# Leakage-Resilient Cryptography in the Standard Model

Stefan Dziembowski
University of Rome
*La Sapienza*

Krzysztof Pietrzak
CWI Amsterdam

May 28, 2008

**Abstract**

We construct a stream-cipher $\mathsf{S}$ whose *implementation* is secure even if arbitrary (adversely chosen) information on the internal state of $\mathsf{S}$ is leaked during computation. This captures *all* possible side-channel attacks on $\mathsf{S}$ where the amount of information leaked in a given period is bounded, but overall can be arbitrary large, in particular much larger than the internal state of $\mathsf{S}$. The only other assumption we make on the *implementation* of $\mathsf{S}$ is that only data that is accessed during computation leaks information. The construction can be based on any pseudorandom generator, and the only computational assumption we make is that this PRG is secure against non-uniform adversaries in the classical sense (i.e. when there are no side-channels).

The stream-cipher $\mathsf{S}$ generates its output in chunks $K_1, K_2, \ldots$, and arbitrary but bounded information leakage is modeled by allowing the adversary to adaptively chose a function $f_\ell : \{0,1\}^* \to \{0,1\}^\lambda$ before $K_\ell$ is computed, she then gets $f_\ell(\tau_\ell)$ where $\tau_\ell$ is the internal state of $\mathsf{S}$ that is accessed during the computation of $K_\ell$. One notion of security we prove for $\mathsf{S}$ is that $K_\ell$ is indistinguishable from random when given $K_1, \ldots, K_{\ell-1}$, $f_1(\tau_1), \ldots, f_{\ell-1}(\tau_{\ell-1})$ and also the complete internal state of $\mathsf{S}$ after $K_\ell$ has been computed (i.e. our cipher is forward-secure).

The construction is based on alternating extraction (previously used in the intrusion-resilient secret-sharing scheme from FOCS'07). We move this concept to the computational setting by proving a lemma that states that the output of any PRG has high HILL pseudoentropy (i.e. is indistinguishable from some distribution with high min-entropy) even if arbitrary information about the seed is leaked. The amount of leakage $\lambda$ that we can tolerate in each step depends on the strength of the underlying PRG, it is at least logarithmic, but can be as large as a constant fraction of the internal state of $\mathsf{S}$ if the PRG is exponentially hard.

# 1 Introduction

When analyzing the security of a cryptosystem, we can either think of the system as a mathematical object, or try to analyze the security of an actual implementation. Traditionally, cryptographers have mostly considered the former view and analyzed the security of the mathematical object, and it is generally believed that our current knowledge of cryptography suffices to construct schemes that, when modeled in this way, are extremely secure. On a theoretical side, we know how to construct secure primitives under quite weak complexity-theoretic assumptions, for example (symmetric) encryption can be based on any one-way function [15]. Also from the practical perspective, the currently used constructions have very strong security properties, e.g. after 30 years of intensive cryptanalytic efforts still the most practical attack on the DES cipher is exhaustive key search.

**Side-Channel Attacks.** The picture is much more gloomy when the security of *real-life implementations* is considered. This is because when considering an implementation of a cryptosystem, one must take into account the possibility of side-channels, which refers to leakage of any kind of information from the cryptosystem during its execution which cannot be efficiently derived from access to the mathematical object alone. In the last decade many attacks against cryptosystems (still assumed to by sound as mathematical objects) have been found exploiting side-channels like running-time [20], electromagnetic radiation [27, 13], power consumption [21], fault detection [4, 3] and many more (see e.g. [28, 25]).

A typical countermeasure against this type of attacks is to design hardware that minimizes the leakage of secret data (e.g. by shielding any electromagnetic emissions), or to look for an algorithm-specific solution, for example by masking intermediate variables using randomization (see [25] for a list of relevant papers). The problem with hardware-based solutions is that protection against all possible types of leakage is very hard to achieve [1], if not impossible. On the other hand, most algorithm-specific methods proposed so far are only heuristic and do not offer any formal security proof (we mention some exceptions in Sect.1.1). Moreover, they are ad-hoc in the sense that they protect only against some specific attacks that are known at the moment, instead of providing security against a large well-defined class of attacks. This raises the following, natural question: is there a systematic method of designing cryptographic schemes so that already their mathematical description guarantees that they are provably-secure, even if they are implemented on hardware that may be subject to a side-channel attack belonging to a large well-defined class of attacks? Ideally, one would like to develop a theory that (1) provides precise definition of such a class of attacks, and (2) shows how to construct systems that are secure in this model (under the assumptions that are as weak as possible). This should be viewed as moving the task of constructing cryptosystems secure against side-channel attacks from the realm of engineering or security research to cryptography, which over the last 3 decades was extremely successful in defining security models, and constructing cryptosystems that are provably-secure in this models.

**General Model for Leakage Resilience.** We propose a model for cryptographic computation where the class of possible side-channel attacks is extremely broad, yet simple and natural. Models similar to ours have been proposed before, in particular Micali and Reyzin [23] explicitly stated the "only computation leaks" assumption we will use. The only other assumption on the *implementation* we make is that the amount of leakage in each round is bounded. This approach is inspired by the bounded-storage and bounded-retrieval models and has to best of our knowledge never been used in this context. We stress however, that the main contribution of this paper is not the definition of the model, but the construction of an actual cryptosystem (a stream-cipher) which is provably secure in this model, details follow.

Consider a cryptosystems $\mathsf{CS}$, let $\mathcal{M}$ denote its memory and $\mathcal{M}^0$ denote the data initially on $\mathcal{M}$ (i.e. the secret key). Clearly the most general side-channel attack against a cryptosystem $\mathsf{CS}(\mathcal{M}^0)$ is one in which the adversary can choose any polynomial-time computable *leakage function* $f$ and retrieve $f(\mathcal{M}^0)$ from the cryptographic machine.[1] Of course no security is achievable in this setting, as defining $f(\mathcal{M}^0) = \mathcal{M}^0$ the adversary learns the complete random key. Thus a necessary restriction we must make on $f$ is that its output range is bounded to $\{0,1\}^\lambda$ where $\lambda \ll |\mathcal{M}^0|$.

We assume that the adversary can apply this attack many times throughout the lifetime of the device. Technically, this will be done by dividing the execution of the algorithm implementing $\mathsf{CS}$ into *rounds*, and allowing the adversary to evaluate a function on the internal state of $\mathsf{CS}$ in each of those rounds (let $f_j$ denote the leakage function that she chooses in the $j$th round, for $j = 1, 2...$). Almost all cryptographic tasks can be divided into rounds in a natural way: in particular, in this paper we will construct a stream cipher that outputs data in chunks of a few bits, and each chunk will be computed in a separate round.

Let $q$ be the number of rounds we want our cryptosystem $\mathsf{CS}$ to run, and let $\mathcal{M}^0$ be the secret key that is used in the scheme. At first sight one may think that to hope for any security we would need to assume that $q \cdot \lambda < \left|\mathcal{M}^0\right|$, as otherwise the adversary can learn the entire $\mathcal{M}^0$, by just retrieving in every round $\lambda$ different bits of it. This trivial attack does not work any more if we consider cryptosystems which occasionally update their state. For this let $\mathcal{M}^j$ denote the state of $\mathsf{CS}$ after round $j$.

Unfortunately, no security is possible even if we allow $\mathsf{CS}$ to update its state (i.e. when $\mathcal{M}^j$ is not necessarily equal to $\mathcal{M}^{j+1}$) if we allow *any* $f_j$, to see this let $t = \lceil |\mathcal{M}|/\lambda \rceil$ and consider $f_j, j \leq t$ where each $f_j$ outputs different $\lambda$ bits of $\mathcal{M}^t$ (note that the function $f_j, j \leq t$ can compute the future state $\mathcal{M}^t$ from the current state $\mathcal{M}^j$). After the $t$th round the adversary has learned the complete state $\mathcal{M}^t$, and no security is possible beyond this point. We call this the *key-precomputation attack*.

Hence, we have to somehow restrict the leakage function if we want security even when the amount of leaked information is (much) larger than the internal state. The restriction that we will use is that in each round, the leakage function $f_j$ only gets as input the part of the state $\mathcal{M}^j$ that is actually accessed in the $j$th round by $\mathsf{CS}$. This translates into a requirement on the implementation: we assume that only computation leaks information, and the "untouched memory cells" are completely secure. As illustrated in Figure 1, in our construction of a stream-cipher, $\mathcal{M}$ will consists of just three parts $\mathcal{M}_0, \mathcal{M}_1$ and $\mathcal{O}$ (where $\mathcal{O}$ is the output tape), and in the $j$th round the circuit $\mathsf{CS}$ (and thus the leakage function $f_j$) will access only $\mathcal{M}_{j \bmod 2}$ and $\mathcal{O}$ . We give the leakage function (in the $j$th round) access to the complete $\mathcal{M}_{j \bmod 2}, \mathcal{O}$, even if the computation of $\mathsf{CS}$ only access a small part of it. Thus in an actual implementation, one only must ensure that in the $j$th round $\mathcal{M}_{j+1 \bmod 2}$ does not leak. This requirement should easily be realizable by an actual implementation having $\mathcal{M}_0$ and $\mathcal{M}_1$ use different static memory cells (here "static" refers to the fact that this memory needs not to be refreshed, and thus should not leak any kind of radiation when not used).[2]

Let us mention that the above restriction is not the only natural restriction that one could make on the leakage functions to avoid the key-precomputation attack. One other option might be to allow the state to be refreshed using external randomness, which would make the key-evolution non-deterministic. Though, this option is a bit difficult to handle for many

[1] Without loss of generality we can assume that the leakage function is applied only to $\mathcal{M}^0$ since all the other internal variables used in computation are deterministic functions of $\mathcal{M}^0$.

[2] Let us mention that this model also covers the case where (the not accessed) $\mathcal{M}_{j+1 \bmod 2}$ does leak in round $j$, as long as this leakage is independent of the leakage of (the accessed) $\mathcal{M}_{j \bmod 2}$ (i.e. when we consider an adversary $\mathsf{Q}'$ who can in round $j$ choose two functions $f'_j$ and $f''_j$ and then gets $f'_j(\mathcal{M}_{j \bmod 2})$ and also $f''_j(\mathcal{M}_{j+1 \bmod 2})$). The reason is that we can simulate $\mathsf{Q}'$ by an adversary $\mathsf{Q}$ who just chooses one function $f_j$ which outputs $f'_j(\mathcal{M}_{j \bmod 2})$ and also $f''_{j+1}(\mathcal{M}_{j \bmod 2})$ (thus $\mathsf{Q}$ in round $j$ simply precomputes the information that $\mathsf{Q}'$ will learn in round $j+1$ on the non-leaking part). Note that it's not a problem that $\mathsf{Q}'$ might compute $f''_{j+1}$ adaptively as a function of the information leaked in round $j$, as the leakage function $f_j$ has this information too, and thus can compute the $f''_{j+1}$ that $\mathsf{Q}'$ would have chosen.

cryptosystems including ciphers, as here one would have to make sure all legitimate parties knowing the secret key use the same randomness (and this randomness cannot be agreed upon initially). Another option is to require that the leakage function is in some very weak complexity class not including the function used for key evolution.[3]

**Leakage Resilient Stream-Cipher.** The main contribution of this paper is the construction of a stream cipher $\mathsf{S}$ which is provably secure in the model described above. Let $\tau_\ell$ denote the data on $\mathsf{S}$'s memory which is accessed in the $\ell$th round, and let $K_\ell$ denote the output written by $\mathsf{S}$ on its output tape $\mathcal{O}$ in the $\ell$th round.

