# Error Correction in The Bounded Storage Model


Yan Zong Ding

College of Computing, Georgia Institute of Technology, 801 Atlantic Drive, Atlanta, Georgia 30332-0280, USA,
ding@cc.gatech.edu



Supported by NSF grant CCR-0205423



**Abstract.** We initiate a study of Maurer's *bounded storage model* (*JoC*, 1992) in presence of transmission errors and perhaps other types of errors that cause different parties to have *inconsistent views of the public random source*. Such errors seem inevitable in any implementation of the model. All previous schemes and protocols in the model assume a perfectly consistent view of the public source from all parties, and do not function correctly in presence of errors, while the private-key encryption scheme of Aumann, Ding and Rabin (*IEEE IT*, 2002) can be extended to tolerate only a $O(1/\log(1/\varepsilon))$ fraction of errors, where $\varepsilon$ is an upper bound on the advantage of an adversary.

In this paper, we provide a general paradigm for constructing secure and error-resilient private-key cryptosystems in the bounded storage model that tolerate a *constant* fraction of errors, and attain the near optimal parameters achieved by Vadhan's construction (*JoC*, 2004) in the errorless case. In particular, we show that any *local fuzzy extractor* yields a secure and error-resilient cryptosystem in the model, in analogy to the result of Lu (*JoC*, 2004) that any local strong extractor yields a secure cryptosystem in the errorless case, and construct efficient local fuzzy extractors by extending Vadhan's sample-then-extract paradigm. The main ingredients of our constructions are *averaging samplers* (Bellare and Rompel, *FOCS '94*), *randomness extractors* (Nisan and Zuckerman, *JCSS*, 1996), *error correcting codes*, and *fuzzy extractors* (Dodis, Reyzin and Smith, *EUROCRYPT '04*).


## 1 Introduction

The *bounded storage model*, introduced by Maurer [Mau92], has seen increasing activities recently. In contrast to the standard complexity-based model for cryptography, this model imposes a bound on the storage space of an adversary rather than its running time. The model does not rely on complexity assumptions, and achieves information-theoretic security by employing a public source emitting random strings whose length exceeds the known space bound of the adversary. The security is guaranteed against a computationally unbounded adversary who stores almost all information about a public random string, while a legitimate user is only required to store a small number of public random bits. In a practical implementation, a good candidate for the public source is a system of high-speed satellites broadcasting random bits at a very high rate.

The bounded storage model has enjoyed success in private-key cryptography [Mau92,CM97b,AR99,ADR02,DR02,DM04b,Lu04,Vad04]. In particular, an important property known as *everlasting security* was observed in [ADR02,DR02], namely the private key can be reused exponentially many times under active attacks, and security is preserved even if after the execution of the protocol, the key is revealed to the adversary and the adversary becomes unbounded in both time and space. Subsequent works [DM04b,Lu04,Vad04] succeeded in constructing highly efficient (in terms of key length and storage requirement) cryptosystems in the model that attain everlasting security, culminating in the near optimal construction of Vadhan [Vad04]. Significant progress has also been made in oblivious transfer [CCM98,Din01,DHRS04] and key agreement [CM97b,DM04a] in the bounded storage model. More recently, it was shown that a primitive known as non-interactive timestamping, which is impossible in standard complexity-based cryptography, can be constructed in the bounded storage model [MST04].

All the above-mentioned works are based an ideal assumption that all the parties have a perfectly consistent view of the public random source. It seems, however, that in any implementation of the bounded storage model, transmission errors and perhaps other types of errors that cause different parties to have *inconsistent views of the public source*, are inevitable. The previous schemes and protocols do not function correctly in presence of such errors. Error-correcting the source might at the first glance appear as a natural solution, however this approach has several disadvantages, and in certain circumstances is infeasible, insufficient, or even impossible: (1) Error-correcting an entire string from the source is infeasible due to its huge size. (2) Encoding the source blockwise does not withstand worst-case adversarial errors that cause too many bits from a same block to be corrupted or erased. Worst-case adversarial errors may at first seem very unnatural. However, considering such errors is necessary, for instance in a setting where a system of several sources is employed, and the adversary compromises a fraction of the sources. (3) The practicality of the bounded storage model is based on the assumption that communications technology allows transmission of data at a rate higher than the storage rate of the adversary. Encoding the source by an error correcting code may significantly slow down the speed of transmission, thereby giving the adversary an advantage in storing information. (4) Error-correcting the source is impossible in implementations which use, for instance, existing natural sources of randomness that cannot be modified. Thus, the ability to cope with errors in the model itself, without an error-corrected source, is natural and fundamental for the bounded storage model.

It was noted by Rabin [Rab02] that the cryptosystem of [ADR02] (the ADR scheme for shorthand), which uses a long private key, can in fact be extended to tolerate a $O(1/\log(1/\varepsilon))$ fraction of errors, where $\varepsilon$ is an upper bound on the advantage of an adversary. Throughout the paper, the error is measured by the maximum relative Hamming distance between the original public source and the source as perceived by a party. The ADR scheme extracts a one-time pad from the source where each bit of the one-time pad is the parity of $O(\log(1/\varepsilon))$ bits of the source at random positions. Thus, if the error in accessing the source is $O(1/\log(1/\varepsilon))$, then with high probability the fraction of corrupted bits in the one-time pad is a constant, and therefore correct decryption can be achieved by error-correcting the message using an asymptotically good error correcting code. It is also easy to see that $O(1/\log(1/\varepsilon))$ is an upper bound on the fraction of errors that can be tolerated by the extended ADR scheme. We note that by a slightly more careful analysis, it can be shown that a similar result also holds for the schemes of Lu [Lu04], which can be viewed as being obtained by derandomizing the ADR scheme.

In this paper, we provide a general paradigm for constructing secure and error-resilient private-key cryptosystems in the bounded storage model that tolerate a *constant* fraction of worst-case errors, and simultaneously attain the near optimal parameters achieved by Vadhan's construction [Vad04] in the errorless case. In particular, we show that any *local fuzzy extractor* yields a secure and error-resilient cryptosystem in the bounded storage model, in analogy to the results of Lu [Lu04] that any local strong extractor yields a secure cryptosystem in the errorless case, and construct efficient local fuzzy extractors by extending Vadhan's sample-then-extract paradigm [Vad04]. Further, for ensuring correct functionality in presence of errors, our cryptosystems only incur a communication overhead that can be made as small as any constant fraction. The main ingredients of our constructions are *averaging samplers* [BR94] and *randomness extractors* [NZ96], two powerful tools from the theory of pseudorandomness that are now standard in bounded-storage cryptography (c.f., [Lu04,Vad04,DHRS04]), as well as *error correcting codes*, and a new primitive known as *fuzzy extractors* recently introduced by Dodis, Reyzin and Smith [DRS04].

Averaging samplers, introduced by Bellare and Rompel [BR94], are procedures that approximate the average of a $[0,1]$-function by taking the average of the function evaluated at sampled points determined by a short random seed. Randomness extractors, introduced by Nisan and Zuckerman [NZ96], are functions that extract near perfect randomness from imperfect random sources using a short random seed. An extractor is *strong* if its output remains near uniform even if the seed is given. See the excellent surveys and tutorials of [NT99,Sha02,Vad02,Gol97] and references therein for constructions, connections, and applications of extractors and samplers.

Recently extractors and averaging samplers have proven fundamental in bounded-storage cryptography. Lu [Lu04] showed that *any* strong extractor yields a secure private-key cryptosystem in the bounded storage model, however due to the huge size of the source, the extractor is required to be *locally computable*, or simply *local*, namely the output of the extractor depends on only a few bits of the source. In [Vad04], Vadhan gave a general *sample-then-extract* paradigm for constructing local extractors from *any* averaging sampler and randomness extractor: first sample a small number of bits from the source using an averaging sampler, then apply an extractor to the sampled bits. By using strong extractors and samplers with near optimal parameters, the construction of [Vad04] yields near optimal local strong extractors.

Fuzzy extractors were introduced by Dodis, Reyzin and Smith [DRS04] recently, motivated by the problem of using biometrics for cryptography. The basic underlying ideas and techniques for constructing such objects have however already been used in the rich literature on information reconciliation and privacy amplification (c.f. [BBR88,BS93,BBCM95,CM97a]). The work of [DRS04] and this work can be seen as revisiting these ideas, using modern terminologies and techniques from the pseudorandomness literature. Informally speaking, a fuzzy extractor is a function which on input $x \stackrel{R}{\leftarrow} X$ where $X$ is an imperfect random source, extracts a near uniform string $Y$ together with a "fingerprint" $P$ using a random seed $K$,[1] such that: (1) $Y$ is near uniform even when given $(K, P)$, and (2) there is a recovery algorithm that recovers $Y$ from $P$, $K$, and any $x'$ "sufficiently close" to $x$. Fuzzy extractors that allow recovery from a constant fraction of errors can be constructed using strong extractors and asymptotically good error correcting codes. ([DRS04]. See also Section 4.4 of this paper.)

### 1.1 An Overview of Our Constructions

We show that any fuzzy extractor yields a secure and error-resilient cryptosystem in the bounded storage model, and construct efficient *local* fuzzy extractors by extending Vadhan's sample-then-extract paradigm. Here the term *local* means that both extraction and recovery depend on a small number of bits from the input source, and further the positions of the bits read for both extraction and recovery are completely determined by the seed $K$ and do not depend on the source $X$. Thus the positions of the bits read can be *preprocessed* using $K$ by a *sampling algorithm.* Therefore we assume that both the extraction algorithm and the recovery algorithm proceed in two phases. In the first phase, both read bits from the source whose positions are determined by the seed. In the second phase, the actual extraction and recovery take place, on the bits read in the first phase along with other inputs. As the local extraction and recovery procedures do not access the entire source, we allow a small recovery error.

