On Everlasting Security in the *Hybrid* Bounded Storage Model[‡]

Danny Harnik[*] Moni Naor[†]

Abstract

The *bounded storage model* (BSM) bounds the storage space of an adversary rather than its running time. It utilizes the public transmission of a long random string $\mathcal{R}$ of length r, and relies on the assumption that an eavesdropper cannot possibly store all of this string. Encryption schemes in this model achieve the appealing property of *everlasting security.* In short, this means that an encrypted message remains secure even if the adversary eventually gains more storage or gains knowledge of (original) secret keys that may have been used. However, if the honest parties do not share any private information in advance, then achieving everlasting security requires high storage capacity from the honest parties (storage of $\Omega(\sqrt{r})$, as shown in [10]).

We consider the idea of a *hybrid bounded storage model* were computational limitations on the eavesdropper are assumed up until the time that the transmission of $\mathcal{R}$ has ended. For example, can the honest parties run a computationally secure key agreement protocol in order to agree on a shared private key for the BSM, and thus achieve everlasting security with low memory requirements? We study the possibility and impossibility of everlasting security in the hybrid bounded storage model. We start by formally defining the model and everlasting security for this model. We show the equivalence of two flavors of definitions: *indistinguishability of encryptions* and *semantic security.*

On the negative side, we show that everlasting security with low storage requirements cannot be achieved by *black-box* reductions in the hybrid BSM. This serves as a further indication to the hardness of achieving low storage everlasting security, adding to previous results of this nature [10, 18]. On the other hand, we show two augmentations of the model that allow for low storage everlasting security. The first is by adding a random oracle to the model, while the second bounds the accessibility of the adversary to the broadcast string $\mathcal{R}$. Finally, we show that in these two modified models, there also exist bounded storage oblivious transfer protocols with low storage requirements.

1 Introduction

1.1 The bounded storage model

The *bounded storage model*, introduced by Maurer [21] postulates a bound on the *space* (memory size) of dishonest players rather than their running time. The model makes use of a long random string $\mathcal{R}$ of length r that is publicly transmitted and accessible to all parties. One can imagine that $\mathcal{R}$ is broadcast at a very high rate by a trusted party or by some natural source or phenomena.

[*] Dept. of Computer Science, Technion, Haifa, Israel. harnik@cs.technion.ac.il. This work was conducted while at the Weizmann Institute.

[†] Incumbent of the Judith Kleeman Professorial Chair. Dept. of Computer Science and Applied Math., Weizmann Institute of Science, Rehovot, Israel. moni.naor@weizmann.ac.il.

[‡] Research supported by a grant from the Israel Science Foundation.

Security in this model relies on the assumption that an adversary cannot possibly store all of the string $\mathcal{R}$ in his memory. For instance, consider the case of honest parties Alice and Bob that want to exchange secret messages in presence of an eavesdropper Charlie. Let the honest parties Alice and Bob use storage of respective size s_A and s_B while an eavesdropper Charlie has a bound of s_C on his storage capacity. Typically we ask that security of the encryption holds in a setting where $s_A, s_B << s_C < r$. That is, the adversary is allowed to have storage space that is much larger than that of the honest players, but still smaller than r. In addition there are no computational restrictions on Charlie.

This model has enjoyed much success for the task of private key encryption. It has been shown that Alice and Bob who share a short private key can exchange messages secretly using only very small storage (a key of length $O(\log r + \log \frac{1}{\varepsilon})$ can be used with storage of size $s_A = s_B = O(\ell + \log r + \log \frac{1}{\varepsilon})$ for an ℓ bit message and ε probability of error). On the other hand an eavesdropper who can store up to a constant fraction of $\mathcal{R}$ (e.g. $\frac{1}{2}r$ bits) cannot learn anything about the messages (this was shown initially in [2] and improved in [1, 9, 11, 20] and ultimately in [27]). These encryption schemes have the important property called *everlasting security* (put forward in [1, 9]), where once the broadcast is over and $\mathcal{R}$ is no longer accessible then the message remains secure even if the private key is exposed and Charlie gains stronger storage capabilities.

In contrast, the situation is far from satisfiable when Alice and Bob do not share any secret information in advance. Cachin and Maurer [4] suggest a method for a key agreement protocol in the bounded storage model. However, this solution requires Alice and Bob to use storage of size at least $\Omega(\sqrt{r})$ which is quite high and renders this approach far less appealing (if not impractical). Dziembowski and Maurer [10] subsequently proved that this is the best one can do.

1.2 Everlasting Security and The Hybrid Bounded Storage Model

The inability to achieve everlasting secure encryption in the bounded storage model with memory requirements smaller than $\sqrt{r}$, has lead to the following appealing suggestion that we call the *hybrid BSM* and is the focus of this paper. Let Alice and Bob agree on their secret key using a computationally secure key agreement protocol (e.g. Diffie-Hellman [6]). The rationale being that while an unbounded eavesdropper will eventually break the key, this is likely to happen only after the broadcast had already occurred. In such a case, the knowledge of the shared key would be useless by that time (this should be expected from the everlasting security property where getting the shared key after the broadcast has ended is useless).

Somewhat surprisingly, Dziembowski and Maurer [10] showed that this rationale may fail. They introduce a specific computationally secure key agreement protocol (containing a non-natural modification based on private information retrieval (PIR) protocols). If this key agreement protocol is used in the hybrid BSM setting with a specific private key scheme, then the eavesdropper can completely decrypt the encrypted message. However, their result does not rule out the possibility that the hybrid idea will work with some other key agreement protocol. For instance, using the plain Diffie Hellman key agreement may still work.

This hybrid model is very natural as we try to achieve everlasting security by adding limitations on the adversary that have a *strict time limit* (expiration date). Assumptions of this sort are generally very reasonable. For instance, in the key agreement example, all that we require is that the computational protocol is not broken in the short time period between its execution and the transmission of $\mathcal{R}$. An assumption such as the Diffie Hellman key agreement [6] cannot be broken within half an hour, can be made with far greater degree of trust than actually assuming the long term security of a computational key agreement protocol.

1.3 This Paper's Contributions

This paper studies the possibility and impossibility of everlasting security in the hybrid bounded storage model. Our contributions are as follows:

- We formally define the model and everlasting security for this model.

- On the negative side we show that everlasting security with low storage requirements cannot be proved by a *black-box* reduction in the hybrid BSM.

- On the other hand, we show two augmentations of the model that allow for low storage everlasting security. The first is by adding a random oracle to the model, while the second bounds the accessibility of the adversary to the broadcast string $\mathcal{R}$.

- Finally, we show that in these two modified models, there also exist oblivious transfer protocols with low storage requirements.

