On Generating the Initial Key in the Bounded-Storage Model

Stefan Dziembowski*
Institute of Informatics, Warsaw University
Banacha 2, PL-02-097 Warsaw, Poland, std@mimuw.edu.pl

Ueli Maurer
Department of Computer Science, ETH Zurich
CH-8092 Zurich, Switzerland, maurer@inf.ethz.ch

Abstract. In the bounded-storage model (BSM) for information-theoretically secure encryption and key-agreement one uses a random string R whose length t is greater than the assumed bound s on the adversary Eve's storage capacity. The legitimate parties Alice and Bob share a short initial secret key K which they use to select and combine certain bits of R to obtain a derived key X which is much longer than K. Eve can be proved to obtain essentially no information about X even if she has infinite computing power and even if she learns K after having performed the storage operation and lost access to R.

This paper addresses the problem of generating the initial key K and makes two contributions. First, we prove that without such a key, secret key agreement in the BSM is impossible unless Alice and Bob have themselves very high storage capacity, thus proving the optimality of a scheme proposed by Cachin and Maurer. Second, we investigate the hybrid model where K is generated by a computationally secure key agreement protocol. The motivation for the hybrid model is to achieve provable security under the sole assumption that Eve cannot break the key agreement scheme *during* the storage phase, even if afterwards she may gain infinite computing power (or at least be able to break the key agreement scheme). In earlier work on the BSM, it was suggested that such a hybrid scheme is secure because if Eve has no information about K during the storage phase, then she has missed any opportunity to know anything about X, even when later learning K. We show that this very intuitive and apparently correct reasoning is false by giving an example of a secure (according to the standard definition) computational key-agreement scheme for which the BSM-scheme is nevertheless completely insecure. One of the surprising consequences of this example is that existing definitions for the computational security of key-agreement and encryption are still too weak and therefore new, stronger definitions are needed.

* Part of this work was done while the first author was a Post-Doc at ETH Zurich, Switzerland. Supported in part by the Polish KBN grant no. 4 T11C 042 25, by the European Communitiy Research Training Network ("GAMES" contract HPRN-CT-2002-00283), and by the Foundation for Polish Science (FNP).

1 Introduction

In the bounded-storage model (BSM) for information-theoretically secure encryption and key-agreement one can prove the security of a scheme based on the sole assumption that the adversary's storage capacity is bounded, say by s bits, even if her computing power is unlimited. Assume that a random t-bit string R is either temporarily available to the public (e.g. the signal of a deep space radio source) or broadcast by one of the legitimate parties. If $s < t$, then the adversary can store only partial information about R. The legitimate parties Alice and Bob, sharing a short secret key K initially, can therefore potentially generate a very long n-bit one-time pad X with $n \gg |K|$ about which the adversary has essentially no information.

1.1 Definition of the Bounded-Storage Model

We define the bounded-storage model for key-expansion (and encryption) more formally. Alice and Bob share a short secret *initial key* K, selected uniformly at random from a key space $\mathcal{K}$, and they wish to generate a much longer n-bit *expanded key* $X = (X_1, \ldots, X_n)$ (i.e. $n \gg \log_2 |\mathcal{K}|$).

In a first phase, a t-bit random string R is available to all parties, i.e., the randomizer space is $\mathcal{R} = \{0,1\}^t$. For instance, R is sent from Alice to Bob or broadcast by a satellite. In fact, R need not be uniformly random, it suffices to know a lower bound on the min-entropy $H_\infty(R)$ of R. Alice and Bob apply a known *key-expansion function*

$$f : \mathcal{R} \times \mathcal{K} \to \{0,1\}^n$$

to compute the expanded (or derived) key as $X = f(R, K)$. Of course, the function f must be efficiently computable and based on only a very small portion of the bits of R such that Alice and Bob need not read the entire string R.

