**Commitment and Oblivious Transfer in the Bounded Storage Model with Errors**

Rafael Dowsley, Felipe Lacerda and Anderson C. A. Nascimento

**Abstract**—In the bounded storage model the memory of the adversary is restricted, instead of its computational power. With this different restriction it is possible to design protocols with information-theoretical (instead of only computational) security. We present the first protocols for commitment and oblivious transfer in the bounded storage model with errors, i.e., the model where the public random sources available to the two parties are not exactly the same, but instead are only required to have a small Hamming distance between themselves. Commitment and oblivious transfer protocols were known previously only for the error-free variant of the bounded storage model, which is harder to realize.

**Index Terms**—Bounded storage model, error correction, commitment, oblivious transfer, unconditional security.

**I. INTRODUCTION**

Commitment schemes are fundamental building blocks of modern cryptography. They are important in the construction of protocols such as identification protocols [2], contract signing [3], zero-knowledge proofs [4], coin flipping over the phone [5], and more generally in two- and multi-party computation protocols [6], [7]. A commitment scheme is a two-stage protocol between two parties, Alice and Bob. First they execute the commit stage, in which Alice chooses a value \( v \) as input and commits to it. Later, they execute the open stage, in which Alice reveals \( v \) to Bob. For the protocol to be secure, it must satisfy two conditions: the hiding property, which means that Bob cannot learn any information about \( v \) before the open stage, and the binding property, which means that after the commit phase, Alice cannot change \( v \) without that being detected by Bob.

Oblivious transfer (OT) is another essential primitive for two- and multi-party computation. It is a two-party protocol in which Alice inputs two strings \( s_0 \) and \( s_1 \), and Bob inputs a bit \( c \). Bob’s output is the string \( s_c \). The protocol is called secure if Alice never learns the choice bit \( c \) and Bob does not learn any information about \( s_{1-c} \). Oblivious transfer is a fundamental building block for multi-party computation and can be used to realize any secure two-party computation [8], [9]. Unconditionally secure commitment and oblivious transfer are impossible in the setting where the parties only communicate through noiseless channels (even if quantum channels are available [10]). However, both of them are possible in the context of computational security (in which the adversaries are restricted to be polynomial-time Turing machines), as long as computational hardness assumptions are made. Commitment can be obtained using generic assumptions such as the existence of pseudorandom generator [11] or (more efficiently) assuming the hardness of various specific computational problems [12], [5], [13]. Oblivious transfer can be obtained from dense trapdoor permutations [14] (which is conjectured to be stronger than pseudorandom generators) or assuming the hardness of many specific computational problems [15], [16], [17], [18], [19], [20], [21].

Physical assumptions, such as the existence of noisy channels, enable one to obtain unconditional security for commitment and oblivious transfer protocols. In this scenario the problem was studied from both the theoretical [22], [23], [24], [25], [26], [27], [28] and the efficient protocol designing [29], [30], [31], [32] points of view.

In this paper, we consider a different setting, the so called bounded storage model (BSM) [33], in which the adversary is assumed to have bounded memory.

**A. The Bounded Storage Model**

The bounded storage model assumes that both parties have access to a public random string, and that a dishonest party cannot store the whole string. This string can be obtained from a natural source, from a trusted third party, or, in some cases even generated by one of the parties.

A variety of cryptographic tasks can be implemented in the bounded storage model. Cachin and Maurer [34] proposed a key agreement protocol in the bounded storage model in which the parties have a small pre-shared key, and use it to select bits from a public random source of size \( n \). Key agreement in this setting is always possible if the pre-shared key has size proportional to \( \log n \), as long as the adversary has bounded memory. They also proposed a protocol for key agreement by public discussion (that is, without a pre-shared key) that requires \( \sqrt{k}n \) (where \( k \) is the key length) samples from the random source and is thus less practical. Later, Dziembowski and Maurer [35] showed that this protocol is optimal, in the sense that one cannot have key agreement by public discussion in the bounded storage model with less than \( O(\sqrt{n}) \) samples.

The first oblivious transfer protocol in the bounded storage model was introduced by Cachin et al. [36]. Ding [37] and Hong et al. [38] proposed improvements in a slightly different model. Ding et al. [39] obtained the first constant-round protocol.
Recently, Shikata and Yamanaka [40] and independently Alves [41] studied the problem of commitment in the bounded storage model and provided protocols for it.

Unfortunately the bounded storage model assumes that there exists a random source that can be reliably broadcasted to all parties, without errors in the transmission, and this is hard to realize in practice.

Consider a scenario where a satellite broadcasts a very large random string to be used in protocols in the bounded storage model. In his Ph.D. thesis, Ding [42] made an analysis of the practicality of this scenario, showing that an antenna with a surface area of $10m^2$ can be used to receive random bits from a geo-stationary satellite at rates up to 50 Gbps. Ding’s analysis did not consider the fact that errors are introduced in the string and that an adversary might be able to jam signal received by a legitimate party.

Our goal with this work is to study two-party protocols under more realistic assumptions.

B. Our contribution

In this work, we consider a more general variant of the BSM, in which errors are introduced in the public random source in arbitrary positions. This setting captures the situation in which the source is partially controlled by an adversary, and also the situation in which there are errors due to noise in the channel. It is only assumed that the fraction of errors, relative to the length of the public string, is not too large. Ding previously studied this model [43] in the context of secret key extension protocols (protocols that extend a pre-shared secret key). These protocols can be modified, at the cost of an efficiency loss, to handle the case of key agreement, when no pre-shared key exists. He defined a general paradigm for BSM randomness extraction schemes and also showed how to incorporate error correction in key agreement extension by using fuzzy extractors [44].

We give a brief introduction to the model and its notation in order to state our results. A transmission phase is executed prior to the realization of the protocols’ main part. In this phase, Alice has access to a sample $x \in \{0, 1\}^n$ from an $\alpha n$-source $X$ (a source with min-entropy at least $\alpha n$), where $0 < \alpha < 1$, and the receiver (Bob) to $x \in \{0, 1\}^n$ such that $\text{HD}(x, \tilde{x}) \leq \delta n$, where $\text{HD}()$ represents the hamming distance. We assume that an adversary has complete control on where to insert the differences between the strings $x$ and $\tilde{x}$, thus capturing both the situation where the source is noisy and the situation where an adversary controls part of the source.

We propose the first protocols for bit commitment and oblivious transfer in the BSM with errors, thus extending the results of [43] to the case of two-party secure protocols. We show that the techniques introduced by [43] originally in the context of key extension give us efficient protocols for implementing oblivious transfer. Our protocol assumes a memory bounded Bob (i.e., he is able to store at most $\gamma n$ bits for $\gamma < \alpha$), but no limitation is put on Alice’s memory. It works based on an efficient linear error correcting code proposed in [45] with rate $\beta$ and achieving the Zyablov bound. We show that as long as $\beta > 1 - \alpha - \gamma$ the protocol works for noise levels $\delta$ as severe as (approximately)

$$\max_{\beta < \beta < 1} \frac{(1 - \beta)\gamma}{2},$$

where $\gamma$ is the unique value in $[0, 1/2]$ so that $h(\gamma) = 1 - \beta/\beta$ and $h(\cdot)$ is the binary entropy function. In case a random linear error correcting code is used an improved noise level can be tolerated

$$h(2\delta) < \alpha - \gamma.$$

This improvement in the resilience comes at the price of making the protocol inefficient from a computational complexity point of view, given the intractability of decoding random linear codes.

The proposed oblivious transfer protocol immediately gives us a commitment scheme. However, using oblivious transfer for obtaining commitment is not a desirable solution. The communication, round and computational complexities of oblivious transfer protocols are usually much higher than the ones for commitment schemes. Moreover, it could be the case that commitment protocols work for different ranges of noise $\delta$.

We propose a direct construction of a non-interactive commitment protocol that does not rely on the framework proposed by Ding [43], does not use error correcting codes at all, implements string commitment and has only one message from Bob to Alice. Again, we assume that Bob has limited memory. No limitations are imposed on Alice whatsoever. The protocol is very efficient and simple and works for

$$h(\delta) < \frac{\alpha - \gamma}{2}.$$

We then show that it is possible to obtain a protocol that works for a much larger range of noise

$$h(\delta) < \alpha - \gamma$$

at the cost of having one additional message in each direction and by using a family of $4k$-universal hash functions. Finally, we show that the use of families of $4k$-universal hash functions can be avoided by imposing a memory bound on Alice, instead of Bob. We note that being able to implement protocols in the memory bounded by bounding any of the parties is an important matter, particularly when one of the parties is much more powerful than the other. This protocol is based on the interactive hashing protocol of [39] and also works for

$$h(\delta) < \alpha - \gamma,$$

but has extra rounds of communication and implements bit rather than string commitment.

The techniques we use in our results are standard in the field: extractors, error-correcting codes, typicality tests, sampling, etc. However, to the best of our knowledge, this is the first time that these techniques are combined to obtain commitment and oblivious transfer protocols in the memory bounded model with errors. Moreover, the study of how much adversarial noise can be tolerated in this model and its relation to round complexity is also original, as far as we know.

Interestingly, the noise levels tolerated by our protocols are
different for oblivious transfer and commitment schemes. This contrasts sharply with the noiseless situation where either one has every possible secure two-party computation or nothing at all.

