extend this protocol to 1-out-of-N OT (where the sender chooses N inputs) in Section 7.4.

Finally, and surprisingly, we implement a (non-interactive) random oblivious transfer protocol using rABB.

▶ **Result 4.** *We present a non-interactive protocol for establishing random string oblivious transfer (*ROT^ℓ*) from sender* $\mathcal{B}$ *to receiver* $\mathcal{A}$ *with the helper* $\mathcal{C}$*.*

The ROT^ℓ functionality provides two uniformly random messages x_0 and x_1 instead of receiving them as input from the sender. Discussions on the non-interactive random string oblivious transfer can be found in Section 7.4.

▶ Remark 4 (New research problems in interaction-limited MPC models). Our use of $\mathsf{rABB}^{P,Q,R}$ can be interpreted as sampling from the joint distribution $(P, Q, R|\Gamma)$ in a preprocessing step, where Γ is the union of the three. Under this interpretation, our research problems translate into research questions in the NICD [34, 35], SNIS [26], SNIR [2], OWSC [20], and PSM [18] models.

[2] The sender can choose to send $x_0 = 11$ and $x_1 = 0m$ for a bit $m \in \{0, 1\}$. The receiver receives the bit m with a probability of $1/2$; otherwise, it is erased with a probability of $1/2$. Therefore, the impossibility of implementing an erasure channel extends to this case.

1. For key agreement, we prove that the uniform distribution achieves the optimal result even against arbitrary independent distributions.
2. For BEC, we prove that our protocol is round optimal for uniform distributions. However, whether a non-interactive protocol exists for some other independent distributions remains an open problem.
3. For random string oblivious transfer, we show that by using specialized distributions P, Q, R, we are able to obtain non-interactive random string oblivious transfer.

1.2 Related Works

1.2.1 Key-Agreement

There are many works focused on key-agreement or developing secure point-to-point links based on anonymous communication. [5] performs key agreements by having each party send a set of position-labeled bits, and discard any identical bits to use the remaining bits as the key. There are also several works that expand upon or improve [5]. For example, [42] expands [5] to work over semi-honest channels. [9] proposes a protocol that only requires k total messages for a k-bit key by utilizing the fact that parties can set the source of the message to be honest or false, and also send messages in random order. [14] similarly proposes a protocol that requires the parties to send messages in a random order by implementing random wait times. [36] considers key-agreement when the receivers (instead of the senders) are anonymous. [19] considers key-agreement in a similar setting, where a “deck of cards” is dealt such that each party has several cards from the deck. The remaining cards are dealt to the adversary. Using this setup, the parties would like to agree on a secret key. Finally, Gilad and Herzberg [21] demonstrate the practical utility of [24] for the IP-level security protocol IPSec.

There has been extensive study of establishing fixed length secret key in the source model in which parties observe i.i.d samples from a joint distribution and the eavesdropper possibly observes some side information from these samples [4, 12, 22, 31–33]. The main objective is to study the achievable key rate when the number of samples tend to infinity.

[25] study the question of bootstrapping anonymous communication. The objective is to communicate a large amount of data using non-anonymous communication and only a small amount of anonymous broadcasts.

1.2.2 Communication-limited MPC Models

Non-interactive Correlation Distillation. In information theory and theoretical computer science, non-interactive correlation distillation (NICD) is a well-studied analytically-tractable problem [8, 10, 34, 35, 41]. NICD also aims to establish secure key agreements. In NICD, each party holds a noise version of some source bits, a particular form of correlated private randomness. It is common in NICD that the failure probability for the key-agreement instances is high. On the other hand, parties have access to ABB that generates a different form of conditional distribution in the rABB-hybrid model. We are the first to choose this distribution and achieve near-optimal key length.

Secure Non-interactive Simulation/Reduction. Secure non-interactive simulation/reduction (SNIS/SNIR) is a cryptographic primitive introduced recently [2, 26]. In this model, parties have i.i.d samples of a source correlated private randomness; the objective is to non-interactively and securely transform these samples into i.i.d samples of another target correlated private randomness. This line of work investigates both the feasibility and efficiency of SNIS/SNIR constructions. We shall employ the techniques to prove the impossibility results in their settings to show the round-complexity of realizing BEC or OT using rABB-hybrid.

One-way Secure Computation. One-way secure computation [3, 20] uses one round of communication to securely transform the samples of the source distribution to the samples of the target distribution.

2 Technical Overview

This section provides a technical overview of our results. For a complete list of notations and backgrounds, refer to Section 3. The formal definition of the anonymous bulletin board (ABB) and its variant are in Section 4.

2.1 Key Agreement

We first present an overview of our (near-optimal) key-agreement protocol in Figure 4. We construct a key agreement protocol in which parties $\mathcal{A}$ and $\mathcal{B}$ receive a set of m messages of n bits each (A and B respectively). Additionally, all parties ($\mathcal{A}$, $\mathcal{B}$, $\mathcal{D}$) receive the set of $2m$ messages ($\Gamma = A \cup B$) from the rABB. The parties first discard any duplicate messages in Γ, resulting in $2m'$ total messages where m' messages belong to each set A and B. Since no duplicate messages exist, only parties $\mathcal{A}$ and $\mathcal{B}$ can identify which of the $2m'$ messages belong to each set A and B. By using a canonical ordering of the $2m'$ messages and assigning messages belonging to A as 1 and messages belonging to B as 0, the two parties can agree on a $2m'$ bit string that is known only to them. Then, by using standard techniques in combinatorics, the two parties can index the agreed upon bit string out of the $\binom{2m'}{m'}$ possible bit strings and agree on a key of length $\log\left(\binom{2m'}{m'}\right)$.

We discuss how the parameter choices m and n affect the expected key length and the failure probability using standard techniques in probability. Additionally, we perform brute force searches to identify the optimal parameters for various key lengths and compare those results with previous state-of-the-art results.

Finally, using techniques on mutual information, we prove that under the setting of arbitrary/unlimited message length, our protocol achieves the optimal expected key length given parameter m.

Additionally, we present a variant of our protocol called duplicate recovery, which is suitable for small values of n. In duplicate recovery, instead of removing all duplicates, the protocol considers the duplicates as part of the possible distributions. We note that in this case, indexing the possible distributions becomes non-trivial. We present such a problem as a new problem in combinatorics, as well as reformulate it as an Integer Programming (IP) problem. We believe this problem may be of independent interest.

The complete description and analysis of the key agreement protocol can be found in Section 5.

2.2 Binary Erasure Channel

Next, we present an overview of our protocol realizing BEC in Figure 7. In our protocol, we have $\mathcal{A}$ as the receiver, $\mathcal{B}$ as the sender, and $\mathcal{C}$ as a helper. $\mathcal{A}$ receives A, $\mathcal{B}$ receives B, while $\mathcal{C}$ receives C from the rABB. Additionally, all parties receive the set $\Gamma = A \cup B \cup C$.

At a high level, $\mathcal{A}$ cannot distinguish between messages belonging to B and C, while $\mathcal{B}$ cannot distinguish between messages belonging to A and C. Therefore, out of all anonymous messages, $\mathcal{B}$ will choose one message belonging to B and another message, then use the information related to which message belonging to B to generate a one-bit key which is used to encrypt a bit that is sent to $\mathcal{A}$. If $\mathcal{B}$ selected a message from B and a message from A, then $\mathcal{A}$ can also identify which message was from B and obtain the key, which allows $\mathcal{A}$ to receive the message. On the other hand, if $\mathcal{B}$ selected a message from B and a message from C, then $\mathcal{A}$ cannot distinguish which message belongs to B, and thus the bit is erased. The erasure probability can be controlled by how many messages A and C each contain, as it directly determines the probability that $\mathcal{B}$ will select a message belonging to A.

Using properties of joint distribution and their eigenvalues, we prove that our protocol is round optimal.

The complete description and analysis of the binary erasure channel protocol can be found in Section 6.

2.3 Chosen Message Random String Oblivious Transfer

Finally, we present an overview of our construction of cmROT in Figure 8. A single bit of random oblivious transfer can be seen as two BEC instances that are correlated in a way such that whenever one of the messages is erased, the other message is delivered.

We use a set of four elements, one belonging to $\mathcal{A}$, one belonging to $\mathcal{C}$, and two belonging to $\mathcal{B}$, that is divided into two subsets that each contain an element belonging to $\mathcal{B}$. $\mathcal{B}$ is able to identify both messages that belong to $\mathcal{B}$ in the two subsets, and can therefore obtain two bits. On the other hand, $\mathcal{A}$ can only identify the element belonging to $\mathcal{B}$ in the subset that contains $\mathcal{A}$'s element. This creates a setting where $\mathcal{B}$ is able to identify two messages while $\mathcal{A}$ is only able to identify one of them.

When we directly perform the above step multiple times, a natural issue arises in which $\mathcal{B}$ is unable to identify what messages $\mathcal{A}$ can obtain, but will instead get a cartesian product of all the possible bits.

The key observation is that security still holds if we set all elements belonging to $\mathcal{A}$ to be even (or odd), all elements belonging to $\mathcal{C}$ to be odd (or even, respectively), and half the elements belonging to $\mathcal{B}$ to be even and half to be odd. This will allow $\mathcal{B}$ to "link" the bits that form the same message, thus identifying the two possible messages that $\mathcal{A}$ can obtain without learning which message $\mathcal{A}$ obtains.

We can also compress the multiple calls to rABB into a single call using sequence identifiers and parallel identifiers (full detail can be found in Section 4.4).

Finally, to ensure that $\mathcal{C}$ learns nothing about either message, $\mathcal{B}$ sends two "correction messages" that get xored with the original message to create the final message to $\mathcal{A}$ through a private authenticated channel (such a private channel can be established in parallel with no additional round using our key-agreement protocol).

The complete description and analysis of the random string oblivious transfer protocol can be found in Section 7.

3 Preliminaries

This section introduces some notations and basic background that will be useful in the later sections.

3.1 Sets

Throughout the paper, we may use the word "set" when we mean "multi-set". We will often use capital letters to denote multi-sets. Alternatively, we will define a multi-set by listing its elements in curly braces. We denote elements of the multi-set using lowercase letters. Whenever we talk about multi-sets, especially the union of sets, we assume that all elements are randomized. For example, for multi-set A and B, by only looking at $A \cup B$, it should be impossible to determine which elements came from A. Additionally, we assume that there exists some canonical ordering of elements in a multi-set. For simplicity, it may help to assume that sets are automatically sorted by their canonical ordering. We note that the desired properties such as not being able to determine which elements came from A by only looking at $A \cup B$ hold for when sets are randomized or when they are sorted according to the canonical ordering.

3.2 Binary Erasure Channel (BEC)

In a binary erasure channel (BEC) with erasure probability p, a sender S sends a binary message $m \in \{0,1\}$ to a receiver R. With probability $1-p$, R receives the message m. With probability p, the message is "erased" and R receives nothing. In this case, we say that R receives $\perp$. The sender is unaware of whether the erasure happened or not.

Roughly, security of BEC requires that the sender does not learn whether the bit was erased or not, and the receiver not learning anything about the bit if it is erased (and it received $\perp$).

3.3 String Oblivious Transfer

String Oblivious Transfer. ℓ-bit string (1-out-of-2) oblivious transfer, denoted as OT^{ℓ}, is a two-party functionality that takes as input $(x_0, x_1) \in \left(\{0,1\}^{\ell}\right)^2$ from Bob, a bit $b \in \{0,1\}$ from Alice, and outputs x_b to Alice. Security of OT requires that Alice learns nothing about the bit b, and Bob learns nothing about x_{1-b}.

Note that when $\ell = 1$, the functionality OT^1 is the (standard) bit oblivious transfer.

Random String Oblivious Transfer. Random oblivious transfer, denoted as ROT^{ℓ}, is a correlation that samples $x_0 \in \{0,1\}^{\ell}, x_1 \in \{0,1\}^{\ell}, b \in \{0,1\}$ uniformly and independently at random. It provides Bob with the secret share $r_B = (x_0, x_1)$ and provides Alice the secret share $r_A = (b, x_b)$.

Chosen Message String Random Oblivious Transfer. Chosen Message Random oblivious transfer, denoted as cmROT^{ℓ}, is a functionality that takes as input $(x_0, x_1) \in \left(\{0,1\}^{\ell}\right)^2$ from Bob, samples a bit b uniformly at random, and outputs (b, x_b) to Alice.

3.4 Entropy and Mutual Information

We shall use mutual information and entropy-based arguments to prove the optimality of our key-agreement protocols.