Eve can store arbitrary s bits of information about R, i.e., she can apply an arbitrary storage function

$$h : \mathcal{R} \to \mathcal{U}$$

for some $\mathcal{U}$ with the only restriction that $|\mathcal{U}| \leq 2^s$.[1] The memory size during the evaluation of h need not be bounded. The value stored by Eve is $U = h(R)$. After storing U, Eve loses the ability to access R. (This is also referred to as the second phase.) All she knows about R is U. In order to prove as strong a result as possible, one assumes that Eve can now even learn K, although in a practical system one would of course keep K secret. This strong security property will be of special importance in this paper.

A key-expansion function f is secure in the bounded-storage model if, with overwhelming probability, Eve, knowing U and K, has essentially no information about X. More precisely, the conditional probability distribution $P_{X|U=u,K=k}$ is

[1] Since for every probabilistic strategy there is a best choice of the randomness, we can without loss of generality consider only deterministic adversary strategies.

very close to the uniform distribution over the n-bit strings, with overwhelming probability over values u and k. Hence X can be used as a secure one-time pad. Of course the security of f depends on Eve's memory size s.

1.2 The Subject of this Paper and Previous Results

The bounded-storage model was proposed initially in 1992 [15], but the really strong (and essentially optimal) security results were proved only recently in a sequence of papers [2, 1, 10, 11, 14, 17]. The first security proof for general storage functions h was obtained by Aumann and Rabin [2], but only for $n = 1$ (i.e., for a scheme in which the derived key X is much shorter than the initial key) or for $s \ll t$ (i.e., when the size of the memory of the adversary is much smaller than the length of the randomizer). The first fully general security proof was given in [11]. Lu [14] and Vadhan [17] showed that a special type of randomness extractor can be used to construct secure schemes, also improving on the size of the initial key K.

In all these papers one assumes that Alice and Bob initially share a secret key K, usually without considering how such a key K is obtained by Alice and Bob. In this paper we address the problem of generating this key K and investigate how this key generation process relates to the security proof of the BSM. We discuss the two most natural approaches to generating K, in a setting where Alice and Bob are connected only by an authenticated communication channel, without a trusted third party that initially distributes the key K.

The first approach is to generate K within the context of the BSM itself or, equivalently, to perform key agreement in the BSM without sharing any secret key K initially. This approach was discussed by Cachin and Maurer in [3] where a scheme was proposed in which both Alice and Bob need storage on the order of $\sqrt{t}$. More precisely, they each store a random subset (with pairwise independent indices) of the bits of R and, after R has disappeared for all parties, publicly agree on which bits they have both stored. With very high probability, Eve has only partial information about these bits, and therefore Alice and Bob can apply privacy amplification (i.e., randomness extraction using a strong extractor with a public extractor parameter) to distill an essentially perfect key X, which they can then use as a one-time pad. We show (Section 3) that the protocol of [3] is essentially optimal (in terms of the ratio between the storage size of the honest parties and the adversary) if s is on the order of t. Since the storage requirement of $\sqrt{t}$ (which is also on the order of $\sqrt{s}$) bits for Alice and Bob may be too high for a reasonable storage bound s for Eve, the practicality of this approach is questionable.

The second approach is to generate K by a computationally secure key-agreement protocol, for instance based on the Diffie-Hellman protocol [7]. At first, this approach may appear to be completely useless since the provable information-theoretic security of the BSM-scheme would be lost: A computationally unbounded adversary could break the computational key-agreement protocol and then play the role of either Alice or Bob, with the same (small) storage requirements. However, at second sight, this approach is quite attractive as it allows to preserve the security of the key agreement protocol, which is only computational, even if the adversary can later break it and even if she gains infinite computing power.