*Construction of Local Fuzzy Extractors.* A local fuzzy extractor LFE can be constructed from any given averaging sampler Samp and fuzzy extractor FE with recovery algorithm Rec, as follows. A seed for the resulting LFE is of form $(K_1, K_2)$, where $K_1$ is a random seed for Samp, and $K_2$ is a random seed for FE. For local fuzzy extraction from $X$, one samples $W = X_{\mathrm{Samp}(K_1)}$ from $X$, then computes and outputs $(Y, P) = \mathrm{FE}(W, K_2)$. For local recovery of $Y$ using $P$, $(K_1, K_2)$, and a string $X'$ that is sufficiently close to $X$ in Hamming distance, one samples $W' = X'_{\mathrm{Samp}(K_1)}$ from $X'$, and recovers $Y = \mathrm{Rec}(W', K_2, P)$. The security (or randomness) property of LFE follows from the fact that for almost all seeds $K_1$ of Samp, the sampler Samp essentially preserves the entropy rate of the source $X$ (see [NZ96,Vad04]), and the security property of FE that output

[1] Our definition of a fuzzy extractor differs slightly from the original definition in [DRS04] in that our fuzzy extractor explicitly uses a random seed, whereas that of [DRS04] does not make the seed explicit yet makes it part of the fingerprint.

$Y$ is near uniform even when $K_2$ and $P$ are given. The local recovery property of LFE follows from the recovery property of FE, and the fact that for almost all seeds $K_1$ of Samp, the sampled substrings $X_{\mathrm{Samp}(K_1)}$ and $X'_{\mathrm{Samp}(K_1)}$ essentially preserve the relative Hamming distance between $X$ and $X'$, i.e. the fraction of positions at which $X$ and $X'$ differ. Details of our construction and analysis will be give in Section 4.3. In Sections 4.4, 4.5 and 4.6, we show that by proper choice of the underlying building blocks, our general paradigm yields a local fuzzy extractor that attains the near optimal seed length and sample complexity of Vadhan's strong local extractor, and produces a very short fingerprint needed for recovery from errors.

*Private-Key Encryption from a Local Fuzzy Extractor.* Given a local fuzzy extractor LFE together with a recovery algorithm REC that allows recovery from a constant fraction of errors, as well as a sampling procedure Samp (see the discussion at the beginning of Section 1.1), a *basic one-time* private-key encryption scheme in the bounded storage model that tolerates a constant fraction of errors can be constructed as follows: The sender Alice and the receiver Bob share a private-key $K$ which is a random seed for LFE. While the public random string $X$ is transmitted, Alice computes $(Y, P) = \mathrm{LFE}(X^A, K)$, and Bob samples $W^B = X^B_{\mathrm{Samp}(K)}$ from $X^B$ required for the recovery of $Y$, where $X^A$ and $X^B$ are the views of $X$ as perceived by Alice and Bob respectively. To encrypt a message $M$, Alice computes $C = M \oplus Y$, and sends $(C, P)$ to Bob. Upon receiving $(C, P)$, Bob decrypts by first recovering the one-time pad $Y = \mathrm{REC}(W^B, K, P)$, then computing $M = C \oplus Y$.

Correct decryption (with high probability) of the resulting basic scheme follows directly from the recovery property of a local fuzzy extractor, and its security, in the case that the key $K$ is used *just once* to encrypt one message, follows immediately from the security property of a local fuzzy extractor. However, an important question is *whether the key can be used many times* as in the errorless case, under the attack of an active space-bounded adversary who at each time step is also given the one-time pads and fingerprints from the past.[2] Recall that in the errorless case, the very general results of [Lu04] and [Vad04] show that *any* strong local extractor yields a cryptosystem in which the key is reusable and everlasting security is attained. In contrast, a moment's thought shows that one *cannot* hope to have such an analogous general result for an *arbitrary* local fuzzy extractor in the case of errors! Consider for instance the following (contrived) counter-example. Let LFE be a local fuzzy extractor constructed by the sample-then-extract paradigm described above, which takes as input a source $X$ and a key $K = (K_S, K_E)$, and outputs $(Y, P) = \mathrm{LFE}(X, K) \stackrel{\Delta}{=} \mathrm{FE}(X_{\mathrm{Samp}(K_S)}, K_E)$, where Samp and FE are the given sampler and fuzzy extractor. Let REC be its recovery algorithm. Now let $\widehat{\mathrm{LFE}}$ be obtained by modifying LFE as follows: on input $(X, K)$, $\widehat{\mathrm{LFE}}$ computes $(Y, P) = \mathrm{LFE}(X, K)$, but outputs $(Y, P')$ where $P' = P \circ K_S$ is the concatenation of $P$ and $K_S$. Let $\widehat{\mathrm{REC}}$ be the same as

[2] The fingerprints are sent in the clear and are thus public to anyone, while the past one-time pads can be obtained by a chosen plaintext or chosen ciphertext attack.

REC, except that $\widehat{\mathrm{REC}}$ uses only $|P|$ bits of the fingerprint $P'$. It is not hard to see that the resulting $\widehat{\mathrm{LFE}}$ is a local fuzzy extractor with recovery algorithm $\widehat{\mathrm{REC}}$: As LFE is a local fuzzy extractor, by definition $Y$ is near uniform even when given $(K, P)$, and thus is also near uniform when given $(K, P \circ K_S)$.[3] The security property of $\widehat{\mathrm{LFE}}$ follows. The recovery property, i.e. the correctness of $\widehat{\mathrm{REC}}$ is obvious. However, if $\widehat{\mathrm{LFE}}$ is employed in the above construction of a private-key encryption scheme, then from a fingerprint $P'$ from a past time period the adversary gets $K_S$, the part of the key used for sampling. If the same key $K = (K_S, K_E)$ is reused, then from this point on, just as the sender and receiver the adversary need only store a small number of bits from the source as specified by $K_S$, and when he obtains $K_E$ later he can simply decrypt just as the receiver. In general, the fingerprint $P$ and the seed $K$ are *dependent*. The definition of a local fuzzy extractor only guarantees that its first output $Y$ is nearly uniform and independent of $(K, P)$. The dependence between $K$ and $P$ renders a generic local fuzzy extractor non-reusable in this context, as the fingerprint $P$, sent in the clear, could give information about the seed $K$.

Note that the above counter-example only shows that a generic local fuzzy extractor does not yield a stateless cryptosystem with a reusable key, and does not answer the question whether the sample-then-extract paradigm, with a general averaging sampler and (non-local) fuzzy extractor, results in such a system. We believe that the answer to the latter question is also negative, for the following reason. First, it can be seen that if the sampled substring $W = X_{\mathrm{Samp}(K_S)}$ were given, then an adversary who stores sufficient information about the source $X$ and has the capability to introduce sufficient errors to $X$, could obtain substantial information about the seed $K_S$ from $W$ and his state. The fingerprint $P$ is a function of $W$ and thus gives partial information about $W$, which together with the adversary's state, may give adequate information about $K_S$.

However, we do note that a stateless encryption scheme under the sample-then-extract paradigm with a reusable key would result from a stronger type of fuzzy extractors, called entropically secure fuzzy extractors recently introduced by Dodis and Smith (see [Smi04]), which would result in a local fuzzy extractor where $(K, Y)$ is essentially uniformly random even conditioned on the fingerprint $P$. Yet, the current constructions of entropically secure fuzzy extractors are not randomness-efficient enough to yield a desired value for key length.

Is there still any hope of using a generic local fuzzy extractor to construct a full-fledged error-resilient encryption scheme, where many messages can be encrypted? The answer is yes, if encryption and decryption are allowed to maintain a state. We circumvent the difficulty described above by *refreshing the key, instead of reusing it*, as follows. Let LFE be an arbitrary local fuzzy extractor, and let Alice and Bob share an initial key $K_1$. At each time $t$, we use the given local fuzzy extractor to extract a few more bits that will be used as the key for time $t+1$. That is, at time $t$, Alice computes $((Y_t^A, K_{t+1}^A), P_t) = \mathrm{LFE}(X^A, K_t^A)$, where $K_t^A$ is Alice's key for time $t$, $Y_t^A$ is Alice's one-time pad for encrypting a

[3] More generally, for any function $f$, $Y$ is near uniform even when given $(K, f(P, K))$.

(single) message at time $t$, and $K_{t+1}^A$ is the new key Alice uses for time $t+1$. The fingerprint $P_t$ is used by Bob to recover $(Y_t^B, K_{t+1}^B)$, where $Y_t^B$ and $K_{t+1}^B$ are respectively Bob's one-time pad for decrypting a ciphertext at time $t$, and Bob's new key for time $t+1$. Ideally we would like to have $(Y_t^A, K_{t+1}^A) = (Y_t^B, K_{t+1}^B)$, although a small recovery error is inevitable. Intuitively, the resulting encryption scheme is secure as the new key $K_{t+1}^A$ is a part of the first output of LFE, which by definition is near uniform given $(K_t^A, P_t)$. Had there been no error from the source, security would have followed from known results [Vad04,Lu04]. The presence of error however does complicate the matter quite substantially, and a careful analysis of security and error-resilience is necessary.

Thus unlike the previous schemes in which the same key is reused, this scheme updates the key at each time step in a *forward-secure* manner (c.f. [And97]), and is therefore *stateful*. Such state-dependence may be viewed as a drawback in some cases. However, in communication settings where communication devices do maintain much state information (e.g. session IDs and counters), such a stateful encryption scheme is reasonable. On the other hand, it remains an interesting problem to construct a stateless error-resilient scheme matching the near optimal parameters achieved by the stateful construction. However, the general negative result described above suggests that resolving this issue may require resorting to and analyzing particular constructions of the building blocks, such as the underlying error correcting code. One promising approach is to derandomize the construction of entropically secure fuzzy extractors in [Smi04].