We elaborate on each of these points:

Defining Everlasting Security in the Hybrid BSM: We give rigorous definitions of the type of security we are pursuing. We first define what a hybrid BSM encryption scheme is and then define everlasting security for such a scheme. Following the common practice (stemming from [17]), we give security definitions by indistinguishability of encryptions and by semantic security. We then prove that these two definitions are equivalent.

Regarding the Impossibility of Hybrid schemes: We have more than one indication that proving everlasting security in the hybrid bounded storage model is hard. In fact, it seems quite plausible that everlasting security is not achievable in this model at all. We survey two results (of [10] and [18]) that contribute to this point of view and provide additional evidence in the form of a black-box impossibility result for the general hybrid schemes.

We show a setting (or "world") in which (general) hybrid schemes with everlasting security do not exist. However, if computational key agreement protocols exist in the plain model, then they also exist in this world (this property holds for any computationally secure protocol and not only for key agreement). Thus, we deduce that there can be no black-box proof of everlasting security, basing hybrid BSM schemes on the security of any computational primitive. In particular, this claim rules out the use of computational key agreement protocols via a black-box proof.

Positive Results: We show two modifications of the model that allow for hybrid everlasting security, with small memory requirements (much smaller than $\sqrt{r}$).

Everlasting Security in the presence of Random Oracles: Suppose that the parties are given access to a random oracle. A random oracle is a different type of public random string than the broadcast string $\mathcal{R}$. It is exponentially long and due to its length an efficient algorithm can only access a small fraction of it. This is unlike the broadcast string $\mathcal{R}$ that can be fully accessed by an efficient algorithm. On the other hand the random oracle alone cannot assure everlasting security as it does not eventually disappear and may always be queried at a later stage. We show that combining these two different types of public random strings (the random oracle and $\mathcal{R}$) is sufficient for achieving low storage everlasting security. Everlasting security in this setting means, in particular, that encrypted messages remain secure even if the adversary queries all of the random oracle entries after the broadcast is over.

There are several interpretations to the above result. If one can assemble a random oracle then this presents a methodology for low storage everlasting security. For example, such an oracle may be assembled using natural phenomena (if the broadcast string $\mathcal{R}$ can be implemented this way then why not a random oracle) or using a distributed protocol with partially (and temporarily) trusted parties, e.g. [23]. The emphasis being that such a random oracle need only be secure up until the time of the broadcast. One can also view this result, somewhat optimistically, as a suggested heuristic that achieves everlasting security "for all practical purposes" (when plugging some hash function instead of the random oracle). On the other hand, in light of the negative results for the general hybrid BSM model, one can view the above statement as a testament against relying (blindly) on random oracles to determine whether a task is feasible at all. This is since it shows a task (everlasting security with low storage requirements) that is achievable with a random oracle but might be impossible altogether (or at least very hard to achieve) without it. This is a different statement than previous results regarding random oracles (such as [5, 16, 22]) that show a specific *protocol* (rather than a task) that becomes insecure if the random oracle is replaced by a function with a small representation.

Bounded Accessibility: The *Bounded Accessability Model* assumes that the adversary cannot actually read the whole broadcast string $\mathcal{R}$. Rather, the adversary may choose (adaptively) γr locations of $\mathcal{R}$ that he would like to read and access only these bits during the broadcast. This model was considered by Maurer [21]. The bounded accessibility assumption can be justified, for example, if the broadcast is at a very high rate, and all the adversary can do during a broadcast is to read a bit and write it in his memory (that is of size γr). Another example is if the source consists of a large number of simultaneous transmissions and one can only record a limited number of sources.

We show that the basic hybrid scheme has everlasting security in this model with memory requirements that are substantially smaller than $\sqrt{r}$. An important observation is that this protocol would not have been successful in a plain bounded access model (without introducing the computational limitations on the adversary). This follows since the lower bound of $\sqrt{r}$ [10] applies for this model as well.

On Oblivious Transfer in the Hybrid BSM: We demonstrate that the hybrid BSM in the two augmented models described above can also achieve oblivious transfer (OT) protocols with everlasting security using only low storage. This is in contrast to OT in the standard BSM (see for example [3, 7, 8]) that requires storage of at least $\Omega(\sqrt{r})$ as the lower bound for key agreement [10] applies also for OT. The hybrid scheme is based on the assumption that there exist computationally secure OT protocols.

Paper Organization: In Section 2 we give rigorous definitions of the hybrid BSM and everlasting security. The positive results are given in Section 3, while the negative results appear in Section 5. Section 4 discusses the results for OT.

2 Hybrid BSM - Setting and Security Definitions

We consider the task of exchanging secret messages between two parties in the presence of an eavesdropper. Following is a precise description of what a hybrid BSM encryption scheme consists of. We consider the honest parties Alice and Bob (A and B) and an eavesdropper Charlie (C). The parties have storage bounds s_A, s_B and s_C respectively. Charlie is limited to run in polynomial time up to the end of the broadcast, at which point he must store at most s_C bits of information. After this point, Charlie is not limited in any way. In the general hybrid scheme we do not restrict Alice and Bob to a specific behavior, but rather allow them to communicate before and after the broadcast. The first communication is not necessarily a key agreement, (as was suggested in the introduction) but rather any protocol. The point is that this protocol should take advantage of the assumption that at this time Charlie is restricted to being a polynomial time algorithm. After the broadcast the parties may communicate again, though Charlie is no longer bounded.[1]

Formally, a hybrid scheme consists of the following:

1. Alice and Bob run an interactive protocol whose transcript is denoted by T_1 (and is accessible to Charlie). Alice's view of this interaction is denoted T_1^A (may also include Alice's private coin flips and private data at this stage) and respectively Bob's view is denoted T_1^B.

2. A long random string $\mathcal{R}$ of length r is transmitted.

3. Alice and Bob store information that is no longer than their respective storage bounds and is *efficiently computable* from R and their view of T_1. Denote their storage algorithms by $S_A = S_A(T_1^A, \mathcal{R})$ and $S_B = S_B(T_1^B, \mathcal{R})$ respectively (with $|S_A| \leq s_A$ and $|S_B| \leq s_B$).

4. Alice and Bob run a second round of interaction, whose transcript is denoted T_2. Alice's view of this interaction is denoted T_2^A (Bob's view is denoted T_2^B).

5. Alice has an efficient encryption algorithm E based on S_A and T_2^A. The encryption of message m is denoted by $E_{S_A,T_2^A}(m)$. Bob has an efficient decryption algorithm $E^{-1}_{S_B,T_2^B}$ such that $E^{-1}_{S_B,T_2^B}(E_{S_A,T_2^A}(m)) = m$.

The eavesdropper Charlie's storage capacity is usually taken to be a constant fraction of r, that is, $s_C \leq \gamma \cdot r$ (e.g. half of the length of $\mathcal{R}$). The eavesdropper's behavior is divided into two parts:

- Up until and throughout the broadcast Charlie runs a polynomial time algorithm We call this the compression algorithm.