It was claimed in [1] (Section IV B) (also, less formally in [10] (p. 5), [9] (p. 11) and [14] (p. 2)) that this implies the security of the hybrid scheme, for the following reason. Let T be the transcript of the key agreement protocol. The adversary has (computationally) no information about K, given T, when performing the storage operation. More precisely, she could not distinguish K from a truly random and independently generated key (as in the pure BSM). Therefore she could just as well forget T and generate a random key K himself, in which case she obviously has no advantage over the pure BSM setting. Since in this setting Eve learns K anyway after finishing the storage operation, it does not hurt in the computational setting if Eve can now break the key-agreement scheme and compute K (from T). Note that all the remaining aspects of the security proof are entirely information-theoretic.

It may come as a surprise that this reasoning is false, as is proved in Section 4. More specifically, we give an example of a computationally secure (according to the standard definition) key-agreement scheme which, when used to generate K in the BSM context, renders the latter completely insecure. This shows that security arguments in a mixed computational/information-theoretic context can be very subtle. More interestingly, it demonstrates that existing definitions for the computational security of key-agreement and encryption are still too weak. Therefore new, stronger definitions are needed.

2 Preliminaries

The treatment in this paper is intentionally quite informal, but it is obvious how all aspects could be formalized in the traditional manner used in cryptography. The computation of a party (or algorithm) can be modelled by a probabilistic Turing machine, a protocol for two parties can be modelled as two interactive Turing machines, cryptographic primitives (such as key agreement) can be modelled as an asymptotic family with a security parameter, efficient can be defined as polynomial time, and negligible can also be defined in the traditional manner.

2.1 Secure Key-Agreement

A key-agreement scheme is a protocol between two parties Alice and Bob, at the end of which each party computes the same key $K \in \mathcal{K}$ (with overwhelming probability), for some key space $\mathcal{K}$. Let T be the transcript of the protocol, i.e., the entire list of exchanged messages. Throughout the paper, we consider security against *passive* attacks, i.e., we assume that Alice and Bob can communicate over an authenticated channel. This is sufficient to illustrate our point, but note that security definitions for key-agreement are much more subtle in a setting with an active adversary who can tamper with messages exchanged between Alice and Bob.

A key-agreement scheme is *computationally secure* if no efficient distinguisher, when given T, can distinguish K from a key K' chosen independently and uniformly at random from the key space $\mathcal{K}$, with non-negligible advantage. For example, the computational security of the Diffie-Hellman key-agreement protocol [7] is equivalent to the so-called Decision-Diffie-Hellman assumption.

A computationally secure key-agreement scheme can also be obtained from any semantically secure public-key encryption scheme: Alice selects a random key $K \in \mathcal{K}$ and sends it to Bob, encrypted with Bob's public key.

2.2 Private Information Retrieval

The idea of private information retrieval (PIR) was introduced in [4]. A PIR scheme is a protocol for two parties, a user U and a database D, allowing the user to access database entries in a way that D cannot learn which information U requested. More precisely, the database content can be modelled as a string $x = (x_1, \ldots, x_l) \in \{0,1\}^l$, and U wants to access the ith bit x_i of x, for some $i \in \{1, \ldots, l\}$, such that D does not learn i. It is not relevant whether U learns more than x_i.

A trivial solution to this problem is that D sends all bits $x_1, \ldots, x_l$ to U, allowing U to pick the bits he wants. The purpose of PIR protocols is to reduce the required communication. Depending on the protocol, the secrecy of i can be computational or information-theoretic. In this paper we consider computationally secure PIR protocols [13].

A typical PIR protocol proceeds in three stages. First, U sends a query, depending on i. Let $\mathcal{Q}(i)$ denote the query for index i. Second, D computes the reply $\mathcal{R}(\mathcal{Q}(i), x)$ and sends it to U. Third, U extracts x_i from $\mathcal{R}(\mathcal{Q}(i), x)$. The scheme is computationally private if no efficient distinguisher can distinguish $\mathcal{Q}(i)$ from $\mathcal{Q}(i')$, for any $i, i' \in \{1, \ldots, l\}$.