The classical security notion for stream ciphers implies that one cannot distinguish $K_\ell$ from a random string given $K_1, \ldots, K_{\ell-1}$, of course our construction satisfies this notion. But we prove much more, namely that $K_\ell$ is indistinguishable from random even when not only given $K_0, \ldots, K_{\ell-1}$, but additionally $\Lambda_1, \ldots, \Lambda_{\ell-1}$ where $\Lambda_j = f_j(\tau_j)$ and each $f_j$ is a function with range $\{0,1\}^\lambda$ chosen adaptively (as a function of $K_1, \ldots, K_{\ell-1}, \Lambda_1, \ldots, \Lambda_{j-1}$) by an adversary. If the adversary also gets $\Lambda_\ell$, we cannot hope that $K_\ell$ is indistinguishable from random any more, as $f_\ell$ could for example simply output the $\lambda$ first bits of $K_\ell$. The best we can hope for in this case, is that $K_\ell$ is unpredictable (or equivalently, has high HILL-pseudoentropy), in the full version of this paper we will show that for our construction this indeed is the case.

**Forward Security.** In many settings, it is not enough that $K_\ell$ is indistinguishable (or unpredictable) given the view of the adversary after round $\ell - 1$ as just described, but it should stay indistinguishable even if $\mathsf{S}$ leaks some information *in the future*. In our construction such "forward-security" comes up naturally, as the key $K_\ell$ is almost independent (in a computational sense) from the state of $\mathsf{S}$ after $K_\ell$ was output. Precise security definitions are given is Sect. 2.

**Our Construction.** The construction, as described in detail in Section 2.2, uses the idea of alternating extraction previously used in the intrusion-resilient secret-sharing scheme from [12]. We move this concept to the computational setting by proving a lemma that states that the output of any PRG has high HILL pseudoentropy (i.e. is indistinguishable from some distribution with high min-entropy) even if arbitrary information about the seed is leaked. Our construction can be instantiated with any pseudorandom-generator, and the amount of leakage $\lambda$ that we can tolerate in each step depends on the strength of the underlying PRG, it is at least logarithmic, but can be as large as a constant fraction of the internal state of $\mathsf{S}$ if the PRG is exponentially secure.

**On (Non-)Uniformity.** Throughout the paper, we always consider non-uniform adversaries.[4] In particular, our stream-cipher is secure against non-uniform adversaries, and we require the PRG used in the construction to be secure against non-uniform adversaries. The only step in the security proof where it matters that we are in a non-uniform setting, is in Section 5, where we use a theorem due to Barak et al. [2] which shows that two notions of pseudoentropy (called HILL and metric-type) are equivalent for circuits. In [2] this equivalence is also proved in a uniform setting, and one could use this to get a stream-cipher secure against uniform adversaries from any PRG secure against uniform adversaries. We won't do so, as for one thing the non-uniform setting is the more interesting one, and also the exact security we could get

[3] Interestingly, that would probably be the first case of a real-life cryptographic application where it makes sense to assume that the computational power of the adversary (in some parts of the attack scenario) is smaller than the computational power needed to execute the scheme. Hence, one could hope to obtain results that do not require any additional intractability assumptions (except, say, that one-way functions exist), since the hierarchy results separating such complexity classes are known to hold unconditionally).

[4] Recall that a uniform adversary can be modelled as a single Turing-machine which as input gets a security parameter, whereas (more powerful) non-uniform adversaries are modelled by a sequence of circuits indexed by the security parameter.

in the uniform setting is much worse (due to the security loss in the reduction from [2] in the uniform setting).

## 1.1 Related work

A general theory of side-channel attacks was put forward in [23], where the authors propose a number of "axioms" on which such a theory should be based. In particular they formulate the assumption, that we also use in this work, that only computation leaks information. As mentioned in the introduction, most published work on securing cryptosystems against side-channel attacks are ad-hoc solutions trying to prevent some particular attack or some general heuristics coming without security proofs. A notable exception is the work of Ishai et al. [17, 16], who propose a general way of making circuits provably secure [17] and even tamper resistant [16] against adversaries who can read/tamper the value of a bounded number of arbitrary wires in the circuit (the more restricted model where an adversary can learn the value of some *input* bits has been extensively investigated [5, 8, 18]). It is interesting to compare the result from this paper with the approach of Ishai et al. On one hand, their results are generic, in the sense that they provide a method to transform any cryptosystem given as a circuit $C$ into another circuit $C_t$ that is secure against an adversary that can read-off up to $t$ wires, whereas we only construct a particular primitive (a stream-cipher). On the other hand, we prove security against any side-channel attack, whereas Ishai et al. consider the particular case where the adversary can read-off the values of a few individual wires. Moreover Ishai et al. require special gates that can generate random bits, we do not assume any special hardware.

The idea to define the set of leakage functions by restricting the length of function's output is taken from the bounded-retrieval model [7, 10, 9, 6, 12], which in turn was inspired by the bounded-storage model [22].[5]

Some constructions of ciphers secure against general leakages were also proposed in the literature, however, their security proofs rely on very strong assumptions. For example [26] works only in the ideal-cipher model, and [23] reduce the existence of such a cipher to one-way permutations which do not leak any information at all.

## 1.2 Notation

We denote with $U_n$ the random variable with distribution uniform over $\{0,1\}^n$. With $X \sim Y$ we denote that $X$ and $Y$ have the same distribution. Let random variables $X_0, X_1$ be distributed over some set $\mathcal{X}$ and let $Y$ be a random variable distributed over $\mathcal{Y}$. Define the *statistical distance between* $X_0$ *and* $X_1$ as $\delta(X_0; X_1) = 1/2 \sum_{x \in \mathcal{X}} |P_{X_0}(x) - P_{X_1}(x)|$. Moreover let $\delta(X_0; X_1|Y) := d(X_0, Y; X_1, Y)$ be the *statistical distance between* $X_0$ *and* $X_1$ *conditioned on* $Y$. If $X$ is distributed over $\{0,1\}^n$ then let $d(X) := \delta(X; U_n)$ denote the *statistical distance of* $X$ *from a uniform distribution (over* $\{0,1\}^n$*)*, and let $d(X|Y) := \delta(X; U_n|Y)$ denote the statistical distance of $X$ from a uniform distribution, *given* $Y$. If $d(X) \leq \epsilon$ then we will say that $X$ is $\epsilon$-close to uniform.

[5] The bounded-storage model is limited in its usability by the fact that the secret key must be larger than the memory of a potential adversary, which means in the range of terabytes. In the bounded-retrieval model, the key must only be larger than the amount of data adversary can retrieve without being detected (say, by having a computer-virus send the data from an infected machine), which means in the range of Mega- or Gigabytes. Whereas in our setting the key length depends on the amount of side-channel information that leaks (in one round) form the cryptosystem considered, which (given a reasonable construction) we can assume to be as small as a few (or a few hundred) bits. In particular, unlike the bounded-storage and bounded-retrieval models, our keys need not to be made artificially huge.

## 2 A Leakage-Resilient Stream-Cipher

We will now formally define our security notions which we already informally discussed and motivated in Sect. 1.

**Initialization.** The secret key of our stream cipher $\mathsf{S}$ consists of the three variables $A, B \in \{0,1\}^r$ and $K_0 \in \{0,1\}^k$. The values $A, B, K_0$ should be sampled uniformly at random, but only $A, B$ must be secret, $K_0$ must not, one can think of $K_0$ as the first $k$ bits of output of $\mathsf{S}$. We also do not really need $A$ and $B$ to be uniformly random, they just must be independent and have sufficiently high min-entropy. In an implementation, the memory of $\mathsf{S}$ is assumed to be split in three parts, $\mathcal{M}_0, \mathcal{M}_1, \mathcal{O}$, and for $j > 0$ we denote with $\mathcal{M}_0^{j-1}, \mathcal{M}_1^{j-1}, \mathcal{O}^{j-1}$ the contents of $\mathcal{M}_0, \mathcal{M}_1, \mathcal{O}$ at the beginning of the $j$th round, in particular the initial state is $\mathcal{M}_0^0 = A, \mathcal{M}_1^0 = B$ and $\mathcal{O}^0 = K_0$.

**Computation.** As illustrated in Fig. 1, in the $j$th round $\mathsf{S}$ does only access (which means reads and possible rewrites) $\mathcal{M}_{j \bmod 2}$ and the output tape $\mathcal{O}$. Let $\tau_\ell$ denote the values (on either $\mathcal{M}_0$ or $\mathcal{M}_1$) that is accessed in the $\ell$th round, and $\overline{\tau}_\ell$ the value which is not accessed, i.e.

$$\tau_\ell \stackrel{\mathsf{def}}{=} \mathcal{M}_{\ell \bmod 2}^{\ell-1} \qquad \overline{\tau}_\ell \stackrel{\mathsf{def}}{=} \mathcal{M}_{\ell+1 \bmod 2}^{\ell-1} \tag{1}$$

We will refer to the output of the $\ell$th round (i.e. the value $\mathcal{O}^\ell$ on the output tape $\mathcal{O}$ at the end of this round) as $K_\ell$.

**Adversary.** As illustrated in Figure 1, we consider adversaries $\mathsf{Q}$ which in the $\ell$th round can adaptively choose a function $f_\ell$ with range $\{0,1\}^\lambda$, and at the end of the round gets the output $K_\ell$ and

$$\Lambda_\ell \stackrel{\mathsf{def}}{=} f_\ell(\tau_\ell)$$

i.e. the output of $f_\ell$ on input the data accessed by $\mathsf{S}$ in this round.

We denote with $\mathcal{A}_\lambda$ adversaries as just described restricted to choose leakage functions with range $\{0,1\}^\lambda$. As we consider non-uniform adversaries, we can without loss of generality assume that they are deterministic. Let $\mathtt{view}_\ell$ denote the view of the adversary after $K_\ell$ has been computed, i.e. $\mathtt{view}_\ell = \{K_0, \ldots, K_\ell, \Lambda_1, \ldots, \Lambda_\ell\}$.

**Indistinguishability.** The security notion we consider requires that $K_\ell$ is *indistinguishable* from random, even when given $\mathtt{view}_{\ell-1}$.

For an adversary $\mathsf{Q} \in \mathcal{A}_\lambda$ we denote with $\mathsf{S}(A, B, K_0) \stackrel{\ell}{\leadsto} \mathsf{Q}$ the random experiment where $\mathsf{Q}$ attacks $\mathsf{S}$ (initialized with a key $A, B, K_0$) for $\ell$ rounds (cf. Figure 1), and with $\mathtt{view}(\mathsf{S}(A, B, K_0) \stackrel{\ell}{\leadsto} \mathsf{Q})$ we denote the view $\mathtt{view}_\ell$ of $\mathsf{Q}$ at the end of the attack. For any circuit $\mathsf{D}_{\mathsf{ind}} : \{0,1\}^* \to \{0,1\}$ (with one bit output), we denote with $\mathsf{AdvInd}(\mathsf{D}_{\mathsf{ind}}, \mathsf{Q}, \mathsf{S}, \ell)$ the advantage of $\mathsf{D}_{\mathsf{ind}}$ in distinguishing $K_\ell$ from random given $\mathtt{view}(\mathsf{S} \stackrel{\ell-1}{\leadsto} \mathsf{Q})$, formally

$$\begin{aligned}
\mathsf{AdvInd}(\mathsf{D}_{\mathsf{ind}}, \mathsf{Q}, \mathsf{S}, \ell) &= |p_{real} - p_{random}| \qquad \text{where} \\
p_{random} &= \Pr_{A,B,K_0}[\mathsf{D}_{\mathsf{ind}}(\mathtt{view}(\mathsf{S}(A, B, K_0) \stackrel{\ell-1}{\leadsto} \mathsf{Q}), U_k) = 1] &(2) \\
p_{real} &= \Pr_{A,B,K_0}[\mathsf{D}_{\mathsf{ind}}(\mathtt{view}(\mathsf{S}(A, B, K_0) \stackrel{\ell-1}{\leadsto} \mathsf{Q}), K_\ell) = 1] &(3)
\end{aligned}$$

In the full version of this paper, we will also consider the case where the distinguisher also gets $\Lambda_\ell$, i.e. we assume that information leaks also in round $\ell$. Although then we can't hope for $K_\ell$ to be indistinguishable from random (as $\Lambda_\ell$ could for example be the first $\lambda$ bits of $K_\ell$), we still can require that $K_\ell$ cannot be completely guessed.