- After the broadcast is over Charlie may run any (unbounded in space or time) procedure (the decompression).

Note that Charlie is a passive adversary, in the sense that he may not intercept or alter the messages being exchanged (an active adversary cannot be handled in the case that nothing is shared in advance since man-in-the-middle attacks cannot be avoided).

[1] We note that this is a description of the most general hybrid scheme we consider, and is used for the impossibility results in order to rule out as many potential solutions as possible. On the other hand, we would like actual protocols to be much simpler and in particular, all of our positive results have schemes that have interaction only before the broadcast.

The above is a description of the most general hybrid scheme we consider, and is used for the impossibility results in order to rule out as many potential solutions as possible. On the other hand, we would like actual protocols to be much simpler and in particular to have interaction only before the broadcast. In fact, all our positive results have schemes of the following form:

1. Alice and Bob run an interactive protocol whose transcript is denoted by T_1.

2. Alice and Bob use their view of T_1 to choose a small fraction of $\mathcal{R}$ that they want to access.

3. The long random string $\mathcal{R}$ is transmitted, and Alice and Bob use the bits they read from $\mathcal{R}$ and their view of T_1 in order to compute a key K.

4. Encryption of a message m is done by $K \oplus m$.

2.1 Everlasting Security of a Hybrid BSM Scheme

We now define everlasting security for general hybrid BSM schemes. We follow the lines of Goldwasser and Micali's seminal paper [17] by giving two equivalent flavors of definitions: *indistinguishability of encryptions* and *semantic security.* We give our definitions with respect to an adversary Charlie that is a non-uniform polynomial time algorithm throughout the broadcast and an unbounded algorithm after it. For consistency with the definitions of [17] (see also [13]), Charlie's polynomial time compression algorithm is modelled differently in the two definitions. In the indistinguishability definition the compression algorithm is modelled by an ensemble of polynomial size circuits $\{C_k\}_{k\in\mathbb{N}}$, as opposed to the semantic security definition, where it is modelled as a non-uniform polynomial-time TM (receiving non-uniform advice). We say that two distributions P and Q over Ω are ϵ-close if for every $A \subseteq \Omega$, $|\Pr_{x\leftarrow P}(A) - \Pr_{x\leftarrow Q}(A)| \leq \epsilon$.

Definition 2.1 (Everlasting Security - Indistinguishability) *A hybrid BSM scheme is said to have (γ, ε)-everlasting security if $\forall$ polynomial-size circuit family $\{C_k\}$, $C_k : \{0,1\}^* \times \{0,1\}^r \rightarrow \{0,1\}^{\gamma r}$ for all sufficiently large k and for every two messages $m_0, m_1 \in \{0,1\}^k$, we have that $\{C_k(T_1, \mathcal{R}), T_2, E_{S_A,T_2^A}(m_0)\}$ is ε-close to $\{C_k(T_1, \mathcal{R}), T_2, E_{S_A,T_2^A}(m_1)\}$. Where the distribution is taken over the randomness of A, B and the broadcast string R.*

Note: Complexity is measured with respect to the security parameter k. Typically r is chosen to be polynomially related to this parameter, ε is negligible in the security parameter, while γ is taken to be some constant (e.g. $\frac{1}{2}$).

We note that main difference between the above definition and the corresponding definition in the computational setting of [17] is that here the two distributions are required to be *statistically* indistinguishable while in the computational case they are required to be *computationally* indistinguishable.

A different approach is semantic security, that bounds the ability of an unbounded "decompression" procedure D to learn anything about the encrypted message m. Charlie is modelled by a pair of algorithms (C, D), where C is a polynomial time algorithm that receives the encryption of a message M taken from a probability ensemble $\{M_k\}$ of k bit messages, along with a non-uniform polynomial size hint h. The output of C has length at most γr. The algorithm D is unbounded in space or time and models Charlie after the broadcast is over. The definition states that whatever can be computed with the encryption, can be computed essentially as well without seeing the encrypted message:

Definition 2.2 (Everlasting Security - Semantic) *A hybrid BSM scheme is said to have (γ, ε)-everlasting security if $\forall$ boolean function $f : \{0,1\}^k \to \{0,1\}$, polynomial size hint $h : \{0,1\}^k \to \{0,1\}^*$ and for every PPT $C : \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^r \to \{0,1\}^{\gamma r}$ and every unbounded procedure $D : \{0,1\}^{\gamma r} \times \{0,1\}^* \to \{0,1\}$ there exist a PPT $C' : \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^r \to \{0,1\}^{\gamma r}$ and an unbounded procedure $D' : \{0,1\}^{\gamma r} \to \{0,1\}$ such that for every probability ensemble $\{M_k\}$ on messages in $\{0,1\}^k$, and for all sufficiently large k we have that:*

$$\Pr\left[D(C(1^k, T_1, \mathcal{R}, h(M_k)), T_2, E_{S_A, T_2^A}(M_k)) = f(M_k)\right]$$
$$< \Pr\left[D'(C'(1^k, T_1, \mathcal{R}, h(M_k)), T_2) = f(M_k)\right] + \varepsilon$$

Where the probability is taken over M_k, the broadcast string R, and the randomness of Alice, Bob and the algorithm C.

In the corresponding computational definition of [17] only a single efficient algorithm C is discussed, unlike our definition that partitions the computation into its efficient computation C and unbounded computation D.

2.2 Equivalence of the Security Definitions

Claim 2.3 *A hybrid scheme has everlasting security by indistinguishability of encryptions (definition 2.1) if and only if it has everlasting semantic security (definition 2.2).*

We will sketch the proof of this claim, which follows the path of the proof of equivalence of these notions in the common setting of probabilistic encryption [17].
Proof Sketch:

Indistinguishability $\Rightarrow$ Semantic security: The goal is to show that for every pair (C, D) as in definition 2.2 there exist algorithms (C', D') that do essentially as well without getting the encrypted message. This is achieved by replacing the algorithm D with an algorithm D' that first generates an encryption of the dummy string 1^k (this is done by simulating a whole conversation between Alice and Bob including a fresh broadcast string and encrypting 1^k accordingly). Then D' executes (C, D) on this dummy encryption (using the hint of the original message from M_k).