In Section 3 we carefully define the bounded storage model with errors, and give a definition of security and error-resilience. In Section 4.2 (Theorem 1), we will show that under the general forward-secure paradigm described above, *any* local fuzzy extractor yields a secure encryption scheme that achieves desired security and error correction properties simultaneously. More precisely, both the adversary's advantage and the probability of a single recovery error in the first $T$ time periods, grow only linearly with $T$, essentially the best one can hope.

## 2 Preliminaries

We use the following standard notations in this paper. For a random variable $X$, the notation $x \stackrel{R}{\leftarrow} X$ denotes that $x$ is chosen according to $X$. For a set $S$, $x \stackrel{R}{\leftarrow} S$ denotes that $x$ is chosen uniformly from $S$. For an integer $n$, we denote by $U_n$ a uniformly distributed random variable on the set $\{0,1\}^n$, and denote by $[n]$ the set $\{1, \dots, n\}$. For a string $x \in \{0,1\}^n$ and a subset $S = \{i_1, \dots, i_l\} \subseteq [n]$, $x_S \stackrel{\Delta}{=} x_{i_1} \dots x_{i_l}$, where $x_i$ is the $i$-th bit of $x$. We denote by $\mathrm{Supp}(X)$ the support of a random variable $X$.

For two strings $x$ and $y$ of the same length, we use $\Delta(x, y)$ to denote their Hamming distance, i.e. the number of bit positions at which $x$ and $y$ differ.

We say that a function (e.g. an extractor, a sampler, or an error correcting code) is explicit if it can be computed by a polynomial-time algorithm.

In the remainder of this section, we give definitions of weak random sources and statistical distance.

**Definition 1 ([CG88,Zuc96]).** *For a random variable $X$ on a finite set $\Omega$, the* min-entropy *of $X$ is defined by:* $\mathrm{H}_\infty(X) = \min_{x\in\Omega} \log(1/\Pr[X = x])$. *We say that $X$ is a $k$*-source *if* $\mathrm{H}_\infty(X) \geq k$. *We say that a random variable $X$ over $\{0,1\}^n$ has* entropy rate $\alpha$ *if $X$ is an $\alpha n$-source.*

**Definition 2.** *For random variables $X$ and $Y$ taking values in $\Omega$, their* statistical distance *is defined as* $\mathrm{SD}(X,Y) \stackrel{\Delta}{=} \max_{A\subseteq\Omega} |\Pr[X \in A] - \Pr[Y \in A]| = \frac{1}{2}\sum_{x\in\Omega} |\Pr[X = x] - \Pr[Y = x]|$. *We say $X$ and $Y$ are* $\varepsilon$-close *if* $\mathrm{SD}(X,Y) \leq \varepsilon$.

# 3 The Model and Definition of Security

In this section we take a closer look at the bounded storage model *with errors*, and define security in the model. In the presentation we use many terminologies and notations from [Vad04].

*The Public Random Source.* The bounded storage model (BSM) employs a public source of random strings, each of length exceeding the storage bound of the adversary. Throughout the paper, we use $N$ to denote the length of a public random string. The public source is thus modeled as a sequence of random variables $X_1, X_2, \ldots, X_t, \ldots$, each distributed over $\{0,1\}^N$. We denote by $\beta N$ the storage bound, where $\beta < 1$ is constant fraction, and call $\beta$ the *storage rate* of the adversary.

The original work of Maurer [Mau92], as well as some early works (c.f., [AR99,ADR02,DR02]) assume that the public source is perfectly random, that is, each $X_t$ is uniformly distributed and independent of the others. It was noted in [Lu04,Vad04] that each $X_t$ need not be uniform, and it is sufficient (and necessary) that each $X_t$ has entropy rate $\alpha > \beta$. Moreover, it was pointed out in [Vad04] that the $X_t$'s need not be independent, and it is sufficient (and necessary) that the sequence of random variables $X_1, X_2, \ldots, X_t, \ldots$ form a *reverse block source*, which is the Chor-Goldreich [CG88] notion of a block source but backwards in time. Namely, in a reverse block source, each $X_t$ has sufficient min-entropy conditioned on the future, whereas in a standard block source of [CG88] each $X_t$ has sufficient min-entropy conditioned on the past. For the model with errors considered in this paper, we slightly strengthen the requirement on the public source by postulating that it be blockwise *both forward and backward*, i.e. it be both a standard block source and a reverse block source. The reason for imposing this forward blockwise structure in addition to its reverse counterpart is that the fingerprints $P_1, \ldots, P_{t-1}$ required for recovery from errors in the past time periods are exposed and depend on the $X_1, \ldots, X_{t-1}$. Therefore it is necessary that $X_t$ has sufficient min-entropy conditioned on $P_1, \ldots, P_{t-1}$. This would certainly be satisfied if the source is (forward) blockwise, that is $X_t$ has sufficient min-entropy conditioned on $X_1, \ldots, X_{t-1}$.

**Definition 3.** *Let $(X_t) = (X_1, X_2, \ldots)$ be a sequence of random variables each distributed over $\{0,1\}^n$. For each $t \in \mathbb{N}$, denote $X\backslash_t = (X_1, \ldots, X_{t-1}, X_{t+1}, X_{t+2}, \ldots)$. We say that $(X_t)$ is a* two-way block source of entropy rate $\alpha$ *if for every $t \in \mathbb{N}$,* *and every* $\boldsymbol{x} = (x_1, \ldots, x_{t-1}, x_{t+1}, x_{t+2}, \ldots) \in \mathrm{Supp}(X\backslash_t)$*, the random variable* $X_t|_{X\backslash_t = \boldsymbol{x}}$ *is an* $\alpha n$*-source.*

Intuitively, this means that $X_t$ has $\alpha n$ bits of information that can not be predicted from the past and will be forgotten in the future. In the special case of $\alpha = 1$, $X_1, X_2, \ldots$ are uniform and independent.

*BSM Randomness Extraction.* An essential ingredient is a *bounded storage model randomness extraction scheme*,[4] or simply a *BSM extraction scheme*. In the errorless case, such an extraction scheme is a function of the form $\mathrm{EXT} : \{0,1\}^N \times \{0,1\}^d \to \{0,1\}^m$. In a private-key setting, such an extraction scheme is typically used as follows. A seed or a key $K \stackrel{R}{\leftarrow} \{0,1\}^d$ is chosen, and shared between two parties. At time $t$, the parties extract a common string $Y_t = \mathrm{EXT}(X_t, K)$, while the adversary $A$ updates and stores his state $S_t = A(S_{t-1}, Y_1, \ldots, Y_{t-1}, X_t)$, where $|S_t| = \beta N$. The scheme EXT is secure if for every adversary $A$ with storage rate $\beta$, $Y_t$ is statistically close to uniform even when given the key $K$, all the previous $Y_1, \ldots, Y_{t-1}$, the adversary's state $S_t$, and the future public random strings $X_{t+1}, X_{t+2}, \ldots$.

In order to be used as a BSM primitive, the extraction scheme needs to be *locally computable*, that is $\mathrm{EXT}(X, K)$ depends only on a few bits of $X$ whose positions are completely determined by $K$. As in the discussion on local fuzzy extractors in Section 1.1, here we also assume that a sampling procedure Samp precomputes positions $\mathrm{Samp}(K)$, the bits $W = X_{\mathrm{Samp}(K)}$ are read when $X$ is transmitted, and the extraction algorithm EXT actually takes $W$ and $K$ as input, and computes $\mathrm{EXT}(W, K)$.

*Incorporating Errors.* We now incorporate errors into the model, and consider the case where two parties Alice and Bob have inconsistent views of the source as a result of errors. We consider *error-resilient BSM randomness extraction with forward security*, as motivated in Section 1.1 of the Introduction. Such an extraction scheme is a pair of algorithms (EXT, REC), where $\mathrm{EXT} : \{0,1\}^N \times \{0,1\}^d \to \{0,1\}^{m+d} \times \{0,1\}^\ell$ is a *local* extraction function, and $\mathrm{REC} : \{0,1\}^N \times \{0,1\}^d \times \{0,1\}^\ell \to \{0,1\}^{m+d}$ is a *local* recovery algorithm. The second output of EXT is a fingerprint that enables recovery of its first output, and the last $d$ bits from the first output of EXT will be used as the key for the next time period. Alice and Bob initially share a common random key $K_1 \stackrel{R}{\leftarrow} \{0,1\}^d$ for EXT. Let $K_1^A = K_1^B = K_1$.

We model errors by having an *unbounded* adversary who at time $t$ for each $t \in \mathbb{N}$, on input $x_t \stackrel{R}{\leftarrow} X_t$, computes $x_t^A$ and $x_t^B$ such that $\Delta(x_t, x_t^A) \le \delta N$ and $\Delta(x_t, x_t^B) \le \delta N$, where $\delta < 1$ is a constant fraction, and sends $x_t^A$ and $x_t^B$ to Alice and Bob respectively. We call $\delta$ the *error rate*.

[4] In [Vad04], such a scheme is called a BSM pseudorandom generator. We choose to call it a BSM randomness extraction scheme, because of the usual computational connotations of "pseudorandom generators".

On input the corrupted string $x_t^A$ and her key $K_t^A$ for time $t$, Alice computes $((Y_t^A, K_{t+1}^A), P_t) = \mathrm{EXT}(x_t^A, K_t^A)$, where $Y_t^A$ is Alice's extracted "one-time pad" for time $t$, $K_{t+1}^A$ is Alice's key for time $t+1$, and $P_t$ is a fingerprint needed by Bob to recover $(Y_t^A, K_{t+1}^A)$. Meanwhile, on input $x_t^B$, Bob reads the substring $w$ of $x_t^B$ at positions in $\mathrm{Samp}(x_t^B, K_t^B)$ needed to recover $(Y_t^A, K_{t+1}^A)$, where Samp is the sampling procedure. Upon receiving $P_t$ from Alice, Bob recovers $(Y_t^B, K_{t+1}^B) = \mathrm{REC}(w, K_t^B, P_t)$. Ideally we would like to have $(Y_t^B, K_{t+1}^B) = (Y_t^A, K_{t+1}^A)$, although we allow a small recovery error which is inevitable.