▶ **Definition 5** (Mutual Information). *Let X and Y be a pair of discrete random variables over the space $\mathcal{X} \times \mathcal{Y}$. If their joint probability distribution is $P_{XY}(x,y)$, the mutual information between them, denoted as $I(X,Y)$, is*

$$I(X,Y) := \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} P_{XY}(x,y) \log \frac{P_{XY}(x,y)}{P_X(x) P_Y(y)}.$$

Moreover, the conditional mutual information of $(X, Y|Z)$ is defined as follows.

$$I(X,Y|Z) := \sum_{z\in\mathcal{Z}} P_Z(z) \sum_{y\in\mathcal{Y}} \sum_{x\in\mathcal{X}} P_{X,Y|Z}(x,y|z) \log \frac{P_{X,Y|Z}(x,y|z)}{P_{X|Z}(x|z)P_{Y|Z}(y|z)}.$$

▶ **Definition 6** (Entropy). *Let X be a discrete random variable distributed according to $P \colon \mathcal{X} \to (0,1)$. The entropy of X, denoted as $H(X)$, is defined as*

$$H(X) := -\sum_{x\in\mathcal{X}} P(x)\log P(x).$$

▶ **Definition 7** (Conditional Entropy). *The conditional entropy of X given Y is defined as*

$$H(X|Y) := -\sum_{x\in\mathcal{X},y\in\mathcal{Y}} P_{XY}(x,y)\log\frac{P_{XY}(x,y)}{P_Y(y)}.$$

▶ **Fact 1.** *It holds that*

$$I(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X).$$

Furthermore,

$$I(X,Y|Z) = H(X|Z) - H(X|Y,Z) = H(Y|Z) - H(Y|X,Z).$$

4 Anonymous Bulletin Board Formalism

In this section, we first formally define Anonymous Bulletin Board (ABB) and Random Anonymous Bulletin Board (rABB). We then discuss the similarities and differences between ABB and rABB, and crucially, show that our protocol is equivalent in the ABB-hybrid and the rABB-hybrid. Additionally, we present a connection of rABB with the offline phase of the offline-online model.

4.1 Anonymous Bulletin Board

We assume all messages are from the domain $\mathbb{Z}_{2^n}$, where $n \in \mathbb{N}$ is called the *message length*. In an anonymous bulletin board (ABB), parties can privately send multiple messages to the ABB. Then, the ABB will broadcast the "set" of messages, with order and sender information removed. Additionally, we note that the ABB waits until it receives all messages before publishing. Therefore, a rushing adversary is impossible.

We define the ideal functionality in Figure 2. It is defined over a four-party setting, $\mathcal{A}$ and $\mathcal{B}$ represent the main participants that are trying to achieve something through interaction with the ABB. $\mathcal{C}$ represents the facilitators that are trying to assist $\mathcal{A}$ and $\mathcal{B}$ through interaction with the ABB. $\mathcal{D}$ represents eavesdroppers that do not send messages to the ABB, but receive the output of the ABB.

We define this ideal functionality as $\mathsf{ABB}_{m_A,m_B,m_C}$. Formally, $\mathsf{ABB}_{m_A,m_B,m_C}$ takes multi-set of inputs $A := \{a_1, \ldots, a_{m_A}\}$ from $\mathcal{A}$, multi-set of inputs $B := \{b_1, \ldots, b_{m_B}\}$ from $\mathcal{B}$, multi-set of inputs $C := \{c_1, \ldots, c_{m_C}\}$ from $\mathcal{C}$, and outputs the multi-set $\Gamma = A \cup B \cup C := \{\gamma_1, \ldots, \gamma_{m_A+m_B+m_C}\}$. In particular, note that there are no "links" or associations between the different messages that A sends.

We note that this model is more powerful than the random public anonymous bulletin board functionality presented next since ABB allows messages to be chosen adaptively, that is, dependent on previous messages.

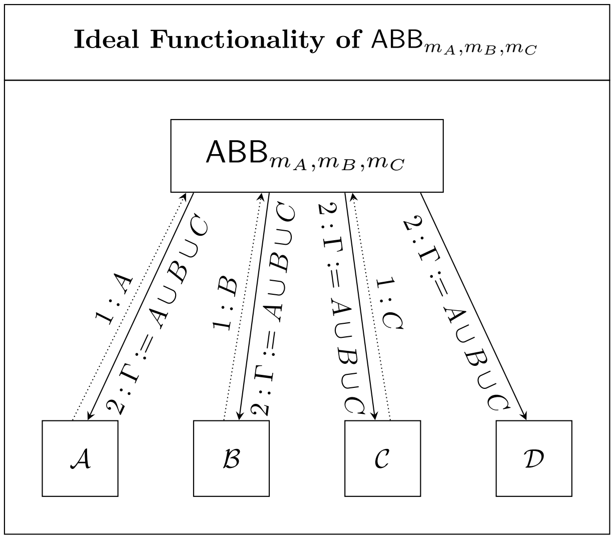

Ideal Functionality of $\mathsf{ABB}_{m_A,m_B,m_C}$
$\mathsf{ABB}_{m_A,m_B,m_C}$
$1 : A$
$2 : \Gamma := A \cup B \cup C$
$1 : B$
$2 : \Gamma := A \cup B \cup C$
$2 : \Gamma := A \cup B \cup C$
$1 : C$
$2 : \Gamma := A \cup B \cup C$
$\mathcal{A}$
$\mathcal{B}$
$\mathcal{C}$
$\mathcal{D}$

Figure 2 Ideal Functionality of Anonymous Bulletin Board (ABB). Dotted lines represent private authenticated channels, while solid lines represent public channels. The number in front of messages shows the order/round in which the messages are sent. A is the multi-set $\{a_1, \ldots, a_{m_A}\}$, B and C are similarly defined.

Additionally, we emphasize that one call to $\mathsf{ABB}_{m,m,m}$ is different from m calls to $\mathsf{ABB}_{1,1,1}$, even when we do not consider adaptive message choosing. Specifically, for m calls to $\mathsf{ABB}_{1,1,1}$, all parties will know that the first 3 messages did not come from the same parties. That is, parties gain additional information on subsets of messages that definitely did not come from the same party. Whereas in a single call to $\mathsf{ABB}_{m,m,m}$, parties do not gain such information.

4.2 Random Anonymous Bulletin Board

We also define a random anonymous bulletin board (rABB) in Figure 3, which takes additional parameters P, Q, R, which are independent distributions, and samples set of messages A according to distribution P, set of messages B according to distribution Q, set of messages C according to distribution R, privately outputs A to $\mathcal{A}$, B to $\mathcal{B}$, C to $\mathcal{C}$ (using private authenticated channels denoted with dashed lines), and outputs the multi-set Γ to all parties (using public channels denoted with solid lines). We define the ideal functionality as $\mathsf{rABB}^{P,Q,R}_{m_A,m_B,m_C}$.

4.3 Comparison between ABB and rABB

We note that the rABB functionality is as powerful as the ABB functionality when messages are not chosen adaptively. Essentially, since messages are not chosen adaptively, the parties should be able to determine the distribution of the messages they want to send before interacting with ABB. Therefore, parties can simply program a rABB with the appropriate P, Q, R in order to mimic the messages they are going to send. We note that P, Q, R can be distributions with sample space of size 1. Effectively making them deterministic.

In all of our protocols, parties sample their inputs uniformly at random (without replacement) and send them to the ABB. We note that this is precisely equivalent to rABB with P, Q, R set to output values uniformly at random without replacement. Intuitively, since the messages are chosen at random by semi-honest parties, it does not matter if the parties

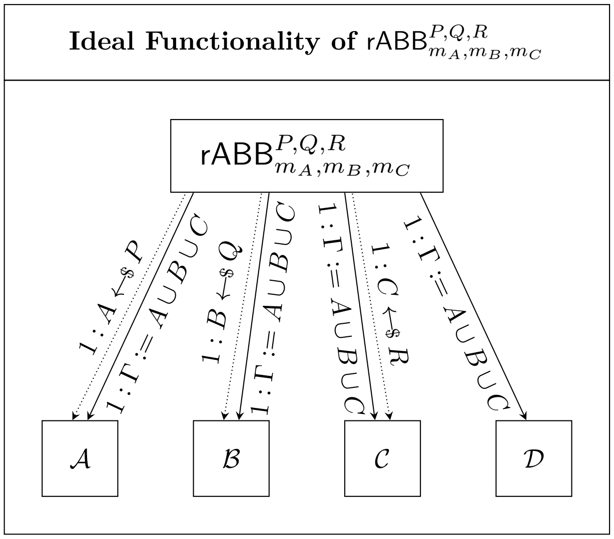

Ideal Functionality of $\mathsf{rABB}^{P,Q,R}_{m_A,m_B,m_C}$
$\mathsf{rABB}^{P,Q,R}_{m_A,m_B,m_C}$
$1 : A \xleftarrow{\$} P$
$1 : \Gamma := A \cup B \cup C$
$1 : B \xleftarrow{\$} Q$
$1 : \Gamma := A \cup B \cup C$
$1 : \Gamma := A \cup B \cup C$
$1 : C \xleftarrow{\$} R$
$1 : \Gamma := A \cup B \cup C$
$\mathcal{A}$
$\mathcal{B}$
$\mathcal{C}$
$\mathcal{D}$

Figure 3 Ideal Functionality of Random Anonymous Bulletin Board (rABB). Dotted lines represent private authenticated channels, while solid lines represent public channels. The number in front of messages shows the order/round in which the messages are sent. A is the multi-set $\{a_1, \ldots, a_{m_A}\}$ sampled according to P, B is sampled according to Q, and C is sampled according to R.

chose them and sent them to ABB, or if rABB chose them and sent them to the parties. The end result is both the ABB/rABB and parties will have the same random values.

4.4 Compression of Non-Adaptive Sequential and Parallel Calls

We also note that when messages are not chosen adaptively, we can compress multiple sequential calls to the ABB or rABB into a single call to the ABB/rABB by including a "sequence identifier" in each message. Furthermore, we can also compress multiple parallel calls to the ABB or rABB into a single call to the ABB or rABB by including a "parallel identifier" in each message.

Using the rABB as an example. Let $\mathsf{rABB}_{i,j}$ represent $\mathsf{rABB}^{P_{i,j},Q_{i,j},R_{i,j}}_{m_{A_{i,j}},m_{B_{i,j}},m_{C_{i,j}}}$.

Let us assume we want to call $\mathsf{rABB}_{1,1}, \mathsf{rABB}_{1,2}, \ldots, \mathsf{rABB}_{1,\pi}$ in the first round, $\mathsf{rABB}_{2,1}$, $\mathsf{rABB}_{2,2}$, $\ldots$, $\mathsf{rABB}_{2,\pi}$ in the second round, up to $\mathsf{rABB}_{\sigma,1}, \mathsf{rABB}_{\sigma,2}, \ldots, \mathsf{rABB}_{\sigma,\pi}$ in the σ^{th} round.

We can instead call $\mathsf{rABB}'_{i,j}$, where $\mathsf{rABB}'_{i,j}$ is $\mathsf{rABB}^{P'_{i,j},Q'_{i,j},R'_{i,j}}_{m_{A_{i,j}},m_{B_{i,j}},m_{C_{i,j}}}$, where $P'_{i,j}$ is the same distribution as $P_{i,j}$, except that every message is prefixed by i, j. $Q'_{i,j}$ and $R'_{i,j}$ are similarly defined.

Given this modification, we can now call $\mathsf{rABB}'' := \mathsf{rABB}^{P'',Q'',R''}_{\sum_{i,j} m'_A,\sum_{i,j} m'_B,\sum_{i,j} m'_C}$ where P'' is the union of all $P'_{i,j}$. Q'' and R'' are similarly defined. m'_A is defined as the sum of all $m'_{A_{i,j}}$, and m'_B and m'_C are similarly defined.

Given that all messages in rABB'' are prefixed by their sequence identifier and parallel identifier, the parties can locally divide them into the appropriate $\mathsf{rABB}'_{i,j}$s, and the result remains the same as the original.

4.5 An Equivalent Reformulation of rABB

In the semi-honest setting, the random anonymous bulletin board can be reformulated as the preprocessing step (offline phase) of the offline-online paradigm. In this step, parties will receive private correlated randomness from a conditional distribution.