Next claim that D' succeeds in computing $f(M_k)$ as well as D does. To prove this assume towards a contradiction, that for infinitely many k, the algorithm D performs better than D' with a polynomial advantage. For each such k, use an averaging argument over the messages in M_k to single out one message $m_k \in M_k$ for which we have the following:

$$|\Pr\left[D(C(1^k, T_1, \mathcal{R}, h(m_k)), T_2, E_{S_A, T_2^A}(m_k)) = f(m_k)\right] -$$
$$\Pr\left[D(C(1^k, T_1, \mathcal{R}, h(m_k)), T_2, E_{S_A, T_2^A}(1^k)) = f(m_k)\right]| > \varepsilon$$

Now define the polynomial-size circuit family $\{C_k\}$ such that for each k the circuit C_k executes C with fixed hint $h(m_k)$ (this is the non-uniform advice of the circuit). We get that there exists a D, and family $\{C_k\}$, such that for infinitely many k, there exists a message m_k such that

$$|\Pr\left[D(C_k(T_1, \mathcal{R}), T_2, E_{S_A, T_2^A}(m^k)) = 1\right] - \Pr\left[D(C_k(T_1, \mathcal{R}), T_2, E_{S_A, T_2^A}(1^k)) = 1\right]| > \varepsilon$$

contradicting the indistinguishability assumption that $\{C_k(T_1, \mathcal{R}), T_2, E_{S_A, T_2^A}(m^k)\}$ is ε-close to $\{C_k(T_1, \mathcal{R}), T_2, E_{S_A, T_2^A}(1^k)\}$.

Semantic security $\Rightarrow$ Indistinguishability: Suppose for contradiction that there exists a polynomial-size circuit family $\{C_k\}$ such that for infinitely many k there exist pairs $m_0, m_1 \in \{0,1\}^k$ such that, $\{C_k(T_1, \mathcal{R}), T_2, E_{S_A,T_2^A}(m_0)\}$ and $\{C_k(T_1, \mathcal{R}), T_2, E_{S_A,T_2^A}(m_1)\}$ are ε-far. This means that for each such k these probability ensembles may be distinguished with probability at least ε by some all powerful distinguisher that we denote by D. Define the probability ensemble $\{M_k\}$ so that for each k, M_k is uniformly distributed over the two messages $\{m_0, m_1\}$. Define the target function f to be such that for every k $f(m) = 1$ iff $m = m_0$. Define the hint $h(m)$ for every $m \in \{0,1\}^k$ to be the description of the circuit C_k. The algorithm C reads the hint and computes the output of the circuit C_k. Putting it all together we get that there exist f,h,C and D such that for infinitely many k:

$$\Pr\left[D((C(T_1, \mathcal{R}, h(M_k), T_2, E_{S_A,T_2^A}(M^k)) = f(M_k)\right] > \frac{1}{2} + \varepsilon$$

But since the hint contains no information about M_k (other than the length k), then the input for any possible C', D' is information theoretically independent from $f(M_k)$ and cannot predict $f(M_k)$ with probability better than $\frac{1}{2}$. Thus C, D do significantly better than any C', D' which contradicts the assumption. $\square$

3 Positive Results

Due to the negative results of Section 5 it seems that proving everlasting security with low storage requirements is out of our reach at this time (or not possible altogether). We next try to modify the model itself in ways that will allow for positive results. We provide two modifications of the standard hybrid model, and prove that under this modifications we can achieve low storage everlasting security.

3.1 Everlasting Security with a Random Oracle

We show a simple scheme that achieves hybrid everlasting security when given access to a random oracle. We view a random oracle RO as an exponentially long list of entries that is publicly available and allows random access. Each entry in the list contains a (relatively short) string of bits. We start by showing a feasibility result that produces just a one bit key and then describe how this can be generalized to give an efficient scheme that outputs a longer key. *Protocol Encrypt*$_{RO}$:

1. Alice and Bob run any computational key agreement (KA) protocol and agree on a key CK of length k.
2. Alice and Bob access the random oracle at the location CK, the output $SK = RO(CK)$ consists of $k \cdot \log r$ bits. We view these bits as a list of k indices $i_1, \ldots, i_k$ where each $i_j \in [r]$.[2]
3. For each $j \in [k]$ Alice and bob store the bit $\mathcal{R}(i_j)$.
4. The output bit $K = \bigoplus_{j \in [k]} \mathcal{R}(i_j)$.
5. The key K is the final key and the encryption of the message m is $m \oplus K$.

[2] We consider the random oracle $RO(\cdot)$ as an exponentially long array broken into cells where each cell contains $k \cdot \log r$ bits.

Theorem 3.1 *The protocol Encrypt$_{RO}$ has $(\gamma, neg(\cdot))$-everlasting security.*

Proof Sketch: (of Theorem 3.1) First observe that with overwhelming probability the eavesdropper does not query the random oracle at cell CK before the broadcast.

Lemma 3.2 *With probability $1 - neg(k)$ the eavesdropper does not query RO at point CK before the broadcast is over.*

Proof: Suppose that the eavesdropper runs an efficient procedure C that queries $RO(CK)$ with probability $\frac{1}{p(k)}$ for some polynomial $p(\cdot)$. Then using C one can predict CK with a polynomial success probability, simply by outputting a random index that C queries in RO. Since C queries RO at most $q(k)$ times (for some polynomial $q(\cdot)$), this will succeed with probability $\frac{1}{p(k)q(k)}$ which contradicts the security of the key agrement KA. □
Thus we get that with probability $1 - neg(k)$ the eavesdropper has no information on which t locations where stored. In [2] (Theorem 1), it was shown that such an eavesdropper has probability of at most $\frac{1}{2} + neg(k)$ in predicting the key. □

The above simple scheme demonstrates that everlasting security can be achieved by when a random oracle is available. However, it is not efficient in the sense that the number of bits read from the random oracle is far larger than the actual key. This may be greatly improved by using one of the known schemes for private key encryption in the regular (non-hybrid) bounded storage model, such as the scheme in [20, 27]. The ensuing hybrid scheme that is described next achieves comparable parameters to the scheme of [27]. Namely, the storage requirement of Alice and Bob is only $O(k + \log \frac{1}{\varepsilon})$.

Loosely speaking, a *locally computable strong extractor* [27] is a strong randomness extractor [3] with the additional property that the extractor computes its output while reading only a small number of bits from the source. A construction of such an extractor is shown in [27] with seed length $O(\log r + \log \frac{1}{\varepsilon})$ that reads only $O(k + \log \frac{1}{\varepsilon})$ bits of the source.

In the efficient scheme, Alice and Bob use the key SK as a seed to a locally computable strong extractor Ext. Thus the overall scheme is:

1. Alice and Bob run any computational key agreement (KA) protocol and agree on a key CK of length k.

2. Alice and Bob access the random oracle at the location CK, the output $SK = RO(CK)$ consists of $O(\log r + \log \frac{1}{\varepsilon})$ bits.

3. When the long random string $\mathcal{R}$ is broadcast, Alice and Bob run a locally computable strong extractor Ext on the string $\mathcal{R}$ with seed SK and store the encryption key $K = Ext(\mathcal{R}, SK)$.