C. Overview

In Section II, we present the main tools used in our protocols. Section III explains the security model. Our commitment protocols are introduced in Sections IV, V and VI, and the oblivious transfer in Section VII. A conference version of this work appeared at the proceedings of ISIT 2014 [1] and only covered the case of oblivious transfer. In this full version a more detailed presentation of the case of oblivious transfer is presented and the case of commitment is entirely new; the other sections are also extended accordingly.

II. Preliminaries

We use calligraphic letters for denoting domains of random variables and other sets, upper case letters for random variables and other sets, upper case letters for random variables and lower case letters for realizations of the random variables. We deal solely with discrete random variables. The probability mass function of a random variable will be denoted by \( P_X \). The set \( \{1, \ldots, n\} \) will be written as \([n]\). If \( x = (x_1, \ldots, x_n) \) is a sequence and \( S = \{s_1, \ldots, s_t\} \subseteq [n] \), \( x^S \) denotes the sequence \( (x_{s_1}, \ldots, x_{s_t}) \). \( U \) denotes that \( u \) is drawn from the uniform distribution over the set \( U \), \( u \leftarrow U \) is the uniformly-distributed \( r \)-bit random variable. \( y \leftarrow F(x) \) denotes the act of running the probabilistic algorithm \( F \) with input \( x \) and obtaining the output \( y \). \( y \leftarrow F(x) \) is similarly used for deterministic algorithms.

If \( x \) and \( y \) are strings, \( \text{HD}(x, y) \) denotes their Hamming distance (that is, the number of positions in which they differ) and \( x \oplus y \) their bitwise exclusive or. Let \( \log x \) denote the logarithm of \( x \) in base 2. The binary entropy function is denoted by \( h : \{0,1\} \to \mathbb{R} \) defined by \( h(x) = -x \log x - (1-x) \log(1-x) \).

By convention, \( 0 \log 0 = 0 \). \( H(X) \) denotes the entropy of \( X \) and \( I(X;Y) \) the mutual information between \( X \) and \( Y \).

The statistical distance is a measure of the distance between two probability distributions. Here, we state its definition for the case of discrete probabilities.

**Definition 2.1 (Statistical distance):** The statistical distance \( \|P_X - P_Y\| \) between two probability mass functions \( P_X, P_Y \) over an alphabet \( \mathcal{X} \) is defined as

\[
\|P_X - P_Y\| = \max_{A \subseteq \mathcal{X}} \left| \sum_{x \in A} P_X(x) - P_Y(x) \right|.
\]

We say \( P_X \) and \( P_Y \) are \( \varepsilon \)-close if \( \|P_X - P_Y\| \leq \varepsilon \).

A. Entropy Measures

The main entropy measure in this work is the min-entropy, which captures the notion of unpredictability of a random variable.

**Definition 2.2 (Min-entropy):** Let \( P_{XY} \) be a probability mass function over \( \mathcal{X} \times \mathcal{Y} \). The min-entropy of \( X \), denoted by \( H_\infty(X) \), and the conditional min-entropy of \( X \) given \( Y \), denoted by \( H_\infty(X|Y) \), are respectively defined as

\[
H_\infty(X) = \min_{x \in \mathcal{X}} (\log P_X(x))
\]

\[
H_\infty(X|Y) = \min_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} (\log P_{X|Y=y}(x))
\]

\( X \) is called a \( k \)-source if \( H_\infty(X) \geq k \).

The conditional min-entropy \( H_\infty(X|Y) \) measures the extractable private randomness from the variable \( X \), given the correlated random variable \( Y \) possessed by an adversary. The min-entropy has the problem of being sensitive to small changes in the probability mass function. Due to this fact the notion of smooth min-entropy [46] will be used.

**Definition 2.3 (Smooth min-entropy):** Let \( \varepsilon > 0 \) and \( P_{XY} \) be a probability mass functions. The \( \varepsilon \)-smooth min-entropy of \( X \) given \( Y \) is defined by

\[
H_\infty^\varepsilon(X|Y) = \max_{x \in \mathcal{X}, y \in \mathcal{Y}} \left\{ P_{X|Y=y}(x) \mid H_\infty(X|Y=y) \leq \varepsilon \right\}
\]

Intuitively, the smooth min-entropy is the maximum min-entropy in the neighborhood of the probability mass function. Similarly, we also define the max-entropy and its smooth version.

**Definition 2.4 (Smooth Max-entropy):** The max-entropy is defined as

\[
H_0(X) = \log |\{ x \in \mathcal{X} | P_X(x) > 0 \}|
\]

and its conditional version is given by

\[
H_0(X|Y) = \max_y \log |\{ x \in \mathcal{X} | P_{X|Y=y}(x) > 0 \}|.
\]

The smooth variants are defined as

\[
H_0^\varepsilon(X) = \max_{x \in \mathcal{X}} \min_{y} \left\{ P_{X|Y=y}(x) \mid H_0(X|Y=y) \leq \varepsilon \right\}
\]

\[
H_0^\varepsilon(X|Y) = \max_{x \in \mathcal{X}} \min_{y \in \mathcal{Y}} \left\{ P_{X|Y=y}(x) \mid H_0(X|Y=y) \leq \varepsilon \right\}
\]

The following inequalities are smooth min-entropy analogues of the chain rule for conditional Shannon entropy [46].

**Lemma 2.5:** Let \( \varepsilon, \varepsilon', \varepsilon'' > 0 \) and \( P_{XYZ} \) be a tripartite probability mass function. Then

\[
H_\infty^{\varepsilon+\varepsilon'}(X,Y|Z) \geq H_\infty^{\varepsilon}(X|Y,Z) + H_\infty^{\varepsilon'}(Y|Z)
\]

\[
H_\infty^{\varepsilon+\varepsilon''}(X,Y,Z) \geq H_\infty^{\varepsilon+\varepsilon''}(X,Y|Z) + H_0^{\varepsilon''}(Y|Z) + \log(1/\varepsilon').
\]

The notion of min-entropy rate and a few results regarding its preservation will be used in the subsequent parts of this work.

**Definition 2.6 (Min-entropy rate):** Let \( X \) be a random variable with an alphabet \( \mathcal{X} \), \( Y \) be an arbitrary random variable, and \( \varepsilon \geq 0 \). The min-entropy rate \( R_\infty(X|Y) \) is defined as

\[
R_\infty(X|Y) = \frac{H_\infty(X|Y)}{\log |\mathcal{X}|}.
\]

The following lemma is a restatement of a lemma in [39] and says that a source with high min-entropy also has high min-entropy when conditioned on a correlated short string. This lemma is what makes the bounded storage assumption limited
information about a string sampled from the public random string.

Lemma 2.7: Let $X \in \{0, 1\}^n$ such that $R_{\infty}(X) \geq \rho$ and $Y$ be a random variable over $\{0, 1\}^m$. Fix $\epsilon > 0$. Then

$$R_{\infty}^{\epsilon'}(X|Y) \geq \rho - \phi - \frac{1 + \log(1/\epsilon')}{n}.$$ 

Proof: Let $\psi = \rho - \phi - \frac{1 + \log(1/\epsilon')}{n}$. By lemma 3.16 in [39] we have that if $R_{\infty}(X) \geq \rho$ then

$$\Pr_y [R_{\infty}^{\epsilon'}(X|Y = y) \geq \psi] \geq 1 - \epsilon' - \sqrt{2\epsilon}.$$ 

To get the desired result, let $G = \{y \in \mathcal{Y} | R_{\infty}^{\epsilon'}(X|Y = y) \geq \psi\}$ and $P_{XY}$ be the joint probability distribution of $X$ and $Y$. Let $P'_{XY}$ be the distribution that is $\sqrt{2\epsilon}$-close to $P_{XY}$ and such that $P'(X = x|Y = y) \leq 2^{-m}$ for any $x \in X, y \in G$. Let $P''_{XY}$ be obtained by letting $P''(X|Y = y) = P'(X|Y = y)$ for $y \in G$ and defining $P''(X|Y = y) = 2^{-n}$ for any $x \in X, y \notin G$. As $P(G) \geq 1 - \epsilon' - \sqrt{2\epsilon}$, it holds that $\|P'_{XY} - P''_{XY}\| \leq \epsilon' + \sqrt{2\epsilon}$ and so $\|P''_{XY} - P_{XY}\| \leq \epsilon' + 2\sqrt{2\epsilon}$. Since $P''(X = x|Y = y) \leq 2^{-m}$ for every $x \in X, y \in Y$, the lemma follows.

B. Averaging Samplers and Randomness Extractors

The sample-then-extract paradigm is usually employed in the bounded storage model - first some positions of the source are sampled and then an extractor is applied on these positions. Note that due to the assumption that it is inexpensive to store the whole source string (the memory bound), it is not possible to apply an extractor to the complete string, the extractor needs to be locally computable [47]. In this context, averaging samplers [48], [49], [50] are a fundamental tool. Intuitively, averaging samplers produce samples such that the average value of any function applied to the sampled string is roughly the same as the average when taken over the original string.

Definition 2.8 (Averaging sampler): A function $\text{Samp}: \{0, 1\}^t \rightarrow [n]^t$ is an $(\mu, \nu, \epsilon)$-averaging sampler if for every function $f: [n] \rightarrow [0, 1]$ with average $\sum_{i \in S} f(i)/n \geq \mu$ it holds that

$$\Pr_{S \xleftarrow{\text{Samp}(t)}} \left[ \frac{1}{t} \sum_{i \in S} f(i) \leq \mu - \nu \right] \leq \epsilon.$$

Averaging samplers enjoy several useful properties. Particularly important to this work is the fact that averaging samplers roughly preserve the min-entropy rate.