As in [Vad04], we use $S_t \in \{0,1\}^{\beta N}$ to denote the state of the adversary at time $t$. For a sequence $Z_1, Z_2, \ldots$, we use the shorthand $Z_{[a,b]} = (Z_a, Z_{a+1}, \ldots, Z_b)$, and $Z_{[a,\infty)} = (Z_a, Z_{a+1}, \ldots)$. At time $t$, we allow the adversary access to the current corrupted strings $x_t^A, x_t^B$, all previous one-time pads $Y_{[1,t-1]}^A, Y_{[1,t-1]}^B$ and keys $K_{[1,t-1]}^A, K_{[1,t-1]}^B$ of *both* Alice and Bob, as well as $P_{[1,t-1]}$. With this information the adversary computes

$$S_t = \mathcal{A}(Y_{[1,t-1]}^A, Y_{[1,t-1]}^B, K_{[1,t-1]}^A, K_{[1,t-1]}^B, P_{[1,t-1]}, S_{t-1}, x_t^A, x_t^B),$$

with $|S_t| = \beta N$.

We now define the security and error correction properties of an error-resilient BSM randomness extraction scheme. In doing so, we use a *real-vs-ideal* paradigm as [Vad04] does.

The real experiment is a real execution of a protocol. For $T \in \mathbb{N}$, the output of our real experiment is $(Y_{[1,T]}^A, Y_{[1,T]}^B, K_{[1,T+1]}^A, K_{[1,T+1]}^B, P_{[1,T]}, S_T, X_{[T+1,\infty)})$, with each component defined above. The ideal experiment is a simulated execution of the protocol in an ideal setting that guarantees security.

In our ideal experiment, for each $t \in [T]$, we choose a uniform one-time pad $Y_t \xleftarrow{R} \{0,1\}^m$, and set $Y_t^A = Y_t^B = Y_t$. Similarly, for each $t \in [T+1]$, we choose a uniform key $K_t \xleftarrow{R} \{0,1\}^d$, and set $K_t^A = K_t^B = K_t$. Thus, in the output of the ideal experiment, each of $Y_{[1,T]}$ and $Y_{[1,T+1]}$ is *uniformly and independently* chosen, and further each $Y_t$ and $K_t$ are *replicated twice* to simulate $Y_t^A, Y_t^B$ and $K_t^A, K_t^B$ respectively, *as if there is no recovery error*. Hence proving security amounts to proving that the outputs of the real and ideal experiments are indistinguishable.

We now precisely define the real and ideal experiments. For both experiments, let $X_1, X_2, \ldots$ be the public random source, let $K_1 \xleftarrow{R} \{0,1\}^d$ be the initial shared key, let $K_1^A = K_1^B = K_1$, $S_0 = 0^{\beta N}$, and let $\mathcal{A}$ be the adversary's algorithm.

*Real Experiment:*

- For $t = 1, \ldots, T$: On $x_t \xleftarrow{R} X_t$:
  Let $(x_t^A, x_t^B) = \mathcal{A}(x_t, Y_{[1,t-1]}^A, Y_{[1,t-1]}^B, K_{[1,t-1]}^A, K_{[1,t-1]}^B, P_{[1,t-1]}, S_{t-1})$, where $\Delta(x_t^A, x_t) \leq \delta N$ and $\Delta(x_t^B, x_t) \leq \delta N$. In this step we allow $\mathcal{A}$ to be unbounded in both time and space.
  Let $((Y_t^A, K_{t+1}^A), P_t) = \mathrm{EXT}(x_t^A, K_t^A)$, and $(Y_t^B, K_{t+1}^B) = \mathrm{REC}(x_t^B, K_t^B, P_t)$.
  Let $S_t = \mathcal{A}(x_t^A, x_t^B, Y_{[1,t-1]}^A, Y_{[1,t-1]}^B, K_{[1,t-1]}^A, K_{[1,t-1]}^B, P_{[1,t-1]}, S_{t-1}) \in \{0,1\}^{\beta N}$.
- Output $Z_T^{\mathrm{real}} = (Y_{[1,T]}^A, Y_{[1,T]}^B, K_{[1,T+1]}^A, K_{[1,T+1]}^B, P_{[1,T]}, S_T, X_{[T+1,\infty)})$.

*Ideal Experiment:*

- For $t = 1, \ldots, T$: On $x_t \xleftarrow{R} X_t$:
  Let $(x_t^A, x_t^B) = \mathcal{A}(x_t, Y_{[1,t-1]}, Y_{[1,t-1]}, K_{[1,t-1]}, K_{[1,t-1]}, P_{[1,t-1]}, S_{t-1})$, where $\Delta(x_t^A, x_t) \leq \delta N$ and $\Delta(x_t^B, x_t) \leq \delta N$.
  Let $((\tilde{Y}_t, \tilde{K}_{t+1}), P_t) = \mathrm{EXT}(X_t^A, K_t)$.
  Choose uniformly and independently $Y_t \xleftarrow{R} \{0,1\}^m$ and $K_{t+1} \xleftarrow{R} \{0,1\}^d$.
  Let $S_t = \mathcal{A}(x_t^A, x_t^B, Y_{[1,t-1]}, Y_{[1,t-1]}, K_{[1,t-1]}, K_{[1,t-1]}, P_{[1,t-1]}, S_{t-1}) \in \{0,1\}^{\beta N}$.
- Output $Z_T^{\mathrm{ideal}} = (Y_{[1,T]}, Y_{[1,T]}, K_{[1,T+1]}, K_{[1,T+1]}, P_{[1,T]}, S_T, X_{[T+1,\infty)})$.

*Notation:* From now on, we denote by $X_t^A$ and $X_t^B$ the induced sources at time $t$ as perceived by Alice and Bob after errors are introduced to $X_t$.

**Definition 4.** *A BSM randomness extraction scheme* (EXT, REC) *is* $\varepsilon$-secure for storage rate $\beta$, entropy rate $\alpha$, and error rate $\delta$ *if for every two-way block source* $(X_t)$ *of entropy rate* $\alpha$*, every adversary* $\mathcal{A}$ *with storage rate* $\beta$*, every means to introduce a* $\delta$*-fraction of errors to the source* $(X_t)$*, and every* $T \in \mathbb{N}$, $\mathrm{SD}(Z_T^{\mathrm{real}}, Z_T^{\mathrm{ideal}}) \leq T \cdot \varepsilon$*, where* $Z_T^{\mathrm{real}}$ *and* $Z_T^{\mathrm{ideal}}$ *are the outputs of the Real and Ideal Experiments respectively.*

We say that (EXT, REC) is $t$-*local* if for every key $K \in \{0,1\}^d$, both the extraction scheme $\mathrm{EXT}(x, K)$ and its recovery algorithm $\mathrm{REC}(x', K, P)$ depend on only $t$-bits of $x$ and $x'$ respectively, whose positions are completely determined by the key $K$.

We refer readers to the remarks after Definition 3.2 of [Vad04] for a discussion on the definition of everlasting security in the errorless model, which apply to the model with errors as well. Below are some more remarks that are important.

*Remarks:*

- A reader may notice that we have not explicitly defined the error correction property of a BSM randomness extraction scheme. However, by a careful inspection, it is not hard to see that *the security property as defined in Definition 4 implies error correction*. That is, if (EXT, REC) is $\varepsilon$-secure, then for every two-way block source $(X_t)$ of entropy rate $\alpha$, for error rate $\delta$, and every $T \in \mathbb{N}$, the probability of a single recovery error in the first $T$ time periods in the Real Experiment, is at most $T\varepsilon$. More precisely, with probability at least $1 - T\varepsilon$ (over the source $(X_t)$ and the initial common key $K_1 \xleftarrow{R} \{0,1\}^d$), we have that for each $t \in [T]$, $(Y_t^A, K_{t+1}^A) = (Y_t^B, K_{t+1}^B)$, where $((Y_t^A, K_{t+1}^A), P_t) = \mathrm{EXT}(X_t^A, K_t^A)$, and $(Y_t^B, K_{t+1}^B) = \mathrm{REC}(X_t^B, K_t^B, P_t)$. This is because in the output $Z_T^{\mathrm{ideal}}$ of the Ideal Experiment, each $Y_t$ and $K_t$ are *replicated* twice. Thus if the probability of a recovery error in the first $T$ time periods in the Real Experiment is greater than $T\varepsilon$, then the distinguisher that simply compares the corresponding components of the two inputs, and outputs 1 if and only if they are the same, distinguishes between $Z_T^{\mathrm{real}}$ and $Z_T^{\mathrm{ideal}}$ with an advantage greater than $T\varepsilon$, contradicting the $\varepsilon$-security of (EXT, REC).

– From Definition 4, it is clear that the output $Y_t$ of an error-resilient BSM extraction scheme can be used in place of a truly random string at time $t$ for general cryptographic purposes. In particular, using each $Y_t$ as a one-time pad for time $t$, such an extraction scheme yields an error-resilient BSM private-key encryption scheme *secure against chosen plaintext attacks and chosen ciphertext attacks* (c.f. [NY90]), with a small decryption error.

# 4 Local Fuzzy Extractors and BSM Extraction

In this section, we construct local fuzzy extractors and error-resilient BSM randomness extraction schemes.