The proof of security of the efficient scheme follows by choosing ε to be negligible (for example $\varepsilon(k) = k^{-\log k}$) and combining Lemma 3.2 with the following proposition:

Proposition 3.3 (Implicit in [27]) *Let Ext be a locally computable strong extractor as above and let C be any (even an unbounded) algorithm $C : \{0,1\}^r \rightarrow \{0,1\}^{\gamma r}$ with constant γ that does not have information on the secret key SK. Then $\{C(\mathcal{R}), Ext(\mathcal{R}, SK), SK\}$ is ε-close to $\{C(\mathcal{R}), U_k, SK\}$.*

[3] roughly speaking, a strong extractor is an efficiently computable function $Ext(\cdot,\cdot)$ such that for a *source* x with sufficient min-entropy, and a truly random *seed* y the pair $(Ext(x,y), y)$ is statistically close to uniform. See, for example, [26] for background on the standard notion of extractors.

3.2 Bounded Accessability and Hybrid Everlasting Security

The crux in the use of the random oracle is that the adversary does not *read* some information. Thus making assumptions on the ability of the adversary to read all of the bits (rather than store them) seems helpful and indeed we show that it is. The *Bounded Accessability Model* (considered already in the original paper of Maurer [21]) assumes that the adversary cannot actually read the whole of the broadcast string $\mathcal{R}$ but rather just a chosen γ fraction of it. That is, the adversary chooses $\gamma \cdot r$ bits that he wishes to read from the string $\mathcal{R}$ and may then store all of these bits.[4] Such an assumption can be justified, for example, if the broadcast is at a very high rate, allowing the parties to read and store just a small fraction that they prepare for in advance. The definitions of everlasting security are the same as in Section 2 only that in the underlying model, the efficient part of Charlie can only decide on $\gamma \cdot r$ locations in $\mathcal{R}$ and store these actual bits from $\mathcal{R}$.

We show that hybrid everlasting security is achievable in this model with memory requirements that are much smaller than $\sqrt{r}$. An important note is that the lower bound of $\Omega(\sqrt{r})$ [10] applies for this model as well. Therefore the hybrid with computational assumptions is essential.

We present a basic hybrid encryption scheme and prove its security. We note that the following example is aimed at showing the feasibility of such a scheme and does not try to optimize the parameters. The output of the scheme is a one bit key, that should be pseudorandom. We discuss how to achieve better parameters towards the end of this section.

Protocol Encrypt$_{BAM}$*:*

1. Alice and Bob run a computational key agreement protocol KA with the security parameter k. They agree on a key $\bar{\imath} = (i_1, \ldots i_k)$ where for each $j \in [k]$ the j^{th} entry is an index $i_j \in \{0,1\}^{\log r}$ (the transcript of KA is denoted by T_{KA}).

2. For each $j \in [k]$ Alice and Bob store the bit $\mathcal{R}(i_j)$.

3. The final key is $K = \bigoplus_{j \in [k]} \mathcal{R}(i_j)$ and a message m is encrypted by $m \oplus K$.

Theorem 3.4 *The protocol Encrypt*$_{BAM}$ *has* $(\gamma, neg(\cdot))$*-everlasting security.*

Proof: (of Theorem 3.4) Intuitively, for all of the $j \in [k]$ the efficient adversary learns nothing about i_j and therefore his chance of choosing to store $\mathcal{R}(i_j)$ is essentially γ. Therefore the probability that he chose to store all of the bits participating in the XOR is approximately γ^k which is exponentially small. In any other case the bit K is uniformly distributed since it contains a XOR with an unknown bit.

The proof is therefore concluded by the following lemma:

Lemma 3.5 *For all polynomial-size circuit family* $\{C_k\}$ *with bounded accessability* γ*, for any polynomial* $p(\cdot)$*, for all sufficiently large* k *and for every fixing of* $\mathcal{R}$ *we have that*

$$\Pr\left[\ \textit{for all } j \in [k], C_k \textit{ stores the bit } \mathcal{R}(i_j)\right] < \frac{1}{p(k)}$$

[4] Note that we allow the choice of locations to store to be an adaptive choice (rely on the answers of the first bits that where read to determine what to read next).

Proof: (of Lemma 3.5) Suppose that there exists a polynomial-size circuit family $\{C_k\}$ with bounded accessability γ, and a polynomial $p(\cdot)$ such that for infinitely many k we have a fixing of $\mathcal{R}$:

$$\Pr[\text{ for all } j \in [k], C_k \text{ stores the bit } \mathcal{R}(i_j)] > \frac{1}{p(k)}$$

Then we can use $\{C_k\}$ to give a polynomial-size circuit family $\{C'_k\}$ that breaks the security of the key agreement protocol (see definition in Appendix A).

Our goal is to break the computational key agreement scheme and this is done by constructing a circuit that distinguishes between the agreed key and a uniform string. That is, we construct a circuit that distinguishes between the distributions $\{T_{KA}, \bar{i}\}$ and $\{T_{KA}, U_\ell\}$ (where $\ell = k \cdot \log r$ is the length of $\bar{i}$). The circuit C'_k takes as input a transcript t_{KA} of KA and a potential key $\bar{i}$ (such a key is list of indices $\bar{i} = (i_1, \ldots, i_k)$). The transcript t_{KA} is set as the KA part in the transcript of the hybrid BSM scheme. The second part being the fixed string $\mathcal{R}$ from the assumption above. This fixing of $\mathcal{R}$ can be given as a non-uniform hint to circuit C'_k. The circuit C'_k executes C_k on the whole transcript, with the exception that it also records the list of indices of all the places that C_k queried in $\mathcal{R}$. Note that this is no longer a bounded storage process, since the indices may take more space than $\mathcal{R}$. However, this is allowed since the key agreement is supposed to also withstand attacks from adversaries with no storage bound. By the assumption above, the list calculated in C'_k contains all the indices $i_1, \cdot, i_k$ with probability at least $\frac{1}{p(k)}$.

Finally, given a pair $(t_{KA}, \bar{i})$ output 1 if all indices in $\bar{i}$ have been stored by C_k. Otherwise output 0. If $\bar{i}$ is indeed the output of the transcript t_{KA} then the output of C'_k is 1 with probability at least $\frac{1}{p(k)}$. On the other hand, if $\bar{i}$ contains a list of random locations, then the probability of outputting 1 is at most γ'^k which is exponentially small. Thus C'_k has a noticeable advantage in distinguishing between $\{T_{KA}, \bar{i}\}$ and $\{T_{KA}, U_k\}$, contradicting the security of the key agreement protocol. □

□

As in the case of the random oracle protocol, an efficient protocol can be achieved by using the computational key agreement to agree on a key CK of $O(\log r + \log \frac{1}{\varepsilon})$ bits, that will serve as the private key for a regular BSM scheme of [27].