In this paper we need an additional property of the PIR scheme, namely that x_i is determined by i, $Q(i)$, and $\mathcal{R}(\mathcal{Q}(i), x)$ (even if it cannot be efficiently computed). Note that in a PIR scheme, U typically holds a secret key which allows to extract x_i efficiently from i and $\mathcal{R}(\mathcal{Q}(i), x)$.

A well-known PIR scheme proposed in [13] makes use of the homomorphic property of the quadratic residues and the computational difficulty of the quadratic residuosity problem. More precisely, U generates an RSA modulus $n = pq$. The string $(x_1, \ldots, x_l)$ is divided into $v = \lceil l/t \rceil$ blocks of length t, for some t. Let $1 \leq j \leq t$ be the index of x_i within its block. The query $\mathcal{Q}(i)$ consists of a list $(y_1, \ldots, y_t)$ of t elements of Z_n^*, all (independent) random quadratic residues, except for y_j which is a random quadratic non-residue with Jacobi symbol 1. The database's reply consists of v elements in Z_n^*, one for each of the v blocks, where for each block D computes the product of all the y_m corresponding to 1's in the block. More precisely, $\mathcal{R}(\mathcal{Q}(i), x)$ consists of one element of Z_n^* for each block, where for the first block $(x_1, \ldots, x_t)$ the value is

$$\prod_{k=1}^{t} y_k^{x_k},$$

for the second block it is $\prod_{k=1}^{t} y_k^{x_{t+k}}$, and similarly for the other blocks. Let $m \in \{1, \ldots, v\}$ be the index of the block to which x_i belongs. It is easy to see that $x_i = 0$ if and only if the reply for the mth block is a quadratic residue. Clearly this can be efficiently checked by the user U (who knows p and q). Note that the user ignores all other received values not relevant for obtaining x_i. The communication complexity of this scheme is as follows: The query consists of t elements of Z_n^* and the reply consists of v elements of Z_n^*. A reasonable trade-off is to let $t \approx \sqrt{l}$.

3 Limitations of Key-Agreement in the BSM

3.1 The Setting

In this section we consider the BSM without an initially shared secret key between Alice and Bob. In this setting, in the first phase when R is available to all parties, Alice and Bob may use a randomized strategy (where the random strings of Alice and Bob are independent and denoted as R_A and R_B, respectively) to execute a protocol resulting in transcript T, and to each store some information about R. Alice stores $M_A = f_A(R, T, R_A)$, and Bob stores $M_B = f_B(R, T, R_B)$, for some functions f_A and f_B. Eve also stores some information $M_E = f_E(R, T, R_E)$ about R, for some random string R_E.

In the second phase, when R has disappeared, Alice and Bob execute a second (probabilistic) protocol based on the stored values M_A and M_B, resulting in a second transcript T' and in both computing the key K.[2] The security requirement is that Eve must have only a negligible amount of information about K, i.e., $I(K; M_E T') \approx 0$. In fact, for the sake of simplicity, we assume here that Eve should obtain zero information about K, i.e.,

$$I(K; M_E T') = 0,$$

but the analysis can easily be generalized to a setting where Eve is allowed to obtain some minimal amount of information about K. The lower bound result changes only marginally.

We prove the following result, which shows that the practicality of such an approach without shared initial key is inherently limited. Alice or Bob must have storage capacity $\sqrt{s}$. The proof is given in Section 3.3.

Theorem 1. *For any key-agreement protocol secure in the BSM with no initial key for which $I(K; M_E T') = 0$, the entropy $H(K)$ of the generated secret key K is upper bounded by*

$$H(K) \leq \frac{s_A s_B}{s},$$

where s_A and s_B are the storage requirements for Alice and Bob, respectively, and s is the assumed storage bound for Eve.

[2] Here we assume that Alice and Bob generate the same key K, but this is of course a requirement of the scheme. The results can easily be generalized to a setting where the two key values must agree only with high probability.