## 4.1 Local Fuzzy Extractors

First we define fuzzy extractors, which were recently introduced by Dodis, Reyzin and Smith [DRS04]. We slightly modify the original definition in [DRS04] to suit our application.

**Definition 5 ([DRS04] - modified).** *A* $(k, \varepsilon, \delta, \gamma)$-fuzzy extractor *is a pair* $\mathrm{FE} = (\mathrm{EXT}, \mathrm{REC})$ *of algorithms, where* $\mathrm{EXT} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m \times \{0,1\}^\ell$ *is an extraction algorithm and* $\mathrm{REC} : \{0,1\}^n \times \{0,1\}^d \times \{0,1\}^\ell \to \{0,1\}^m$ *is a recovery algorithm satisfying*

– **(Security)** *For every $k$-source $X$, $(K, Y, P)$ is $\varepsilon$-close to $(K, U_m, P)$, where $(Y, P) = \mathrm{EXT}(X, K)$, $K \xleftarrow{R} \{0,1\}^d$ is a uniformly chosen seed independent of $X$, and $U_m$ is independent of $K$ and $P$.*
– **(Recovery)** *For every $x, x' \in \{0,1\}^n$ with $\Delta(x, x') \le \delta n$, $\Pr[\mathrm{REC}(x', K, P) = Y] \ge 1 - \gamma$, where $(Y, P) = \mathrm{EXT}(x, K)$, and the probability is taken over $K \xleftarrow{R} \{0,1\}^d$.*

A fuzzy extractor $\mathrm{FE} = (\mathrm{EXT}, \mathrm{REC})$ is *$t$-local* if for every seed $r \in \{0,1\}^d$, both the extraction algorithm $\mathrm{EXT}(x, r)$ and the recovery algorithm $\mathrm{REC}(x', r, p)$ depend on only $t$-bits of $x$ and $x'$ respectively, whose positions are completely determined by the seed $r$.

## 4.2 Error-Resilient BSM Extraction from Local Fuzzy Extractors

The following main theorem of this paper states that any $t$-local fuzzy extractor yields a $t$-local error-resilient BSM randomness extraction scheme.

**Theorem 1.** *For every $t \in \mathbb{N}$, if $\mathrm{LFE} = (\mathrm{EXT}, \mathrm{REC})$ is a $t$-local $(k, \varepsilon, 2\delta, \gamma)$-fuzzy extractor for $\gamma < 1/2$ and $k = (\alpha - \beta - \mathrm{H}(\delta))N - \log(1/\varepsilon)$, where $\mathrm{H}(\delta) \stackrel{\Delta}{=} -\delta \log \delta - (1-\delta)\log(1-\delta)$ is the binary entropy function, and $\mathrm{EXT}$ is of the form $\mathrm{EXT} : \{0,1\}^N \times \{0,1\}^d \to \{0,1\}^{m+d} \times \{0,1\}^\ell$, then $\mathrm{LFE}$ is a $t$-local $4(\varepsilon + \gamma)$-secure BSM randomness extraction scheme for storage rate $\beta$, entropy rate $\alpha$, and error rate $\delta$.*

*Proof.* (Sketch) Let LFE = (EXT, REC) be a $(k, \varepsilon, 2\delta, \gamma)$-fuzzy extractor where $k = (\alpha - \beta - \mathrm{H}(\delta))N - \log(1/\varepsilon)$, $\gamma < 1/2$, and EXT is of the form EXT : $\{0,1\}^N \times \{0,1\}^d \to \{0,1\}^{m+d} \times \{0,1\}^\ell$. We prove the theorem by induction on $T$. The proof builds on the framework developed in [Vad04,Lu04].

As in [Vad04], we use superscripts to distinguish between random variables in the Real Experiment and the Ideal Experiment, e.g. $K_t^{\mathrm{real}}$ vs. $K_t^{\mathrm{ideal}}$. We prove by induction on $T$ that for every $T$, the random variable

$$Z_T^{\mathrm{real}} = (Y_{[1,T]}^A, Y_{[1,T]}^B, K_{[1,T+1]}^A, K_{[1,T+1]}^B, P_{[1,T]}^{\mathrm{real}}, S_T^{\mathrm{real}}, X_{[T+1,\infty)})$$

is $T \cdot 4(\varepsilon + \gamma)$-close to

$$Z_T^{\mathrm{ideal}} = (Y_{[1,T]}^{\mathrm{ideal}}, Y_{[1,T]}^{\mathrm{ideal}}, K_{[1,T+1]}^{\mathrm{ideal}}, K_{[1,T+1]}^{\mathrm{ideal}}, P_{[1,T]}^{\mathrm{ideal}}, S_T^{\mathrm{ideal}}, X_{[T+1,\infty)}),$$

where $Z_T^{\mathrm{real}}$ and $Z_T^{\mathrm{ideal}}$ are the output of the Real Experiment and the Ideal Experiment respectively.

Recall that for each $t$, $Y_t^{\mathrm{ideal}} \equiv U_m^{(t)}$ and $K_t^{\mathrm{ideal}} \equiv U_d^{(t)}$, where $U_m^{(t)}$ (resp. $U_d^{(t)}$) is an independent copy of $U_d$ (resp. $U_m$). Note again that each $Y_t^{\mathrm{ideal}}$ and $K_t^{\mathrm{ideal}}$ are replicated twice in $Z_T^{\mathrm{ideal}}$.

As the induction hypothesis, suppose that $Z_{T-1}^{\mathrm{real}}$ and $Z_{T-1}^{\mathrm{ideal}}$ are $(T-1) \cdot 4(\varepsilon + \gamma)$-close. It follows from the definition of the Real Experiment that $Z_T^{\mathrm{real}}$ is obtained from $Z_{T-1}^{\mathrm{real}}$ by applying the function $f_T$ that:

- Computes
  $(X_T^A, X_T^B) = \mathcal{A}(X_T, Y_{[1,T-1]}^A, Y_{[1,T-1]}^B, K_{[1,T-1]}^A, K_{[1,T-1]}^B, P_{[1,T-1]}^{\mathrm{real}}, S_{T-1}^{\mathrm{real}})$,
  where $\Delta(X_T^A, X_T) \leq \delta N$ and $\Delta(X_T^B, X_T) \leq \delta N$.
- Computes $((Y_T^A, K_{T+1}^A), P_T^{\mathrm{real}}) = \mathrm{EXT}(X_T^A, K_T^A)$,
  and $(Y_T^B, K_{T+1}^B) = \mathrm{REC}(X_T^B, K_T^B, P_T^{\mathrm{real}})$.
- Updates state
  $S_T^{\mathrm{real}} = \mathcal{A}(X_T^A, X_T^B, Y_{[1,T-1]}^A, Y_{[1,T-1]}^B, K_{[1,T-1]}^A, K_{[1,T-1]}^B, P_{[1,T-1]}^{\mathrm{real}}, S_{T-1}^{\mathrm{real}}) \in \{0,1\}^{\beta N}$ .
- Removes $X_T$.
- Outputs $Z_T^{\mathrm{real}} = (Y_{[1,T]}^A, Y_{[1,T]}^B, K_{[1,T+1]}^A, K_{[1,T+1]}^B, P_{[1,T]}^{\mathrm{real}}, S_T^{\mathrm{real}}, X_{[T+1,\infty)})$.

Applying the same function $f_T$ to $Z_{T-1}^{\mathrm{ideal}}$, we get the random variable $f_T(Z_{T-1}^{\mathrm{ideal}})$ as follows:

- Let $(X_T^A, X_T^B) = \mathcal{A}(X_T, Y_{[1,T-1]}^{\mathrm{ideal}}, Y_{[1,T-1]}^{\mathrm{ideal}}, K_{[1,T-1]}^{\mathrm{ideal}}, K_{[1,T-1]}^{\mathrm{ideal}}, P_{[1,T-1]}^{\mathrm{ideal}}, S_{T-1}^{\mathrm{ideal}})$,
  where $\Delta(X_T^A, X_T) \leq \delta N$ and $\Delta(X_T^B, X_T) \leq \delta N$.
- Let $((\tilde{Y}_T^A, \tilde{K}_{T+1}^A), P_T^{\mathrm{ideal}}) = \mathrm{EXT}(X_T^A, K_T^{\mathrm{ideal}})$,
  and $(\tilde{Y}_T^B, \tilde{K}_{T+1}^B) = \mathrm{REC}(X_T^B, K_T^{\mathrm{ideal}}, P_T^{\mathrm{ideal}})$.
- Update state $S_T^{\mathrm{ideal}} =$
  $\mathcal{A}(X_T^A, X_T^B, Y_{[1,T-1]}^{\mathrm{ideal}}, Y_{[1,T-1]}^{\mathrm{ideal}}, K_{[1,T-1]}^{\mathrm{ideal}}, K_{[1,T-1]}^{\mathrm{ideal}}, P_{[1,T-1]}^{\mathrm{ideal}}, S_{T-1}^{\mathrm{ideal}}) \in \{0,1\}^{\beta N}$.
- Remove $X_T$.
- Output $f(Z_{T-1}^{\mathrm{ideal}}) =$
  $(Y_{[1,T-1]}^{\mathrm{ideal}}, \tilde{Y}_T^A, Y_{[1,T-1]}^{\mathrm{ideal}}, \tilde{Y}_T^B, K_{[1,T]}^{\mathrm{ideal}}, \tilde{K}_{T+1}^A, K_{[1,T]}^{\mathrm{ideal}}, \tilde{K}_{T+1}^B, P_{[1,T]}^{\mathrm{ideal}}, S_T^{\mathrm{ideal}}, X_{[T+1,\infty)})$.