Reformulation of $\mathsf{rABB}^{P,Q,R}_{m_A,m_B,m_C}$. The bulletin board samples (A, B, C) from the distribution (P, Q, R), where $A = \{a_1, a_2, \dots, a_{m_A}\}$, $B = \{b_1, b_2, \dots, b_{m_B}\}$, and $C = \{c_1, c_2, \dots, c_{m_C}\}$. Alice receives the multi-set A, Bob receives B, and $\mathcal{C}$ receives C. The bulletin board sends $\Gamma = A \cup B \cup C$ to all parties $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$. So, party $\mathcal{D}$ gets some side information about the correlated randomness of $\mathcal{A}, \mathcal{B}, \mathcal{C}$. We note that from the perspective of the parties, (A, B) is sampled according to the conditional distributions $(P, Q|\Gamma)$.

Discussion. Unlike the standard offline-online model in which correlated private randomness is sampled from a joint distribution, our model samples them from a conditional distribution, which is a family of joint distributions.

5 Key Agreement Protocols

This section presents our optimal length key agreement protocols in the rABB-hybrid in which party $\mathcal{C}$ does not send any messages to the bulletin board. We start by defining the problem setting.

5.1 Problem Setting

Suppose parties are in $\mathsf{rABB}^{P,Q}_{m_A,m_B}$-hybrid (without the helper $\mathcal{C}$). That is, the parties have access to a single instance of $\mathsf{rABB}^{P,Q}_{m_A,m_B}$, which they can call one time. We note that this is different from having access to $\max(m_A, m_B)$ different $\mathsf{rABB}^{P',Q'}_{1,1}$-hybrid. That is, party $\mathcal{A}$ has $A = \{a_1, a_2, \dots, a_{m_A}\}$ sampled according to P, party $\mathcal{B}$ has $B = \{b_1, b_2, \dots, b_{m_B}\}$ sampled according to Q, and party $\mathcal{D}$ has $A \cup B$. Every message is n-bit. Parties $\mathcal{A}$ and $\mathcal{B}$ are allowed to communicate with each other through a public noiseless channel. Party $\mathcal{D}$ (the eavesdropper) can see the messages sent between $\mathcal{A}$ and $\mathcal{B}$. At the end of the protocol, two parties $\mathcal{A}$ and $\mathcal{B}$ agree on a sample space Ω_L for the key, and $\mathcal{A}$ outputs a (variable-length) key $K_A \in \Omega_L$, and $\mathcal{B}$ outputs a key $K_B \in \Omega_L$. So, the key length is $\log|\Omega_L|$, where log denotes the logarithmic with base two. We define the *expected key length* of the protocol as the expectation of $\log|\Omega_L|$, where the expectation is taken over the randomness of samples A and B.

The protocol is correct if $K_A = K_B = K$ with high probability and K is close to a uniform distribution over Ω_L. It is secure if the eavesdropper $\mathcal{D}$ learns almost nothing about the key K. More formally, the statistical distance between two distributions $(K_A, K_B, T, A \cup B)$ and $(U_\Omega, U_\Omega, T, A \cup B)$ is small, where T is the transcript of the protocol (messages sent between $\mathcal{A}$ and $\mathcal{B}$).

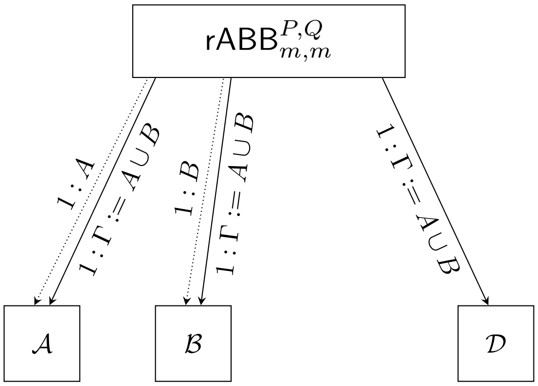
Given a desired key length k, we define the *failure probability* as

$$\min\left(\Pr[\mathsf{length}(K_A) < k], \Pr[\mathsf{length}(K_B < k)]\right),$$

where the probabilities are taken over the randomness of K_A and K_B, respectively. Typically, the failure probability is negligible. We define the *communication cost* as $m \cdot n + \mathsf{length}(T)$, where $\mathsf{length}(T)$ denotes the length of the protocol's transcript.

Objectives. In this work, we focus on the following objectives. Given a desired length $k \in \mathbb{N}$ and a failure probability δ, we are interested in constructing key agreement protocols

Key-Agreement Protocol



$\mathcal{A}$
1. Receive $A \in (\{0,1\}^n)^m := \{a_1, \ldots, a_m\}$ and $\Gamma = A \cup B := \{\gamma_1, \ldots, \gamma_{2m}\}$ from rABB.
2. Disregard γ_i, γ_j for all $i \neq j$ s.t. $\gamma_i = \gamma_j$, obtaining set $\Gamma' := \gamma'_1, \ldots, \gamma'_{2m'}$.
3. Compute $A' = A \cap \Gamma'$.
4. Compute $k :=$ *Algorithm* $1(A', \Gamma')$.

$\mathcal{B}$
1. Receive $B \in (\{0,1\}^n)^m := \{b_1, \ldots, b_m\}$ and $\Gamma = A \cup B := \{\gamma_1, \ldots, \gamma_{2m}\}$ from rABB.
2. Disregard γ_i, γ_j for all $i \neq j$ s.t. $\gamma_i = \gamma_j$, obtaining set $\Gamma' := \gamma'_1, \ldots, \gamma'_{2m'}$.
3. Compute $A' = \Gamma' - (B \cap \Gamma')$.
4. Compute $k :=$ *Algorithm* $1(A', \Gamma')$.

Figure 4 Key-agreement protocol between parties $\mathcal{A}$ and $\mathcal{B}$ in presence of an eavesdropper $\mathcal{D}$ in the $\mathsf{rABB}^{P,Q}_{m,m}$-hybrid, where P, Q are independent uniform distributions.

that output a (variable-length) key of length at least k with failure probability at most δ and with least communication costs.

Remark. Our protocols always achieve perfect correctness and perfect secrecy even when the key length is less than the desired threshold k. Our problem setting is similar to the key agreement model considered in [30]. The main difference is that parties get components A and B, respectively, of a conditional distribution of the form $(P, Q|Z)$ in our setting; while parties get A and B sampled according to a joint distribution of the form (P, Q) in their setting. Similar to the setting considered in another line of work [4,31,32], the eavesdropper has some side information about the samples of $\mathcal{A}$ and $\mathcal{B}$. The difference is that parties have access to multiple i.i.d samples in their setting, while parties have access to only one sample of a (large) joint distribution.

Remark. In our problem setting, parties agree on a variable-length key. One can rely on the asymptotic equipartition property and apply standard extraction procedures to obtain a fixed-length key.

5.2 Our Protocol

Next, we present our protocol in Figure 4, as well as provide an overview of the protocol below.

The protocol is similar to the example presented in [24]. At a high level, parties $\mathcal{A}$ and $\mathcal{B}$ will each receive m random values from the rABB, which are sampled from P and Q respectively. For security, P and Q are independent identical copies, and for optimal performance, P is a uniform distribution over $(\mathbb{Z}_{2^n})^m$ under the constraint that elements do not repeat. That is, P produces a set $A := \{a_1, \ldots, a_m\}$ such that all elements are equally likely to be in the set, and that for all $i \neq j, a_i \neq a_j$. Q produces a similar set B. Once they see the set of values, $\mathcal{A}$ and $\mathcal{B}$ can easily distinguish between $\mathcal{A}$'s values and $\mathcal{B}$'s values, while the eavesdropper cannot. Using this information, $\mathcal{A}$ and $\mathcal{B}$ can agree on a key K determined by the positions of $\mathcal{A}$'s values. Intuitively, there are $\binom{2m}{m}$ possible cases of which ones are $\mathcal{A}$'s values, and only $\mathcal{A}$ and $\mathcal{B}$ can identify one of the $\binom{2m}{m}$ possible cases.

We can also efficiently assign key values to the identified cases. We do so by assuming a canonical ordering of the elements in the set, and the more $\mathcal{A}$'s values are towards the "front" of the set, the higher the value of K is.

To efficiently compute the value of K, we employ standard techniques for analyzing fixed-weight bitstrings using combinatorial[3]. These techniques and algorithms have long been known and described in various places such as [7].

For completeness, we present such an algorithm in Algorithm 1 and briefly explain the logic behind the algorithm.

Algorithm 1 Key Determination

```
Parameters :
Input      : $A', \Gamma'$
Output     : $K$
We assume there exists some canonical ordering of elements in a set.
Identifies which of $\gamma'_1, \ldots, \gamma'_{2m'}$ are in the set $A'$, and create a bitstring $x$ such that if
  the $i^{th}$ element is in $A'$, then the $i^{th}$ bit is set to 1.
Compute $m' = size(A')$
$K = 0$
$c = m'$
for $i$ in range($2 \cdot m'$): do
    if $x[i] == 1$ then
        $K = K + \text{binomial}(2 \cdot m' - i - 1\ ,\ c)$
        $c = c - 1$
return $K$
```

5.2.0.1 Explanation of Algorithm 1

The idea behind Algorithm 1 is to first convert the set of elements into a binary string using the canonical ordering and assigning 1s at $\mathcal{A}$'s inputs and 0s to $\mathcal{B}$'s input. The algorithm can determine which bit string this is by examing all the 1 bits that appear and counting how many bit strings are skipped. For example, let us look at a simple case of $m = 3$, and the bit-string being 101010. Upon seeing the first 1, the algorithm knows all bit-strings of the form $0*****$ have been skipped over, where the $*****$ consists of 3 1s, and 2 0s. There are $\binom{5}{3}$ of them. Then, upon seeing the next 1, the algorithm knows all bit-strings of the form $100***$ have been skipped over, where the $***$ represents 2 1s and 1 0s. There are $\binom{3}{2}$ of them. By continuing through the entire bit-string, the algorithm can determine the value of this bit-string, thus determining the value of the key K.

[3] We thank our anonymous reviewers for pointing us in the direction of these techniques.

5.3 Performance Analysis and Parameter Choices

Now, we use concentration bounds to show that the expected key length in our protocol is highly concentrated around the mean that is large. Therefore, our protocol outputs a long key with high probability.

▶ **Theorem 8.** *Fix $m, n \in \mathbb{N}$, and $0 \leqslant \varepsilon \leqslant 1$. Our non-interactive protocol in Figure 4 uses m messages (each being n-bit strings) to help parties agree on a key of length at least $k = \log \binom{2m'}{m'} = (2m') \cdot (1 - o(1))$ with failure probability at most $\exp\left(-\varepsilon^2 \cdot m \cdot \left(1 - \frac{m}{2^n}\right)/2\right)$, where $m' = m \cdot \left(1 - \frac{m}{2^n}\right) \cdot (1 - \varepsilon)$.*

Proof. Let us fix the elements of $\mathcal{B}$. For each element of $\mathcal{A}$, there is at least $\left(1 - \frac{m}{2^n}\right)$ probability that such an element of $\mathcal{A}$ will not be a duplicate of an element of $\mathcal{B}$. Therefore, we can apply a simplified Chernoff Bound. For $2m'$ unique elements, $\mathcal{A}$ and $\mathcal{B}$ can agree on a $\log \binom{2m'}{m'} \approx \log \left(\frac{4^m}{\sqrt{m\pi}}\right)$-bit key.

▶ **Theorem 9.** *Fix $m, n \in \mathbb{N}$, satisfying $m = o\left(2^{n/3}\right)$. Using m messages of size n-bit each, our protocol in Figure 4 allows parties to agree on a $k = \log \binom{2m}{m} = 2m \cdot (1 - o(1))$ bit key with probability at least $1 - \exp\left(-\frac{m^2}{2 \cdot 2^n}\right)$.*

Proof. By the birthday bound, with probability at least $1 - \exp\left(-\frac{m^2}{2 \cdot 2^n}\right)$, all $2m$ values will be unique. For $2m$ unique elements, $\mathcal{A}$ and $\mathcal{B}$ can agree on a $\log \binom{2m}{m}$-bit key.

Parameter Choices. Our protocol has two main parameters, m and n. The expected key length increases with m and n. At the same time, the communication cost increases with m and n as well. However, we note that this is a simple optimization problem and that automatic searches for optimal parameters can be done. Furthermore, for common key-length such as 128 or 256 bits, the search only has to be performed once.