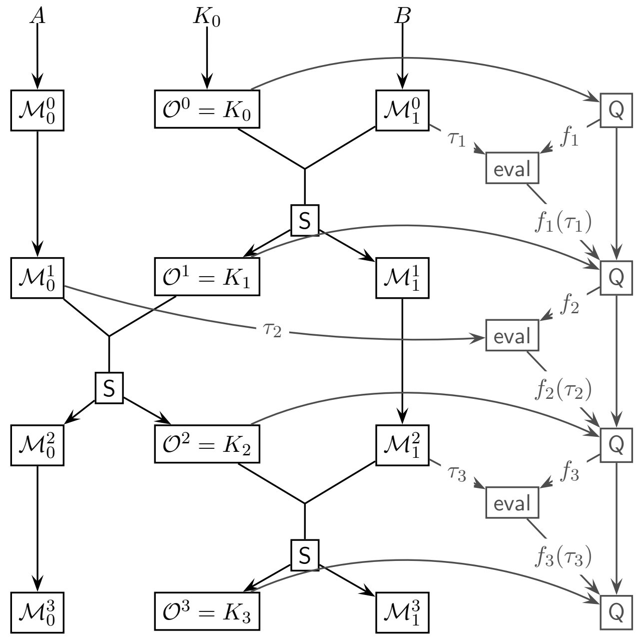


Figure 1: General structure of the random experiment $\mathsf{S}(A, K_0, B) \overset{3}{\rightsquigarrow} \mathsf{Q}$ (the regular evaluation of $\mathsf{S}$, generating output $K_1, K_2, \ldots$, is shown in black. The attack related part is shown in gray). Here $\mathsf{Q}$ is the adversary and $\mathsf{eval}$ denotes a circuit which on input the description of a function $f$ and some input $\tau$ outputs $f(\tau)$. The adversary can, before $K_i$ is computed, adaptively choose a leakage function $f_i : \{0,1\}^r \to \{0,1\}^\lambda$, after the $i$th round she then gets $K_i$ and $\Lambda_i \overset{\mathsf{def}}{=} f_i(\tau_i)$.

**Forward Security.** As motivated in the introduction, we'll also consider "forward-secure" notions of the above definition. Informally, we'd like to extend the definitions $\mathsf{AdvInd}$ just given, but additionally give the attacker $\mathsf{D}_{\mathsf{ind}}$ the complete state $\mathcal{M}_0^\ell, \mathcal{O}^\ell, \mathcal{M}_1^\ell$ of $\mathsf{S}$ after $K_\ell$ has been computed. Of course then $K_\ell = \mathcal{O}^\ell$ cannot be secure in any way as it is given to $\mathsf{D}_{\mathsf{ind}}$ entirely. We could simply not give $\mathcal{O}^\ell$ to $\mathsf{D}_{\mathsf{ind}}$, but then we cannot claim that we leaked the state of $\mathsf{S}$ completely, as in our construction $\mathcal{O}^\ell$ is needed to compute the future outputs of $\mathsf{S}$. There are at least two ways around this problem. We could relax our requirement on forward security, and not leak the state after round $\ell$, but only after round $\ell+1$ (in terms of the implementation, this would mean that the output $K_\ell$ is indistinguishable, if in rounds $\ell$ *and* $\ell+1$ nothing leaked, even given the complete state of $\mathsf{S}$ after round $\ell + 1$).

Another possibility, which we we'll use, is to split the value on the output tape into two parts $\mathcal{O}^\ell = K_\ell = K_\ell^{\mathsf{nxt}} \| K_\ell^{\mathsf{out}}$, such that only the $K_\ell^{\mathsf{nxt}}$ part is actually used by $\mathsf{S}$ to compute the future state. We then require that the $K_\ell^{\mathsf{out}}$ (and not the entire $K_\ell$) is indistinguishable from random if in round $\ell$ nothing leaked, even when given the state of $\mathsf{S}$ after round $\ell$, where $K_\ell^{\mathsf{out}}$ is not considered to be part of the state.

Let $\mathtt{state}_\ell := \{\mathcal{M}_0^\ell, K_\ell^{\mathsf{nxt}}, \mathcal{M}_1^\ell\}$ denote the state of $\mathsf{S}$ after round $\ell$ (not containing $K_\ell^{\mathsf{out}}$ as just explained). The forward secure indistingusihability notion is given by

$$\begin{aligned}
\mathsf{AdvIndFwd}(\mathsf{D}_{\mathsf{ind}}, \mathsf{Q}, \mathsf{S}, \ell) &= |p_{real}^{fwd} - p_{random}^{fwd}| \qquad \text{where} \\
p_{random}^{fwd} &= \Pr_{A,B,K_0}[\mathsf{D}_{\mathsf{ind}}(\mathtt{view}(\mathsf{S}(A, B, K_0) \overset{\ell-1}{\rightsquigarrow} \mathsf{Q}, \mathtt{state}_\ell), U_{|K^{\mathsf{out}}|}) = 1] \\
p_{real}^{fwd} &= \Pr_{A,B,K_0}[\mathsf{D}_{\mathsf{ind}}(\mathtt{view}(\mathsf{S}(A, B, K_0) \overset{\ell-1}{\rightsquigarrow} \mathsf{Q}, \mathtt{state}_\ell), K_\ell^{\mathsf{out}}) = 1]
\end{aligned}$$

Note that the only difference to $\mathsf{AdvInd}$ is that now $\mathsf{D}_{\mathsf{ind}}$ additionally gets $\mathtt{state}_\ell$, and we only require $K_\ell^{\mathsf{out}}$ (and not the whole $K_\ell$) to be indistinguishable. Thus, if forward security is an issue, one must always discard the first $K_\ell^{\mathsf{nxt}}$ part of $\mathsf{S}$'s output $K_\ell$. In our construction, $K_\ell^{\mathsf{nxt}}$

will be just a random seed for an extractor, using existing constructions we can make this part logarithmic in the total length of $K_\ell$, thus the efficiency loss one has to pay to get forward security is marginal.

### 2.1 The Ingredients

The main ingredients of our construction is the concept of alternating extraction introduced in the intrusion-resilient secret-sharing scheme of [12] (which again was based on ideas from the bounded storage model [11, 22, 29]), combined with the concept of HILL-pseudoentropy (cf. Def. 3, Sect. 5) which we use to get a *computational* version of alternating extraction.

**Alternating Extraction.** Let $\mathsf{ext} : \{0,1\}^{k_{\mathsf{ext}}} \times \{0,1\}^r \rightarrow \{0,1\}^k$ be an $(\epsilon_{\mathsf{ext}}, n_{\mathsf{ext}})$-extractor (cf. Def. 1, Sect. 4.1). Consider some uniformly random $A, B \in \{0,1\}^r$ and some random $K_0 \in \{0,1\}^k$. As illustrated in Figure 3 in Sect. 4, let $K_1, K_2, \ldots$ be computed as $K_i = \mathsf{ext}(K_{i-1}^{\mathsf{nxt}}, C_i)$ (where $K^{\mathsf{nxt}}$ denotes the $k_{\mathsf{ext}}$ first bits of $K$ and $C_i = B$ if $i$ is odd and $C_i = A$ otherwise). So the $K_i$'s are computed by alternately extracting from $A$ and $B$. It is not hard to show that $K_i = \mathsf{ext}(K_{i-1}^{\mathsf{nxt}}, C_i)$ is $i\epsilon_{\mathsf{ext}}$ close to uniformly random given $K_0, \ldots, K_{i-1}$ while $C_i$ has still enough min-entropy for our extractor (i.e. $\mathbf{H}_\infty(C_i | K_1, \ldots, K_{i-1}) \geq n_{\mathsf{ext}}$).

As shown in [12], the key $K_i$ is even close to uniformly random when not only given $K_1, \ldots, K_{i-1}$ but also some values $f_1(C_1), \ldots, f_{i-1}(C_{i-1})$ for arbitrary functions $f_i$ as long as $C_i$ has min-entropy at least $n_{\mathsf{ext}}$ (conditioned on $K_0, \ldots, K_{i-1}$, and $f_1(C_1), \ldots, f_{i-1}(C_{i-1})$).

Consider a "stream cipher" $\mathsf{S}^*(A, B, K_0)$ which outputs $K_1, K_2, \ldots$ computed as described above, and an adversary $\mathsf{Q}$ which, before $K_i$ is computed, can adaptively choose a function $f_i$ and then gets $K_i, f_i(C_i)$ (as $K_{i-1}$ can be hard-coded into $f_i$, this function has access to all the data accessed during the computation of $K_i = \mathsf{ext}(K_{i-1}^{\mathsf{nxt}}, C_i)$. As explained in the previous paragraph, we can give the following security guarantee for $\mathsf{S}^*$: as long as the min-entropy of $C_i$ is at least $n_{\mathsf{ext}}$ (given the adversary's view), the next output $K_i$ is close to uniformly random (given the view of the adversary so far).

**Pseudoentropy.** The stream cipher $\mathsf{S}^*$ just described is not very useful, as it only provides security (in the sense that the next output looks random given the current view as required by our $\mathsf{AdvInd}$ security notion) as long as the output (i.e. the $K_i$'s plus the leaked information) is shorter (by at least $n_{\mathsf{ext}}$ bits) than the initial key.

To get security beyond that bound, we will "refresh" the values $A, B$ after we extracted from them. Let $A_i = \mathcal{M}_0^i$ and $B_i = \mathcal{M}_1^i$ denote the values on $\mathcal{M}_0$ and $\mathcal{M}_0$ after round $i$ respectively. In round $i$ (we assume $i$ is odd, otherwise replace the role of $A$ and $B$) we extract $(K_i, X_i) = \mathsf{ext}(K_{i-1}^{\mathsf{nxt}}, B_{i-1})$, and use $X_i$ to compute the fresh $B_i := \mathsf{prg}(X_i)$ using a pseudorandom generator $\mathsf{prg}$ as illustrated in Figure 2. If at the beginning of the $i$th round $B_{i-1}$ has min-entropy at least $n_{\mathsf{ext}}$ (given the adversaries view), $K_{i-1}^{\mathsf{nxt}}$ is pseudorandom (given $B_i$) and we assume that during this $i$th round no information is leaked, then $X_i$, and thus also $B_i = \mathsf{prg}(X_i)$ is pseudorandom given the view of the adversary.