Therefore the *only places* where $f(Z_{T-1}^{\text{ideal}})$ and $Z_T^{\text{ideal}}$ differ are $Y_T^{\text{ideal}}$ vs. $\tilde{Y}_T^A$, $Y_T^{\text{ideal}}$ vs. $\tilde{Y}_T^B$, $K_{T+1}^{\text{ideal}}$ vs. $\tilde{K}_{T+1}^A$, and $K_{T+1}^{\text{ideal}}$ vs. $\tilde{K}_{T+1}^B$.

Since $Z_T^{\text{real}} = f_T(Z_{T-1}^{\text{real}})$, and $\mathrm{SD}(Z_{T-1}^{\text{real}}, Z_{T-1}^{\text{ideal}}) \le (T-1) \cdot 4(\varepsilon + \gamma)$, by basic properties of statistical distance, we have

$$\begin{aligned}
\mathrm{SD}(Z_T^{\text{real}}, Z_T^{\text{ideal}}) &= \mathrm{SD}(f_T(Z_{T-1}^{\text{real}}), Z_T^{\text{ideal}}) \\
&\le \mathrm{SD}(f_T(Z_{T-1}^{\text{real}}), f_T(Z_{T-1}^{\text{ideal}})) + \mathrm{SD}(f_T(Z_{T-1}^{\text{ideal}}), Z_T^{\text{ideal}}) \\
&\le \mathrm{SD}(Z_{T-1}^{\text{real}}, Z_{T-1}^{\text{ideal}}) + \mathrm{SD}(f_T(Z_{T-1}^{\text{ideal}}), Z_T^{\text{ideal}}) \\
&\le (T-1) \cdot 4(\varepsilon + \gamma) + \mathrm{SD}(f_T(Z_{T-1}^{\text{ideal}}), Z_T^{\text{ideal}}).
\end{aligned}$$

Thus to prove that $\mathrm{SD}(Z_T^{\text{real}}, Z_T^{\text{ideal}}) \le T \cdot 4(\varepsilon + \gamma)$, it suffices to show that $\mathrm{SD}(f_T(Z_{T-1}^{\text{ideal}}), Z_T^{\text{ideal}}) \le 4(\varepsilon + \gamma)$.

Let

$$Z'_T \triangleq f_T(Z_{T-1}^{\text{ideal}}) \backslash (\tilde{Y}_T^B, \tilde{K}_{T+1}^B),$$

that is, obtained from $f_T(Z_{T-1}^{\text{ideal}})$ by removing $\tilde{Y}_T^B$ and $\tilde{K}_{T+1}^B$. Let

$$Z''_T \triangleq (Y_{[1,T]}^{\text{ideal}}, Y_{[1,T-1]}^{\text{ideal}}, K_{[1,T+1]}^{\text{ideal}}, K_{[1,T]}^{\text{ideal}}, P_{[1,T]}^{\text{ideal}}, S_T^{\text{ideal}}, X_{[T+1,\infty]})$$

be obtained from $Z_T^{\text{ideal}}$ by the same procedure, that is, by removing the second $Y_T^{\text{ideal}}$ and the second $K_{T+1}^{\text{ideal}}$ from $Z_T^{\text{ideal}}$. Thus, $Z'_T$ and $Z''_T$ are respectively $f_T(Z_{T-1}^{\text{ideal}})$ and $Z_T^{\text{ideal}}$ without simulating Bob's recovery of $(Y_T^B, K_{T+1}^B)$, and the *only places* where $Z'_T$ and $Z''_T$ differ are $Y_T^{\text{ideal}}$ vs. $\tilde{Y}_T^A$, and $K_{T+1}^{\text{ideal}}$ vs. $\tilde{K}_{T+1}^A$.

The next basic fact, which follows from simple counting, states that if a source $X$ has "sufficient" entropy, and if a source $X'$ is obtained from $X$ by changing at most a $\delta$ fraction of bits in each $x \leftarrow X$, then as long as $\delta$ is not too large, $X'$ still has sufficient entropy.

**Proposition 1.** *Let $\delta$ and $\alpha$ satisfy $0 \le \delta < 1/2$ and $\mathrm{H}(\delta) < \alpha \le 1$, where $\mathrm{H}(\cdot)$ is the binary entropy function. If $X$ is an $\alpha N$-source taking values in $\{0,1\}^N$, and source $X'$ is obtained from $X$ by changing at most $\delta N$ bits of each $x \leftarrow X$, then $X'$ is a $(\alpha - \mathrm{H}(\delta))N$-source.*

By Proposition 1 and the two-way block structure of $(X_t)$, we have

**Corollary 1.** *For each $t$, the random variable $X_t^A$, conditioned on all other $X_{t'}$ for $t' \neq t$, has entropy rate at least $\alpha - \mathrm{H}(\delta)$.*

The following technical claims follow by manipulating statistical distance and weak random sources.

**Claim 1** $\mathrm{SD}(Z'_T, Z''_T) \le 2\varepsilon$.

The proof of Claim 1 is similar to the reasoning in the proof of Lemma 3.3 of [Vad04]. Claim 1 follows from Corollary 1, the definition of the Ideal Experiment, the security property of a local fuzzy extractor, and basic properties of statistical distance and weak random sources.

Let $\mathcal{S}_T$ denote the event that $(\tilde{Y}_T^A, \tilde{K}_{T+1}^A) = (\tilde{Y}_T^B, \tilde{K}_{T+1}^B)$, i.e. the event of correct recovery at time $T$ in the Ideal Experiment.

**Claim 2** $\Pr[\mathcal{S}_T] \geq 1 - \gamma$.

Claim 2 follows from the definition of the Ideal Experiment, and the recovery property of a local fuzzy extractor. The next claim follows from Claims 1 and 2, as well as basic properties of statistical distance.

**Claim 3** $\mathrm{SD}(f_T(Z_{T-1}^{\text{ideal}})|_{\mathcal{S}_T}, Z_T^{\text{ideal}}|_{\mathcal{S}_T}) < 4\varepsilon + 2\gamma$.

Therefore by Claims 2 and 3, and basic properties of statistical distance,

$$\mathrm{SD}(f_T(Z_{T-1}^{\text{ideal}}), Z_T^{\text{ideal}}) \ < \ 4\varepsilon + 2\gamma + \gamma \ = \ 4\varepsilon + 3\gamma \ < \ 4 \cdot (\varepsilon + \gamma),$$

and the theorem follows.

### 4.3 Construction of Local Fuzzy Extractors

In this section we construct a local fuzzy extractor from any given averaging sampler and fuzzy extractor.

*Averaging Samplers.* Averaging samplers are procedures that approximate the average of a $[0,1]$-function by taking the average of the function evaluated at sampled points determined by a random seed. We adopt the following variant of definition in [Vad04] that makes the dependence on $\mu$ explicit.

**Definition 6 ([BR94,Vad04]).** *A function* $\mathrm{Samp} : \{0,1\}^r \to [n]^t$ *is a* $(\mu, \theta, \gamma)$*-*averaging sampler *if for every function* $f : [n] \to [0,1]$ *with average value* $\overline{\mu} = \frac{1}{n} \cdot \sum_{i=1}^{n} f(i) \geq \mu$,

$$\Pr_{(i_1,\ldots,i_t) \leftarrow \mathrm{Samp}(U_r)} \left[ \frac{1}{t} \cdot \sum_{j=1}^{t} f(i_j) \ < \ \overline{\mu} - \theta \right] \leq \gamma. \tag{1}$$

Samp *has* distinct samples *if for every* $x \in \{0,1\}^r$, $\mathrm{Samp}(x)$ *produces* $t$ *distinct samples.*

The following result, analogous to Theorem 6.3 of [Vad04], states that combining an averaging sampler and a fuzzy extractor scheme yields a local fuzzy extractor.

**Theorem 2.** *Let* $\alpha, \tau, \delta, \theta > 0$ *be constants satisfying relations* $\tau < \alpha/3$ *and* $\theta = \tau / \log(1/\tau) < 1 - \delta$. *Let* $\mathrm{Samp} : \{0,1\}^r \to [n]^t$ *be a* $(\mu, \theta, \gamma)$*-averaging sampler with distinct samples with* $\mu = \min\{(\alpha - 2\tau)/\log(1/\tau), 1 - \delta\}$, *and let* $\mathrm{FE} = (\mathrm{Ext}, \mathrm{Rec})$ *be a* $((\alpha - 3\tau)t, \varepsilon, \delta + \theta, \gamma')$*-fuzzy extractor, where* Ext *is of the form* $\mathrm{Ext} : \{0,1\}^t \times \{0,1\}^d \to \{0,1\}^m \times \{0,1\}^\ell$. *Define* $\mathrm{EXT} : \{0,1\}^n \times \{0,1\}^{r+d} \to \{0,1\}^m \times \{0,1\}^\ell$ *as*

$$\mathrm{EXT}(x, (k_1, k_2)) \stackrel{\Delta}{=} \mathrm{Ext}(x_{\mathrm{Samp}(k_1)}, k_2),$$

*and define* $\mathrm{REC} : \{0,1\}^n \times \{0,1\}^{r+d} \times \{0,1\}^\ell \to \{0,1\}^m$ *as*

$$\mathrm{REC}(x', (k_1, k_2), p) \stackrel{\Delta}{=} \mathrm{Rec}(x'_{\mathrm{Samp}(k_1)}, k_2, p).$$

*Then* $(\mathrm{EXT}, \mathrm{REC})$ *is a* $t$*-local* $(\alpha n, \varepsilon + 2 \cdot (\gamma + 2^{-\Omega(\tau n)}), \delta, \gamma + \gamma')$*-fuzzy extractor.*

### 4.4 Construction of the Underlying Fuzzy Extractor

In this section, we describe a construction of (non-local) fuzzy extractors from any given strong extractor and linear error correcting code with an efficient syndrome decoding algorithm. The underlying ideas in the construction have already been used in information reconciliation and privacy amplification (c.f. [BBR88,BS93,BBCM95,CM97a]). This construction also appears in [DRS04].