Lemma 2.9 ([47]): Let $X \in \{0, 1\}^n$ be such that $R_{\infty}(X|Y) \geq \rho$. Let $\tau$ be such that $1 \geq \rho \geq 3\tau > 0$ and $\text{Samp}: \{0, 1\}^t \rightarrow [n]^t$ be an $(\mu, \nu, \epsilon)$-averaging sampler with distinct samples for $\mu = (\rho - 2\tau)/\log(1/\tau)$ and $\nu = \tau/\log(1/\tau)$. Then for $S \xleftarrow{\text{Samp}(t)}$

$$R_{\infty}(X^S|S, Y) \geq \rho - 3\tau$$

where $\epsilon' = \epsilon + 2^{\Omega(\tau)}n$.

For $t < n$, the uniform distribution over subsets of $[n]$ of size $t$ is an averaging sampler, also called the $(n, t)$-random subset sampler.

Lemma 2.10: Let $0 < t < n$. For any $\mu, \nu > 0$, the $(n, t)$-random subset sampler is a $(\mu, \nu, e^{-t\nu^2/2})$-averaging sampler.

Proof: It is just a restatement of Lemma 5.5 in [51].

A randomness extractor is a function that takes a string with high min-entropy as an input and outputs a string that is close (in the statistical distance sense) to a uniformly distributed string.

Definition 2.11 (Strong extractor): A function $\text{Ext}: \{0, 1\}^n \times \{0, 1\}^r \rightarrow \{0, 1\}^m$ is a $(k, \varepsilon)$-strong extractor if for every $k$-source $X$, we have

$$\|P_{\text{Ext}(X, U_r), U_r} - P_{\text{Ext}, U_r}\| \leq \varepsilon.$$ 

The following lemma specifies the parameters of an explicit strong extractor construction [50].

Lemma 2.12 ([50]): Let $\rho, \psi > 0$ be arbitrary constants. For every $n \in \mathbb{N}$ and every $\varepsilon > e^{-n/2^{O(\log n)}}$, there is an explicit construction of a $(\rho n, \varepsilon)$-strong extractor $\text{Ext}: \{0, 1\}^n \times \{0, 1\}^r \rightarrow \{0, 1\}^m$ with $m = (1 - \psi)n$ and $r = O(\log n + \log(1/\varepsilon))$.

The oblivious transfer protocol presented in this work uses a variant of a strong extractor, called a fuzzy extractor [44]. Intuitively, fuzzy extractors are noise-resilient extractors, that is, extractors such that the extracted string can be reproduced by any party with a string that is close (in the Hamming distance sense) to the original source.

Definition 2.13 (Fuzzy extractor): A pair of functions $\text{Ext}: \{0, 1\}^n \times \{0, 1\}^r \rightarrow \{0, 1\}^m \times \{0, 1\}^q$, $\text{Rec}: \{0, 1\}^n \times \{0, 1\}^r \times \{0, 1\}^q \rightarrow \{0, 1\}^m$ is an $(k, \varepsilon, \delta, \beta)$-fuzzy extractor if:

- For every $k$-source $X \in \{0, 1\}^t$, $(Q, Y) \leftarrow \text{Ext}(X, U_r)$.
- Then $\|P_{Q, Y|Q, U_r} - P_{Y|Q, U_r}\| \leq \varepsilon$.
- For every $x, x' \in \{0, 1\}^t$ such that $\text{HD}(x, x') \leq \delta\ell$, let $r \xleftarrow{\text{Ext}, (x, r)} \leftarrow \text{Rec}(x, r)$. Then it should hold that $\Pr_{r \xleftarrow{\text{Ext}, (x, r)}} [\text{Rec}(x', r) = y] \geq 1 - \beta$.

Fuzzy extractors are a special case of one-way key-agreement schemes [52], [53]. Ultimately they are equivalent to performing information reconciliation followed by privacy amplification [54]. Since there is a restriction to close strings with respect to the Hamming distance, syndrome-based fuzzy extractors can be used, as summarized in the following lemma from Ding [43].

Lemma 2.14 ([43]): Let $1 \geq \rho, \psi > 0$ and $1/4 > \delta > 0$ be arbitrary constants. There is a constant $\beta$, depending on $\delta$, such that for every sufficiently large $n \in \mathbb{N}$, and every $\varepsilon > e^{-n/2^{O(\log n)}}$, there exists an explicit construction of a $(\rho n, \varepsilon, \delta, 0)$-fuzzy extractor $\text{Ext}, \text{Rec}$, where Ext is of the form $\text{Ext}: \{0, 1\}^n \times \{0, 1\}^r \rightarrow \{0, 1\}^m \times \{0, 1\}^p$ with

$$m = (1 - \psi)n, \quad r = O\left(\log n + \log \frac{1}{\varepsilon}\right), \quad p \leq \frac{1 - \beta}{(1 - \psi)^2}n.$$ 

Remark 2.15: The parameters $\beta, \delta$ refer to the error-correcting code used in the construction, specifically, a code of size $n$ with rate $\beta$ that can correct $\delta n$ errors. It is known [55] that, for a given $\nu$ with $0 < \nu < 1/2$ and $0 \leq \mu \leq 1 - h(\nu)$,
there exists a random linear code with minimum distance $\nu n$ and $\beta \geq 1 - h(\nu) - \mu$ (i.e., it matches the Gilbert-Varshamov bound). However this construction has no known efficient decoding. We can instead use the concatenated solution in Theorem 4 of [45], which achieves the Zyablov bound. The construction provides a code with linear-time encoding and decoding. We can instead use the concatenated solution in the literature [58]. It is, however, enough to prove security of our oblivious transfer protocol. This more general definition allows for the possibility of using the constant-round protocol of Ding et al. [39] for interactive hashing.

**Lemma 2.17 ([39]):** Let $t, m$ be positive integers such that $t \geq \log m + 2$. Then there exists a four-message $(2^{-m})$-secure $(t, 2^{-(m-t)} + O(\log m))$-secure interactive hashing protocol.

The following lemma is a result by [59]. It is 0-uniform (that is, $W_{1-d}$ is distributed uniformly), and achieves near-optimal security [58], but has the disadvantage of taking $m - 1$ rounds to execute.

**Lemma 2.18 ([59]):** There exists a 0-uniform $(t, a \cdot 2^{-(m-t)})$-secure interactive hashing protocol for some constant $a > 0$.

A secure interactive hashing scheme guarantees that one of the outputs is random; however, in the oblivious transfer protocols, the two binary strings are not used directly, but as encodings of subsets of sequences. Thus for the protocol to succeed, both outputs need to be valid encodings of subsets of $(\binom{n}{i})$. The original protocol of Cachin et al. [36] for oblivious transfer used an encoding scheme that has probability of success 1/2, thus requiring that the protocol be repeated several times to guarantee correctness. Later, Ding et al. [39] proposed a “dense” encoding of subsets, ensuring that most $m$-bit strings are valid encodings. More precisely, they showed the following result.

**Lemma 2.19:** Let $\ell \leq n, m \geq \lceil \log \binom{n}{i} \rceil$, $t_m = \lceil 2^m / \binom{n}{i} \rceil$. Then there exists an injective mapping $F': \binom{n}{i} \times [t_m] \rightarrow [2^m]$ with $|\text{Im}(F)| > 2^m - \binom{n}{i}$.

**D. Miscellaneous**

Universal hash functions were introduced by Carter and Wegman [60] and are very useful in cryptography.

**Definition 2.20 (t-universal hash functions):** A family of functions $G = \{g: \mathcal{H} \rightarrow \mathcal{L}\}$ is called a family of $t$-universal hash functions if for $g \leftarrow G$ and any $x_1, \ldots, x_t \in \mathcal{H}$, the induced distribution on $(g(x_1), \ldots, g(x_t))$ is uniform over $\mathcal{L}^t$.

For any $\mathcal{H} = \{0, 1\}^\ell$ and $\mathcal{L} = \{0, 1\}^t$, there exists a $t$-universal family of hash functions for which the function description has size $\text{poly}(h, t)$ bits, and the sampling and computing times are in $\text{poly}(h, t)$.

The following is a basic fact that follows from simple counting.

**Lemma 2.21:** Let $0 \leq \delta < 1/2$ and let $x, y \in \{0, 1\}^n$ such that $\text{HD}(x, y) \leq \delta n$ and $H_\infty(X) \geq \alpha n$ where $0 < \alpha < 1$. Then $H_\infty(Y) \geq (\alpha - h(\delta)) n$.

The next lemma shows that the restrictions of two tuples to random subsets of their positions have relative Hamming distances that are close to the one between the entire tuples.

**Lemma 2.22:** Let $x, y \in \{0, 1\}^n, S$ be a random subset of $[n]$ of size $r$ and consider any $\nu \in [0, 1]$. On one hand, if $\text{HD}(x, y) \leq \delta n$, then $\text{HD}(x^S, y^S) < (\delta + \nu)r$ except with probability $e^{-r\nu^2/2}$. On the other hand, if $\text{HD}(x, y) \geq \delta n$, then $\text{HD}(x^S, y^S) > (\delta - \nu)r$ except with probability $e^{-r\nu^2/2}$.