Of course assuming that the refreshing phase does not leak any information is completely unjustified, and we do not want to make such an assumption. As we give $\Lambda_i = f_i(B_i)$ to the adversary, we cannot hope for $B_i$ to be pseudorandom (just consider the case where $f_i(B_i)$ are the $\lambda$ first bits of $B_i$). Fortunately, $B_i$ needs not to be (pseudo)random to apply alternating extraction, all we need is that $B_i$ has high min-entropy. Of course $B_i = \mathsf{prg}(X_i)$ cannot even have min-entropy more than $X_i$, but as we consider computationally bounded adversaries, it is enough if $B_i$ is indistinguishable from some distribution with high min-entropy. Such a variable which is computationally indistinguishable from some variable with min-entropy $k$ is said to have HILL-pseudoentropy $k$. It is not hard to see that a pseudorandom value $B_i$ has high HILL-pseudoentropy when given $f_i(B_i)$ for some efficient function $f_i$, but this is not enough for our application, as the leakage function $f_i$ is given access to $B_{i-1}$ (and not just $B_i$), from which it can compute the seed $X_i$ used to compute $B_i = \mathsf{prg}(X_i)$. We will prove (Lemma 8) that for any pseudorandom generator $\mathsf{prg}$, the output of $\mathsf{prg}(X)$ on a random seed $X$ has high HILL-pseudoentropy even if some function (with sufficiently short output) of $X$ (and not only $\mathsf{prg}(X)$) is leaked.

Using this lemma, we can prove that refreshing using a PRG as just described actually works, and will result in a "fresh" value $B_i$ (or $A_i$ for even $i$) having high HILL-pseudoentropy.

### 2.2 The construction

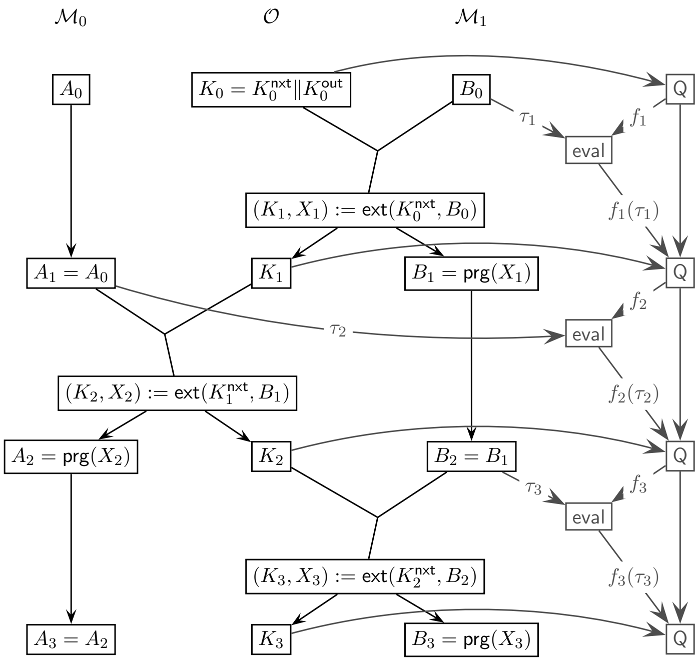


Figure 2: The random experiment $\mathsf{S}(A_0, B_0, K_0) \overset{3}{\rightsquigarrow} \mathsf{Q}$ for our construction of a stream cipher $\mathsf{S}$ as described in Section 2.2. The outputs $K_i = K_i^{\mathsf{nxt}} \| K_i^{\mathsf{out}}$ are computed by alternating extraction $(K_i, X_i) = \mathsf{ext}(K_{i-1}^{\mathsf{nxt}}, \tau_i)$ from the values in $\mathcal{M}_0$ and $\mathcal{M}_1$ and written on the output tape $\mathcal{O}$. The value $\tau_i$ (on $\mathcal{M}_{i \bmod 2}$) which is accessed in round $i$, is replaced with a "fresh" value computed as $\mathsf{prg}(X_i)$.

We will now formally define the construction just outlined, based on an extractor $\mathsf{ext} : \{0,1\}^{k_{\mathsf{ext}}} \times \{0,1\}^r \to \{0,1\}^{m_{\mathsf{ext}}}$ and a pseudorandom generator $\mathsf{prg} : \{0,1\}^{k_{\mathsf{prg}}} \to \{0,1\}^r$.

**State:** The state of $\mathsf{S}$ at the beginning of round $\ell$ is $\mathcal{M}_0^{\ell-1}, \mathcal{M}_1^{\ell-1}, \mathcal{O}^{\ell-1}$.

**Get Key:** Read $K_{\ell-1} = \mathcal{O}^{\ell-1}$ and parse it as $K_{\ell-1} = (K_{\ell-1}^{\mathsf{nxt}}, K_{\ell-1}^{\mathsf{out}}) \in \{0,1\}^{k_{\mathsf{ext}}} \times \{0,1\}^{k_{out}}$.

**Extract next Key and Seed:** Compute $\mathsf{ext}(K_{\ell-1}^L, \mathcal{M}_{\ell \bmod 2}^{\ell-1})$ and parse it as $(K_\ell, X_\ell) \in \{0,1\}^k \times \{0,1\}^{k_{\mathsf{prg}}}$.

**Write output:** Write $K_\ell$ on $\mathcal{O}$ (so $\mathcal{O}^\ell = K_\ell$).

**Refresh:** Compute $\mathsf{prg}(X_\ell)$ and write it on $\mathcal{M}_{\ell \bmod 2}$.

## 3 Security of $\mathsf{S}$

**Total Size.** We denote with $\mathtt{size}(\mathsf{D})$ the size size of the circuit $\mathsf{D}$. For an adversary $\mathsf{Q} \in \mathcal{A}_\lambda$, $\mathtt{size}(\mathsf{S} \overset{\ell}{\rightsquigarrow} \mathsf{Q})$ denotes the size of a circuit needed to implement the experiment $\mathsf{S} \overset{\ell}{\rightsquigarrow} \mathsf{Q}$.

**Theorem 1 (Main Result: Security of $\mathsf{S}$)** *Let* $\mathsf{ext} : \{0,1\}^{k_{\mathsf{ext}}} \times \{0,1\}^r \to \{0,1\}^{m_{\mathsf{ext}}}$ *be an* $(\epsilon_{\mathsf{ext}}, n_{\mathsf{ext}})$ *extractor, and let* $\mathsf{prg} : \{0,1\}^{k_{\mathsf{prg}}} \to \{0,1\}^r$ *be an* $(\epsilon_{\mathsf{prg}}, s_{\mathsf{prg}})$ *pseudorandom generator. Consider any* $\epsilon_{\mathsf{HILL}} > 0$ *and let* $\hat{s} \approx \epsilon^2_{\mathsf{HILL}} s_{\mathsf{prg}}/8r$.[6] *Consider any* $\epsilon_{\mathsf{gap}} > 0, \Delta > 0$ *where*

$$\epsilon_{\mathsf{prg}} \leq \frac{\epsilon^2_{\mathsf{gap}}}{2^\lambda} - 2^{-\Delta} \quad \textit{and} \quad n_{\mathsf{ext}} \leq r - \Delta - (\lambda + m_{\mathsf{ext}}) - 2\log(1/\epsilon_{\mathsf{gap}}) \tag{4}$$

*Then for all adversaries* $\mathsf{Q} \in \mathcal{A}_\lambda$ *and* $\mathsf{D}$ *where* $\mathtt{size}(\mathsf{S} \overset{\ell}{\rightsquigarrow} \mathsf{Q}) + \mathtt{size}(\mathsf{D}) \leq \hat{s}$ *with* $\delta_\ell \overset{\mathsf{def}}{=} \ell^2(3\epsilon_{\mathsf{gap}} + \epsilon_{\mathsf{HILL}} + \epsilon_{\mathsf{ext}})$

$$\mathsf{AdvInd}(\mathsf{D}, \mathsf{Q}, \mathsf{S}, \ell) \leq \delta_\ell \qquad \textit{and} \qquad \mathsf{AdvIndFwd}(\mathsf{D}, \mathsf{Q}, \mathsf{S}, \ell) \leq \delta_\ell \tag{5}$$

*We actually do not even need the initial key to* $\mathsf{S}$ *to be uniformly random, but only require a weaker condition as give by equations (27) and (28).*

The proof of Theorem 1 is split in three parts. The first part in Section 4 on alternating extraction is information theoretic and uses ideas from the intrusion-resilient secret-sharing scheme from [12]. In the second part (Section 5) we revisit some notions and results on computational pseudoentropy. We then prove that the output of any pseudorandom generator has high HILL pseudoentropy even if information about the seed is leaked. In Section 6 we prove Theorem 1 by using the result from Section 5 to get a computational version of alternating extraction from Section 4.

**How Much Leakage can we Tolerate?** The amount of leakage we can tolerate is bounded by (4) as $\epsilon_{\mathsf{prg}} \leq \epsilon^2_{\mathsf{gap}}/2^\lambda - 2^{-\Delta}$. For concreteness, assume we set $\Delta$ such that $2^{-\Delta} \leq \epsilon_{\mathsf{prg}}/2$ and $\epsilon_{\mathsf{gap}} \geq \sqrt[4]{\epsilon_{\mathsf{prg}}/4}$, then we can set

$$\lambda = \left\lfloor \frac{\log \epsilon^{-1}_{\mathsf{prg}}}{2} \right\rfloor$$

To see what this means it is convenient to take an asymptotic viewpoint and think of $\mathsf{S}$ as a *family* of stream ciphers indexed by a security parameter which we identify with $k_{\mathsf{prg}}$, i.e. the input length to $\mathsf{prg}$. If $\mathsf{prg}$ is secure against polynomial-size circuits, then $\epsilon_{\mathsf{prg}} = 2^{-\omega(\log k_{\mathsf{prg}})}$ (and thus $\lambda \in \omega(\log k_{\mathsf{prg}})$), and if $\mathsf{prg}$ is secure against exponential size circuits, then $\epsilon_{\mathsf{prg}} = 2^{-\Theta(k_{\mathsf{prg}})}$ (and $\lambda \in \Theta(k_{\mathsf{prg}})$).

Let us mention that already the case where $\lambda \in \omega(\log k_{\mathsf{prg}})$, i.e. the leakage is super-logarithmic, covers quite a large class of real-life attacks. In particular, many attacks that are based on measuring the power consumption result in logarithmic-size leakages. For example in a so-called *Hamming weight attack* (see e.g. [19]) the adversary just learns the number of wires carrying the bit 1. Of course this value is of logarithmic length in the size of the circuit, and hence also in $k_{\mathsf{prg}}$.

The case where $\mathsf{prg}$ is exponentially hard and thus $\lambda \in \Theta(k_{\mathsf{prg}})$ actually means that we can leak a constant fraction of the entire state of $\mathsf{S}$ (it needs some work to check that using existing constructions of extractors, one can set the parameters such that the entire state has size $\Theta(k_{\mathsf{prg}})$).

# 4 Random Keys by Alternating Extraction

We first prove an information theoretic result which is inspired by the security proof of the intrusion-resilient secret-sharing scheme from [12]. Basically, we consider the random experiment $\mathsf{S} \overset{\ell}{\rightsquigarrow} \mathsf{Q}$ but without the refreshing. For this let $\mathsf{S}^*$ denoted the construction $\mathsf{S}$ but without

[6] See Lemma 7 as to what $\hat{s}$ exactly is.