We perform this automated search and present our results in Figure 5. Concretely, using 702-bits of communication, $\mathcal{A}$ and $\mathcal{B}$ can agree on a 128-bit key in expectation, and using 1550-bits of communication, $\mathcal{A}$ and $\mathcal{B}$ can agree on a 256-bit key in expectation.

5.4 Comparison With previous State-of-the-Art

Recall that we are in the setting where parties have access to a single ABB or rABB that they can call once. To the best of our knowledge, the best previous state-of-the-art key agreement protocol that works in this setting is [5]. Other protocols such as [9] and [14] require that parties send their message in a random order (which requires additional assumptions on the synchrony of the system model), and are often presented as using sequential calls to the ABB in order to perform key agreement on more than one bit. The key agreement protocols presented in [24] either require more communication rounds and communication costs than the one in [5], or are fundamentally similar to the one in [5].

Therefore, we focus our concrete communication cost analysis on comparison with [5]. Additionally, we present our analysis using the expected value of the key length. Similar analyses can be done on achieving the desired key length with high probability instead of in expectation by using standard techniques on concentration bounds, etc.

To achieve a k-bit key in expectation, [5] requires each party to send $2k$ messages, each of length $\log(2k) + 1$, where $\log(2k)$ bits are used to represent the sequence identifier and 1-bit is used to represent the random bit chosen. This results in a total communication cost of $2k \cdot (\log(2k) + 1)$. Even when we assume that parties do not send the leading 0s in the sequence identifiers, the total communication cost is still $2k + \sum_{i=1}^{\log(2k)} i \cdot 2^{i-1} = 2k \log(2k) + 1$.

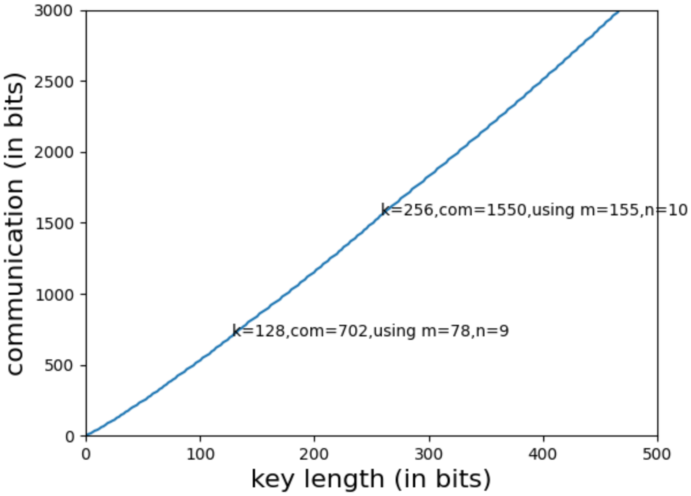


Figure 5 For various key lengths k, we plot the communication required (in bits) such that the expected key length is at least k. We also label two points, one for 128-bit keys and one for 256-bit keys. We label the key length and communication required, as well as the parameter choices of m and n used.

We then compare this communication cost for various key lengths against the result from our automated search results presented in Figure 5 by taking a ratio of their communication cost divided by our communication cost, which produced Figure 1 shown in our contribution at the beginning of the paper.

For example, for 128-keys, their protocol required $2 \cdot 128 \log(2 \cdot 128) + 1 = 2049$ bits of communication while our protocol required 702 bits of communication, resulting in a ratio of $\frac{2049}{702} = 2.9188$.

For completeness, we also analyze the sequential version of the key-agreement protocol in [5]. In the sequential protocol, parties use number of rounds linear in the length of the key to achieve a lower communication cost. To obtain a k bit key, each party sends 1 bit in each of the $2k$ rounds, resulting in $2k$ total bits of communication per party.

5.5 On the Optimality of Key Length

In this section, we analyze protocols under the setting of having arbitrary/unlimited message length and show that our protocol in $Figure$ 4 is near optimal in terms of the expected key length based on the number of messages. In fact, we will prove a much stronger result. That is, the expected key length of any interactive protocol for key agreement in the $\mathsf{rABB}^{P,Q}_{m,m}$ is at most $2m - \mathsf{poly}(\log \mathsf{m})$, where P, Q are arbitrary independent distributions over $(\{0,1\}^n)^m$, and n is the message length. By Theorem 8, for any $\varepsilon > 0$, our protocol achieves $2(1-\varepsilon)m$-bit key length with an exponentially small failure probability (depending on ε). This means that our *non-interactive protocol* asymptotically achieves the optimal key length of the best interactive protocol. We provide detailed proof below.

First, we upper bound the expected key length by the mutual information. Then, we show that the mutual information of any $\mathsf{rABB}^{P,Q}_{m,m}$ is at most $\log \binom{2m}{m}$.

▶ **Theorem 10.** *Let $m, n \in \mathbb{N}$ and P, Q be independent distributions over $(\{0,1\}^n)^m$. Suppose parties are in the random public anonymous bulletin board hybrid $\mathsf{rABB}^{P,Q}_{m,m}$. Then,* *the expected key length in any key agreement protocol (allowing interaction) is at most* $I(\mathsf{rABB}^{P,Q}_{m,m}) + 1 + \log 3$*.*

We shall employ the techniques developed recently in [29, 30] to prove the theorem above. We say that Alice and Bob are in (X, Y)-correlation hybrid if Alice has x and Bob has y, where (x, y) is sampled according to the joint distribution (X, Y). The following result shall be useful for the proof.

▶ **Theorem 11.** *[29, 30] Let (X, Y) be a joint distribution. Then, the maximal expected key length in the (X, Y)-correlation hybrid (allowing an arbitrary amount of communication) is at most $I(X, Y) + 1 + \log 3$.*

Proof of Theorem 10. Recall the reformulation of rABB as correlated private randomness in Section 4.5. The correlation $\mathsf{rABB}^{P,Q}_{m,m}$ is a conditional distribution of the form $(X, Y|Z)$, where Z the random variable denoting the eavesdropper's view (the set $A \cup B \cup C$). Conditioned on fixing the eavesdropper's view ($Z = z$), applying Theorem 11 to the joint distribution $(X, Y|Z = z)$ yields that the key length is at most $I(X, Y|Z = z) + 1 + \log 3$. Thus, the expected key length is at most

$$\mathbb{E}_z[I(X, Y|Z = z) + 1 + \log 3] = I(X, Y|Z) + 1 + \log 3.$$

Next, we bound the mutual information of the rABB.

▶ **Lemma 12.** *Let $(X, Y|Z)$ be the correlation corresponding to the random public bulletin board $\mathsf{rABB}^{P,Q}_{m,m}$. For each z in the sample space of the random variable Z, let ℓ_z be the length of z after removing all duplicate elements. Then*

$$I(X, Y|Z) = \sum_z p_Z(z) \cdot \log \binom{2\ell_z}{\ell_z} = \mathbb{E}_z \left[\log \binom{2\ell_z}{\ell_z} \right].$$

Proof. First, note that $Z = X \cup Y$. Thus, $H(X|Y, Z) = 0$ since X is completely determined conditioned on knowing Y and Z. We have

$$\begin{aligned} I(X, Y|Z) &= \sum_z p_Z(z) \cdot I(X, Y|Z = z) \\ &= \sum_z p_Z(z) \cdot (H(X|Z = z) - H(X|Y, Z = z)) && (\mathit{Fact}\ 1) \\ &= \sum_z p_Z(z) \cdot H(X|Z = z) \end{aligned}$$

For each $x = \{a_1, a_2, \ldots, a_m\}$ in the sample space of X, there is no duplicates in x; that is $a_i \neq a_j$ for every $i \neq j$. Conditioned on $Z = z = \{a_1, \ldots, a_m, b_1, \ldots, b_m\}$, which might contain duplicates, the number of x that are consistent with z is $\binom{2\ell_z}{\ell_z}$. Thus, the support's size of the random variable $(X|Z = z)$ is $\binom{2\ell_z}{\ell_z}$. Observe that the random variable $(X|Z = z)$ is uniform over its support. This implies that $H(X|Z = z) = \log \binom{2\ell_z}{\ell_z}$, for every z such that $p_Z(z) > 0$. Therefore, we have

$$H(X, Y|Z) = \sum_z p_Z(z) \cdot \log \binom{2\ell_z}{\ell_z},$$

which completes the proof. ◀

By our construction in Figure 4, it is clear that the expected key length of our protocol is the quantity $E_z \log \binom{2\ell_z}{\ell_z}$ defined above. The following results are consequences of Lemma 12.

▶ **Corollary 13.** *The expected key length of the protocol in Figure 4 is exactly $I(\mathsf{rABB}_{m,m}^{P,Q})$, where P and Q are the distribution that samples m messages randomly without replacement.*

▶ **Corollary 14.** *Let $m, n \in \mathbb{N}$ and let P, Q be arbitrary distributions over $(\{0,1\}^n)^m$. Then, the expected key length of any protocol in the $\mathsf{rABB}_{m,m}^{P,Q}$ is at most $\log\binom{2m}{m}$.*

5.6 Security Analysis

▶ **Theorem 15.** *The key-agreement protocol in Figure 4 securely establishes a shared key K between $\mathcal{A}$ and $\mathcal{B}$, with $\mathcal{D}$ learning no information regarding the key.*

Proof. We give an outline of the security proof. The correctness comes from the fact that $\mathcal{A}$ can determine which elements were received by $\mathcal{A}$ and thus belong to A, while $\mathcal{B}$ can determine which elements were received by $\mathcal{B}$ and thus belong to B. Since $\Gamma := A \cup B$, $\mathcal{B}$ is also able to determine which elements were received by $\mathcal{A}$ and thus belong to A. Therefore, the information that $\mathcal{A}$ and $\mathcal{B}$ have are the same, and will allow them to agree on the same key K. Regarding privacy, note that due to the property of rABB, only $\mathcal{A}$ and $\mathcal{B}$ can determine which elements were received by $\mathcal{A}$ and thus belong to A. To $\mathcal{D}$, elements belonging to A and B look indistinguishable. Therefore, only $\mathcal{A}$ and $\mathcal{B}$ will know the value of the key K. We also note that since A and B are chosen according to the same distribution (uniform distribution in this case), all key values are equally likely to occur and the adversary gains no information. ◀

5.7 Duplicate Recovery

We discuss a variant of our protocol named duplicate-recovery variant that is especially useful in settings where m is relatively close to 2^n, which means that many duplicates are likely to occur. This protocol allows parties to consider duplicates as opposed to disregarding them and is optimal in these settings. (We note that under the same m, having duplicates will decrease the key length k. This variant is designed to "recover" slightly from cases when duplicates exist. However, increasing n such that no duplicates occur will result in a larger key length.)

We also highlight a combinatorial problem that naturally arises in this variant that may be of independent interest.

Protocol Overview. This variant of the protocol is very similar to the original key-agreement protocol, with the key difference being P and Q allows sample with replacement, i.e. duplicate messages from the same party are possible and parties do not discard the duplicates. Then, when parties want to determine the key k, they have to consider all possible cases. (Note that for security, P cannot sample values independently and uniformly at random, instead, P samples in a way such that every "set" occurs with equal probability. For example, $\{0,0\}$ and $\{0,1\}$ have the same probability of occurring. With this, the security of this variant closely follows the security of the original protocol.)

In general, given a multi-set of values Γ, one can list all possible values of A and B that can produce such a multi-set. However, a direct listing requires exponential computation as there are exponentially many possible cases. Given a generic algorithm for counting the number of possible cases, one can apply the same idea as Algorithm 1 to recursively determine the k value of a given set of inputs. We note that this generic algorithm for counting the number of possible cases given Γ may be of independent interest in the field of combinatorics.