4 On Hybrid Everlasting Security for Oblivious Transfer

Oblivious transfer (OT) (originally defined by Rabin [24] and presented here using the definition of [12]) is a protocol between Alice holding two secrets s_0 and s_1, and Bob holding a choice bit c. At the end of the protocol Bob should learn the secret of his choice (i.e., s_c) but learn nothing about the other secret. Alice, on the other hand, should learn nothing about Bob's choice c. Oblivious transfer is an important building block for construction of secure computation. (e.g., [28, 14, 15, 19]). Cachin, Crepeau and Marcil [3] showed an implementation of OT in the Bounded Storage model with everlasting security in the sense that once the broadcast is over, no party can learn additional information about the secrets, even if the party has gained more power (or storage space) since. This protocol was subsequently improved in [7] and ultimately in [8]. We refer the reader to [8] for rigorous definitions of OT in the bounded storage model.

The main drawback of all bounded storage OT schemes is that the honest parties are required to use storage of $\sqrt{r}$ bits. This requirement is tight since the lower bound of [10] holds also for OT (as OT implies key agreement). We next show that the idea of the hybrid BSM may be useful for implementing OT with low storage requirements. More precisely, we show such schemes under the two modified models discussed for the case of encryption. That is, if random oracles are allowed and in the bounded accessability model.

The idea, in both cases, is to run a computational string OT between Alice and Bob. At the end of this protocol Bob will have learnt one of the two strings that Alice holds. Alice then uses the two strings as secret keys in a classical BSM private key encryption scheme (e.g., as seeds to a locally computable strong extractor in the scheme of [20, 27]). Alice encrypts her two secrets using the two respective keys. Bob can decrypt one (as he received one of the keys) but not the other. For example, in the bounded accessibility model, the protocol is as follows:

Protocol $\mathbf{OT}_{BAM}(s_0, s_1; c)$:

1. Alice chooses two random strings r_0 and r_1 of length $O(\log r + \log \frac{1}{\varepsilon})$ bits.

2. Alice and Bob run a computational OT protocol with Alice inputs r_0, r_1 and Bob's choice bit c.

3. When the long random string $\mathcal{R}$ is broadcast, Alice sends Bob the two encryptions $s_0 \oplus Ext(\mathcal{R}, r_0)$ and $s_1 \oplus Ext(\mathcal{R}, r_1)$ (where Ext is a locally computable strong extractor).

4. Bob uses his knowledge of r_c to decrypt the secret s_c.

The OT protocol when using a random oracle is similar except that it uses the value $RO(r_i)$ as the seed for the extractor. The proofs of security in the two models follow the respective proofs for the case of encryption.

It is interesting that this protocol does not follow the path of the oblivious transfer in the bounded storage model. Oblivious transfer is inherently an asymmetric protocol, and in the above protocol the asymmetry follows only from the computational protocol. Thus not utilizing the mechanism designed for this task in the bounded storage model.

5 Negative Results

We have more than one indication that proving everlasting security in the hybrid bounded storage model is hard. In fact, it seems quite plausible that everlasting security is not achievable in this model at all. In this section we survey two results contributing to this point of view and provide additional evidence in the form of a black-box impossibility result for the general hybrid schemes.

The [DM04] Example: Dziembowski and Maurer [10] show an example that proves that basic hybrid schemes cannot be blindly trusted. The example takes any combination of a computational key agreement protocol and a private key BSM encryption scheme, and replaces the key agreement with a non-natural, yet secure key agreement protocol, that renders the overall scheme insecure. The new key agreement consists of the old one, and an additional "hint" as to what an adversary should store from the string $\mathcal{R}$. This hint is based on private information retrieval (PIR) protocols, and does not give the computationally bounded adversary any information on the underlying key. It does allow him to store a function of $\mathcal{R}$ that contains the necessary information about what the honest players stored. This information can be extracted at a later stage once the adversary is no longer bounded.

This result can be viewed as saying such a basic hybrid scheme cannot work in a black-box manner, taking any key agreement with any private key BSM scheme. It falls short though of saying that hybrid schemes using specific protocols are insecure. Neither does it rule out a black-box general hybrid scheme (we show such an impossibility result in Section 5.1).

On Compression of $\mathcal{NP}$ instances and Hybrid Schemes: In [18] we define the following problem regarding the compression of $\mathcal{NP}$ instances: Consider $\mathcal{NP}$ problems that have long instances but relatively short witnesses. The question is, can one efficiently compress an instance and store a shorter representation that maintains the information of whether the original input is in the language or not. For example, the compression of the language SAT is formulated as follows: A compression for SAT is an efficient algorithm and a polynomial $p(\cdot,\cdot)$ such that the algorithm takes as input a CNF formula Φ with m clauses over n variables (where $m >> n$). The output should be a formula Ψ of size $p(n, \log m)$ such that Ψ is satisfiable if and only if Φ is satisfiable.

In [18], a family of $\mathcal{NP}$ languages is defined that includes, for example, the languages SAT and Clique. It is shown that if there exists a compression algorithm for any language in this family, then the hybrid BSM is no more powerful (with respect to everlasting security) than the standard BSM. Conversely, in order to prove everlasting security in the hybrid model with low storage requirements, one must prove (or assume) that certain languages are incompressible.

5.1 Black-Box Impossibility

We next show that there exists no black-box *proof* of everlasting security of a hybrid encryption scheme, based on the security of a computational primitive. This forms a wider black-box impossibility than that of the [10] result, as it captures any general hybrid scheme (rather than just a combination of a key agreement and BSM encryption scheme).

We prove the result by introducing a setting (or "world") containing a specialized oracle where any hybrid scheme with low storage requirements (that does not use the oracle) may be broken, but otherwise computational primitives are left unaffected. As a corollary we get that there can be no black-box proof of everlasting security of a hybrid scheme based on a computational protocol (such as key agreement, but also others, e.g., oblivious transfer). The reason is that a security proof is a reduction showing that if the hybrid BSM can be broken then so can the computational scheme. Such a black-box proof would also apply in a world as described above and would thus prove the insecurity of the computational scheme. Therefore, if there existed such a black-box proof based on key agreement (for example) then it would serve as a proof that there exists no computational key agreement protocols altogether. In particular this means that there is no fully black-box reduction (in the terminology of [25]).

We note though that this result rules out an even wider family of reductions (that was not named in [25]). In a fully black-box reduction both the construction itself and the proof are required to be black-box. In contrast, our statement asserts that there can be no black-box *proof* of security, even if the construction itself is allowed to do anything. In particular this holds for non-black-box constructions and even for constructions based on specific assumptions.