We note that this bound also implies a bound on the memory of the adversary in the protocol for the oblivious transfer in the bounded-storage model.[3] Namely, if the memory of the honest parties is s_A then the memory of a cheating party has to be much smaller than s_A^2. This shows that the protocol of [8] is essentially optimal and answers the question posted in [8, 9].

3.2 The Cachin-Maurer Scheme

Indeed, as shown in [3], key agreement can be possible in such a BSM setting where Alice and Bob share no secret initial key . In this scheme, Alice and Bob each stores an (independent) random subset (with pairwise independent indices) of the bits of R. After R has disappeared for all parties, they publicly check which bits they have stored. Eve has only partial information about these bits (with overwhelming probability), no matter what she has stored. Therefore Alice and Bob can use privacy amplification using an extractor to distill an essentially perfect key K.

In this scheme, due to the birthday paradox, the number of bits stored by Alice and Bob must be greater than $\sqrt{t}$ since otherwise the number of bits known to both Alice and Bob would be very small with high probability. This also shows that for s on the order of t, the scheme of [3] has parameters close to the lower bound given by Theorem 1.[4]

3.3 Proof of Theorem 1

We first need the following information-theoretic lemma.

Lemma 1. *Consider a random experiment with random variables $Y, Z, Z_1, \ldots, Z_n$ such that conditioned on Y, the variables $Z, Z_1, \ldots, Z_n$ are independent and identically distributed, i.e.,*

$$P_{ZZ_1,\ldots,Z_n|Y}(z, z_1, \ldots, z_n, y) = P_{Z|Y}(z, y) \prod_{i=1}^{n} P_{Z|Y}(z_i, y)$$

for all $y, z, z_1, \ldots, z_n$ and for some conditional probability distribution $P_{Z|Y}$.[5] *Then*

$$I(Y; Z|Z_1 \cdots Z_n) \ \leq \ \frac{H(Y)}{n+1}.$$

---

[3] This is because there exists a black-box reduction of the key-agreement problem to the oblivious transfer problem [12]. (It is easy to see that the reduction of [12] works in the bounded-storage model.)

[4] It should be mentioned that the security analysis given in [3] is quite weak, but this could potentially be improved by a better scheme or a tighter security analysis.

[5] In other words, $Z, Z_1, \ldots, Z_n$ can be considered as being generated from Y by sending Y over $n+1$ independent channels specified by $P_{Z|Y}$, i.e.,

$$P_{Z|Y}(z, y) = P_{Z_1|Y}(z, y) = \cdots = P_{Z_n|Y}(z, y)$$

for all y and z.

Proof. The random experiment has a strong symmetry between the random variables $Z, Z_1, \ldots, Z_n$, both when considered without Y, and also when considered conditioned on Y. Any information-theoretic quantity involving some of the random variables $Z, Z_1, \ldots, Z_n$ (and possibly Y) depends only on how many of these random variables occur, but not which ones. Let therefore $H(u)$ denote the entropy of (any) u of the random variables $Z, Z_1, \ldots, Z_n$. Similarly, we can define $H(u|v)$ as the conditional entropy of u of them, given any other v of them. The quantities $H(Y, u|v)$, $H(u|Y, v)$, and $I(Y; u|v)$ can be defined analogously. We refer to [5] for an introduction to information theory.