Counting Solutions. We elaborate more on the problem of counting the number of possible sets that exist for a given Γ. One can easily do so by enumerating all possible solutions. For

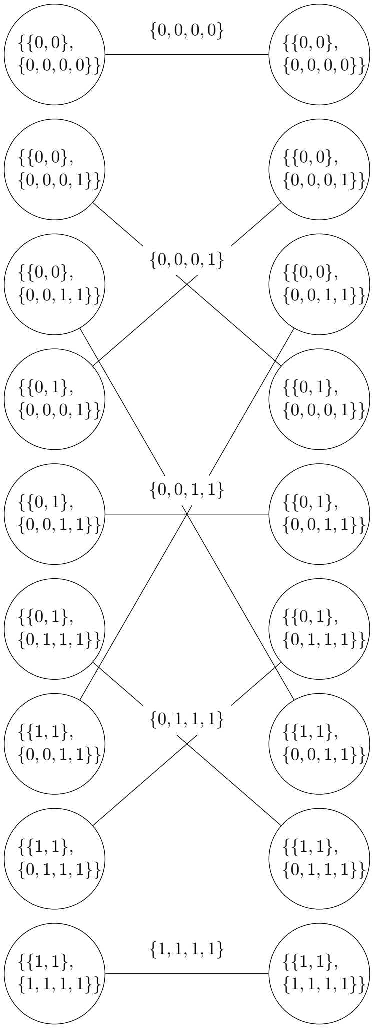


Figure 6 Bipartite-graph for parties view and transcript for $m = 2$, $n = 1$. The nodes at the left represent the view of $\mathcal{A}$, which includes its private input (A) and the transcript it sees (Γ). Similarly, the nodes at the right represent the view of $\mathcal{B}$. Edges represent consistent views (transcript matches the private inputs produce correct transcript), with the transcript (Γ) labeled above the edges.

example, let us look at the simple case of when $m = 2$ and $n = 1$. Figure 6 shows all possible Γ and the corresponding possible A and B. We can clearly see that when $\Gamma = \{0, 0, 1, 1\}$, there are 3 possible solutions for A, B, and when $\Gamma = \{0, 1, 1, 1\}$, there are 2 possible solutions for A, B.

Alternatively, we can model it as an Integer Programming (IP) problem. Let us define $z_1, \ldots, z_{N-1}$ as z_i being the amount of the element i showing up in Γ. Similarly, define $x_1, \ldots, x_{N-1}$ as x_i being the amount of i showing up in A, and $y_1, \ldots, y_{N-1}$ being the amount of i in B. Note that $\forall i, x_i \in \mathbb{Z}, y_i \in \mathbb{Z}, z_i \in \mathbb{Z}$.

We need to find the number of possible solutions to the x_is and y_js satisfying the equations and constraints

$$\sum_{i=0}^{N-1} x_i = m$$

$$\sum_{i=0}^{N-1} y_i = m$$

$$\forall i, x_i + y_i = z_i$$

$$\forall i, x_i \geqslant 0$$

$$\forall i, y_i \geqslant 0$$

To the best of our knowledge, there are no works specifically answering this question. We believe this question may be of independent interest. We also note that when there are no duplicates, that is $\forall i, z_i \leqslant 1$, this reduces to a simple binomial problem with the solution being $\binom{2m}{m}$.

6 Binary Erasure Channel

Rabin and Crépeau [11, 37, 38] showed that binary erasure channels suffice for general secure computation using interaction. These elegant noise sources provide uncluttered access to abstract the primary hurdles in achieving security. This section focuses on constructing binary erasure channels in the rABB-hybrid. We present our round-optimal secure protocols.

6.1 Problem Setting.

Suppose parties are in the $\mathsf{rABB}^{P,Q,R}_{m_A,m_B,m_C}$ hybrid (with the helper $\mathcal{C}$). That is, party $\mathcal{A}$ has $A = \{a_1, a_2, \ldots, a_{m_A}\}$ sampled according to P, party $\mathcal{B}$ has $B = \{b_1, b_2, \ldots, b_{m_B}\}$ sampled according to Q, and $\mathcal{C}$ has $C = \{c_1, c_2, \ldots, c_{m_C}\}$ sampled according to R. Furthermore, all parties have the set $A \cup B \cup C$. Parties $\mathcal{A}$ and $\mathcal{B}$ want to establish a binary erasure channel $\mathsf{BEC}(p)$ between them, with $\mathcal{A}$ being the receiver and $\mathcal{B}$ being the sender. Parties $\mathcal{A}$ and $\mathcal{B}$ can communicate via an authenticated channel. We are in the semi-honest adversary model; that is, parties follow the protocol description but are curious to learn more from the protocol's transcript. Unlike in the key agreement protocols, the adversary here can corrupt a party.

For the binary erasure channel, without loss of generality, we assume that $\mathcal{B}$ is the sender and $\mathcal{A}$ is the receiver. So $\mathcal{B}$ will send a bit β to $\mathcal{A}$. The protocol is correct if $\mathcal{A}$ receives β with probability $(1-p)$, and $\mathcal{A}$ receives $\perp$ (nothing) with erasure probability p. We define security following the standard simulation-based definition. Intuitively, it is secure against the corrupted sender $\mathcal{B}$ if $\mathcal{B}$ does not know whether the sender bit gets erased or not; it is secure against the corrupted receiver if the sender bit is uniformly random in the receiver's view whenever she outputs $\perp$. We say that the protocol is ε-statistical secure if the simulation error is at most ε, and perfectly secure if $\varepsilon = 0$.

We assume that no duplicates exist throughout the rest of the section. In the event that duplicates occur, parties can abort and rerun the protocol. Practically, by setting n to be large enough, we can ensure that with a high probability, no duplicates exist.

Remark. We note that with the key-agreement protocols in the previous section, $\mathcal{A}$ and $\mathcal{B}$ can establish a private authenticated channel from an authenticated channel. Furthermore, using the technique in Section 4.4, we can establish the private authenticated channel in parallel with the BEC protocol without using any additional rounds of communication.

▶ **Theorem 16.** *Let $p \in (0, 1)$ be a rational number. There is a perfectly secure one-round protocol for* $\mathsf{BEC}(p)$ *in the* rABB*-hybrid.*

Remark. The communication cost in our protocol is proportional to the denominator of the erasure probability. We left determining the minimum communication cost as an open problem.

Note that any irrational number can be approximated by a rational number with arbitrary precision. For example, this can be done by using Drichilet's approximation algorithm. Therefore, we have the following result as a corollary.

▶ **Corollary 17.** *For any erasure probability $p \in (0,1)$ and $\varepsilon \in (0,1)$, there is a ε-statistical secure one-round protocol for* $\mathsf{BEC}(p)$ *in the* rABB*-hybrid.*

6.2 Construction

We present our protocol in Figure 7, as well as provide an overview of the protocol below.

The main idea behind the protocol is that given a set of two values, one from A and one from B, both $\mathcal{B}$ and $\mathcal{A}$ can distinguish and identify the owner of the value, and agree on a one-bit key used to send a one-bit message. On the other hand, if the set of two inputs is from B and C, then $\mathcal{A}$ learns nothing about the key and thus nothing about the message.

Therefore, in the protocol, $\mathcal{A}$ and $\mathcal{C}$ will each receive several values from the rABB, while $\mathcal{B}$ receives one value. $\mathcal{B}$ will then select two values from the multi-set published by the rABB, ensuring that one of them is his value, and encrypt the bit message using the key derived from those two values. If he selected his value and one of $\mathcal{C}$'s values, then the message is erased. If he selected his value and one of $\mathcal{A}$'s values, then $\mathcal{A}$ receives the message. Note that $\mathcal{B}$ cannot distinguish between $\mathcal{A}$'s and $\mathcal{C}$'s value, so $\mathcal{B}$ will not know if the message was erased, and $\mathcal{A}$ cannot distinguish between $\mathcal{B}$ and $\mathcal{C}$'s value, so the message can indeed be erased.

6.3 Correctness and Security Proofs

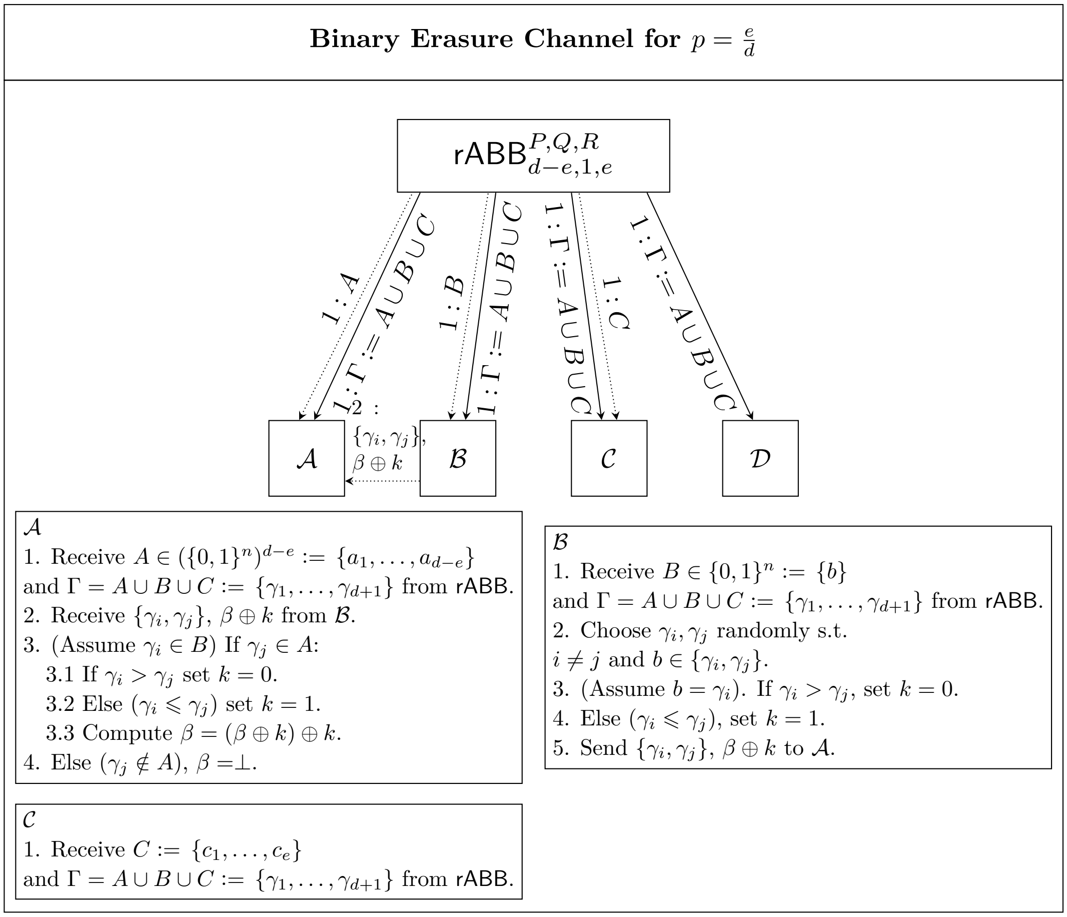
Correctness. Observe that $\gamma_i \neq \gamma_j$ since there are no collisions at all. Thus, $\gamma_i > \gamma_j$ with probability $1/2$ and $\gamma_i < \gamma_j$ with probability $1/2$. Thus, the bit k is a uniformly random bit. Observe that $\gamma_i \in A$ with probability $(d-e)/d$. So, $\mathcal{A}$'s output is β with probability $1-e/d$ and $\perp$ with probability e/d. Therefore, the protocol is perfectly correct.

Security. For security against a corrupted $\mathcal{B}$, whether the bit gets erased or not depends entirely on the event $\gamma_j \in A$ that $\mathcal{B}$ knows nothing about. Therefore, $\mathcal{B}$ does not know whether $\mathcal{A}$'s output is $\perp$ (erased) or β. For security against a corrupted $\mathcal{A}$, we need to show that when $\mathcal{A}$ outputs $\perp$, the bit β is uniformly random in the view of $\mathcal{A}$. $\mathcal{A}$ outputs $\perp$ when $\gamma_j \notin A$. In $\mathcal{A}$'s view, the event $\gamma_i > \gamma_j$ is uniformly random. It means that $\mathcal{A}$ has β masking with a uniformly random bit. Hence, it follows from the property of the one-time pad that the bit β is uniformly random in $\mathcal{A}$'s view.

6.4 On the Round Optimality of Our Protocols

Our protocols use only one round of communication from $\mathcal{B}$ to $\mathcal{A}$. This section will show that it is impossible to securely implement BEC in the rABB-hybrid without communication. In fact, we will show that it is impossible to implement the BEC with randomized inputs, a weaker functionality. We shall employ the techniques from secure non-interactive simulation (SNIS/SNIR), recently introduced in [2, 26, 27], to prove the following theorem.