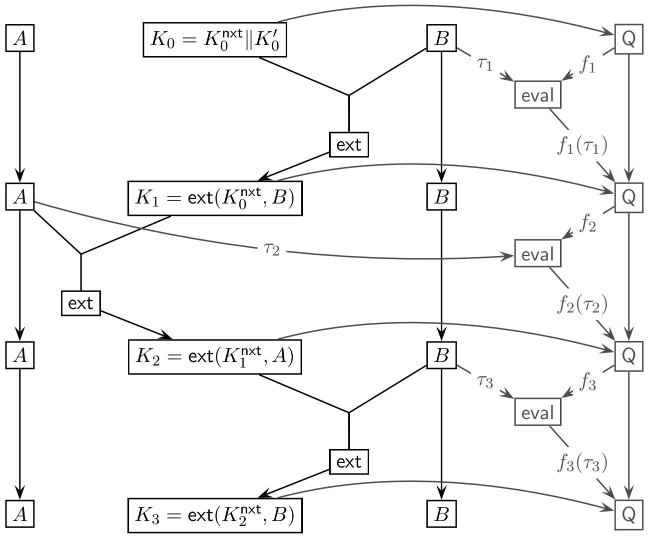


Figure 3: The "alternating extraction" random experiment $\mathsf{S}^*(A, B, K_0) \overset{3}{\rightsquigarrow} \mathsf{Q}$ as considered in Lemma 1.

refreshing: thus in the random experiment $\mathsf{S}^*(A, B, K_0) \overset{\ell}{\rightsquigarrow} \mathsf{Q}$ where $\mathsf{Q} \in \mathcal{A}_\lambda$, in the $j$th round Q chooses a function $f_j : \{0,1\}^r \to \{0,1\}^\lambda$ and as output gets $K_j = \mathsf{ext}(K_{j-1}^{\mathsf{nxt}}, \tau_j)$ and $\Lambda_j = f_j(\tau_j)$ where $\tau_j = B$ if $j$ is odd and $\tau_j = A$ otherwise.

As Q attacks $\mathsf{S}^*$, she learns information on $A$ and $B$, and thus the min-entropy of $A$ and $B$ degrades (recall that a variable $X$ has min-entropy $k$, denoted $\mathbf{H}_\infty(X) = k$, if $\max_x \Pr[X = x] = 2^{-k}$). We show that as long as the min-entropy of $A$ and $B$ is high enough (which means more than $n_{\mathsf{ext}}$ as required by the extractor ext), the next key $K_j$ to be output is close to uniformly random when given the view after $K_{j-1}$ has been computed.

Lemma 1 belows similar to Lemma 8 from [12] (for the special case of two players). One difference is that in [12] the variables $A, B, K_0$ were all independent and uniformly random, whereas in the Lemma below this is somewhat relaxed (cf. equation (6)). Further in [12] there was no conditioning on $\tau_\ell$ in equation (8). For the security proof of S, we'll only need the lemma for the special case $\ell = 1$, we prove this more general version as for one thing it's not significantly harder, and it can be used to prove the security of variations of S where one does not refresh in every round.

**Definition 1 (Extractor)** *A function* ext : $\{0,1\}^{k_{\mathsf{ext}}} \times \{0,1\}^r \to \{0,1\}^{m_{\mathsf{ext}}}$ *is an* $(\epsilon_{\mathsf{ext}}, n_{\mathsf{ext}})$ *extractor if for any* $X$ *with* $\mathbf{H}_\infty(X) \geq n_{\mathsf{ext}}$ *and* $K \sim U_{k_{\mathsf{ext}}}$ *we have that* $d((\mathsf{ext}(K, X), K) \leq \epsilon_{\mathsf{ext}}$.

**Lemma 1 (Alternating Extraction)** *Let* ext : $\{0,1\}^{k_{\mathsf{ext}}} \times \{0,1\}^r \to \{0,1\}^{m_{\mathsf{ext}}}$ *be an* $(\epsilon_{\mathsf{ext}}, n_{\mathsf{ext}})$-*extractor. Let* $A, B \in \{0,1\}^r$ *and* $K_0 \in \{0,1\}^k$ *be random variables where* $A$ *and* $B$ *are independent and*

$$d(K_0|B) \leq \epsilon_0 \qquad \mathbf{H}_\infty(A) \geq r - \Delta \qquad \mathbf{H}_\infty(B) \geq r - \Delta, \tag{6}$$

*Consider any* $\lambda, \Delta, r \geq 0$ *and* $1 \geq \epsilon_{\mathsf{gap}} > 0$ *which satisfy*

$$n_{\mathsf{ext}} \leq r - \Delta - \lceil \ell/2 \rceil (\lambda + m_{\mathsf{ext}}) - \log(1/\epsilon_{\mathsf{gap}}). \tag{7}$$

*Consider any adversary* $\mathsf{Q} \in \mathcal{A}_\lambda$ *and the random experiment* $\mathsf{S}^*(A, B, K_0) \overset{\ell}{\rightsquigarrow} \mathsf{Q}$ *and let* $\mathtt{view}_\ell = \{K_0, \ldots, K_\ell, \Lambda_1, \ldots, \Lambda_\ell\}$ *and* $\tau_\ell = B$ *if* $\ell$ *is odd and* $\tau_\ell = B$ *otherwise. Then*

$$d(K_{\ell+1}|\mathtt{view}_\ell, \tau_\ell) \leq (\ell + 1)\epsilon_{\mathsf{ext}} + 2\epsilon_{\mathsf{gap}} + \epsilon_0, \tag{8}$$

*i.e. given the view of* Q *after the computation of* $K_\ell$ *and* $\tau_\ell$, *the next key* $K_{\ell+1}$ *to be output by* S *is* ($\ell\epsilon_{\mathsf{ext}} + 2\epsilon_{\mathsf{gap}} + \epsilon_0$)*-close to uniformly random.*

### 4.1 Basic Lemmata

We state some basic information theoretic lemmata which will be needed in the proof of the alternating extraction lemma. The following Lemma similar to Lemma 5 from [12] (the latter can be seen as a special case where $X_i = X_j$ and $Y_i = Y_j$ for all $i \neq j$). Its proof appears in Appendix A

**Lemma 2** *Let $X_0, Y_0$ be independent random variables, and $\phi_1, \phi_2, \ldots$ be any sequence of functions. Let $X_1, X_2, \ldots$, $Y_1, Y_2, \ldots$ and $V_1, V_2, \ldots$ be defined as*

$$\begin{array}{rcll} ((X_{i+1}, V_{i+1}), Y_{i+1}) & := & (\phi_{i+1}(X_i, V_1, \ldots, V_i), Y_i) & \textit{if } i \textit{ is even} \\ (X_{i+1}, (V_{i+1}, Y_{i+1})) & := & (X_i, \phi_{i+1}(Y_i, V_1, \ldots, V_i)) & \textit{otherwise} \end{array}$$

*Then $Y_i \to \{V_1, \ldots, V_i\} \to X_i$ (and $X_i \to \{V_1, \ldots, V_i\} \to Y_i$) is a Markov chain (or equivalently, $X_i$ and $Y_i$ are independent given the $V_1, \ldots, V_i$)*

By identifying $X_\ell, Y_\ell$ from Lemma 2 with $A, B$ in the random experiment $\mathsf{S}^*(A, B, K_0) \overset{\ell}{\rightsquigarrow} \mathsf{Q}$, or with $A_\ell, B_\ell$ in the random experiment $\mathsf{S}(A, B, K_0) \overset{\ell}{\rightsquigarrow} \mathsf{Q}$, we get the following corollary.

**Corollary 1** *(i) In $\mathsf{S}^*(A, B, K_0) \overset{\ell}{\rightsquigarrow} \mathsf{Q}$, $A$ and $B$ are independent given $\mathtt{view}_\ell$. (ii) In $\mathsf{S}(A, B, K_0) \overset{\ell}{\rightsquigarrow} \mathsf{Q}$, $A_\ell = \mathcal{M}_0^\ell$ and $B_\ell = \mathcal{M}_1^\ell$ are independent given $\mathtt{view}_\ell$.*

The proofs of Lemmata 3–6 appear in Appendix A

**Lemma 3** *Let $X$ and $Y$ be (in general dependent) random variables where $Y \in \{0,1\}^\mu$, and let $Z$ be a random variable which is independent of $X$ given $Y$, i.e. $\mathbf{I}(X, Z|Y) = 0$. Then for any $\epsilon_{\mathsf{gap}} > 0$ we have that $\Pr_{y:=Y}[\mathbf{H}_\infty(X|Z, Y = y) \leq \mathbf{H}_\infty(X) - \mu - \log(1/\epsilon_{\mathsf{gap}})] \leq \epsilon_{\mathsf{gap}}$.*

**Lemma 4** *For random variables $A, B$ and any function $\phi$, $d(A|B) = d(A|B, \phi(B))$ and $d(A|B) \geq d(A|\phi(B))$.*

**Lemma 5** *Let $K, \widetilde{K}, R, T$ be random variables such that $K$ is uniformly random, and let $\phi$ be any function. Then $d(\phi(\widetilde{K}, R)|\widetilde{K}, T) \leq d(\phi(K, R)|K, T) + d(\widetilde{K}|T)$.*

**Lemma 6** *Consider any random variables $K', V, R$ where $d(K'|V, R) \leq \epsilon$ and $\mathbf{H}_\infty(R|V) \geq n_{\mathsf{ext}}$, and let $\mathsf{ext}$ be an $(\epsilon_{\mathsf{ext}}, n_{\mathsf{ext}})$-extractor, then $d(\mathsf{ext}(K', R), K'|V) \leq \epsilon + \epsilon_{\mathsf{ext}}$.*

### 4.2 Proof of Lemma 1 (alternating extraction)

By the following claim, the min-entropy of $A$ and $B$ is unlikely to fall below $n_{\mathsf{ext}}$ during the random experiment.

**Claim 1**

$$\Pr_{v=\mathtt{view}_\ell}[\mathbf{H}_\infty(A|\mathtt{view}_\ell = v) \leq n_{\mathsf{ext}}] \leq \epsilon_{\mathsf{gap}} \qquad \Pr_{v=\mathtt{view}_\ell}[\mathbf{H}_\infty(B|\mathtt{view}_\ell = v) \leq n_{\mathsf{ext}}] \leq \epsilon_{\mathsf{gap}}$$

*Proof of Claim:* For $b \in \{0,1\}$, let $\mathtt{view}_\ell^b = \{K_i, \Lambda_i : i \geq 1, i = 0 \bmod b\}$, then $\mathtt{view}_\ell = K_0 \cup \mathtt{view}_\ell^0 \cup \mathtt{view}_\ell^1$. The claim now follows by Lemma 3 and eq. (7), observing that $|\mathtt{view}_\ell^b| \leq \lceil \ell/2 \rceil (\lambda + m_{\mathsf{ext}})$ and that $A$ is independent $\mathtt{view}_\ell^1$ given $\mathtt{view}_\ell^0, K_0$ (and $B$ is independent $\mathtt{view}_\ell^0$ given $\mathtt{view}_\ell^1, K_0$). △

By the above claim, the probability that the min-entropy of $A$ or $B$ drops below $n_{\mathsf{ext}}$ during the random experiment is at most $2\epsilon_{\mathsf{gap}}$. Below we will assume that this will not be the case, the $2\epsilon_{\mathsf{gap}}$ term in (8) accounts for this assumption. We will show by induction on $i$ that

$$d(K_i|\mathtt{view}_{i-1}, \tau_{i-1}) \leq i \cdot \epsilon_{\mathsf{ext}} + \epsilon_0. \tag{9}$$

Note that is the statement of the Lemma for $i = \ell$ (modulo the $2\epsilon_{\mathsf{gap}}$ term). We first prove that (9) holds for $i = 1$. The left side of (9) for $i = 1$ by definition is

$$d(K_1|\mathtt{view}_0, A) = d(K_1|K_0, A) = d(\mathsf{ext}(K_0^{\mathsf{nxt}}, B)|K_0, A)$$