In this notation, the lemma states that

$$I(Y; 1|n) \leq \frac{H(Y)}{n+1}.$$

The chain rule for conditional information[6] implies that

$$I(Y; n+1) = \sum_{i=0}^{n} I(Y; 1|i). \tag{1}$$

We next show that

$$I(Y; 1|i) \leq I(Y; 1|i-1). \tag{2}$$

This can be seen as follows:

$$\begin{aligned} I(Y; 1|i-1) - I(Y; 1|i) &= H(Y, i-1) + H(i) - H(Y, i) - H(i-1) \\ &\quad -(H(Y, i) + H(i+1) - H(Y, i+1) - H(i)) \\ &= \underbrace{2H(i) - H(i-1) - H(i+1)}_{=I(1;1|i-1)} \\ &\quad - \underbrace{(2H(Y, i) - H(Y, i-1) - H(Y, i+1))}_{I(1;1|Y,i-1)=0} \\ &\geq 0 \end{aligned}$$

The first step follows from

$$I(U; V|W) = H(UW) + H(VW) - H(UVW) - H(W),$$

the second step by rearranging terms, and the last step since $I(1; 1|i-1) \geq 0$ but $I(1; 1|Y, i-1) = 0$. This last fact follows since when given Y, any disjoint sets of Z-variables are independent.

Now using $I(Y; n+1) \leq H(Y)$ and combining (1) and (2) completes the proof since the right side of (2) is the sum of $n+1$ terms, the smallest of which is $I(Y; 1|n)$. □

[6] Recall that the chain rule for information (see eg. [5], Theorem 2.5.2) states that for arbitrary random variables $V_1, \ldots, V_n$, and U we have

$$I(U; V_1, \ldots, V_n) = \sum_{i=1}^{n} I(U; V_i|V_{i-1}, \ldots, V_1)$$

To prove Theorem 1, recall that s_A, s_B and s, are the storage capacities of Alice, Bob, and Eve, respectively. We have to specify a strategy for Eve to store information (i.e., the function f_E). Such an admissible strategy is the following: For the fixed observed randomizer $R = r$ and transcript $T = t$, Eve generates $\lfloor s/s_B \rfloor$ independent copies of what Bob stores, according to the distribution $P_{M_B|R=r,T=t}$. In other words, Eve plays, independently, $\lfloor s/s_B \rfloor$ times the role of Bob. We denote Eve's stored information (consisting of β parts) as M_E. The above lemma implies that

$$I(M_A; M_B | M_E) \leq \frac{H(M_A)}{\left\lfloor \frac{s}{s_B} \right\rfloor + 1} \leq \frac{H(M_A)}{\frac{s}{s_B}} \leq \frac{s_A s_B}{s}.$$

The last step follows from $H(M_A) \leq s_A$. Now we can apply Theorem 3 in [16] which considers exactly this setting, where Alice, Bob, and Eve have some random variables M_A, M_B, and M_E, respectively, jointly distributed according to some distribution $P_{M_A M_B M_E}$. The theorem states that the entropy of a secret key K that can be generated by public discussion is upper bounded as

$$H(K) \ \leq \ \min\big(I(M_A; M_B), I(M_A; M_B|M_E)\big),$$

i.e., in particular by $I(M_A; M_B|M_E)$. This concludes the proof. □

4 The Hybrid Model

As described in Section 1.2, some authors have suggested that one can securely combine the BSM with a (computationally secure) public-key agreement scheme KA used to generate the initial key K. We call this model the *hybrid model*. The motivation for the hybrid model is to achieve provable security under the sole assumption that Eve cannot break the key agreement scheme *during* the storage phase, even if afterwards she may gain infinite computing power, or for some other reason might be able to break the key agreement scheme. The BSM can hence potentially be used to preserve the security of a computationally secure scheme for eternity. The reason is that because if Eve has no information about K during the storage phase, she has missed any opportunity to know anything about the derived key X, even if she later learns K. Note that in the standard BSM Eve learns K by definition, but in the hybrid scheme she may at a later stage learn it because she can possibly break the key agreement scheme based on the stored transcript.

We show that this very intuitive and apparently correct reasoning is false by giving an example of a secure (according to the standard definition) computational key-agreement scheme for which the BSM-scheme is nevertheless completely insecure.