▶ **Theorem 18.** *Let $p \in (0,1)$ be the erasure probability. Any zero round protocol implementing* $\mathsf{BEC}(p)$ *in* $\mathsf{rABB}^{U_A,U_B,U_C}_{m_A,m_B,m_C}$*-hybrid has constant insecurity, where U_A, U_B, U_C are uniform*


Binary Erasure Channel for $p = \frac{e}{d}$
$\mathsf{rABB}^{P,Q,R}_{d-e,1,e}$
$1 : A$
$1 : \Gamma := A \cup B \cup C$
$1 : B$
$1 : \Gamma := A \cup B \cup C$
$1 : \Gamma := A \cup B \cup C$
$1 : C$
$1 : \Gamma := A \cup B \cup C$
$2 :$
$\{\gamma_i, \gamma_j\}$,
$\beta \oplus k$
$\mathcal{A}$
$\mathcal{B}$
$\mathcal{C}$
$\mathcal{D}$

```
$\mathcal{A}$
1. Receive $A \in (\{0,1\}^n)^{d-e} := \{a_1, \dots, a_{d-e}\}$
and $\Gamma = A \cup B \cup C := \{\gamma_1, \dots, \gamma_{d+1}\}$ from rABB.
2. Receive $\{\gamma_i, \gamma_j\}$, $\beta \oplus k$ from $\mathcal{B}$.
3. (Assume $\gamma_i \in B$) If $\gamma_j \in A$:
    3.1 If $\gamma_i > \gamma_j$ set $k = 0$.
    3.2 Else ($\gamma_i \leqslant \gamma_j$) set $k = 1$.
    3.3 Compute $\beta = (\beta \oplus k) \oplus k$.
4. Else ($\gamma_j \notin A$), $\beta = \perp$.
```

```
$\mathcal{B}$
1. Receive $B \in \{0,1\}^n := \{b\}$
and $\Gamma = A \cup B \cup C := \{\gamma_1, \dots, \gamma_{d+1}\}$ from rABB.
2. Choose $\gamma_i, \gamma_j$ randomly s.t.
$i \neq j$ and $b \in \{\gamma_i, \gamma_j\}$.
3. (Assume $b = \gamma_i$). If $\gamma_i > \gamma_j$, set $k = 0$.
4. Else ($\gamma_i \leqslant \gamma_j$), set $k = 1$.
5. Send $\{\gamma_i, \gamma_j\}$, $\beta \oplus k$ to $\mathcal{A}$.
```

```
$\mathcal{C}$
1. Receive $C := \{c_1, \dots, c_e\}$
and $\Gamma = A \cup B \cup C := \{\gamma_1, \dots, \gamma_{d+1}\}$ from rABB.
```

Figure 7 Binary Erasure Channel from $\mathcal{B}$ to $\mathcal{A}$ using a helper $\mathcal{C}$ and in presence of an eavesdropper $\mathcal{D}$

distribution over $(\{0,1\}^n)^{m_A}, (\{0,1\}^n)^{m_B}, (\{0,1\}^n)^{m_C}$ *respectively, and* n *is the message length.*

Proof Sketch. We prove this by contradiction. Suppose that it is possible to get $\mathsf{BEC}(p)$ from the $\mathsf{rABB}^{U_A,U_B,U_C}_{m_A,m_B,m_C}$. It follows from [2,26] that if it is possible to implement the randomized inputs $\mathsf{BEC}(p)$ from some other distribution (X,Y), then the eigenvalues of $\mathsf{BEC}(p)$ must be a subset of eigenvalues of the distribution (X,Y). Note that the eigenvalues of $\mathsf{BEC}(p)$ are 1 and $\sqrt{1-p}$. The correlation $\mathsf{rABB}_{m_A,m_B,m_C}$ is a family of joint distributions of the form $(X,Y|Z)$. Therefore, it must be the case that $\sqrt{1-p}$ is an eigenvalue of the correlation $(X,Y|Z=z)$, for every z in support of the random variable Z. This implies that $\sqrt{1-p}$ is an eigenvalue of all the conditional distributions $(X,Y|Z=z)$, which is impossible.

We provide elaborated arguments on Appendix A.

7 Random String Oblivious Transfer

In this section, we consider the construction of chosen message random oblivious transfer (cmROT) (see Section 3 for definitions) in the rABB-bybrid.

In the rABB-hybrid with the helper $\mathcal{C}$, we construct an efficient one-round protocol. We start by describing the problem setting as follows.

Problem Setting. $\mathcal{A}$ and $\mathcal{B}$ will like to establish a chosen message random string oblivious transfer (cmROT^ℓ) for ℓ-bit strings between them with $\mathcal{A}$ being the receiver and $\mathcal{B}$ being the sender. They have access to a rABB and a helper party $\mathcal{C}$, as well as a public authenticated channel from $\mathcal{B}$ to $\mathcal{A}$. We note that with the key-agreement protocol, $\mathcal{A}$ and $\mathcal{B}$ can turn the public authenticated channel into a private authenticated channel. Furthermore, by using the technique in Section 4.4, this key-agreement can be done in parallel with the first step of the cmROT^ℓ protocol and does not require any additional round.

We also note that in our setting, we assume that no duplicates exist. Practically, by setting n to be large enough, we can ensure that with a high probability, no duplicates exist. In the event that duplicates occur, parties can simply abort and rerun the protocol. Therefore, we assume that no duplicates exist throughout the rest of the section.

We shall prove the following theorem.

▶ **Theorem 19.** *For any $\ell \in \{1, 2, \dots\}$, there is a perfectly secure one-round protocol for* cmROT^ℓ *in the* rABB*-hybrid (with the helper).*

7.1 Construction

Intuition. We begin with some intuition. The main idea behind the protocol is that given a set of two values, one from A and one from B, both $\mathcal{B}$ and $\mathcal{A}$ can distinguish and identify the owner of the values, and agree on a random one-bit message. On the other hand, if the set of two values is from B and C, then $\mathcal{A}$ learns nothing about the message. Additionally, observe that this still holds if we randomly set all messages from A to be even (or odd) and C to be odd (or even respectively), while messages from B contain both even and odd values.

Additionally, by using the technique discussed in Section 4.4, we can effectively perform several parallel calls to the rABB in the same round.

Formally, the parties utilize $\mathsf{rABB}^{P,Q,R}$, where $P = P_1 \cup P_2 \cup \cdots \cup P_\sigma$, $Q = Q_1 \cup Q_2 \cup \cdots \cup Q_\sigma$, and $R = R_1 \cup R_2 \cup \cdots \cup R_\sigma$ for some parameter $\sigma \in \{1, 2, \dots\}$ (chosen appropriately later).

Independently, each P_i is in "even mode" with probability $\frac{1}{2}$, which means $P_i = P_{i,1}^{(EVEN)} \cup P_{i,2}^{(EVEN)} \cup \cdots \cup P_{i,\ell}^{(EVEN)}$, where each $P_{i,j}^{(EVEN)}$ samples a n bit even value (with least significant bit being 0) uniformly at random, denoted as $\alpha_{i,j}^{(EVEN)}$, and outputs the tuple $\left(i, j, \alpha_{i,j}^{(EVEN)}\right)$. Similarly, each P_i is in "odd mode" with probability $\frac{1}{2}$, which means $P_i = P_{i,1}^{(ODD)} \cup P_{i,2}^{(ODD)} \cup \cdots \cup P_{i,\ell}^{(ODD)}$, where each $P_{i,j}^{(ODD)}$ samples a n bit odd value (with least significant bit being 1) uniformly at random, denoted as $\alpha_{i,j}^{(ODD)}$, and outputs the tuple $\left(i, j, \alpha_{i,j}^{(ODD)}\right)$.

R_i is defined as an independent copy of P_i. Similarly, we denote the output of $Q_{i,j}^{(EVEN)}$ as $\left(i, j, \omega_{i,j}^{(EVEN)}\right)$.

$Q_i = Q_{i,1} \cup Q_{i,2} \cup \cdots \cup Q_{i,\ell}$, where each $Q_{i,j}$ independently samples one even n-bit value uniformly at random and samples one odd n-bit value uniformly at random, and outputs the set of the two tuples $\left\{\left(i, j, \beta_{i,j}^{(EVEN)}\right), \left(i, j, \beta_{i,j}^{(ODD)}\right)\right\}$.

Parties first invoke $\mathsf{rABB}^{P,Q,R}$, which samples A according to distribution P, samples B according to distribution Q, and samples C according to distribution R. $\mathsf{rABB}^{P,Q,R}$ then sends A to $\mathcal{A}$, B to $\mathcal{B}$, and C to $\mathcal{C}$, as well as send $\Gamma = A \cup B \cup C$ to $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$.

$\mathcal{A}$ now has $\sigma\ell$ values in the form of $\left(i, j, \alpha_{i,j}^{()}\right)$. $\mathcal{C}$ now has $\sigma\ell$ values in the form of $\left(i, j, \omega_{i,j}^{()}\right)$. $\mathcal{B}$ now has $2\sigma\ell$ values in the form of $\left(i, j, \beta_{i,j}^{()}\right)$. All parties also see the set containing all values.

Using the index information, parties can locally separate the values according to i, j. Each i, j should now contain $\left(i, j, \alpha_{i,j}^{()}\right)$, $\left(i, j, \beta_{i,j}^{(EVEN)}\right)$, $\left(i, j, \beta_{i,j}^{(ODD)}\right)$, and $\left(i, j, \omega_{i,j}^{()}\right)$.

The parties will first look at $i, j = 1$, and find the lexicographically smallest i such that the four values they see contain exactly two even and two odd values. We denote this as i^*. The parties will now disregard all $i \neq i^*$, and focus only on i^*, j. Note that this means that P_{i^*} and R_{i^*} were in different modes, that is, if P_{i^*} was in even mode, then R_{i^*} was in odd mode, or vice versa. Furthermore, $\mathcal{A}$, having access to $\alpha_{i^*,j}^{()}$, can identify whether P_{i^*} was in even mode or odd mode. Without loss of generality, let us assume P_{i^*} was in even mode, and R_{i^*} is in odd mode. Each i^*, j now contains $\left(i^*, j, \alpha_{i^*,j}^{(EVEN)}\right)$, $\left(i^*, j, \beta_{i^*,j}^{(EVEN)}\right)$, $\left(i^*, j, \beta_{i^*,j}^{(ODD)}\right)$, and $\left(i^*, j, \omega_{i^*,j}^{(ODD)}\right)$.

$\mathcal{B}$ now computes the two "intermediate message" y_{EVEN} and y_{ODD} (we can equivalently think of them as y_0 and y_1). To compute the j^{th} bit of y_{EVEN}, denoted as $y_{EVEN,j}$, $\mathcal{B}$ looks at $(i^*, j, \beta_{i^*,j,EVEN})$ and $(i^*, j, \alpha_{i^*,j,EVEN})$. If $\beta_{i^*,j,EVEN} \geqslant \alpha_{i^*,j,EVEN}$, then $y_{EVEN,j} = 0$, else $y_{EVEN,j} = 1$. Note that $\mathcal{B}$ simply compares the other even value against $\beta_{i^*,j,EVEN}$. In particular, $\mathcal{B}$ does not know whether he is comparing against $\alpha_{i^*,j,EVEN}$ or $\omega_{i^*,j,EVEN}$. Similarly, if $\beta_{i^*,j,ODD}$ is the greater of the two odd values ($\beta_{i^*,j,ODD} \geqslant \omega_{i^*,j,ODD}$ in this case), then $y_{ODD,j} = 0$, else $y_{ODD,j} = 1$.

$\mathcal{B}$ can do this for all $i^*, 1$ to i^*, ℓ, and obtain two ℓ-bit messages y_{EVEN} and y_{ODD}. $\mathcal{B}$ then computes ℓ-bit "key" r_{EVEN} such that $y_{EVEN} \oplus r_{EVEN} = x_{EVEN} = x_0$ and r_{ODD} such that $y_{ODD} \oplus r_{ODD} = x_{ODD} = x_1$.

Since $\mathcal{A}$ has $\alpha_{i^*,j,EVEN}$, $\mathcal{A}$ similarly computes y_{EVEN} by comparing the two even values against each other and setting $y_{EVEN,j} = 0$ if $\alpha_{i^*,j}^{(EVEN)}$ is smaller than $\beta_{i^*,j}^{(EVEN)}$, and 1 otherwise. Note that as $\mathcal{A}$ cannot distinguish between $\beta_{i^*,j}^{(ODD)}$ and $\omega_{i^*,j}^{(ODD)}$, and thus cannot compute y_{ODD}.

Then, using a private authenticated channel, $\mathcal{B}$ sends (r_{EVEN}, r_{ODD}) to $\mathcal{A}$, in that order. $\mathcal{A}$ can now compute $x_0 = x_{EVEN} = y_{EVEN} \oplus r_{EVEN}$. Additionally, $\mathcal{A}$ computes $b = 0$ if $\alpha_{i^*,j}$ is even, and $b = 1$ otherwise.