To specify the world mentioned above we define an oracle Z. Loosely speaking, the oracle Z takes as input an $\mathcal{NP}$ relation L and a (presumed) instance x of L, and generates several random witnesses to the fact that $x \in L$ (assuming such witnesses exist). However, the oracle does not actually output these witnesses. Instead it returns some form of encryption of them. The point is that a polynomial time adversary essentially gains nothing from the presence of the oracle Z. Thus any primitive that is secure against a polynomial time adversary remains secure even when the adversary is given access to the oracle Z. But a hybrid BSM encryption cannot have everlasting security since a polynomial time adversary with access to Z can save encrypted information for future use.

The Oracle Z: For every $\ell \in \mathbb{N}$ let $\pi_\ell : \{0,1\}^\ell \to \{0,1\}^\ell$ be a random permutations. For every $\ell, k \in \mathbb{N}$, let $R_{\ell,k}$ be a $2^\ell \times 2^k$ matrix of entries in $\{0,1\}^\ell$. The entries of the matrix $R_{\ell,k}$ are random strings subject to the sole restriction that $\bigoplus_{i\in[2^k]} R_{\ell,k}[y,i] = \pi_\ell^{-1}(y)$ (where $y \in \{0,1\}^\ell$ is also viewed as an index $y \in [2^\ell]$ and the XOR is performed bit by bit).

1. The oracle takes an input (L, x, n, k) where L is the description of an $\mathcal{NP}$ relation $L(\cdot,\cdot)$, x is a string (presumably in L), n is a parameter and k is a limit on witness length. If there is witness w of size at most k such that $L(x,w) = 1$, then the oracle computes n witnesses $w_1, ..., w_n$ that are randomly chosen under the restriction that $L(x, w_i) = 1$ for all $1 \le i \le n$ (suppose w.l.o.g. that each of the witnesses is of length exactly k). The output of the oracle is $y = \pi_{nk}(w_1, \dots, w_n)$. In case there is no witness of size k for x, then the output is simply a random string y.
2. On input (y, i) where $y \in \{0,1\}^{nk}$ and $i \in [2^k]$, the output is $R_{nk,k}[y,i]$.

Thus the oracle Z is defined by the ensemble of permutations π_ℓ and matrices $R_{\ell,k}$ as well as the randomness used to sample the witnesses.

Theorem 5.1 *In a world containing the Oracle Z (and no other oracle) we have:*

- *Any cryptographic primitive that is computationally secure in the plain model is secure in this world as well. More precisely, for any polynomial time adversary A with access to Z, there exists a polynomial time adversary A' in the plain model (*without *access to Z) so that any polynomial time environment interacting with the adversary without access to Z, cannot distinguish between A and A'.*
- *Any hybrid BSM encryption scheme in which (i) Alice and Bob require storage of size at most $o(\sqrt{r})$, and (ii) Alice and Bob make no calls to Z, cannot have everlasting security.*

The proof idea: For a computationally bounded adversary, the oracle Z is indistinguishable from a random oracle. This is because unless a whole row of R is read, then all of the outputs of Z are truly random strings. Thus, the oracle is of no use to a computationally bounded adversary as such an adversary can simulate the random oracle on his own, simply by tossing random coins whenever it queries the oracle.

On the other hand, the oracle is very handy in breaking any hybrid scheme. This is shown by designing a specific $\mathcal{NP}$ relation L and invoking a Lemma of Dziembowski and Maurer [10], that essentially states that a large enough number of random witnesses to L are sufficient to break a bounded storage scheme with low memory requirements. The adversary Charlie in the hybrid model can query for these witnesses from Z, but gets only an encrypted version via π_{nk} which is useless at the point. However, after the broadcast is over, Charlie may use his unbounded powers to extract the required information from the oracle Z, by decrypting π_{nk} using $R_{nk,k}$, and break the scheme. The actual proof follows:

Proof: (of Theorem 5.1) The first property follows from the fact that any computationally bounded adversary cannot distinguish between Z and a random oracle. The oracle Z receives two types of queries. On input (L, x, n, k) it answers a random string $y \in \{0,1\}^{nk}$, which is exactly the behavior of a random oracle. What differs Z from a random oracle is that on input (y, i) it returns values that are related to the inverse of y. However, these values are truely random when viewing anything short of a full row of $R_{nk,k}$. Therefore, for a computationally bounded adversary all of the answers of Z are truely random, since a whole row of $R_{nk,k}$ cannot be queried in polynomial time (requires 2^k queries). Finally, any adversary A with access to Z, can simulate the answers of Z on his own. An adversary A' simply runs A and whenever a new query to Z is required, A' simply chooses a random answer (by flipping coins locally) and stores this answer (in case the same query is repeated twice). Since A cannot distinguish Z from a random oracle, then A' behaves exactly as A does. In particular, any environment that queries A or A' (the environment has no access to Z) cannot tell the two apart.

To see the second property, we make use of the proof of the lower bound in [10], starting with some notations: Let $K(T_1, \mathcal{R}, T_2)$ denote the encryption key that is agreed on when the protocol is run with transcripts T_1, T_2 and randomness $\mathcal{R}$. Because agreement is guaranteed then this key must be well defined (for simplicity we assume w.l.o.g. that the agreement in the protocol is perfect). Denote by A_{T_1} the set of all possible randomness r_A of Alice that are consistent with the transcript T_1. Let $S = S_A(T_1, \mathcal{R}, r_A)$ denote the bits that Alice stores at the end of the broadcast when running with randomness r_A, transcript T_1 and broadcast string $\mathcal{R}$. Finally, denote by $\mathbf{S}_A(T_1, \mathcal{R})$ the random variable that takes the value $S_A(T_1, \mathcal{R}, r_A)$ for a uniform choice of $r_A \in A_{T_1}$. That is, $\mathbf{S}_A(T_1, \mathcal{R})$ is randomly chosen from all possible S that Alice might have stored when running with transcript T_1 and broadcast string $\mathcal{R}$.

Lemma 5.2 *([10]) Let $\mathbf{S}_A(T_1, \mathcal{R})$ and $K(T_1, \mathcal{R}, T_2)$ be defined as above. For any $\mathcal{R}$ and T_1 Let $\mathbf{S}_C(T_1, \mathcal{R})$ denote the random variables that takes s_B independent samples of $\mathbf{S}_A(T_1, \mathcal{R})$. Then:*

$$H(K(T_1, \mathcal{R}, T_2) | \mathbf{S}_C(T_1, \mathcal{R})) \leq s_A s_B / r.$$

In other words, as long as s_A and s_B are significantly smaller than r, then a strategy for an eavesdropper is to store s_B independent samples of the random variable $\mathbf{S}_A(T_1, \mathcal{R})$. This strategy guarantees that the eavesdropper will have stored (with high probability) enough information on the encryption key K.