**Proof:** Let $S$ begin with the first part of the Lemma. By Lemma 2.10, a random subset sampler is an $(\mu, \nu, e^{-r\nu^2/2})$-averaging sampler for any $\mu, \nu > 0$. Hence for any $f: [n] \rightarrow \{0, 1\}$.
where the last inequality is valid for $\sigma < \frac{1}{n}$ with $\frac{1}{n} \sum_{i=1}^{n} f(i) \geq \mu$

$$\Pr \left[ \frac{1}{|S|} \sum_{i \in S} f(i) \leq \mu - \nu \right] \leq e^{-r^2/2}, \quad (3)$$

Let

$$f(i) = \begin{cases} 0, & \text{if } x_i \neq y_i, \\ 1, & \text{otherwise.} \end{cases}$$

Fix $\mu = 1 - \delta$. Note that $\frac{1}{|S|} \sum_{i \in S} f(i) = 1 - \frac{\text{HD}(x^S, y^S)}{r}$ and $\frac{1}{n} \sum_{i=1}^{n} f(i) = 1 - \frac{\text{HD}(x, y)}{r} \geq \mu$. Thus by Equation (3)

$$e^{-r^2/2} \geq \Pr \left[ \frac{1}{|S|} \sum_{i \in S} f(i) \leq \mu - \nu \right] = \Pr \left[ 1 - \frac{\text{HD}(x^S, y^S)}{r} \leq 1 - \delta - \nu \right] = \Pr \left[ \text{HD}(x^S, y^S) \geq (\delta + \nu)r \right]$$

which proves the first part of the Lemma.

The second part of the Lemma uses the same idea, but now the function $f$ is

$$f(i) = \begin{cases} 0, & \text{if } x_i = y_i, \\ 1, & \text{otherwise}. \end{cases}$$

Fixing $\mu = \delta$ it holds that $\frac{1}{|S|} \sum_{i \in S} f(i) = \frac{\text{HD}(x^S, y^S)}{r}$ and $\frac{1}{n} \sum_{i=1}^{n} f(i) = \frac{\text{HD}(x, y)}{r} \geq \mu$ and hence

$$e^{-r^2/2} \geq \Pr \left[ \frac{1}{|S|} \sum_{i \in S} f(i) \leq \mu - \nu \right] = \Pr \left[ \frac{\text{HD}(x^S, y^S)}{r} \leq \delta - \nu \right] = \Pr \left[ \text{HD}(x^S, y^S) \leq (\delta - \nu)r \right]$$

which finishes the proof of the lemma.

The following statement of the birthday paradox is standard.

**Lemma 2.23**: Let $A, B \subset \{0, 1\}^n$ chosen independently at random, with $|A| = |B| = 2\sqrt{n}$. Then

$$\Pr[|A \cap B| < \ell] < e^{-\ell/4}$$

**Proof**: See corollary 3 in [37].

The following useful lemma will also be needed in the subsequent sections.

**Lemma 2.24 ([61]):** Let $0 < \sigma < 1/2$. Then

$$\sum_{i=0}^{\sigma k} \binom{k}{i} \leq 2^{h(\sigma)k}.$$ 

**Proof**: It holds that

$$2^{-h(\sigma)k} = 2^{\sigma(\log \sigma + (1-\sigma) \log(1-\sigma))k} = \sigma^{\sigma k (1 - \sigma)^{1 - \sigma}k} \leq \sigma^i (1 - \sigma)^{k-i} \text{ for } i = 0, \ldots, \sigma k.$$ 

where the last inequality is valid for $\sigma < 1/2$.

Hence

$$2^{-h(\sigma)k} \sum_{i=0}^{\sigma k} \binom{k}{i} \leq \sum_{i=0}^{\sigma k} \binom{k}{i} \sigma^i (1 - \sigma)^{k-i} = 1.$$ 

This proves the lemma.

The following lemma by Rompel will be also useful.

**Lemma 2.25 ([62]):** Suppose $t$ is a positive even integer, $X_1, \ldots, X_n$ are $t$-wise independent random variables taking values in the range $[0, 1]$, $X = \sum_{i=1}^{n} x_i$, $\mu = E[X]$, and $A > 0$. Then

$$\Pr [ |X - \mu| > A ] < O \left( \left( \frac{t \mu^2 + t^2}{A^2} \right)^{1/2} \right).$$

### III. Security Model

#### A. Bounded Storage Model

We work in a two-party scenario, where two players (Alice and Bob) engage in cryptographic protocols, more specifically commitment and oblivious transfer protocols. We assume that one of the players has an upper bound on the available memory. As usual in the cryptographic literature, we assume an adversary that can corrupt either party. We will call a corrupted party dishonest. Parties that have not been corrupted will be called honest.

Cryptographic protocols in the bounded storage model run a transmission phase prior to their main part. We briefly describe this phase here.

**Transmission Phase**: In this phase, the sender (Alice) has access to a sample $x \in \{0, 1\}^n$ from an $\alpha$-source $X$, where $0 < \alpha < 1$, and the receiver (Bob) to $\bar{x} \in \{0, 1\}^n$ such that $\text{HD}(x, \bar{x}) \leq \delta n$. Note that this captures both the situation where the source is noisy and the situation where an adversary controls part of the source. In the bounded storage model normally a memory bound is imposed on both parties during this phase, but we are able to prove the security of our protocols while imposing a memory bound on only one of them (which one depends on the protocol). For a memory bounded Alice, she computes a randomized function $f(x)$ with output size at most $\gamma n$ for $\gamma < \alpha$, stores its output and discards $x$. Similarly, for a memory bounded Bob, he computes a randomized function $f(\bar{x})$ with output size at most $\gamma n$ for $\gamma < \alpha$, stores its output and discards $\bar{x}$. We should mention that in the proposed protocol the honest parties only have to store a bounded amount of information. It should also be highlighted that even if the memory bounded party gains infinite storage power after the transmission phase is over and the source is not available anymore, this does not affect the security of the protocol, i.e., it has everlasting security.

#### B. Secure Commitment

The main part of a commitment protocol has two phases: commitment and opening.

**Commitment Phase**: Alice has an input string $v \in \mathcal{V}$ (which is a realization of a random variable $V$) that she wants to commit to. The parties exchange messages, possibly in several rounds. Let $\text{trans}^{CP}(v)$ denote all the communication in this phase and $\text{view}^{CP}(v)$ Bob’s view at the end of this phase. These random variables are a function of $v$, the functions that the parties computed from the public random source and the parties’ local randomness.
Opening Phase: Alice sends Bob the string $\tilde{v}$ that she claims she committed to. The parties can then exchange messages in several rounds. Let $\text{trans}^{\text{OP}}(\tilde{v})$ denote all the communication in this phase. In the end Bob performs a test

$$\text{test} \left( \text{view}^{\text{CP}}_{\text{Bob}}(v), \text{trans}^{\text{OP}}(\tilde{v}) \right)$$

that outputs 1 if Bob accepts Alice’s commitment and 0 otherwise.

Security. A commitment protocol is called $(\lambda_C, \lambda_B, \lambda_A)$-secure if it satisfies the following properties:

1) $\lambda_C$-correct: if Alice and Bob are honest, then for every possible $v$, the probability that the protocol aborts is at most $\lambda_C$

$$\Pr \left[ \text{no aborts and test} \left( \text{view}^{\text{CP}}_{\text{Bob}}(v), \text{trans}^{\text{OP}}(v) \right) \right] = 1 \geq 1 - \lambda_C.$$

2) $\lambda_H$-hiding: if Alice is honest then Bob’s knowledge on her committed value is at most $\lambda_H$

$$I(V; \text{view}^{\text{CP}}_{\text{Bob}}(V)|\tilde{X}) \leq \lambda_H.$$

3) $\lambda_B$-binding: if Bob is honest, then there are no $v$ and $\tilde{v} \neq \tilde{v}$ that can be successfully open,

$$\Pr \left[ \text{test} \left( \text{view}^{\text{CP}}_{\text{Bob}}(v), \text{trans}^{\text{OP}}(\tilde{v}) \right) = 1 \right] \geq \lambda_B$$

and

$$\Pr \left[ \text{test} \left( \text{view}^{\text{CP}}_{\text{Bob}}(v), \text{trans}^{\text{OP}}(\tilde{v}) \right) = 1 \right] \geq \lambda_B.$$

C. Secure Oblivious Transfer

We use the definition of oblivious transfer presented in [39]. An oblivious transfer protocol is a protocol between two players. Alice and Bob, in which Alice inputs two strings $s_0, s_1 \in \mathcal{V}$ and outputs nothing, and Bob inputs $c \in \{0, 1\}$ and outputs $s \in \{\bot, s_1\}$. Let $\text{view}^{\text{OT}}_{\text{Alice}}(s_0, s_1; c)$ denote the view of an Alice that interacts with an honest Bob. Similarly, let $\text{view}^{\text{OT}}_{\text{Bob}}(s_0, s_1; c)$ denote the view of a Bob that interacts with an honest Alice.

Intuitively, the protocol will be secure for Bob if the view of Alice does not depend on the choice bit $c$, and secure for Alice if Bob cannot obtain any information about $s_1 \cdots c$. However this is tricky to formalize, because a malicious Bob could choose to play with a different bit, depending on the public random source and the messages exchanged before any secret is used by Alice.

In order to have more generality, the main part of the oblivious transfer protocol is divided in two phase: the setup phase, which encompass all communication before Alice first uses her secrets, and the transfer phase, which happens from that point on. Two pairs of inputs $(s_0, s_1), (s'_0, s'_1)$ are called $I$-consistent if $s_i = s'_i$. By the end of the setup phase there should exist a random variable $I$, such that for any two $I$-consistent pairs of inputs, the resulting view of Bob is statistically close.