The hybrid model can be formalized as follows. During the first phase, Eve is computationally bounded (typically a polynomial bound), and Alice and Bob carry out a key agreement protocol, resulting in transcript T. Eve performs an efficient computation on R and T (instead of performing an unbounded computation on R alone), and stores the result U of the computation (which is again bounded to be at most s bits). Then she loses access to R and obtains infinite computing power. Without much loss of generality we can assume that she stored T as part of U and hence she can now compute K.

Theorem 2. *Assume that a computationally secure PIR scheme*[7] *exists, and assume its communication complexity is at most $l^{2/3}$, where l is the size of the database. Then there exists a key-expansion function f secure in the standard BSM but insecure in the hybrid model, for the same bound on Eve's storage capacity.*

Clearly, the scheme of [13] (described in Section 2.2) satisfies the requirements stated in the theorem. The key-expansion function f (whose existence we claim in the theorem) can be basically any key-expansion function proven secure in the literature.

Proof (of Theorem 2). We are going to construct a (rather artificial) key agreement scheme KA such that f is not secure in the hybrid model. To construct KA we will use an arbitrary computationally secure key agreement scheme KA$'$. In [12, 6] it was shown that the existence of computationally secure PIR schemes implies the existence of a key agreement scheme. Therefore we can assume that such a scheme exists (since we assume a secure PIR scheme). It is also reasonable to assume that the communication complexity of this scheme is small when compared to the size of the randomizer. One can also have in mind a concrete key-agreement scheme, for instance the Diffie-Hellman protocol, in which case the transcript consists of g^x and g^y (for x and y chosen secretly be Alice and Bob, respectively) and the resulting shared key is g^{xy}. This protocol is secure under the so-called decision Diffie-Hellman assumption.

Let us fix some PIR scheme. For the key-expansion we will use an arbitrary function f (secure in the BSM) with the property that the number m of bits accessed by the honest parties is much smaller than the total length t of the randomizer, say $m \leq t^{1/4}$ (without essential loss of generality as any practical scheme satisfies this). We assume that f is secure in the BSM against an adversary who can store at most $s = t/2$ bits. An example of a function satisfying these requirements is the function f of [11] (for a specific choice of the parameters).

In our scenario Eve will be able (at the end of the second phase) to reconstruct each bit accessed by the honest parties. The basic idea is to execute m times independently and in parallel the PIR query protocol. More precisely the protocol KA is defined as follows:

1. Alice and Bob invoke the given key-agreement scheme KA$'$. Let K be the agreed key and let T' be the transcript of the key agreement scheme.

[7] as defined in Section 2.2

2. Let $\kappa_1, \ldots, \kappa_m$ be the indices of the bits in the randomizer that are accessed by the parties (for the given initial key K and the BSM scheme f). Alice sends to Bob a sequence $\mathcal{Q}(\kappa_1), \ldots, \mathcal{Q}(\kappa_m)$ of m PIR queries, where each query is generated independently (with fresh random coins).
3. Alice and Bob (locally) output K as their secret key.

It is not hard to see that the security of KA$'$ and the privacy of PIR imply the security of KA. Step 2 is an artificial extension of KA$'$ needed to make the resulting scheme insecure in the hybrid BSM, i.e., to encode into the transcript some useful information that can be used by Eve. Her strategy (for a given transcript of KA and a randomizer R) is as follows. In the first phase she computes the answers to the queries („acting" as a database). She does not send them anywhere, but simply stores them in her memory. She also stores the queries and the transcript T' of the key-agreement scheme KA$'$. In other words:

$$U \ := \ ((\mathcal{Q}(\kappa_1), \mathcal{R}(\mathcal{Q}(\kappa_1), R)), \ldots, (\mathcal{Q}(\kappa_m), \mathcal{R}(\mathcal{Q}(\kappa_m), R), T).$$

(where R denotes the randomizer). Since the PIR is efficient, so is Eve's computation. Because of the communication efficiency of the PIR scheme and of the key-agreement protocol KA$'$, the length of U is at most $m \cdot t^{2/3} + |T|$, which is at most $t^{1/4+2/3} + |T|$. Since $|T|$ is much smaller than t, this value has to be smaller than $s = \frac{1}{2} \cdot t$ for sufficiently large t.