*Randomness Extractor.* Randomness extractors are functions that extract near perfect randomness from imperfect random sources using a short random seed. An extractor is *strong* if its output remains near uniform even if the seed is given.

**Definition 7 ([NZ96]).** *A function* $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is a* strong $(k, \varepsilon)$-extractor *if for every $k$-source $X$, $(U_d, \mathrm{Ext}(X, U_d))$ is $\varepsilon$-close to $(U_d, U_m)$.*

*Syndrome Decoding.* We quickly review syndrome decoding of a linear error correcting code. Background and details on error correcting codes can be found in standard texts (e.g. [vL99]). Let $C : \{0,1\}^k \to \{0,1\}^n$ be a linear code over $\mathbb{F}_2$ with minimum distance at least $2d+1$. Let $H$ be the $(n-k) \times n$ parity check matrix of $C$. For $x \in \{0,1\}^n$, the *syndrome* of $x$ is defined as $\mathrm{Syn}_C(x) \stackrel{\Delta}{=} Hx$. It is clear that for any *codeword* $y \in C$ and any $e \in \{0,1\}^n$, $\mathrm{Syn}_C(y \oplus e) = \mathrm{Syn}_C(e)$, as $H(y \oplus e) = Hy \oplus He = He$. It is not hard to see that for any $e \in \{0,1\}^n$ with $\mathrm{wt}(e) \leq d$, for every $r \in \{0,1\}^n$ such that $\mathrm{Syn}_C(r) = \mathrm{Syn}_C(e)$, we have $\mathrm{wt}(r) > d \geq \mathrm{wt}(e)$. Hence for any $e \in \{0,1\}^n$ with $\mathrm{wt}(e) \leq d$, $e$ is the *unique* (minimum-weight) vector whose syndrome is $\mathrm{Syn}_C(e)$ and whose weight is at most $d$. A *syndrome decoder* for $C$ that decodes up to $d$ errors is an algorithm $D$ that for every error pattern $e \in \{0,1\}^n$ with $\mathrm{wt}(e) \leq d$, on input $\mathrm{Syn}_C(e)$, outputs $D(\mathrm{Syn}_C(e)) = e$. It is well known that any decoder for a linear code can be converted to a syndrome decoder.

As an important application, syndrome decoding yields a communication efficient protocol for recovering a string $x$ held by a remote party, using a string $y$ that is sufficiently close to $x$ in Hamming distance. Suppose Alice holds $x \in \{0,1\}^n$, Bob holds $y \in \{0,1\}^n$, and $\Delta(x,y) \leq d$. Let $C : \{0,1\}^k \to \{0,1\}^n$ be a linear code over $\mathbb{F}_2$ with minimum distance at least $2d+1$, and an efficient syndrome decoding algorithm $D$ that decodes up to $d$ errors. In order for Bob to recover $x$,

1. Alice sends $\mathrm{Syn}_C(x)$ to Bob.
2. Bob computes $s = \mathrm{Syn}_C(x) \oplus \mathrm{Syn}_C(y) = \mathrm{Syn}_C(x \oplus y)$. Since $\Delta(x,y) \leq d$, $\mathrm{wt}(x \oplus y) \leq d$.
3. Bob then decodes $x \oplus y = D(s)$, and recovers $x = x \oplus y \oplus y$.

Thus Alice sends only $|\mathrm{Syn}_C(x)| = n - k$ bits, as opposed to $n$ bits, to Bob. The correctness of the protocol follows from the correctness of the syndrome decoder $D$: For any $x, y \in \{0,1\}^n$ such that $\Delta(x,y) \leq d$, $\mathrm{wt}(x \oplus y) = \Delta(x,y) \leq d$. Thus $D(\mathrm{Syn}_C(x \oplus y)) = x \oplus y$, and correct recovery follows.

We use $\mathrm{Rep}(D, p, y)$ to denote Bob's algorithm in Steps 2 and 3 above, i.e. on input $s$ and $y$, $\mathrm{Rep}(D, p, y)$ computes $s = p \oplus \mathrm{Syn}_C(y)$, and outputs $D(s) \oplus y$.

*Syndrome-Based Fuzzy Extractor.* This communication efficient recovery protocol above suggests the following fuzzy extractor construction. We adopt an unconventional terminology and say that a code $C : \{0,1\}^{\rho n} \to \{0,1\}^n$ of rate $\rho$ is a $(n, \rho, \delta)$-*code* if it has minimum distance at least $2\delta n + 1$.

**Lemma 1.** *Let $C : \{0,1\}^{\rho n} \to \{0,1\}^n$ be a linear $(n, \rho, \delta)$-code with an efficient syndrome decoder $D$ that decodes up to $\delta n$ errors. Let* $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a strong $(k', \varepsilon)$-extractor, where $k' = k - (1-\rho)n - \log(1/\varepsilon')$. Define* $\mathrm{EXT} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m \times \{0,1\}^{(1-\rho)n}$ *as*

$$\mathrm{EXT}(x, K) \stackrel{\Delta}{=} (\mathrm{Ext}(x, K), \mathrm{Syn}_C(x)),$$

*and define*

$$\mathrm{REC}(x', K, p) \stackrel{\Delta}{=} \mathrm{Ext}(\mathrm{Rep}(D, p, x'), K),$$

*where* $\mathrm{Rep}(.,.,.)$ *is defined above. Then* $\mathrm{FE} = (\mathrm{EXT}, \mathrm{REC})$ *is a $(k, \varepsilon + \varepsilon', \delta, 0)$-fuzzy extractor.*

### 4.5 Choice of Ingredients

*Averaging Sampler.* We use the averaging sampler of Vadhan [Vad04] that is near optimal in both randomness and sample complexity for constant $\mu$ and $\theta$.

**Theorem 3 ([Vad04]).** *For every $n \in \mathbb{N}$, $1 > \mu > \theta > 0$, $\gamma > 0$, there is an explicit $(\mu, \theta, \gamma)$-averaging sampler* $\mathrm{Samp} : \{0,1\}^r \to [n]^t$ *that uses*

- $t$ distinct *samples for any $t \in \left[O(\frac{1}{\theta^2} \cdot \log \frac{1}{\gamma}), n\right]$;*
- $r = \log \frac{n}{t} + \log \frac{1}{\gamma} \cdot \mathrm{poly}(\frac{1}{\theta})$ *random bits.*

*Strong Extractor.* We use the near optimal extractor of Zuckerman [Zuc97] for constant entropy rate.

**Theorem 4 ([Zuc97]).** *For every constant $\alpha, \nu > 0$, for every $n$, and every $\varepsilon > \exp\left(-n/2^{O(\log^* n)}\right)$, there is an explicit strong $(\alpha n, \varepsilon)$-extractor* $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *with $d = O(\log n + \log \frac{1}{\varepsilon})$ and $m = (1-\nu) \cdot \alpha n$.*

*Linear Code.* We need an asymptotically good linear code with rate close to 1 and with an efficient syndrome decoder. Explicit constructions of such codes are well known. In particular, it has been shown in [CRVW02] that the expander codes of Sipser and Spielman [SS96], using a *lossless* expander of [CRVW02], achieve a constant rate $\rho$ that is *arbitrarily* close to 1, and a constant $\delta < 1$.

**Lemma 2 ([SS96,CRVW02]).** *For every constant $\rho < 1$ and every $n \in \mathbb{N}$, there is an explicit linear $(n, \rho, \delta(\rho))$-code $C : \{0,1\}^{\rho n} \to \{0,1\}^n$, where $\delta = \delta(\rho)$ is a constant (depending on $\rho$). Further, $C$ has a linear time syndrome decoder that decodes up to $\delta n$ errors.*

### 4.6 Putting Pieces Together.

In this section, we put all pieces together to yield our final local fuzzy extractor and BSM randomness extraction scheme. First as a corollary of Lemmas 1 and 2, and Theorem 4, we have our final (non-local) fuzzy extractor.

**Lemma 3.** *For every constant $1 \geq \alpha, \gamma, \nu > 0$, there is a constant $\delta > 0$ such that for every sufficiently large $n \in \mathbb{N}$, and every $\varepsilon > \exp\left(-n/2^{O(\log^* n)}\right)$, there is an explicit $(\alpha n, \varepsilon, \delta, 0)$-fuzzy extractor* $(\mathrm{EXT}, \mathrm{REC})$*, where* EXT *is of the form* $\mathrm{EXT} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m \times \{0,1\}^\ell$*, with*

- $d = O(\log n + \log(1/\varepsilon))$,
- $m = (1-\nu)\alpha n$, *and*
- $\ell \leq \gamma m$.

Next, plugging into Theorem 2 the averaging sampler of Theorem 3 and the fuzzy extractor of Lemma 3, we have our final local fuzzy extractor.

**Theorem 5.** *For every constant $1 \geq \alpha, \gamma, \nu > 0$, there is a constant $\delta$ such that for every sufficiently large $N \in \mathbb{N}$, $\varepsilon > \exp\left(-m/2^{O(\log^* m)}\right)$, and $m \leq (1-\nu)\alpha N$, there is an explicit $t$-local $(\alpha N, \varepsilon, \delta, \varepsilon)$-fuzzy extractor* $\mathrm{FE} = (\mathrm{EXT}, \mathrm{REC})$*, where* EXT *is of the form* $\mathrm{EXT} : \{0,1\}^N \times \{0,1\}^d \to \{0,1\}^m \times \{0,1\}^\ell$*, with*

- *seed length* $d = \log N + O(\log m + \log(1/\varepsilon))$,
- *sample size* $t = (1+\nu)m/\alpha$, *and*
- *fingerprint length* $\ell \leq \gamma m$.