Security: A protocol is called $(\lambda_C, \lambda_B, \lambda_A)$-secure if it satisfies the following properties:

1) $\lambda_C$-correct: if Alice and Bob are honest, then

$$\Pr \left[ \text{no aborts and } s = s_c \right] \geq 1 - \lambda_C.$$

2) $\lambda_B$-secure for Bob: for any strategy used by Alice,

$$\Pr \left[ \text{view}^{\text{OP}}_{\text{Alice}}(s_0, s_1; 0) \neq \text{view}^{\text{OP}}_{\text{Alice}}(s_0, s_1; 1) \right] \leq \lambda_B.$$

3) $\lambda_A$-secure for Alice: for any strategy used by Bob with input $c$, there exists a random variable $I$, defined at the end of the setup stage, such that for every two $I$-consistent pairs $(s_0, s_1), (s'_0, s'_1)$, we have

$$\Pr \left[ \text{view}^{\text{OP}}_{\text{Bob}}(s_0, s_1; c) \neq \text{view}^{\text{OP}}_{\text{Bob}}(s'_0, s'_1; c) \right] \leq \lambda_A.$$

IV. A Simple String Commitment Protocol

Next we present a quite simple string commitment protocol that only involves one message from Bob to Alice. A memory bound on Bob is assumed. The scheme works as follows. First, both parties sample a number of bits from the public source. Alice then extracts the randomness of her sample and uses it to conceal her commitment before sending it to Bob. This guarantees the hiding condition. She also computes a hash of her sample, where the hash function is chosen by Bob. Alice sends Bob the concealed commitment along with the hash value. In the open phase, Alice sends her committed value and her sampled string. Bob then performs a number of checks for consistency. These checks enforce bindingness. The details of the protocol are presented below.

The security parameter is $\ell$ and $k$ is set as $k = 2\sqrt{n}$. Fix $\varepsilon' > 0$ and let $\rho = \alpha - \gamma - \frac{1 + \log(1, \ell)}{\ell}$. Fix $\tau$ such that $\frac{\rho}{\ell} \geq \tau > 0$, and $\omega, \zeta > 0$ such that $\rho - 3\tau > \omega > 2h(\delta + \zeta)$ and $\delta + \zeta < 1/2$. Let $k_E = (3 - \rho - \omega)k$ and for $\psi > 0$, $m = (1 - \psi)k_E$. The message space is $\mathcal{V} = \{0, 1\}^m$. It is assumed that the following functionalities, which are possible due to the lemmas in Section II, are available to the parties:

- A family $\mathcal{G}$ of 2-universal hash functions $g : \{0, 1\}^k \rightarrow \{0, 1\}^{\omega k}$

- A $(k_E, \varepsilon_E)$-strong extractor $\text{Ext} : \{0, 1\}^{k} \times \{0, 1\}^r \rightarrow \{0, 1\}^m$, for an arbitrary $\varepsilon_E > e^{-k/2^{O(h^\omega k)}}$.

Remark 4.1: Note that it should hold that $2h(\delta) < \omega + 3\tau < \rho < \alpha - \gamma$, so the protocol is only possible if $2h(\delta) < \alpha - \gamma$.

Transmission phase:

1) Alice chooses uniformly $k$ positions from $X$. Similarly, Bob samples $k$ positions from $\tilde{X}$. We call their sets of positions $\mathcal{A}$ and $\mathcal{B}$, respectively.

Commit phase:

1) Alice announces $\mathcal{A}$ to Bob.

2) Bob chooses $g \leftarrow \mathcal{G}$ and sends its description to Alice.

3) Alice computes $p \leftarrow g(x^A)$, $u \leftarrow \{0, 1\}^r$, and $y \leftarrow \text{Ext}(x^A, u)$. She then computes $z = v \oplus y$ and sends $(z, p, u)$ to Bob in order to commit to $v$. 
Open phase:

1) Alice sends $v'$ and $w$ to Bob, which are defined as $v' = v$ and $w = x^A$ in the case that she is honest.

2) Let $C = A \cap B, c = |C|$ and $w^C$ be the restriction of $w$ to the positions corresponding to the set $C$. Bob verifies whether $c \geq \ell$, $\text{HD}(w^C, \tilde{x}^C) \leq (\delta + \zeta)c, p = g(w)$ and $v' = \text{Ext}(w, u) \oplus z$. If any verification fails Bob outputs 0, otherwise he outputs 1.

Theorem 4.2: The protocol is $(\lambda_C, \lambda_H, \lambda_B)$-secure for $\lambda_C, \lambda_H$ and $\lambda_B$ negligible in $\ell$.

Proof: Correctness: It is clear that if both Alice and Bob are honest, the protocol will fail only in the case that $c < \ell$ or $\text{HD}(x^C, \tilde{x}^C) > (\delta + \zeta)c$. By Lemma 2.23, $c \geq \ell$ except with probability at most $e^{-\ell/4}$. By Lemma 2.22, $\text{HD}(x^C, \tilde{x}^C) \leq (\delta + \zeta)c$ except with probability at most $e^{-\omega^2/2}$, which is negligible in $\ell$ if $c \geq \ell$.

Hiding: After the commit phase, (a possibly malicious) Bob possesses $(z, p, A, w)$ and the output of a function $f(\cdot)$ of $\tilde{x}$, where $f(\tilde{x}) \leq gn$ with $g < \alpha$. The only random variable that can provide mutual information about $V$ when conditioned on $\tilde{x}$ is $Z$, but we prove that $Z$ is almost uniform from Bob’s point of view, and so it works as an one-time pad and only negligible information can be leaked.

By Lemma 2.7, $\Pr[\exists w, w' \text{ s.t.} \begin{align*}
&g(w) = g(w') \\
&\text{HD}(w^C, \tilde{x}^C) \leq \sigma k \\
&\text{HD}(w', \tilde{x}^C) \leq \sigma k
\end{align*}]
= \sum_{w : \text{HD}(w, \tilde{x}^C) \leq \sigma k} \left( \sum_{w' : \text{HD}(w', \tilde{x}^C) \leq \sigma k} 2^{\omega^2/2} \right)
$ where Lemma 2.24 was used to obtain the inequality. By design, it holds that $\omega > 2h(\sigma)$, therefore the probability that Alice successfully cheats by finding two strings that are at distance at most $\sigma k$ from $\tilde{x}^A$ and hash to the same value is negligible in $k$.

Now considering the second case, by assumption $w$ has Hamming distance $(\sigma + \psi)k$ from $\tilde{x}^A$ for some $\psi > 0$. Since Bob is honest, $B$ is chosen randomly. Hence Lemma 2.22 can be applied and thus the probability that $\text{HD}(w^C, \tilde{x}^C) \leq \sigma c$ is smaller than $e^{-\omega^2/2}$.

V. Extending the Feasibility Region

While the previous protocol is simple, efficient and round optimal, it works for a rather limited range of noise: $\mu(\delta) < (\alpha - \gamma)/2$. We next present a more elaborate protocol that works for a much larger range of noise $\mu(\delta) < \alpha - \gamma$ at the cost of increasing the rounds of communication. The memory bound is still on Bob. The idea for guaranteeing the binding property is to use two rounds of hash challenge-responses in order to guarantee the binding condition. Consider the initial set of viable strings that Alice can possibly send to Bob during the commit phase that would pass the Hamming distance test. The first hash challenge-response round binds Alice to one specific output of the hash function, and thus restrict the set of viable strings to be polynomial in the security parameter. The second hash challenge-response round then binds Alice to one specific value for the commitment. Our solution is based on families of 4k-universal hash functions. This approach has been used before in a different context [22].

The security parameter is $\ell$ and $k$ is set as $k = 2\sqrt{\ell n}$. Fix $\epsilon' > 0$ and let $\rho = \alpha - \gamma - \frac{1 + \log(1/\epsilon')}{n}$. Fix $\tau$ such that $\frac{\ell}{4} \geq \tau > 0$, and $\omega_1, \omega_2, \omega_3 > 0$ such that $\rho - 3\tau > \omega_1 + \omega_2, \omega_1 > h(\delta + \zeta)$, and $\delta + \zeta < 1/2$. Let $k_E = (\rho - 3\tau - \omega_1 - \omega_2)k$ and for $\psi > 0, m = (1 - \psi)k_E$. The message space is $V = \{0, 1\}^m$. It is assumed that the following functionalities, which are possible due to the lemmas in Section II, are available to the parties:

- A family $G_1$ of 4k-universal hash functions $g_1 : \{0, 1\}^k \rightarrow \{0, 1\}^{\omega_1 k}$.
- A family $G_2$ of 2-universal hash functions $g_2 : \{0, 1\}^k \rightarrow \{0, 1\}^{\omega_2 k}$.
- A $(k_E, \epsilon_E)$-strong extractor $\text{Ext} : \{0, 1\}^k \times \{0, 1\}^* \rightarrow \{0, 1\}^\mu$, for an arbitrary $\epsilon_E > e^{-k/2^{\Omega(k \log^* k)}}$.
Remark 5.1: Note that it should hold that $h(\delta) < \omega_1 + 3\tau < \rho < \alpha - \gamma$, so the protocol is only possible if $h(\delta) < \alpha - \gamma$.

Transmission phase:
1) Alice chooses uniformly $k$ positions from $X$. Similarly, Bob samples $k$ positions from $X$. We call their sets of positions $A$ and $B$, respectively.