An attack on any hybrid BSM protocol is given by the following strategy: Define the $\mathcal{NP}$ relation L, where the instance consists of a transcript T_1 and a broadcast string $\mathcal{R}$ and a witness consists of a string s and a string r_A. L is defined by $L((T_1, \mathcal{R}); (s, r_A)) = 1$ if and only if $r_A \in A_{T_1}$ and also $s = S_A(T_1, \mathcal{R}, r_A)$. That is, the first part of a random witness to $(T_1, \mathcal{R}) \in L$ coincides exactly with the output of the random variable $\mathbf{S}_A(T_1, \mathcal{R})$.

The attack of Charlie is thus to query the oracle Z with L, $(T_1, \mathcal{R})$, s_B and receive $\pi_{nk}(w_1, \ldots, w_n)$. By Lemma 5.2, the witnesses $w_1, \ldots, w_n$ contain sufficient information to break the scheme. Charlie stores only $y = \pi(w_1, \ldots, w_n)$, and once the broadcast is over, uses his unbounded running time to extract the values $(w_1, \ldots, w_n) = \pi^{-1}(y)$ from the oracle Z, thus breaking the scheme. □

6 Open Problems

The obvious open problem is to settle the possibility or impossibility of low storage everlasting security in the general hybrid model. This would likely entail resolving the existence or in-existence of relevant compression algorithms (following the formulation of [18]). Another problem is to come up with additional (reasonable) models where the notion of everlasting security may be achieved.

Finally, our solution for oblivious transfer (OT) requires the use of computationally secure oblivious transfer protocols. An interesting question is can one achieve low storage everlasting security OT protocols based on weaker assumptions than computational OT such as key agreement, for instance. It is tempting to think such a protocol exists since bounded storage OT may be achieved with no computational assumption at all (albeit, with high storage requirements).

References

[1] Y. Aumann, Y.Z. Ding, and M. O. Rabin. Everlasting security in the bounded storage model. *IEEE Transactions on Information Theory*, 48(6):1668–1680, 2002.

[2] Y. Aumann and M. O. Rabin. Information theoretically secure communication in the limited storage space model. In *Advances in Cryptology – CRYPTO '99, Lecture Notes in Computer Science*, volume 1666, pages 65–79. Springer, 1999.

[3] C. Cachin, C. Crépeau, and J. Marcil. Oblivious transfer with a memory-bound receiver. In *39th IEEE Symposium on Foundations of Computer Science*, pages 493–502, 1998.

[4] C. Cachin and U. Maurer. Unconditional security against memory-bound adversaries. In *Advances in Cryptology – CRYPTO '97, Lecture Notes in Computer Science*, volume 1294, pages 292–306. Springer, 1997.

[5] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. *Journal of the ACM*, 51(4):557–594, 2004.

[6] W. Diffie and M.E. Hellman. New directions in cryptography. *IEEE Transaction on Information Theory*, 22:644–654, 1976.

[7] Y.Z. Ding. Oblivious transfer in the bounded storage model. In *Advances in Cryptology - CRYPTO '01, Lecture Notes in Computer Science*, volume 2139, pages 155–170. Springer, 2001.

[8] Y.Z. Ding, D. Harnik, A. Rosen, and R. Shaltiel. Constant round oblivious transfer in the bounded storage model. In *The 1st Theory of Cryptography Conference – (TCC '04)*, volume 2951 of *Lecture Notes in Computer Science*, pages 446–472, 2004.

[9] Y.Z. Ding and M.O. Rabin. Hyper-encryption and everlasting security. In *Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 1–26, 2002.

[10] S. Dziembowski and U. Maurer. On generating the initial key in the bounded-storage model. In *Advances in Cryptology – EUROCRYPT ' 2004, Lecture Notes in Computer Science*, volume 3027, pages 126–137. Springer, 2004.

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

[12] S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. *Communications of the ACM*, 28(6):637–647, 1985.

[13] O. Goldreich. **Foundations of Cryptography** - *Volume 2.* Cambridge University Press, 2004.

[14] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game - a completeness theorem for protocols with honest majority. In *19th ACM Symposium on the Theory of Computing*, pages 218–229, 1987.

[15] O. Goldreich and R. Vainish. How to solve any protocol problem - an efficiency improvement. In *Advances in Cryptology - CRYPTO '87, Lecture Notes in Computer Science*, volume 293, pages 73–86. Springer-Verlag, 1987.

[16] S. Goldwasser and Y. Tauman Kalai. On the (in)security of the fiat-shamir paradigm. In *44th IEEE Symposium on Foundations of Computer Science*, pages 102–111, 2003.

[17] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of the ACM*, 28(4):270–299, 1984.

[18] D. Harnik and M. Naor. On the compressibility of NP instances and cryptographic applications. In *Electronic Colloquium on Computational Complexity (ECCC), TR06-022*, 2006.

[19] J. Kilian. Founding cryptography on oblivious transfer. In *20th ACM Symposium on the Theory of Computing*, pages 20–31, 1988.

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

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

[22] U. Maurer, R. Renner, and C. Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In *The 1st Theory of Cryptography Conference – (TCC '04)*, volume 2951 of *Lecture Notes in Computer Science*, pages 21–39, 2004.

[23] M. Naor, B. Pinkas, and O. Reingold:. Distributed pseudo-random functions and kdcs. In *Advances in Cryptology - EUROCRYPT '99, Lecture Notes in Computer Science*, volume 1592, pages 327–346. Springer, 1999.

[24] M. O. Rabin. How to exchange secrets by oblivious transfer. TR-81, Harvard, 1981.

[25] O. Reingold, L. Trevisan, and S. Vadhan. Notions of reducibility between cryptographic primitives. In *The 1st Theory of Cryptography Conference – (TCC '04)*, volume 2951 of *Lecture Notes in Computer Science*, pages 1–20, 2004.

[26] R. Shaltiel. Recent developments in explicit constructions of extractors. *Bulletin of the EATCS*, 77:67–95, 2002.

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

[28] A. C. Yao. How to generate and exchange secrets. In *27th IEEE Symposium on Foundations of Computer Science*, pages 162–167, 1986.

A Definition: Computational Key Agreement

For completeness we provide a definition of a computationally secure key agreement protocol that is secure against non-uniform polynomial-time adversaries.

Definition A.1 (Computational Key Agreement) *A protocol KA between Alice and Bob with transcript T_{KA} is a Key Agreement protocol if it takes as input a security parameter k and each of the two parties computes an output key (w.l.o.g. this is a key of length k).*

1. *At the end of the protocol Alice and Bob agree on the same key $K(T_KA)$.*
2. *$\forall$ polynomial-size circuit family $\{C_k\}$, $C_k : \{0,1\}^* \to \{0,1\}^*$ for all sufficiently large k, we have that the distributions $\{C_k(T_{KA})), K(T_{KA}\}$ and $\{C_k(T_{KA}), U_k\}$ are computationally indistinguishable.*