Role of $\mathcal{C}$. We briefly discuss the role of $\mathcal{C}$, and why $\mathcal{C}$ is necessary for BEC and ROT, but not for key agreement. Essentially, $\mathcal{C}$ serves to create confusion and cause some information to be lost/erased. In BEC, with a certain probability, the message needs to be erased, and that happens precisely when $\mathcal{C}$ influences the protocol ($\mathcal{C}$'s value was selected by $\mathcal{B}$). In ROT, one of the two messages needs to be erased, and it's the message that is affected by $\mathcal{C}$ that ends up being lost. On the other hand, for key agreement, we want to preserve as much information as possible in order to obtain a larger key. Thus removing the participation of $\mathcal{C}$ from the key agreement allows the best protocol performance.

Our Construction. Figure 8 presents our protocol constructing cmROT^ℓ in the rrABB-hybrid. For every message of the form (i, j, u), we call that (i, j) the identifier of the message and u the message's payload. The distributions P, Q, R are defined as follows. Let $\sigma, \ell \in \{1, 2, \dots\}$. Define $S_n^{(EVEN)} := \{x \colon x \in \{0,1\}^n, x_n = 0\}$ – the set containing all n-bit even value and $S_n^{(ODD)} = \{x \colon x \in \{0,1\}^n, x_n = 1\}$ – the set containing all n-bit odd value.

For every $1 \leqslant i \leqslant \sigma$ and $1 \leqslant j \leqslant \ell$, define $P_{i,j}^{(EVEN)}$ is the uniform distribution over the sample space

$$S_{i,j,n}^{(EVEN)} = \{(i, j, \alpha) : \alpha \in S_n^{(EVEN)}\}.$$

Similarly, define $P_{i,j}^{(ODD)}$ as the uniform distribution over the sample space

$$S_{i,j,n}^{(ODD)} = \{(i, j, \alpha) : \alpha \in S_n^{(ODD)}\},$$

Input. $\mathcal{B}$ has input $(x_0, x_1) \in (\{0,1\}^\ell)^2$ and $\mathcal{A}$ has no inputs.
Hybrid. Parties are in $\mathsf{rABB}^{P,Q,R}$-hybrid with appropriate P, Q, R such that

1. $\mathcal{A}$ has a set A containing $\sigma\ell$ messages of the form (i, j, u) with distinct identifiers (i, j) and payloads (u's) are all even or all odd, where $1 \leqslant i \leqslant \sigma$ and $1 \leqslant j \leqslant \ell$.
2. $\mathcal{B}$ has a set B containing $2\sigma\ell$ messages such that, for any $1 \leqslant i \leqslant \sigma$ and $1 \leqslant j \leqslant \ell$, there are two messages with identifier (i, j) such that their payloads have different parity.
3. $\mathcal{C}$ has a set C containing $\sigma\ell$ messages of the form (i, j, u) with distinct identifiers (i, j) and payloads (u's) are all even or all odd.

Every party also receives $\Gamma = A \cup B \cup C$.

One-Round Protocol.

- Both $\mathcal{A}$ and $\mathcal{B}$ identify the smallest $i^* \in \{1, 2, \ldots, \sigma\}$ such that the payloads of any message in A and any message in C have different parity.
- For $1 \leqslant j \leqslant \ell$, from the two sets B and Γ, party $\mathcal{B}$ identifies the four messages with identifier (i^*, j). There are exactly two of them in B and exactly two of them whose payloads are even. Then, he sets $y_{0,j} = 0$ if his even payload is bigger than the other even one and $y_{0,j} = 1$ otherwise. Similarly, he sets $y_{1,j} = 0$ if his odd payload is bigger than the other odd and $y_{1,j} = 1$ otherwise.
- $\mathcal{B}$ sends (r_0, r_1) to $\mathcal{A}$ using the private authenticated channel, where $r_0 = y_0 \oplus x_0, r_1 = y_1 \oplus x_1$.

Output Computation. $\mathcal{A}$ receives (r_0, r_1) from $\mathcal{B}$. For $1 \leqslant j \leqslant \ell$, party $\mathcal{A}$ identifies the two messages with identifier (i^*, j) such that (1) one of them is in A and (2) their payloads have the same parity.

1. Case 1: If the two payloads are even, $\mathcal{A}$ sets $\tilde{y}_{0,j} = 0$ if his payload is smaller than the other and sets $\tilde{y}_{0,j} = 1$ otherwise.
2. Case 2: If the two payloads are odd, $\mathcal{A}$ sets $\tilde{y}_{1,j} = 0$ if his payload is smaller than the other and sets $\tilde{y}_{1,j} = 1$ otherwise.

Then $\mathcal{A}$ outputs $(b = 0, \tilde{y}_0 \oplus r_0)$ in case 1, and $\mathcal{A}$ outputs $(b = 1, \tilde{y}_1 \oplus r_1)$ in case 2. In any case, $\mathcal{B}$ outputs nothing.

Figure 8 Realizing cmROT^ℓ in $\mathsf{rABB}^{P,Q,R}$-hybrid with appropriately chosen independent distributions P, Q, R. The parameter σ is chosen large enough so that the probability for the existence of such i^* is negligible.

and define $Q_{i,j}$ as the uniform distributrion over the sample sapce

$$T_{i,j,n} := S_{i,j,n}^{(EVEN)} \times S_{i,j,n}^{(ODD)}.$$

Now, for every $1 \leqslant i \leqslant \sigma$, define P_i is the uniform distribution over the sample space

$$S_{i,n} := S_{i,1,n}^{(EVEN)} \times S_{i,2,n}^{(EVEN)} \times \ldots \times S_{i,\ell,n}^{(EVEN)} \cup S_{i,1,n}^{(ODD)} \times S_{i,2,n}^{(ODD)} \times \ldots \times S_{i,\ell,n}^{(ODD)}.$$

Define R_i as an i.i.d of P_i. For every $1 \leqslant i \leqslant \sigma$, define Q_i as the joint distribution of independent random variables $X_1, X_2, \ldots, X_\ell$ distributed according to $Q_{i,1}, Q_{i,2}, \ldots, Q_{i,\ell}$, respectively.

Next, define P as the joint distribution $(P_1, P_2, \ldots, P_\ell)$. Similarly, $Q := (Q_1, Q_2, \ldots, Q_\ell)$ and $R := (R_1, R_2, \ldots, R_\ell)$.

7.2 Correctness and Security Proofs

We provide a high-level proof overview of the security and correctness of our protocol.

$\mathcal{A}$ and $\mathcal{B}$ can both distinguish between $\beta_{i^*,j,EVEN}$ and $\alpha_{i^*,j,EVEN}$ (or $\beta_{i^*,j,ODD}$ and $\alpha_{i^*,j,ODD}$) and can thus agree on the same message x_b.

A corrupt $\mathcal{A}$ cannot distinguish between $\beta_{i^*,j,EVEN}$ and $\omega_{i^*,j,EVEN}$ (or $\beta_{i^*,j,ODD}$ and $\omega_{i^*,j,ODD}$), therefore $\mathcal{A}$ does not learn anything about the x_{1-b}

A corrupt $\mathcal{B}$ cannot distinguish between the case where the two values are $\alpha_{i^*,j,EVEN}$ and $\omega_{i^*,j,ODD}$, or the case where the two values are $\alpha_{i^*,j,ODD}$ and $\omega_{i^*,j,EVEN}$. Therefore, $\mathcal{B}$ learns nothing about which message $\mathcal{A}$ received (the bit b).

A corrupt $\mathcal{C}$ does not see r_{EVEN} and r_{ODD}. Therefore, although $\mathcal{C}$ can learn one of y_{EVEN}, y_{ODD}, $\mathcal{C}$ learns nothing about x_{EVEN} or x_{ODD}.

While our protocol does have a small failure probability, all such failures are publicly detectable. Conditioned on the fact that failure does not occur, our protocol is unconditionally secure. Furthermore, given the protocol output, a reverse sampling of the view of the parties is efficient. Therefore, we can trivially construct a simulator that simulates the view of the corrupt party.

7.3 Performance Analysis

We provide a brief discussion on how our protocol compares to the protocol presented in [24]. As stated in [24], once we can obtain key agreement using rABB, we can then implement general honest majority MPC to obtain primitives such as random OT.

Concretely, to get random OT assuming shared keys between all pairs of $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$, at least two rounds of communication are required. In contrast, our protocol achieves chosen random OT in a single round of communication.

Regarding our communication cost, each party receives at most $2 \cdot \sigma \cdot \ell \cdot n \cdot \log(\sigma) \cdot \log(\ell)$ bits from the rABB (excluding Γ) and sends at most $2 \cdot \ell$ bits through the private authenticated channel. Therefore, our communication cost is $O(\sigma \ell \log(\sigma) \log(\ell) n)$.

The failure probability of our protocol is upper bounded by $2^{-\sigma}$ (where all P_i and Q_i are in the same mode) plus $1 - \left(1 - 2^{n-1}\right)^{2\ell}$ (probability of having at least 1 collision in the messages).

▶ Remark 20 (Optimality of Round Complexity). We note that as we can trivially get BEC(0.5) from a chosen message random oblivious transfer, the existence of a non-interactive chosen message random oblivious transfer will imply the existence of a non-interactive BEC protocol, which contradicts Theorem 18. This proves that our chosen message random oblivious transfer is round optimal.

7.4 Generalizations and Extensions

We generalize and extend our protocol in the following two ways.

Non-interactive Random String Oblivious Transfer We extend our protocol to a non-interactive random string oblivious transfer where both the message and the choice are random. In our protocol, $\mathcal{B}$ and $\mathcal{A}$ used our non-interactive key-agreement protocol to agree on a random shared key that is used to establish a private authenticated channel in which $\mathcal{B}$ is able to send r_{EVEN} and r_{ODD} to $\mathcal{A}$ in order to determine the message. Instead of sending r_{EVEN} and r_{ODD} using the shared key, $\mathcal{B}$ and $\mathcal{A}$ can directly derive r_{EVEN} and r_{ODD} from the key in a non-interactive way. Since the shared key is random, the resulting r_{EVEN} and r_{ODD} will also be random. This results in a non-interactive random oblivious transfer protocol.

1-out-of-N Chosen Message Random Oblivious Transfer In our protocol, we partitioned values into even and odd, which results in a 1-out-of-2 cmROT^{ℓ}. Our protocol can be generalized to any partitioning scheme Π_N where values are partitioned into N subsets. Our protocol will be modified to give $\mathcal{A}$ one value belonging to one of the subsets, $\mathcal{B}$ N value where each value belonging to a different subset (one value each from each subset), and $\mathcal{C}$ $N-1$ values that belong to all subsets except the one $\mathcal{A}$ received. The rest of the protocol is modified accordingly. This allows us to achieve one round 1-out-of-N cmROT^{ℓ}. However, note that the failure probability will now be $N^{-\sigma}$. In order to keep the failure probability low, σ needs to be increased accordingly, which increases the communication complexity of our protocol.

References

1 Ittai Abraham, Benny Pinkas, and Avishay Yanai. Blinder - scalable, robust anonymous committed broadcast. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, USA, November 9-13, 2020*, pages 1233–1252. ACM, 2020. doi:10.1145/3372297.3417261.

2 Pratyush Agarwal, Varun Narayanan, Shreya Pathak, Manoj Prabhakaran, Vinod M. Prabhakaran, and Mohammad Ali Rehan. Secure non-interactive reduction and spectral analysis of correlations. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part III*, volume 13277 of *LNCS*, pages 797–827. Springer, Heidelberg, May / June 2022. doi:10.1007/978-3-031-07082-2_28.

3 Shweta Agrawal, Yuval Ishai, Eyal Kushilevitz, Varun Narayanan, Manoj Prabhakaran, Vinod M. Prabhakaran, and Alon Rosen. Cryptography from one-way communication: On completeness of finite channels. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part III*, volume 12493 of *LNCS*, pages 653–685. Springer, Heidelberg, December 2020. doi:10.1007/978-3-030-64840-4_22.

4 Rudolf Ahlswede and Imre Csiszár. Common randomness in information theory and cryptography - I: secret sharing. *IEEE Trans. Inf. Theory*, 39(4):1121–1132, 1993. doi:10.1109/18.243431.

5 Bowen Alpern and Fred B. Schneider. Key exchange using 'keyless cryptography'. *Information Processing Letters*, 16(2):79–81, 1983. URL: https://www.sciencedirect.com/science/article/pii/0020019083900297, doi:https://doi.org/10.1016/0020-0190(83)90029-7.