Commit phase:
1) Alice announces $A$ to Bob.
2) Bob chooses $g_1 \leftarrow G_1$ and sends its description to Alice.
3) Alice computes $p_1 \leftarrow g_1(x^A)$ and sends it to Bob.
4) Bob chooses $g_2 \leftarrow G_2$ and sends its description to Alice.
5) Alice computes $p_2 = g_2(w)$ and $v' = Ext(w, u)$ and sends $(z, p_2, u)$ to Bob in order to commit to $v$.

Open phase:
1) Alice sends $w'$ and $w$ to Bob, which are defined as $w' = v$ and $w = x^A$ in the case that she is honest.
2) Let $C = A \cap B$, $c = |C|$ and $w^C$ be the restriction of $w$ to the positions corresponding to the set $C$. Bob verifies whether $c \geq \ell$, $HD(w^C, \tilde{w}^C) \leq (\delta + \zeta)c$, $p_1 = g_1(w)$, $p_2 = g_2(w)$, and $v' = Ext(w, u) \oplus z$. If any verification fails Bob outputs 0, otherwise he outputs 1.

Theorem 5.2: The protocol is $(\lambda_C, \lambda_H, \lambda_B)$-secure for $\lambda_C, \lambda_H, \lambda_B$ negligible in $\ell$.

Proof: Correctness: Same as in Theorem 4.2.

Hiding: Follows the same lines as in Theorem 4.2. The difference is that here $k_E = (\rho - 3\tau - \omega_1 - \omega_2)k$ in order to account for the entropy loss due to the output of both hash functions $g_1$ and $g_2$ (instead of $k_E = (\rho - 3\tau - \omega)$ in Theorem 4.2 that accounts for the output of a single hash function $g$).

Binding: The protocol is binding if, after the commit phase, Alice cannot choose between two different values to successfully open. Let $\sigma = \delta + \zeta$. The only way Alice can cheat is if she can come up with two different strings $w, w'$ that pass all tests performed by Bob during the opening phase. Either $HD(w, \tilde{w}^A) \leq \sigma k$ and $HD(w', \tilde{w}^A) \leq \sigma k$; or Alice can compute $w$ (without knowing the set $B$ that together with $A$ determines $C$) such that $HD(w, \tilde{w}^A) > \sigma k$ and $HD(w, \tilde{w}^C) \leq \sigma c$. The probability that Alice succeeds in cheating in the latter case can be upper bounded as in Theorem 4.2. Below we upper bound her cheating success probability in the former case and prove that it decreases exponentially with the security parameter $\ell$ (or, equivalently in $k$).

Let the viable set dynamically denote the strings that Alice can possibly send to Bob with non-negligible probability of successful opening. Before the first round of hash challenge-response, the viable set consists of all $w$ such that $HD(w, \tilde{w}^A) \leq \sigma k$. Now let's consider an arbitrary fixed value $p_1$ for the output of the first hash. Considering the $j$-th viable string before the first hash challenge-response round, define $I_j$ as 1 if the $j$-th viable string is mapped by $g_1$ to $p_1$; otherwise $I_j = 0$. And define $I = \sum_j I_j$. Clearly $\mu = E[I] < 1$, as $g_1$ is chosen from a $4k$-universal family of hash functions with range of size $\{0, 1\}^{\omega_1}$ for $\omega_1 > h(\delta + \zeta)$. Let $p_1$ be called bad if $I$ is bigger than $8k + 1$. Using the fact that $g_1$ is $4k$-wise independent and applying Lemma 2.25 with $t = 4k$ and $A = 2t = 8k$, we get

$$Pr[I > 8k + 1] < O\left(\frac{t\mu + 2}{2t^2}\right)^{t/2} < O\left(\frac{1 + t}{4t}\right)^{t/2} < O\left(2^{-t/2}\right).$$

Then the probability that any $p_1$ is bad is upper bounded by

$$O\left(2^{e_k}2^{-t/2}\right) < O\left(2^{-k}\right).$$

If the viable set is reduced to at most $8k + 1$ elements after the first hash challenge-response round, then the probability that some of those collide in the second hash challenge-response round is upper bounded by

$$(8k + 1)^2 2^{-\omega_2},$$

which is negligible in $k$.

VI. ALTERNATIVE BIT COMMITMENT PROTOCOL

Next we design a bit commitment protocol where the memory bound is imposed on Alice instead of Bob. The protocol works for $h(\delta) < \alpha - \gamma$ and uses families of 2-universal hash functions, instead of the costly $4k$-universal hash functions. The central idea is to use an interactive hashing execution to perform the bit commitment [40].

Before describing our solution, we remark that is important to obtain protocols that work for memory bounded Alice and protocols that work for memory bounded Bob. This is particularly interesting in the case of an asymmetry of power between the parties. For example, when one of the parties is the government of the United States. It makes sense to impose the bound on the weak party, whenever it is the sender of the commitment (Alice) or the receiver of the commitment (Bob).

Alice has a bit $v$ which she wants to commit to. The security parameter is $\ell$ and $k$ is set as $k = 2\sqrt{\ell}$. Fix $\epsilon' > 0$ and $\xi > 0$ such that $\delta + \xi < 1/2$, and let $\rho = \alpha - \gamma - 1 - \log(1/\epsilon')$. Fix $0 < \xi < 1$ and $\tau$ such that $\frac{\tau}{\xi} \geq \tau > 0$. Let $\nu = \frac{\tau}{\log(1/\tau)}$, $\nu' = e^{\nu} = \frac{\log(1/\tau)}{\log(1/\epsilon')}$, and $\nu'' = e^{-\nu}/2 - 2^{-\Omega(\tau n)}$, where the last term comes from Lemma 2.9. Fix $m \geq \ell (\log k + 1)$ and $m - O(\ell) \geq t \geq m - \log(1/(\epsilon' + \epsilon''))$. It is assumed that the following functionality, which is possible due to the lemmas in Section II-C, is available to the parties:

- An $2^{-m}$-uniform $(t, 2^{-m-t})$-secure interactive hashing protocol with input domain $W = \{0, 1\}^m$ and an associated dense encoding of subsets $F$ for tuples of size $k$ and subsets of size $\ell$.

The following bit commitment protocol is correct and secure if $h(\delta + \xi) < \rho - 3\tau$. 
Transmission phase:
1) Alice chooses uniformly \( k \) positions from \( X \). Similarly, Bob samples \( k \) positions from \( \tilde{X} \). We call their sets of positions \( \mathcal{A} \) and \( \mathcal{B} \), respectively.

Commit phase:
1) Bob announces \( B \) to Alice. Alice computes \( \mathcal{D} = \mathcal{A} \cap \mathcal{B} \). If \( |\mathcal{D}| < \ell \), Alice aborts. Otherwise, Alice picks a random subset \( \mathcal{C} \) of \( \mathcal{D} \) of size \( \ell \).
2) Alice computes the encoding \( w \) of \( \mathcal{C} \) (as a subset of \( \mathcal{B} \)). Alice and Bob interactively hash \( w \), producing two strings \( w_0, w_1 \). They compute the subsets \( \mathcal{C}_0, \mathcal{C}_1 \subset \mathcal{B} \) that are respectively encoded in \( w_0, w_1 \). If either encoding is invalid, they abort.
3) Alice sends \( p = v \oplus d \) to Bob, where \( w_d = w \).

Open phase:
1) Alice sends \( v' \) and \( x^{\mathcal{C}_i} \) to Bob, which are defined as \( v' = v \) and \( x^{\mathcal{C}_i} = x^{\mathcal{C}} \) in the case that she is honest.
2) Bob computes \( d' = p \oplus v' \) and checks whether \( \text{HD}(x^{\mathcal{C}_i}, \tilde{x}^{\mathcal{C}_i}) \leq (\delta + \xi)\ell \). If the verification fails Bob outputs 0, otherwise he outputs 1.

Theorem 6.1: The protocol is \((\lambda_\mathcal{C}, 0, \lambda_\mathcal{B})\)-secure for \( \lambda_\mathcal{C} \) and \( \lambda_\mathcal{B} \).

Proof: Correctness: If both participants are honest, the protocol fails only in the following cases: (1) \( |\mathcal{D}| < \ell \); (2) \( \text{HD}(x^{\mathcal{C}_i}, \tilde{x}^{\mathcal{C}_i}) > (\delta + \xi)\ell \) or (3) \( w_0 \) or \( w_1 \) is an invalid encoding of a subset. By Lemma 2.23, \( |\mathcal{D}| \geq \ell \) except with probability at most \( e^{-\ell/4} \). By Lemma 2.22, \( \text{HD}(x^{\mathcal{C}_i}, \tilde{x}^{\mathcal{C}_i}) \leq (\delta + \xi)\ell \) except with probability at most \( e^{-\ell \xi/2} \). Finally, since \( w_d = w \) is the encoding of \( \mathcal{C} \), one of the two outputs of the interactive hashing protocol is always a valid encoding. The other output \( W_{1-d} \) is \( 2^{-m} \)-close to distributed uniformly over the \( 2^{-m} \) strings different from \( w_d \). Since it is a dense encoding, Lemma 2.19 implies that the probability that it is not a valid encoding is thus less than or equal to

\[
2^{-m} + \frac{k}{2^{m} - 1} \leq 2^{-m} + 2^{\ell \log k - m + 1} \\
\leq 2^{-\ell \log k - \ell} + 2^{-\ell + 1} \\
\leq 2^{-\ell + 2}
\]

for \( m \geq \ell \log (k + 1) \).

Putting everything together this proves the correctness.