Now, as by Corollary 1 the variables $A$ and $B$ are independent and we already condition on $K_0$, this is equivalent to $d(\mathsf{ext}(K_0^{\mathsf{nxt}}, B)|K_0)$. Further, using Lemma 5 (below $K \sim U_k$), we can upper bound this by $\leq d(\mathsf{ext}(K^{\mathsf{nxt}}, B)|K) + d(K_0|B) \leq \epsilon_{\mathsf{ext}} + \epsilon_0$, which proves (9) for $i = 1$. Assuming (9) holds for $i-1$, we can show that it holds also for $i$ as follows (the steps will be explained in detail below, here again $K \sim U_k$)

$$\begin{aligned}
d(K_i|\mathtt{view}_{i-1}, \tau_{i-1}) &= d(\mathsf{ext}(K_{i-1}^{\mathsf{nxt}}, \tau_i)|\mathtt{view}_{i-2}, K_{i-1}, f_{i-1}(\tau_{i-1}), \tau_{i-1}) && (10)\\
&= d(\mathsf{ext}(K_{i-1}^{\mathsf{nxt}}, \tau_i)|\mathtt{view}_{i-2}, K_{i-1}, \tau_{i-1}) && (11)\\
&= d(\mathsf{ext}(K_{i-1}^{\mathsf{nxt}}, \tau_i)|\mathtt{view}_{i-2}, K_{i-1}) && (12)\\
&\leq d(\mathsf{ext}(K^{\mathsf{nxt}}, \tau_i)|\mathtt{view}_{i-2}, K) + d(K_{i-1}|\mathtt{view}_{i-2}) && (13)\\
&\leq \epsilon_{\mathsf{ext}} + d(K_{i-1}|\mathtt{view}_{i-2}) && (14)\\
&\leq \epsilon_{\mathsf{ext}} + (i-1)\epsilon_{\mathsf{ext}} + \epsilon_0 \leq i \cdot \epsilon_{\mathsf{ext}} + \epsilon_0 && (15)
\end{aligned}$$

Step (10) just uses the definition. The next step (11) uses Lemma 4 (observing that $f_{i-1}$ can be computed from $\mathtt{view}_{i-2}$). Step (12) follows as $\tau_i$ and $\tau_{i-1}$ are independent by Corollary 1. Step (13) uses Lemma 5. In step (14) we use our assumption that $\mathbf{H}_\infty(\tau_i|\mathtt{view}_{i-2}) \geq \mathbf{H}_\infty(\tau_i|\mathtt{view}_\ell) \geq n_{\mathsf{ext}}$ and the fact that $\mathsf{ext}$ is an $(\epsilon_{\mathsf{ext}}, n_{\mathsf{ext}})$-extractor. The last step (15) follow from the induction hypothesis (9) for $i-1$. □

# 5 Pseudoentropy

In this section we will prove that the output of a PRG has high HILL-pseudoentropy even if some function of the seed is leaked. We only actually prove this result for a weaker notion of pseudoentropy called "metric-type", and then use the equivalence of metric-type and HILL-pseudoentropy (Lemma 7) to get our lower bound for HILL-pseudoentropy.

**Basic Definitions** We denote with $\delta^{\mathsf{D}}(X;Y)$ the advantage of a circuit $\mathsf{D}$ in distinguishing the random variables $X, Y$, i.e.: $\delta^{\mathsf{D}}(X;Y) \stackrel{\mathsf{def}}{=} \frac{1}{2}|\Pr[\mathsf{D}(X) = 1] - \Pr[\mathsf{D}(Y) = 1]|$. With $\delta_s(X;Y)$ we denote $max_D \delta^{\mathsf{D}}(X;Y)$ where the maximum is over all circuits $\mathsf{D}$ of size $s$. For a random variable $X$ over $\{0,1\}^z$, $d_s(X) \stackrel{\mathsf{def}}{=} \delta_s(X;U_z)$.

**Definition 2 (Pseudorandom Generator)** *A function* $\mathsf{prg} : \{0,1\}^n \to \{0,1\}^m$ *is a* $(\delta, s)$*-secure pseudorandom generator (PRG) if* $d_s(\mathsf{prg}(U_n)) \leq \delta$.

**Definition 3 (HILL pseudoentropy[15, 2])** *We say* $X$ *has* HILL *pseudoentropy* $k$*, denoted by* $\mathbf{H}^{\mathsf{HILL}}_{\epsilon,s}(X) \geq k$*, if there exists a distribution* $Y$ *where* $\mathbf{H}_\infty(Y) \geq k$ *and* $\delta_s(X,Y) \leq \epsilon$.

The above definition requires that there exists a distribution $Y$ with high min-entropy that is indistinguishable from $X$ by all distinguishers. It is natural to consider a definition where the quantifiers are exchanged, i.e. to allow the distribution to depend on the distinguisher.

**Definition 4 (Metric-type pseudoentropy [2])** *We say* $X$ *has* metric-type *pseudoentropy* $k$*, denoted* $\mathbf{H}^{\mathsf{Metric}}_{\epsilon,s}(X) \geq k$*, if for every circuit* $\mathsf{D}$ *of size* $s$ *there exists a distribution* $Y$ *with* $\mathbf{H}_\infty(Y) \geq k$ *and* $\delta^{\mathsf{D}}(X,Y) \leq \epsilon$.

Barak et al. [2] use the von Neumann's min-max theorem [24] to prove the equivalence of $\mathbf{H}^{\mathsf{HILL}}$ and $\mathbf{H}^{\mathsf{Metric}}$.

**Lemma 7 (Equivalence of $\mathbf{H}^{\mathsf{HILL}}$ and $\mathbf{H}^{\mathsf{Metric}}$ for Circuits (Thm.5.2 from [2]))** *Let $X$ be a distribution over $\{0,1\}^n$. For every $\epsilon, \epsilon_{\mathsf{HILL}} > 0$ and $k$, if $\mathbf{H}^{\mathsf{Metric}}_{\epsilon,s}(X) \geq k$ then $\mathbf{H}^{\mathsf{HILL}}_{\epsilon+\epsilon_{\mathsf{HILL}},\hat{s}}(X) \geq k$ where $s \in O(n\hat{s}/\epsilon^2_{\mathsf{HILL}})$ or equivalently $\hat{s} \in \Omega(\epsilon^2_{\mathsf{HILL}} s/n)$. More precisely (by inspection of the proof of Thm.5.2 in [2]) $s \leq 8n\hat{s}/\epsilon^2_{\mathsf{HILL}} - \zeta$ where $\zeta$ is the size of a circuit needed to compute the majority of $8n/\epsilon^2_{\mathsf{HILL}}$ bits.*

## 5.1 Pseudoentropy of a PRG

By the following lemma, the output of a PRG has high metric-type pseudoentropy (and thus by Lemma 7 also high HILL-pseudoentropy) even if some function of its input is leaked.

**Lemma 8 (Metric/HILL Pseudoentropy of a PRG)** *Let $\mathsf{prg} : \{0,1\}^n \to \{0,1\}^m$ and $f : \{0,1\}^n \to \{0,1\}^\lambda$ (where $\lambda < n < m$) be any functions. If $\mathsf{prg}$ is a $(\epsilon_{\mathsf{prg}}, s_{\mathsf{prg}})$-secure pseudorandom-generator, then for any $\epsilon, \Delta > 0$ satisfying[7] $\epsilon_{\mathsf{prg}} \leq \frac{\epsilon^2}{2^\lambda} - 2^{-\Delta}$, we have with $X \sim U_n$*

$$\Pr_{y:=f(X)}[\mathbf{H}^{\mathsf{Metric}}_{\epsilon,s_{\mathsf{prg}}}(\mathsf{prg}(X)|f(X)=y) \geq m - \Delta] \geq 1 - \epsilon \tag{16}$$

*and for any $\epsilon_{\mathsf{HILL}} > 0$*

$$\Pr_{y:=f(X)}[\mathbf{H}^{\mathsf{HILL}}_{\epsilon+\epsilon_{\mathsf{HILL}},\hat{s}}(\mathsf{prg}(X)|f(X)=y) \geq m - \Delta] \geq 1 - \epsilon \tag{17}$$

*where $\hat{s} \approx \epsilon^2_{\mathsf{HILL}} s_{\mathsf{prg}}/8m$.*

**Proof** Eq. (17) follows from (16) by Lemma 7. To prove (16) assume for contradiction that it does not hold. Hence, by Def. 4, there exists a subset

$$\mathcal{S} \subseteq \{0,1\}^\lambda \quad \text{where} \quad \Pr[f(U_n) \in \mathcal{S}] > \epsilon \tag{18}$$

such that for each $a \in \mathcal{S}$ there exists a distinguisher $\mathsf{D}_a$ of size at most $s_{\mathsf{prg}}$ such that for every random variable $Z$ with $\mathbf{H}_\infty(Z) \geq m - \Delta$ we have (again $X \sim U_n$)

$$|\Pr[\mathsf{D}_a(Z) = 1] - \Pr[\mathsf{D}_a(\mathsf{prg}(X)) = 1|f(X) = a]| \geq \epsilon \tag{19}$$

Consider some $a \in \mathcal{S}$ for which

$$\Pr[f(U_n) = a] > 2^{-\lambda}\epsilon \tag{20}$$

Such an $a$ exists by (18) and by the fact that the range of $f$ has a size $2^\lambda$. Our distinguisher for the PRG $\mathsf{prg}$ will be the distinguisher $\mathsf{D}_a$ satisfying (19) and (20). It remains to prove that $\mathsf{D}_a$ breaks $\mathsf{prg}$ with advantage more than $\epsilon_{\mathsf{prg}}$. For $b \in \{0,1\}$ let $\mathcal{I}_b := \{x \in \{0,1\}^m : \mathsf{D}_a(x) = b\}$

**Claim 2** *For some $\beta \in \{0,1\}$ we have $|\mathcal{I}_\beta| < 2^{m-\Delta}$*

*Proof of Claim:* Assume for contradiction that $|\mathcal{I}_b| \geq 2^{m-\Delta}$ for $b = 0$ and $b = 1$. For $b \in \{0,1\}$ and $X \sim U_n$ let $p_b = \Pr[\mathsf{D}_a(\mathsf{prg}(X)) = b|f(X) = a]$. Let $Z'$ be a random variable distributed uniformly over $S'_0 \cup S'_1$ where $S'_b$ is an arbitrary subset of $\mathcal{I}_b$ of size $p_b 2^{m-\Delta}$ (here we use the fact that $|\mathcal{I}_b| \geq 2^{m-\Delta}$). Clearly since $|S'_0 \cup S'_1| = 2^{m-\Delta}$ we have that $\mathbf{H}_\infty(Z') = m - \Delta$ and by construction (with $X \sim U_n$)

$$\underbrace{\Pr[\mathsf{D}_a(Z') = 1]}_{=\Pr[Z' \in S'_1] = p_1} - \Pr[\mathsf{D}_a(\mathsf{prg}(X)) = 1|f(X) = a] = 0$$

contradicting (19). This finishes the proof of the claim. △

[7]For the lemma to be non-trivial, one should choose $\Delta$ such that $\epsilon^2/2^\lambda > 2^{-\Delta}$

For $\beta$ as guaranteed by the above claim, we have

$$\Pr[\mathsf{D}_a(U_m) = \beta] = |\mathcal{I}_\beta|/2^m < 2^{-\Delta}. \tag{21}$$

By equation (19), using that $\mathbf{H}_\infty(U_m) = m > m - \Delta$ we get:

$$|\underbrace{\Pr[\mathsf{D}_a(U_m) = \beta]}_{<2^{-\Delta}} - \Pr[\mathsf{D}_a(\mathsf{prg}(X)) = \beta | f(X) = a]| \geq \epsilon$$