6 Megumi Ando, Anna Lysyanskaya, and Eli Upfal. On the complexity of anonymous communication through public networks. In Stefano Tessaro, editor, *2nd Conference on Information-Theoretic Cryptography, ITC 2021, July 23-26, 2021, Virtual Conference*, volume 199 of *LIPIcs*, pages 9:1–9:25. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ITC.2021.9.

7 Daniel Beer, Nov 2015. URL: https://dlbeer.co.nz/articles/kwbs.html.

8 Andrej Bogdanov and Elchanan Mossel. On extracting common random bits from correlated sources. *IEEE Trans. Inf. Theory*, 57(10):6351–6355, 2011. doi:10.1109/TIT.2011.2134067.

9 Claude Castelluccia and Pars Mutaf. Shake them up! a movement-based pairing protocol for CPU-constrained devices. In *Third International Conference on Mobile Systems, Applications, and Services (MobiSys2005)*, Seattle, WA, June 2005. USENIX Association. URL: https://www.usenix.org/conference/mobisys2005/shake-them-movement-based-pairing-protocol-cpu-constrained-devices.

10 Siu On Chan, Elchanan Mossel, and Joe Neeman. On extracting common random bits from correlated sources on large alphabets. *IEEE Trans. Inf. Theory*, 60(3):1630–1637, 2014. doi:10.1109/TIT.2014.2301155.

11 Claude Crépeau. Equivalence between two flavours of oblivious transfers. In Carl Pomerance, editor, *CRYPTO'87*, volume 293 of *LNCS*, pages 350–354. Springer, Heidelberg, August 1988. doi:10.1007/3-540-48184-2_30.

12 Imre Csiszár and Prakash Narayan. Secrecy capacities for multiple terminals. *IEEE Trans. Inf. Theory*, 50(12):3047–3061, 2004. doi:10.1109/TIT.2004.838380.

13 Debajyoti Das, Sebastian Meiser, Esfandiar Mohammadi, and Aniket Kate. Anonymity trilemma: Strong anonymity, low bandwidth overhead, low latency - choose two. In *2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA*, pages 108–126. IEEE Computer Society, 2018. doi:10.1109/SP.2018.00011.

14 Roberto Di Pietro and Gabriele Oligeri. Coke crypto-less over-the-air key establishment. *IEEE Transactions on Information Forensics and Security*, 8(1):163–173, 2013. doi:10.1109/TIFS.2012.2226718.

15 Claudia Diaz, Harry Halpin, and Aggelos Kiayias. The nym network the next generation of privacy infrastructure. Technical report, Nym Technologies SA, 2021 [Online]. URL: `https://nymtech.net/nym-whitepaper.pdf`.

16 Roger Dingledine, Nick Mathewson, and Paul F. Syverson. Tor: The second-generation onion router. In Matt Blaze, editor, *USENIX Security 2004*, pages 303–320. USENIX Association, August 2004.

17 Saba Eskandarian, Henry Corrigan-Gibbs, Matei Zaharia, and Dan Boneh. Express: Lowering the cost of metadata-hiding communication with cryptographic privacy. In Michael Bailey and Rachel Greenstadt, editors, *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, pages 1775–1792. USENIX Association, 2021. URL: `https://www.usenix.org/conference/usenixsecurity21/presentation/eskandarian`.

18 Uriel Feige, Joe Kilian, and Moni Naor. A minimal model for secure computation (extended abstract). In *26th ACM STOC*, pages 554–563. ACM Press, May 1994. `doi:10.1145/195058.195408`.

19 Michael J. Fischer and Rebecca N. Wright. Multiparty secret key exchange using a random deal of cards. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 141–155. Springer, Heidelberg, August 1992. `doi:10.1007/3-540-46766-1_10`.

20 Sanjam Garg, Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography with one-way communication. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 191–208. Springer, Heidelberg, August 2015. `doi:10.1007/978-3-662-48000-7_10`.

21 Yossi Gilad and Amir Herzberg. Plug-and-play IP security - anonymity infrastructure instead of PKI. In Jason Crampton, Sushil Jajodia, and Keith Mayes, editors, *Computer Security - ESORICS 2013 - 18th European Symposium on Research in Computer Security, Egham, UK, September 9-13, 2013. Proceedings*, volume 8134 of *Lecture Notes in Computer Science*, pages 255–272. Springer, 2013. `doi:10.1007/978-3-642-40203-6\_15`.

22 Amin Aminzadeh Gohari and Venkat Anantharam. Information-theoretic key agreement of multiple terminals: part I. *IEEE Trans. Inf. Theory*, 56(8):3973–3996, 2010. `doi:10.1109/TIT.2010.2050832`.

23 Ryan Henry, Amir Herzberg, and Aniket Kate. Blockchain access privacy: Challenges and directions. *IEEE Secur. Priv.*, 16(4):38–45, 2018. `doi:10.1109/MSP.2018.3111245`.

24 Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography from anonymity. In *47th FOCS*, pages 239–248. IEEE Computer Society Press, October 2006. `doi:10.1109/FOCS.2006.25`.

25 Sune K. Jakobsen and Claudio Orlandi. How to bootstrap anonymous communication. In Madhu Sudan, editor, *ITCS 2016*, pages 333–344. ACM, January 2016. `doi:10.1145/2840728.2840743`.

26 Hamidreza Amini Khorasgani, Hemanta K. Maji, and Hai H. Nguyen. Secure non-interactive simulation: Feasibility and rate. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part III*, volume 13277 of *LNCS*, pages 767–796. Springer, Heidelberg, May / June 2022. `doi:10.1007/978-3-031-07082-2_27`.

27 Hamidreza Amini Khorasgani, Hemanta K. Maji, and Hai H. Nguyen. Secure non-interactive simulation from arbitrary joint distributions. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part II*, volume 13748 of *LNCS*, pages 378–407. Springer, Heidelberg, November 2022. `doi:10.1007/978-3-031-22365-5_14`.

28 Albert Kwon, Henry Corrigan-Gibbs, Srinivas Devadas, and Bryan Ford. Atom: Horizontally scaling strong anonymity. In *Proceedings of the 26th Symposium on Operating Systems Principles, Shanghai, China, October 28-31, 2017*, pages 406–422. ACM, 2017. `doi:10.1145/3132747.3132755`.

29 Cheuk Ting Li and Venkat Anantharam. One-shot variable-length secret key agreement approaching mutual information. In *56th Annual Allerton Conference on Communication,*

Control, and Computing, Allerton 2018, Monticello, IL, USA, October 2-5, 2018, pages 259–266. IEEE, 2018. `doi:10.1109/ALLERTON.2018.8635830`.

30 Cheuk Ting Li and Venkat Anantharam. One-shot variable-length secret key agreement approaching mutual information. *IEEE Trans. Inf. Theory*, 67(8):5509–5525, 2021. `doi:10.1109/TIT.2021.3087963`.

31 Ueli M. Maurer. Protocols for secret key agreement by public discussion based on common information. In Ernest F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 461–470. Springer, Heidelberg, August 1993. `doi:10.1007/3-540-48071-4_32`.

32 Ueli M. Maurer and Stefan Wolf. Unconditionally secure key agreement and the intrinsic conditional information. *IEEE Trans. Inf. Theory*, 45(2):499–514, 1999. `doi:10.1109/18.748999`.

33 Ueli M. Maurer and Stefan Wolf. Information-theoretic key agreement: From weak to strong secrecy for free. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 351–368. Springer, Heidelberg, May 2000. `doi:10.1007/3-540-45539-6_24`.

34 Elchanan Mossel and Ryan O'Donnell. Coin flipping from a cosmic source: On error correction of truly random bits. *Random Structures & Algorithms*, 26(4):418–436, 2005. `doi:10.1002/rsa.20062`.

35 Elchanan Mossel, Ryan O'Donnell, Oded Regev, Jeffrey E Steif, and Benny Sudakov. Non-interactive correlation distillation, inhomogeneous markov chains, and the reverse bonami-beckner inequality. *Israel Journal of Mathematics*, 154(1):299–336, 2006.

36 Andreas Pfitzmann and Michael Waidner. Networks without user observability — design options. In Franz Pichler, editor, *Advances in Cryptology — EUROCRYPT' 85*, pages 245–253, Berlin, Heidelberg, 1986. Springer Berlin Heidelberg.

37 Michael O. Rabin. How to exchange secrets by oblivious transfer. *Technical Memo TR-81*, 1981.

38 Michael O. Rabin. How to exchange secrets with oblivious transfer. Cryptology ePrint Archive, Report 2005/187, 2005. `https://eprint.iacr.org/2005/187`.

39 xx Foundation. xx network white paper. Technical report, xx Foundation, 2021 [Online]. URL: `https://xx.network/wp-content/uploads/2021/10/xx-whitepaper-v2.0.pdf`.

40 xx Foundation. xx network white paper xx cmix. Technical report, xx Foundation, 2021 [Online]. URL: `https://xx.network/wp-content/uploads/2021/10/xx-whitepaper-v2.0.pdf`.

41 Ke Yang. On the (im)possibility of non-interactive correlation distillation. In Martin Farach-Colton, editor, *LATIN 2004*, volume 2976 of *LNCS*, pages 222–231. Springer, Heidelberg, April 2004.

42 Mordechai M. Yung. A secure and useful "keyless cryptosystem". *Information Processing Letters*, 21(1):35–38, 1985. URL: `https://www.sciencedirect.com/science/article/pii/0020019085901061`, `doi:https://doi.org/10.1016/0020-0190(85)90106-1`.

A On the Optimality of Our Constructions

▶ **Theorem 21.** *Suppose $p = e/d$ for $e, d \in \{1, 2, \dots\}$ and $e < d$. Any zero round protocol implementing $\mathsf{BEC}(p)$ in $\mathsf{rABB}^{U_A,U_B,U_C}_{d-e,1,e}$-hybrid has constant insecurity, where U_A, U_B, U_C are uniform distribution over $(\{0,1\}^n)^{d-e}, \{0,1\}^n, (\{0,1\}^n)^e$ respectively, and n is the message length.*

Proof. We prove this by contradiction. Suppose there is a non-interactive secure protocol for $\mathsf{BEC}(p)$ in rABB-hybrid. We introduce some terminologies and notations. Let $m = m_A + m_B + m_C$. Let A, B, C be random variables sampled according to the distributions U_A, U_B, U_C, respectively. Recall that $A \cup B \cup C$ contains no duplicate with high probability over the random choices of A, B, C. Let Γ be a set containing m distinct elements in $\{0,1\}^n$. Consider the conditional distribution $(A, B|A \cup B \cup C = \Gamma)$. After removing all zero rows and columns, the probability mass function of $(A, B|A \cup B \cup C = \Gamma)$ is a matrix of size $\binom{m}{m_A} \times \binom{m}{m_B}$. From now, we refer to $(A, B|A \cup B \cup C = \Gamma)$ as the distribution after removing all these zero rows and columns. Observe that any protocol realizing $\mathsf{BEC}(p)$ from $(A, B|A \cup B \cup C = \Gamma)$ has constant insecurity if there is no perfectly secure protocol realizing $\mathsf{BEC}(p)$ from that distribution. Now, observe that for any two Γ and Γ' each containing m distinct elements, the probability mass functions $(A, B|A \cup B \cup C = \Gamma)$ and $(A, B|A \cup B \cup C = \Gamma')$ are the same (up to permutations of rows and columns). Therefore, there must exist a perfect secure protocol for $\mathsf{BEC}(p)$ from $(A, B|A \cup B \cup C = \Gamma)$.

Now, we employ the technique developed recently in secure non-interactive simulation/reduction [26]. Let T and $\bar{T}$ be the Markov and the adjoint Markov operator associated with the conditional distribution $(A, B|A \cup B \cup C = \Gamma)$ (refer to [26] for definitions).

There is a perfectly secure protocol if and only if there are functions f, g such that

$$Tg = f, \text{ and } \bar{T}f = (1-p)g.$$

Combining two equations together yields $T\bar{T}f = (1-p)f$. This implies that $(1-p)$ is an eigenvalue of the operator $T\bar{T}$ with associated eigenvector f. We shall show that these two conditions yield a contradiction. Observe that any column of $T\bar{T}$ is a permutation of any other column of $T\bar{T}$. Following the approach in [26, 27], the function f must have only two output values 1 or -1. These two facts together give the contradiction. ◀