In the second phase the adversary can easily compute (from T') the value of K and therefore she can obtain $\kappa_1, \ldots, \kappa_m$. For every i she also knows $(\mathcal{Q}(\kappa_i, R))$, thus she can (using the unlimited computing power) compute the bit of R at position κ_i.[8] Therefore she can compute the value of $f(K, R)$. □

5 Discussion

One of the surprising consequences of Theorem 2 is that existing definitions for the computational security of key-agreement and encryption are too weak to cover settings in which the adversary's computing power may change over time, as is the case in real life. We thus need a new security definition of a key-agreement scheme and a public-key cryptosystem which implies, for example, the security in the BSM, as discussed above. It is quite possible that existing schemes such as the Diffie-Hellman protocol satisfy such a stronger security definition, and that only artificial schemes as the one described in the proof of Theorem 2 fail to be secure according to the stronger definition. It is an interesting problem to formalize in a more general context how and when security can be preserved even though a scheme gets broken a a certain point in time.

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.

[8] Recall that in the definition of a PIR scheme (Section 2.2) we assumed that the values $i, \mathcal{Q}(i), \mathcal{R}(\mathcal{Q}(i), x)$ determine the value of x_i.

2. Y. Aumann and M. O. Rabin. Information theoretically secure communication in the limited storage space model. In *Advances in Cryptology - CRYPTO '99*, pages 65–79, 1999.
3. C. Cachin and U. Maurer. Unconditional security against memory-bounded adversaries. In *Advances in Cryptology - CRYPTO '97*, pages 292–306, 1997.
4. B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. *Journal of the ACM*, 45(6):965–981, 1998.
5. T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley and Sons, Inc., 1991.
6. G. Di Crescenzo, T. Malkin, and R. Ostrovsky. Single database private information retrieval implies oblivious transfer. In *Advances in Cryptology - EUROCRYPT 2000*, pages 122–138, 2000.
7. W. Diffie and M.E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.
8. Y. Z. Ding. Oblivious transfer in the bounded storage model. In *Advances in Cryptology - CRYPTO 2001*, pages 155–170, 2001.
9. Y. Z. Ding. *Provable Everlasting Security in the Bounded Storage Model*. PhD thesis, Harvard University, 2001.
10. Y. Z. Ding and M. O. Rabin. Hyper-encryption and everlasting security. In *STACS 2002, 19th Annual Symposium on Theoretical Aspects of Computer Science*, pages 1–26, 2002.
11. S. Dziembowski and U. Maurer. Tight security proofs for the bounded-storage model. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 341–350, 2002.
12. Y. Gertner, S. Kannan, T. Malkin, O. Reingold, and M. Viswanathan. Relationship between public key encryption and oblivious transfer. In *41st Annual Symposium on Foundations of Computer Science*, pages 325–339, 2000.
13. E. Kushilevitz and R. Ostrovsky. Replication is not needed: Single database, computationally-private information retrieval. In *38th Annual Symposium on Foundations of Computer Science*, pages 364–373, 1997.
14. C. Lu. Hyper-encryption against space-bounded adversaries from on-line strong extractors. In *Advances in Cryptology - CRYPTO 2002*, pages 257–271, 2002.
15. U. Maurer. Conditionally-perfect secrecy and a provably-secure randomized cipher. *Journal of Cryptology*, 5(1):53–66, 1992.
16. U. Maurer. Secret key agreement by public discussion. *IEEE Transactions on Information Theory*, 39(3):733–742, 1993.
17. S. Vadhan. On constructing locally computable extractors and cryptosystems in the bounded storage model. In *Advances in Cryptology - CRYPTO 2003*, pages 61–77, 2003.