We can assume that $\epsilon \geq \epsilon^2/2^\lambda > 2^{-\Delta}$ as otherwise the Lemma is trivial. As for any $x, y, \epsilon \geq 0$ we have that $|x - y| \geq \epsilon$ and $\epsilon > x$ implies $y \geq \epsilon$, the above equation implies $\Pr[\mathsf{D}_a(\mathsf{prg}(X) = \beta | f(X) = a] \geq \epsilon$, and further with $X \sim U_n$

$$\Pr[\mathsf{D}_a(\mathsf{prg}(X)) = \beta] \geq \underbrace{\Pr[\mathsf{D}_a(\mathsf{prg}(X)) = \beta | f(X) = a]}_{\geq \epsilon} \cdot \underbrace{\Pr[f(X) = a]}_{\geq 2^{-\lambda} \cdot \epsilon \text{ by}(20)} > \frac{\epsilon^2}{2^\lambda}.$$

By (21) and the above equation, the advantage of $\mathsf{D}_a$ for $U_m$ and $\mathsf{prg}(U_n)$ is at least

$$\Pr[\mathsf{D}_a(\mathsf{prg}(U_n)) = \beta] - \Pr[\mathsf{D}_a(U_m) = \beta] > \frac{\epsilon^2}{2^\lambda} - 2^{-\Delta} \geq \epsilon_{\mathsf{prg}}.$$

□

## 6 Putting Things Together: Proof of Theorem 1

In this section we will prove the security of $\mathsf{S}$ based on $\mathsf{ext}, \mathsf{prg}$ as stated in Theorem 1, recall that

- $\mathsf{ext} : \{0,1\}^{k_{\mathsf{ext}}} \times \{0,1\}^r \to \{0,1\}^{m_{\mathsf{ext}}}$ is an $(\epsilon_{\mathsf{ext}}, n_{\mathsf{ext}})$ extractor.
- $\mathsf{prg} : \{0,1\}^{k_{\mathsf{prg}}} \to \{0,1\}^r$ is an $(\epsilon_{\mathsf{prg}}, s_{\mathsf{prg}})$ pseudorandom generator.

further we set $k_{out} := m_{\mathsf{ext}} - k_{\mathsf{ext}} + k_{\mathsf{prg}}$ (thus $m_{\mathsf{ext}} = k_{\mathsf{ext}} + k_{out} + k_{\mathsf{prg}}$) and parameters $\Delta, \epsilon_{\mathsf{gap}}, \lambda$ satisfying

$$\epsilon_{\mathsf{prg}} \leq \frac{\epsilon_{\mathsf{gap}}^2}{2^\lambda} - 2^{-\Delta} \quad \text{and} \quad n_{\mathsf{ext}} \leq r - \Delta - (\lambda + m_{\mathsf{ext}}) - \log(1/\epsilon_{\mathsf{gap}}) \tag{22}$$

We also fix some $\epsilon_{\mathsf{HILL}} > 0$ and set $\hat{s} := \epsilon_{\mathsf{HILL}}^2 s_{\mathsf{prg}}/8r$.

The following lemma quantifies how much security is "lost" by one round of our stream cipher. Let $\mathtt{size}_i$ denote the size of the circuit realizing the $i$th round of the experiment $\mathsf{S} \overset{\ell}{\rightsquigarrow} \mathsf{Q}$, then $\sum_{i=1}^{\ell} \mathtt{size}_i = \mathtt{size}(\mathsf{S} \overset{\ell}{\rightsquigarrow} \mathsf{Q})$.

**Lemma 9 (The $i$th round)** *Consider the random experiment $\mathsf{S} \overset{\ell}{\rightsquigarrow} \mathsf{Q}$. Then if before round $i \leq \ell$ (recall that $\overline{\tau}_{i-1} = \tau_i = B_i$ if $i$ is odd and $\overline{\tau}_{i-1} = \tau_i = A_i$ otherwise) for some $s_{i-1} \leq \hat{s}$ and $\epsilon' \overset{\text{def}}{=} \epsilon_{\mathsf{HILL}} + \epsilon_{\mathsf{gap}}$*

$$\mathbf{H}^{\mathsf{HILL}}_{\epsilon', s_{i-1}}(A_{i-1} | \mathtt{view}_{i-1}, B_{i-1}) \geq r - \Delta \qquad \mathbf{H}^{\mathsf{HILL}}_{\epsilon', s_{i-1}}(B_{i-1} | \mathtt{view}_{i-1}, A_{i-1}) \geq r - \Delta \tag{23}$$

$$d_{s_{i-1}}(K_{i-1} | \mathtt{view}_{i-2}, \overline{\tau}_{i-1}) \leq \epsilon_{i-1} \tag{24}$$

*then with $s_i \overset{\text{def}}{=} s_{i-1} - \mathtt{size}_i$, $s'_i \overset{\text{def}}{=} s_{i-1} - \mathtt{size}(\mathsf{ext})$ and $\epsilon_i \overset{\text{def}}{=} \epsilon_{i-1} + \epsilon_{\mathsf{ext}} + \epsilon_{\mathsf{gap}} + \epsilon'$.*

$$d_{s'_i}(K_i, X_i | \mathtt{view}_{i-1}, \overline{\tau}_i) \leq \epsilon_i \tag{25}$$

*with probability $1 - \epsilon_{\mathsf{gap}} - \epsilon_i$*

$$\mathbf{H}^{\mathsf{HILL}}_{\epsilon', s_i}(A_i | \mathtt{view}_i, B_i) \geq r - \Delta \qquad \mathbf{H}^{\mathsf{HILL}}_{\epsilon', s_i}(B_i | \mathtt{view}_i, A_i) \geq r - \Delta \tag{26}$$

**Proof** We prove the statement for odd $i$, in this case $(K_i, X_i) := \mathsf{ext}(K_{i-1}^{\mathsf{nxt}}, B_{i-1})$, $B_i := \mathsf{prg}(X_i)$ and $A_i := A_{i-1}$. The case for even $i$ is identical, we just have to swap the role of $A$ and $B$.

The bound for $A_i$ given by the first equation of (26) follows from the bound on $A_{i-1}$ (23), as $A_i = A_{i-1}$, we just have to replace $s_{i-1}$ by $s_i = s_{i-1} - \mathtt{size}_i$ to account for the work done in round $i$ (i.e. computing $\mathtt{view}_i, B_i$ from $\mathtt{view}_{i-1}, B_{i-1}$, which was done by a circuit of size $\mathtt{size}_i$).

We'll now prove the bound on $B_i$ given by (26). For this, let us first consider random variables $\tilde{B}_{i-1}, \tilde{K}_{i-1}$ which satisfy the preconditions for $B_{i-1}, K_{i-1}$ as stated in (23) and (24), but in an information theoretic sense, and not just relative to circuits of size $s_{i-1}$

$$\Pr[\mathbf{H}_\infty(\tilde{B}_{i-1}|\mathtt{view}_{i-1}, A_{i-1}) \geq r - \Delta] \geq 1 - \epsilon' \qquad d(\tilde{K}_{i-1}|\mathtt{view}_{i-2}, \tilde{B}_{i-1}) \leq \epsilon_{i-1}$$

Let $(\tilde{K}_i, \tilde{X}_i) := \mathsf{ext}(\tilde{K^L}_{i-1}, \tilde{B}_{i-1})$ and $\tilde{B}_i = \mathsf{prg}(\tilde{X}_i)$. Then by Lemma 1 (with $\ell = 1$, taking into account that $\tilde{K}_{i-1}$ is only $\epsilon_{i-1}$ close to uniform, and $\tilde{B}_{i-1}$ has high min-entropy only with prob. $1 - \epsilon'$)

$$d([\tilde{K}_i, \tilde{X}_i]|\mathtt{view}_{i-1}, A_i) \leq \epsilon_{i-1} + \epsilon_{\mathsf{ext}} + \epsilon_{\mathsf{gap}} + \epsilon' = \epsilon_i$$

The above directly implies $d(\tilde{X}_i|\tilde{K}_i, \mathtt{view}_{i-1}, A_i) \leq \epsilon_i$, and by Lemma 8 (and the fact that $\tilde{X}_i$ is $\epsilon_i$ close to uniform), we get that with probability $1 - \epsilon_{\mathsf{gap}} - \epsilon_i$

$$\mathbf{H}^{\mathsf{HILL}}_{\epsilon',\hat{s}}(\underbrace{\mathsf{prg}(\tilde{X}_i)}_{\tilde{B}_i} \,|\, \underbrace{\Lambda_i, \tilde{K}_i, \mathtt{view}_{i-1}}_{\mathtt{view}_i}, A_i) \geq r - \Delta$$

The actual bounds follow from the above by taking into account that $B_{i-1}, K_{i-1}$ are only indistinguishable by circuits of size $s_{i-1}$, we omit the details. □

We'll now see how Theorem 1 is implied by this lemma. Let $\epsilon_0 = 0$ and $s_0 = \hat{s}$, then $\epsilon_\ell = \ell(2\epsilon_{\mathsf{gap}} + \epsilon_{\mathsf{ext}} + \epsilon_{\mathsf{HILL}})$ and $s_\ell = \hat{s} - \mathtt{size}(\mathsf{S} \overset{\ell}{\rightsquigarrow} \mathsf{Q})$. If the initial key $A_0, B_0, K_0$ satisfies

$$\mathbf{H}^{\mathsf{HILL}}_{\epsilon',s_0}(A_0|B_0) \geq r - \Delta \qquad \mathbf{H}^{\mathsf{HILL}}_{\epsilon',s_0}(B_0|A_0) \geq r - \Delta \tag{27}$$

$$d(K_0|B_0) = \epsilon_0 \tag{28}$$

Which is the case for the precondition of Theorem 1, then by Lemma 9, with probability $1 - \sum_{i=1}^{\ell}(\epsilon_{\mathsf{gap}} + \epsilon_i) \geq 1 - \ell(\epsilon_{\mathsf{gap}} + \epsilon_\ell)$.

$$d_{s'_\ell}(K_\ell, X_\ell|\mathtt{view}_{\ell-1}, \overline{\tau}_\ell) \leq \epsilon_\ell \tag{29}$$

This proves (note that $s'\ell < s\ell$) the bound for the $\mathsf{AdvInd}$ as stated in Theorem 1. to prove the bound for $\mathsf{AdvIndFwd}$ we move $X_\ell, K_\ell^{\mathsf{nxt}}$ to the conditioned part (recall that $K_\ell = K_\ell^{\mathsf{nxt}} \| K_\ell^{\mathsf{out}}$)

$$d_{s'_\ell}(K_\ell^{\mathsf{out}}|K_\ell^{\mathsf{nxt}}, X_\ell, \mathtt{view}_{\ell-1}, \overline{\tau}_\ell) \leq \epsilon_\ell$$

Then we apply the $\mathsf{prg}$ to $X_\ell$

$$d_{\underbrace{s'_\ell - |\mathsf{prg}|}_{>s_\ell}}(K_\ell^{\mathsf{out}}|K_\ell^{\mathsf{nxt}}, \mathtt{view}_{\ell-1}, \overbrace{\underbrace{\overline{\tau}_{\ell+1}}_{\mathsf{prg}(X_\ell)}, \overline{\tau}_\ell}^{A_\ell, B_\ell}) \leq \epsilon_\ell \tag{30}$$

Which proves the bound on $\mathsf{AdvIndFwd}$ as stated in Theorem 1.

## References


[1] Ross Anderson and Markus Kuhn. Tamper resistance: a cautionary note. In *WOEC'96: Proceedings of the 2nd conference on Proceedings of the Second USENIX Workshop on Electronic Commerce*, pages 1–11, Berkeley, CA, USA, 1996. USENIX Association.

[2] Boaz Barak, Ronen Shaltiel, and Avi Wigderson. Computational analogues of entropy. In *RANDOM-APPROX*, pages 200–215, 2003.

[3] Eli Biham and Adi Shamir. Differential fault analysis of secret key cryptosystems. In *CRYPTO*, pages 513–525, 1997.