Hiding: There are two possibilities: either the protocol does not abort; or it aborts due to \( |\mathcal{D}| < \ell \) or an invalid encoding. If the protocol aborts, Alice still has not sent \( p = v \oplus d \), so Bob’s view is independent from \( V \). On the other hand, if the protocol does not abort, then \( w_{1-d} \) is a valid encoding of some set \( \mathcal{C}_i \). Due to the properties of the interactive hashing protocol, Bob’s view is then consistent with both
1) Alice committing to \( v \) and \( \mathcal{C} \) being the subset for which she knows the positions of \( x \), and
2) Alice committing to \( 1-v \) and \( \mathcal{C}_i \) being the subset for which she knows the positions of \( x \).

Hence Bob’s view is independent of \( V \).

Binding: The strategy of the proof is to demonstrate that there is an \( i \) such \( X^{\mathcal{C}_i} \) has high enough min-entropy from Alice’s point of view so that she cannot guess (except with negligible probability) a string \( X^{\mathcal{C}_i} \) that is close enough to \( \tilde{X}^{\mathcal{C}_i} \). Hence she will not be able to successfully use this output of the interactive hashing during the opening phase and will thus be bounded to use the other output of the interactive hashing.

By the bounded storage assumption, the bounded information \( f(X) \) stored by Alice is such that \( |f(X)| \leq \gamma n \) with \( \gamma < \alpha \).

Then, by Lemma 2.7,

\[
R_{\infty}^{\epsilon}(X|f(X)) \geq \alpha - \gamma - 1 + \log(1/\epsilon^3)/n = \rho.
\]

Since Bob is honest, \( B \) is randomly chosen. Let’s consider a random subset \( \tilde{C} \) of \( B \) such that \( |\tilde{C}| = \ell \). This is an \((\mu, \nu, e^{-\nu^2/2})\)-averaging sampler for any \( \mu, \nu > 0 \) according to Lemma 2.10. By setting \( \mu = \frac{\rho - 2\tau}{\log(1/\tau)} \), we have by Lemma 2.9 that

\[
R_{\infty}^{\epsilon'}(X|^\tilde{C}B, \tilde{C}, f(X)) \geq \rho - \tau,
\]

for \( \epsilon' = e^{-\nu^2/2} - 2^{-\Omega(\tau n)} \). For \( \tilde{C} = (\epsilon' + \epsilon'')^{1-i} \), let \( B\tilde{A}D \) be the set of \( \tilde{C} \)'s such that \( R_{\infty}^{\epsilon'}(X|^\tilde{C}B, \tilde{C}, f(X)) \) is not \( \epsilon \)-close to \( (\rho - \tau) \)-min entropy rate. Due to the above equation the density of \( B\tilde{A}D \) is at most \((\epsilon' + \epsilon'')^2 \). Hence the properties of the interactive hashing protocol guarantee that with overwhelming probability there will be an \( i \) such that

\[
R_{\infty}^{\epsilon'}(X|^\tilde{C}B, \tilde{C}_i, f(X), M_{1H}) \geq \rho - \tau,
\]

where \( M_{1H} \) are the messages exchanged during the interactive hashing protocol.

However, if \( h(\delta + \xi) < \rho - 3\tau \) and the min-entropy rate is at least \( \rho - 3\tau \), then fixing \( 0 < \hat{\epsilon} < \rho - 3\tau - h(\delta + \xi) \), for large enough \( \ell \), the probability that Alice guesses one of the strings \( X^{\mathcal{C}_i} \), that would be accepted by Bob as being close enough to \( \tilde{X}^{\mathcal{C}_i} \), is upper bounded by

\[
2^{(h(\delta + \xi) - \rho + 3\tau - \hat{\epsilon}) \ell}
\]

which is a negligible function of \( \ell \).

By fixing the parameters as small as possible we have that for large enough \( \ell \) the protocol works for values \( \alpha, \gamma, \delta \) which satisfy \( h(\delta) < \alpha - \gamma \).

VII. Oblivious Transfer Protocol

Our OT protocol imposes a memory bound on Bob, but we would like to point out that it is trivial to revert the direction of OT protocols [63]. We first present the intuition behind our protocol before a detailed description. Initially, both parties sample positions from the public random source. Then the parties use an interactive hashing protocol (with an associated dense encoding) to select two subsets of the positions initially sampled by Alice. Bob inputs into the interactive hashing protocol one subset for which he has also sampled the public random source in the same positions. The other subset is out of Bob’s control due to the properties of the interactive hashing.
protocol. Finally the positions specified by the two subsets are used as input to a fuzzy extractor in order to obtain one-time pads. Bob sends one bit indicating which input string should be XORed with which one-time pad. The security for Alice follows from the correctness of the fuzzy extractor. The correctness follows from the correctness of the fuzzy extractor.

The security parameter is $\ell$ and $k$ is set as $k = 2\sqrt{\ell n}$. Fix $\varepsilon', \hat{\varepsilon} > 0$ and $\xi > 0$ such that $\gamma = 1/\delta + \xi > 0$ and let $\rho = \alpha - \gamma - 1 + \log(1/\varepsilon')$. Fix $0 < \zeta < 1$ and $\tau$ such that $\zeta \geq \tau > 0$. Let $\mu = \frac{\rho - \gamma}{\log(1/\tau)}$, $\nu = \frac{\gamma}{\log(1/\tau)}$ and $\varepsilon'' = \frac{\varepsilon - \varepsilon'}{2} = \frac{\varepsilon - \varepsilon'}{2}$. Where the last term comes from Lemma 2.9. Fix $m = \ell (\log k + 1)$ and $m - O(\ell) \geq t \geq m - \xi \log(1/(\varepsilon' + \varepsilon''))$. For $\beta$ depending on $\delta + \xi$ (see comments about the code rate below), let $k_F$ and $m_F$ be such that $k_F = n + \beta - 3\tau - 2m_F - \frac{1}{2} - \frac{1}{2} \log(1/\varepsilon'')$ and $0 < m_F < k_F$. The message is $V = \{0, 1\}^{m_F}$. We assume the following functionalities are available to the parties (see the lemmas in Sections II-B and II-C):

- A pair of functions $\text{Ext}: \{0, 1\}^y \times \{0, 1\}^r \rightarrow \{0, 1\}^{m_F} \times \{0, 1\}^y$ and $\text{Rec}: \{0, 1\}^r \times \{0, 1\}^y \rightarrow \{0, 1\}^{m_F}$ that constitutes an $(k_F, \ell, \varepsilon, \beta, \xi, 0)$-fuzzy extractor where $\varepsilon = (1 - R)\ell$, $\varepsilon = 2^{-\ell}$ is an arbitrary number with $\varepsilon > e^{-\ell/2^{k(\log k)^2}}$.
- An $2^{-m}$-uniform $(t, 2^{-(m+t)} + O(\log m))$-secure interactive hashing protocol with input domain $\mathcal{V} = \{0, 1\}^m$ and an associated dense encoding of subsets $F$ for tuples of size $k$ and subsets of size $\ell$.

Recall (Remark 2.15) that there is a tradeoff between the fraction of errors $\delta + \xi$ that the fuzzy extractor can tolerate and the rate $\beta$ of the code used in the construction. The construction given in Theorem 4 of [45] has linear-time encoding and decoding and achieves the Zyablov bound: for given $1 > \beta > 0$ and $\mu > 0$, the code has rate $\beta$ and

$$\delta + \xi \geq \max_{\beta < \beta < 1} \frac{(1 - \beta - \mu)y}{2}$$

where $y$ is the unique number in $[0, 1/2]$ with $h(y) = 1 - \beta - \mu$ and $\delta + \xi$ the amount of errors that can be corrected by the code.

Note that in order for $k_F$ to be positive, we need to have $\rho + \beta > 1$; since $\rho$ approaches $\alpha - \gamma$ from below in the asymptotic limit, we can obtain an upper bound for $\delta$ by setting $\beta > 1 - \alpha + \gamma$ and $\mu = 0$ in Equation (4).

Random linear codes achieve a better bound, namely, the Gilbert-Varshamov bound: for a given relative distance $\nu$ and $\mu > 0$, a random code has (with high probability) rate $\beta \geq 1 - h(\nu) - \mu$. Applying again the constraint that $\rho + \beta > 1$ and that $\rho \rightarrow \alpha - \gamma$ in the asymptotic limit, and using the fact that a code that can correct $\delta n$ errors has relative distance $\nu = 2\delta + 1/n \rightarrow 2\delta$, this gives an upper bound for $\delta$: we must have $h(2\delta) \leq \alpha - \gamma$. However, as noted in Remark 2.15, the random linear code construction does not have efficient decoding. It is an open question whether an efficient construction can achieve better parameters than the one from [45].

**Transmission phase:**
- Alice chooses uniformly $k$ positions from $X$. Similarly, Bob samples $k$ positions from $X$. We call their sets of positions $\mathcal{A}$ and $\mathcal{B}$, respectively.

**Setup phase:**
- Alice sends $\mathcal{A}$ to Bob. Bob computes $D = \mathcal{A} \cap B$. If $|D| < \ell$, Bob aborts. Otherwise, Bob picks a random subset $C$ of $D$ of size $\ell$.
- Bob computes the encoding $w$ of $C$ (as a subset of $\mathcal{A}$). Alice and Bob interactively hash $w$, producing two strings $w_0, w_1$. They compute the subsets $C_0, C_1 \subset A$ that are respectively encoded in $w_0$, $w_1$. If either encoding is invalid, they abort.