Theorem 5 is the "fuzzy" analogue of Theorem 8.5 of [Vad04]. The seed length and sample complexity (i.e. the value of $t$) of our local fuzzy extractor match those of Vadhan's (non-fuzzy) local extractor [Vad04], and thus are optimal up to constant factors.

Finally as a corollary of Theorem 1 and Theorem 5, we have

**Theorem 6.** *For every constant $\alpha > 0$, $\beta < \alpha$, $\gamma > 0$, and $\nu > 0$, there is a constant $\delta$ such that for every sufficiently large $N \in \mathbb{N}$, sufficiently large $m \leq (1-\nu)(\alpha - \beta - \mathrm{H}(\delta))N$, and $\varepsilon > \exp\left(-m/2^{O(\log^* m)}\right)$, there is an explicit $\varepsilon$-secure $t$-local BSM randomness extraction scheme* $(\mathrm{EXT}, \mathrm{REC})$ *for storage rate $\beta$, entropy rate $\alpha$, and error rate $\delta$, where* EXT *is of the form* $\mathrm{EXT} : \{0,1\}^N \times \{0,1\}^d \to \{0,1\}^{m+d} \times \{0,1\}^\ell$*, with*

- *key length* $d = \log N + O(\log m + \log(1/\varepsilon))$,
- *sample size* $t = (1+\nu)m/\alpha'$, *where* $\alpha' = \alpha - \beta - \mathrm{H}(\delta)$, *and*
- *fingerprint length* $\ell \leq \gamma m$.

## 5 Conclusion

We initiate a study of the bounded storage with errors from the public random source that cause parties to have inconsistent view of the source. We provide a general paradigm for constructing error-resilient BSM cryptosystems based on averaging samplers and fuzzy extractors. By proper choice and construction of the underlying building blocks, our general paradigm yields BSM cryptosystems that tolerate a constant fraction of errors, attain near optimal key length and sample complexity (i.e. the number of bits read from the source), and incur a very small communication overhead. It is interesting to study whether the communication overhead can be further reduced.

The recovery property of our local fuzzy extractor can be further improved by taking advantage of the shared randomness between the extraction and the recovery algorithms. By the method of [Lan04], a local fuzzy extractor can be based on any explicit and *list decodable* (as opposed to uniquely decodable) asymptotically good linear code with rate arbitrarily close to 1, while the seed length increases by only $O(\log t + \log 1/\gamma)$ bits, where $t$ is the number of bits read from the source, and $\gamma$ is the recovery error.

Our general paradigm also yields efficient error-resilient message authentication codes (MAC) in the bounded storage model. By combining the BSM extraction scheme of Theorem 6 and an efficient information-theoretically secure MAC (c.f. that of Krawczyk [Kra95]), we obtain an efficient error-resilient BSM MAC that is secure against *chosen message attacks* [GMR89]. Our paradigm can also be used to construct efficient error-resilient protocols for other cryptographic primitives, such as oblivious transfer and key agreement in the bounded storage model. We leave details to the full version.

Our cryptosystems are stateful. That is, our cryptosystems do not reuse the key, but instead update the key in a forward-secure manner. It is an interesting open problem to construct a stateless error-resilient BSM cryptosystem with a reusable key that matches the near optimal parameters achieved by the stateful construction. One promising approach is to derandomize the construction of entropically secure fuzzy extractors in [Smi04].

Another interesting open problem is to construct efficient local fuzzy extractors for other natural metrics, such as editing distance, where the sample-then-extract paradigm fails.

## Acknowledgment

I thank Michael Rabin and Salil Vadhan for numerous insightful discussions on the bounded storage model, in particular the one that brought up the issue of error correction. I thank Yevgeniy Dodis, Adam Smith, Salil Vadhan and anonymous TCC referees for very helpful comments on this paper.

## References


[ADR02] Yonatan Aumann, Yan Zong Ding, and Michael O. Rabin. Everlasting security in the bounded storage model. *IEEE Transactions on Information Theory*, 48(6):1668–1680, June 2002.

[And97] Ross Anderson. Two remarks on public key cryptology. Invited Lecture. In *4th ACM Conference on Computer and Communications Security*, 1997.

[AR99] Yonatan Aumann and Michael O. Rabin. Information theoretically secure communication in the limited storage space model. In *Advances in Cryptology - CRYPTO '99*, pages 65–79. Springer-Verlag, 1999.

[BBCM95] C. Bennett, G. Brassard, C. Crépeau, and U. Maurer. Generalized privacy amplification. *IEEE Transactions on Information Theory*, 41(6):1915 – 1923, 1995.

[BBR88] C. Bennett, G. Brassard, and J. Roberts. Privacy amplification by public discussion. *SIAM Journal on Computing*, 17(2):210–229, 1988.

[BR94] Mihir Bellare and John Rompel. Randomness-efficient oblivious sampling. In *35th Annual IEEE Symposium on Foundations of Computer Science*, pages 276–287, November 1994.

[BS93] Gilles Brssard and Louis Salvail. Secret-key reconciliation by public discussion. In *Advances in Cryptology - EUROCRYPT '93*, pages 410–423. Springer-Verlag, 1993.

[CCM98] Christian Cachin, Claude Crépeau, and Julien Marcil. Oblivious transfer with a memory-bounded receiver. In *39th Annual IEEE Symposium on Foundations of Computer Science*, pages 493–502, November 1998.

[CG88] Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 17(2):230–261, April 1988.

[CM97a] Christian Cachin and Ueli Maurer. Linking information reconciliation and privacy amplification. *Journal of Cryptology*, 10(2):97–110, 1997.

[CM97b] Christian Cachin and Ueli Maurer. Unconditional security against memory bounded adversaries. In *Advances in Cryptology - CRYPTO '97*, pages 292–306. Springer-Verlag, 1997.

[CRVW02] Michael R. Capalbo, Omer Reingold, Salil P. Vadhan, and Avi Wigderson. Randomness conductors and constant-degree lossless expanders. In *34th Annual ACM Symposium on the Theory of Computer Science*, pages 659–668, 2002.

[DHRS04] Yan Zong Ding, Danny Harnik, Alon Rosen, and Ronen Shaltiel. Constant-round oblivious transfer in the bounded storage model. In *1st Theory of Cryptography Conference – TCC '04*, pages 446–472, 2004.

[Din01] Yan Zong Ding. Oblivious transfer in the bounded storage model. In *Advances in Cryptology – CRYPTO '01*, pages 155–170. Springer-Verlag, August 2001.

[DM04a] Stefan Dziembowski and Ueli Maurer. On generating the initial key in the bounded storage model. In *Advances in Cryptology - EUROCRYPT '04*. Springer-Verlag, 2004.

[DM04b] Stefan Dziembowski and Ueli Maurer. Optimal randomizer efficiency in the bounded-storage model. *Journal of Cryptology*, 17(1):5–26, 2004.

[DR02] Yan Zong Ding and Michael O. Rabin. Hyper-encryption and everlasting security (extended abstract). In *19th Annual Symposium on Theoretical Aspects of Computer Science*, pages 1–26. Springer-Verlag, March 2002.

[DRS04] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors and cryptography, or how to use your fingerprints. In *Advances in Cryptology - EUROCRYPT '04*. Springer-Verlag, 2004.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, February 1989.

[Gol97] Oded Goldreich. A sample of samplers: A computational perspective on sampling. Technical Report TR97-020, Electronic Colloquium on Computational Complexity, May 1997.

[Kra95] Hugo Krawczyk. New hash functions for message authentication. In *Advances in Cryptology - EUROCRYPT '95*, pages 301–310. Springer-Verlag, 1995.

[Lan04] Michael Langberg. Private codes or succinct random codes that are (almost) perfect. In *45th Annual Symposium on Foundations of Computer Science*, 2004.

[Lu04] Chi-Jen Lu. Encryption against space-bounded adversaries from on-line strong extractors. *Journal of Cryptology*, 17(1):27–42, 2004.

[Mau92] Ueli Maurer. Conditionally-perfect secrecy and a provably-secure randomized cipher. *Journal of Cryptology*, 5(1):53–66, 1992.

[MST04] Tal Moran, Ronen Shaltiel, and Amnon Ta-Shma. Non-interactive timestamping in the bounded storage model. In *Advances in Cryptology - CRYPTO '04*, pages 460–476. Springer-Verlag, 2004.

[NT99] Noam Nisan and Amnon Ta-Shma. Extracting randomness: A survey and new constructions. *Journal of Computer and System Sciences*, 58(1):148–173, 1999.

[NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd Annual ACM Symposium on the Theory of Computer Science*, pages 427–437, 1990.

[NZ96] Noam Nisan and David Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, 1996.

[Rab02] Michael O. Rabin. Personal communication, 2002.

[Sha02] Ronen Shaltiel. Recent developments in explicit constructions of extractors. *Bulletin of the European Association for Theoretical Computer Science*, 77:67–95, 2002.

[Smi04] Adam Smith. Maintaining secrecy when information leakage is unavoidable. Ph.D. Thesis, MIT, 2004.

[SS96] Michael Sipser and Daniel A. Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, 1996.

[Vad02] Salil P. Vadhan. Randomness extractors and their many guises. In *43rd Annual IEEE Symposium on Foundations of Computer Science*, pages 9–, November 2002. Presentation available at http://www.eecs.harvard.edu/~salil/extractor-focs.ppt.

[Vad04] Salil P. Vadhan. Constructing locally computable extractors and cryptosystems in the bounded storage model. *Journal of Cryptology*, 17(1):43–77, 2004.

[vL99] J.H. van Lint. *Introduction to Coding Theory*. Spring, 1999.

[Zuc96] David Zuckerman. Simulating BPP using a general weak random source. *Algorithmica*, 16(4/5):367–391, 1996.

[Zuc97] David Zuckerman. Randomness-optimal oblivious sampling. *Random Structures & Algorithms*, 11(4):345–367, 1997.