**Transfer phase:**
- Bob sends $p = c \oplus d$, where $w_d = w$.
- For $i \in \{0, 1\}$, Alice picks $r_i \leftarrow \{0, 1\}^r$, computes $(y_i, z_i) \leftarrow \text{Ext}(x_i, r_i)$ and $z_i = s_i \oplus p \oplus y_i$, and sends $(z_i, r_i, q_i)$ to Bob.
- Bob computes $y' = \text{Rec}(\hat{x}, r, q) \oplus d$ and outputs $s = y' \oplus z_d$.

**Theorem 7.1:** The protocol is $(\lambda_\mathcal{C}, 0, \lambda_\mathcal{A})$-secure for $\lambda_\mathcal{C}$ and $\lambda_\mathcal{A}$ negligible in $\ell$.

**Proof: Correctness:** We first analyze the probability of an abort. The protocol aborts if either $|D| < \ell$, or if one string obtained in the interactive hashing protocol is an invalid encoding of subsets of $\mathcal{A}$. By Lemma 2.23, $\Pr[|D| < \ell] < e^{-\ell/4}$. Since $w_d = w$ is the encoding of $\mathcal{C}$, one of the two strings is always a valid encoding. The other output $w_{1-d}$ is $2^{-m}$-close to distributed uniformly over the $2^{-m} - 1$ strings different from $w_d$. Since it is a dense encoding, Lemma 2.19 implies that the probability that it is not a valid encoding is thus less than or equal to

$$2^{-m} + \frac{\ell}{2^{m-1}} \leq 2^{-m} + 2^{\ell \log k - m + 1} \leq 2^{-\ell \log k - \ell + 2^{\ell + 1}} \leq 2^{-\ell \log k - \ell + 2^{\ell + 1}}$$

for $m \geq \ell (\log k + 1)$. If both parties are honest and there is no abort, then $s = s_e$ if and only if $\text{Rec}(\hat{x}, r, q) = y_d$. By the properties of the employed fuzzy extractor, this last event happens if $\text{HD}(x, \hat{x}) \leq (\delta + \xi)\ell$. By Lemma 2.22, $\text{HD}(x, \hat{x}) > (\delta + \xi)\ell$ with probability at most $e^{-\ell^2/2}$. Putting everything together this proves the correctness.

**Security for Bob:** There are two possibilities: either the protocol aborts or not. If the protocol aborts in the setup phase, Bob still has not sent $p = c \oplus d$, so Alice’s view is independent from $C$. On the other hand, if the protocol does not abort, then $w_{1-d}$ is a valid encoding of some set $C'$. Due to the properties of the interactive hashing protocol, Alice’s view is then consistent with both

1) Bob choosing $c$ and $C$, and
2) Bob choosing $1 - c$ and $C'$.

Hence Alice’s view is independent of $C$. 
Security for Alice: There should be an index $i$ (determined at the setup stage) such that for any two pairs $(s_0, s_1), (s'_0, s'_1)$ with $s_i = s'_i$, Bob’s view of the protocol executed with $(s_0, s_1)$ is close to his view of the protocol executed with $(s'_0, s'_1)$.

The proof’s strategy is to show that for $i$, $X^{C_{i-1}}$ has high enough min-entropy, given Bob’s view of the protocol, in such a way that $Y_{1-i}$ is indistinguishable from an uniform distribution. Indistinguishability of Bob’s views will then follow.

By the bounded storage assumption, $|\bar{f}(\bar{X})| \leq \gamma n$ with $\gamma < \alpha$. Then, by Lemma 2.7,

$$R_{\infty}^{\varepsilon'}(X|\bar{f}(\bar{X})) \geq \alpha - \gamma - \frac{1 + \log(1/\varepsilon')}{n} = \rho.$$

Since Alice is honest, $A$ is randomly chosen. Let consider a random subset $\tilde{C}$ of $A$ such that $|\tilde{C}| = \ell$. This is an $(\mu, \nu, \varepsilon - \ell \nu^2/2)$-averaging sampler for any $\mu, \nu > 0$ according to Lemma 2.10. By setting $\mu = \frac{\varepsilon - \ell \nu^2/2}{\log(1/\nu)}$, $\nu = \frac{\tau}{\log(1/\nu)}$, we have by Lemma 2.9 that

$$R_{\infty}^{\varepsilon'' + \varepsilon'''}(X^{\tilde{C}}|A, \tilde{C}, \bar{f}(\bar{X})) \geq \rho - 3 \tau$$

for $\varepsilon'' = \varepsilon - \ell \nu^2/2 - 2^{-\Omega(\nu \tau)}$. For $\bar{\varepsilon} = (\varepsilon' + \varepsilon''/\ell)$, let $BAD$ be the set of $\tilde{C}$’s such that $R_{\infty}^{\tilde{C}}(X^{\tilde{C}}|A, \tilde{C}, \bar{f}(\bar{X}))$ is not $\bar{\varepsilon}$-close to $(\rho - 3 \tau)$-min entropy rate. Due to the above equation the density of $BAD$ is at most $(\varepsilon' + \varepsilon''/\ell)^\ell \leq 2\ell$. Hence the properties of the interactive hashing protocol guarantee that with overwhelming probability there will be an $i$ such that

$$R_{\infty}^{\varepsilon'}(X^{C_{i-1}}|A, C_{i-1}, \bar{f}(\bar{X}), M_{IH}) \geq \rho - 3 \tau$$

where $M_{IH}$ are the messages exchanged during the interactive hashing protocol. We now show that $X^{C_{i-1}}$ has high min-entropy even when given $Z_i, Y_i, Q_i$. We can see $(Z_i, Y_i, Q_i)$ as a random variable over $\{0, 1\}^{2m_{r} + 1 - \beta \ell}$. Then, by Lemma 2.7,

$$R_{\infty}^{\varepsilon' + \varepsilon''}(X^{C_{i-1}}|A, C_{i-1}, \bar{f}(\bar{X}), M_{IH}, Z_i, Y_i, Q_i) \geq$$

$$\geq \rho + \beta - 3 \tau - 2m_{F} - 1 - \frac{1 + \log(1/\varepsilon)}{\ell} = k_{F}.$$

Thus setting $\varepsilon'$ and $\varepsilon''$ to be negligible in $\ell$, the use of the $(k_{F}, \varepsilon_{F}, \delta + \xi, 0)$-fuzzy extractor to obtain $y_i$, that is used as an one-time pad guarantees that only negligible information about $s_{i}\bar{f}_{i}$ can be leaked and that the protocol is $\lambda_{A}$-secure for Alice for negligible $\lambda_{A}$.

VIII. Discussion

In this section we briefly discuss the protocols we obtained in terms of their robustness against noise.

For the case of oblivious transfer, our best protocol works for levels of noise such that $h(2\delta) < \alpha - \gamma$. Putting $\alpha = 1$ and $\gamma = 0.5$ (that means, a public string that is perfectly random and the bound on the memory equal to half the length of the publicly available string), we have that oblivious transfer is possible if $\delta < 0.055$. Figure 1 presents the maximum supported values of noise for $\alpha - \gamma$ ranging from 0 to 1.

Fig. 1. Acceptable levels of noise as a function of $\alpha - \gamma$ for the oblivious transfer protocol

Our non-interactive commitment protocol works for $h(\delta) < (\alpha - \gamma)/2$. For $\alpha = 1$ and $\gamma = 0.5$ we have that non-interactive commitments are possible in the noisy memory bounded model if $\delta < 0.041$. Figure 2 presents the maximum supported values of noise for $\alpha - \gamma$ ranging from 0 to 1.

Finally, interactive commitments are possible if $h(\delta) < \alpha - \gamma$. For the same settings ($\alpha = 1$ and $\gamma = 0.5$), this gives us a maximum noise rate of $\delta < 0.11$. Figure 3 presents the maximum supported values of noise for $\alpha - \gamma$ ranging from 0 to 1.

Fig. 2. Acceptable levels of noise as a function of $\alpha - \gamma$ for the non-interactive commitment protocol

Fig. 3. Acceptable levels of noise as a function of $\alpha - \gamma$ for the interactive commitment protocols
IX. Conclusion

In this work we presented the first protocols for commitment and oblivious transfer in the bounded storage model with errors, thus extending the previous results existing in the literature for key agreement [43]. As expected, our protocols work for a limited range of values of the noise parameter \( \delta \). The allowed range for our commitment schemes is different than the one for the OT protocol. For the case of commitment schemes, the range of noise that could be tolerated depended on the round complexity of the proposed protocols; extra rounds helped tolerating a more severe noise.

There are many open questions that follow our results here:

- To prove the impossibility of commitment protocols when \( h(\delta) > \alpha - \gamma \).
- To obtain efficient OT protocols that work for the range of noise achieved by our protocols based on random linear codes.
- What is the best range of noise that can be achieved by non-interactive commitment protocols?
- Is there an intrinsic difference in the level of noise tolerated by bit commitment and OT protocols?

We do conjecture that there exists an intrinsic difference between OT and commitment schemes in the sense that there exist levels of noise so that one of them is possible but not the other. If this conjecture is proven, this would sharply contrast with the noise-free bounded memory model, where there is an all-or-nothing situation: either one has OT and bit commitment or one has nothing. Our main argument in support of this conjecture is the need for error correction in the case of oblivious transfer protocols in the bounded storage model. In the case of commitment protocols error correction is not needed, the main tool used to prevent Alice from cheating is a typicality test.

References