[4] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the importance of checking cryptographic protocols for faults (extended abstract). In *EUROCRYPT*, pages 37–51, 1997.

[5] Ran Canetti, Yevgeniy Dodis, Shai Halevi, Eyal Kushilevitz, and Amit Sahai. Exposure-resilient functions and all-or-nothing transforms. In *EUROCRYPT*, pages 453–469, 2000.

[6] David Cash, Yan Zong Ding, Yevgeniy Dodis, Wenke Lee, Richard J. Lipton, and Shabsi Walfish. Intrusion-resilient key exchange in the bounded retrieval model. In Salil P. Vadhan, editor, *TCC*, volume 4392 of *Lecture Notes in Computer Science*, pages 479–498. Springer, 2007.

[7] Giovanni Di Crescenzo, Richard J. Lipton, and Shabsi Walfish. Perfectly secure password protocols in the bounded retrieval model. In Halevi and Rabin [14], pages 225–244.

[8] Yevgeniy Dodis, Amit Sahai, and Adam Smith. On perfect and adaptive security in exposure-resilient cryptography. In *EUROCRYPT*, pages 301–324, 2001.

[9] Stefan Dziembowski. Intrusion-resilience via the bounded-storage model. In Halevi and Rabin [14], pages 207–224.

[10] Stefan Dziembowski. On forward-secure storage. In Cynthia Dwork, editor, *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 251–270. Springer, 2006.

[11] Stefan Dziembowski and Ueli M. Maurer. On generating the initial key in the bounded-storage model. In *EUROCRYPT*, pages 126–137, 2004.

[12] Stefan Dziembowski and Krzysztof Pietrzak. Intrusion-resilient secret sharing. In *FOCS*, pages 227–237, 2007.

[13] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic analysis: Concrete results. In *CHES*, pages 251–261, 2001.

[14] Shai Halevi and Tal Rabin, editors. *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*. Springer, 2006.

[15] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[16] Yuval Ishai, Manoj Prabhakaran, Amit Sahai, and David Wagner. Private Circuits II: Keeping Secrets in Tamperable Circuits. In *EUROCRYPT*, pages 308–327, 2006.

[17] Yuval Ishai, Amit Sahai, and David Wagner. Private Circuits: Securing Hardware against Probing Attacks. In *CRYPTO*, pages 463–481, 2003.

[18] Jesse Kamp and David Zuckerman. Deterministic extractors for bit-fixing sources and exposure-resilient cryptography. *SIAM J. Comput.*, 36(5):1231–1247, 2007.

[19] John Kelsey, Bruce Schneier, David Wagner, and Chris Hall. Side channel cryptanalysis of product ciphers. In Jean-Jacques Quisquater, Yves Deswarte, Catherine Meadows, and Dieter Gollmann, editors, *ESORICS*, volume 1485 of *Lecture Notes in Computer Science*, pages 97–110. Springer, 1998.

[20] Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *CRYPTO*, pages 104–113, 1996.

[21] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *CRYPTO*, pages 388–397, 1999.

[22] Ueli M. Maurer. A provably-secure strongly-randomized cipher. In *EUROCRYPT*, pages 361–373, 1990.

[23] Silvio Micali and Leonid Reyzin. Physically observable cryptography (extended abstract). In *TCC*, pages 278–296, 2004.

[24] John Von Neumann. Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, 100(1):295–320, 1928.

[25] European Network of Excellence (ECRYPT). The side channel cryptanalysis lounge. http://www.crypto.ruhr-uni-bochum.de/en_sclounge.html. retrieved on 29.03.2008.

[26] Christophe Petit, Francois-Xavier Standaert, Olivier Pereira, Tal G. Malkin, and Moti Yung. A block cipher based prng secure against side-channel key recovery. Cryptology ePrint Archive, Report 2007/356, 2007. http://eprint.iacr.org/.

[27] Jean-Jacques Quisquater and David Samyde. Electromagnetic analysis (ema): Measures and counter-measures for smart cards. In *E-smart*, pages 200–210, 2001.

[28] Jean-Jaques Quisquater and Franois Koene. Side channel attacks: State of the art, October 2002. [25].

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

# A Proofs of Lemmata

**Lemma 3** *Let $X$ and $Y$ be (in general dependent) random variables where $Y \in \{0,1\}^\mu$, and let $Z$ be a random variable which is independent of $X$ given $Y$, i.e. $\mathbf{I}(X, Z|Y) = 0$. Then for any $\epsilon_{\mathsf{gap}} > 0$ we have that*

$$\Pr_{y:=Y}[\mathbf{H}_\infty(X|Z, Y = y) \le \mathbf{H}_\infty(X) - \mu - \log(1/\epsilon_{\mathsf{gap}})] \le \epsilon_{\mathsf{gap}}. \tag{31}$$

**Proof** As $Z$ is independent of $X$ given $Y$, we can ignore it, i.e. eq.(31) is equivalent to

$$\Pr_{y:=Y}[\mathbf{H}_\infty(X|Y = y) \le \mathbf{H}_\infty(X) - \mu - \log(1/\epsilon_{\mathsf{gap}})] \le \epsilon_{\mathsf{gap}}. \tag{32}$$

Let $\mathcal{Y}$ be the set of all the elements $y \in \{0,1\}^\mu$ such that

$$\mathbf{H}_\infty(X|Y = y) \le \mathbf{H}_\infty(X) - \mu - \log(1/\epsilon_{\mathsf{gap}}). \tag{33}$$

For the sake of contradiction suppose (32) does not hold, in other words: $\Pr[Y \in \mathcal{Y}] > \epsilon_{\mathsf{gap}}$. Hence there exits $y' \in \mathcal{Y}$ such that

$$\Pr\left[Y = y'\right] > 2^{-\mu} \cdot \epsilon_{\mathsf{gap}}. \tag{34}$$

By definition of min-entropy there exists $x'$ such that $\Pr[X = x' \mid Y = y'] \geq 2^{-\mathbf{H}_\infty(X)+\mu+\log(1/\epsilon_{\mathsf{gap}})} = 2^{-\mathbf{H}_\infty(X)+\mu} \cdot \epsilon_{\mathsf{gap}}^{-1}$, and therefore

$$\begin{aligned}\Pr\left[X = x'\right] &= \Pr\left[X = x' \mid Y = y'\right] \cdot \overbrace{\Pr\left[Y = y'\right]}^{>2^{-\mu}\cdot\epsilon_{\mathsf{gap}} \text{ by}(34)} \\ &> 2^{-\mathbf{H}_\infty(X)},\end{aligned}$$

which contradicts the definition of min-entropy. □

The remaining lemmata in this section are standard and were already proven e.g. the full version of [12]. For completeness we include the proofs here. Before proving Lemma 4 we show the following.

**Lemma 10** *For every random variables $A$ and $A'$ over $\mathcal{A}$, and every function $\gamma : \mathcal{A} \to \mathcal{B}$ we have $\delta(A; A') \geq \delta(\gamma(A); \gamma(A'))$.*

**Proof**

$$\begin{aligned}\delta(A; A') &= \tfrac{1}{2}\sum_{a\in\mathcal{A}} \left|\Pr[A = a] - \Pr\left[A' = a\right]\right| \\ &= \tfrac{1}{2}\sum_{b\in\mathcal{B}} \sum_{a \text{ such that } \gamma(a) = b} \left|\Pr[A = a] - \Pr\left[A' = a\right]\right| \\ &\geq \tfrac{1}{2}\sum_{b\in\mathcal{B}} \left| \sum_{a \text{ such that } \gamma(a) = b} \Pr[A = a] - \sum_{a \text{ such that } \gamma(a) = b} \Pr\left[A' = a\right] \right| \quad (35) \\ &= \tfrac{1}{2}\sum_{b\in\mathcal{B}} \left|\Pr[\gamma(A) = a] - \Pr\left[\gamma(A') = a\right]\right| \\ &= \delta(\gamma(A); \gamma(A')),\end{aligned}$$

where (35) comes from the triangle inequality. □

**Lemma 4** *Let $A, B$ be random variables and let $\phi$ be any function. Then $d(A|B) \geq d(A|\phi(B))$. In particular $d(A|B, C) \geq d(A|B)$ for any random variable $C$.*

**Proof** Define $\gamma$ as $\gamma(X, Y) = (X, \phi(Y))$. Let $U$ be uniform over $\mathcal{A}$, now by Lemma 10

$$\underbrace{\delta(A, B; U, B)}_{d(A|B)} \geq \delta(\gamma(A, B); \gamma(U, B)) = \underbrace{\delta(A, \phi(B); U, \phi(B))}_{d(A|\phi(B))}$$

□

**Lemma 5** *Let $K, \widetilde{K}, R, T$ be random variables such that $K$ is uniformly random, and let $\phi$ be any function. Then*

$$d(\phi(\widetilde{K}, R)|\widetilde{K}, T) \leq d(\phi(K, R)|K, T) + \delta(\widetilde{K}|T).$$

**Proof** Let $\mathcal{Y}$ denote the range of $\phi$ and $Y$ be uniform over $\mathcal{Y}$, we have

$$\begin{aligned}
& d(\phi(K,R)|K,T) \\
= & \ \tfrac{1}{2} \sum_{(k,y,t)\in\mathcal{K}\times\mathcal{Y}\times\mathcal{T}} |\Pr[\phi(K,R)=y, K=k, T=t] - \Pr[Y=y, K=k, T=t]| \\
= & \ \tfrac{1}{2} \sum_{(k,y,t)\in\mathcal{K}\times\mathcal{Y}\times\mathcal{T}} \Pr[K=k,T=t]\cdot|\Pr[\phi(K,R)=y \mid K=k,T=t] - \Pr[Y=y \mid K=k,T=t]| \\
= & \ \tfrac{1}{2} \sum_{(k,y,t)\in\mathcal{K}\times\mathcal{Y}\times\mathcal{T}} \Pr[K=k,T=t]\cdot\left|\Pr[\phi(k,R)=y \mid T=t] - \frac{1}{|\mathcal{Y}|}\right| \qquad (36)
\end{aligned}$$

By the same argument

$$d(\phi(\widetilde{K},R)|\widetilde{K},T) = \tfrac{1}{2} \sum_{(k,y,t)\in\mathcal{K}\times\mathcal{Y}\times\mathcal{T}} \Pr\left[\widetilde{K}=k,T=t\right]\cdot\left|\Pr[\phi(k,R)=y \mid T=t] - \frac{1}{|\mathcal{Y}|}\right|. \qquad (37)$$

Now, (37) minus (36) is equal to

$$\begin{aligned}
& \tfrac{1}{2} \sum_{(k,y,t)\in\mathcal{K}\times\mathcal{Y}\times\mathcal{T}} \left(\Pr\left[\widetilde{K}=k,T=t\right] - \Pr[K=k,T=t]\right)\cdot\left|\Pr[\phi(k,R)=y \mid T=t] - \frac{1}{|\mathcal{Y}|}\right| \\
= & \ \tfrac{1}{2} \sum_{(k,t)\in\mathcal{K}\times\mathcal{T}} \left(\Pr\left[\widetilde{K}=k,T=t\right] - \Pr[K=k,T=t]\right)\cdot\underbrace{\left|\sum_{y\in\mathcal{Y}}\Pr[\phi(k,R)=y \mid T=t] - \frac{1}{|\mathcal{Y}|}\right|}_{=1} \\
\leq & \ \delta(K,T;\widetilde{K},T) = d(\widetilde{K}|T).
\end{aligned}$$

□

**Lemma 6** *Consider any random variables $K^{\mathsf{nxt}}, V, R$ where*

$$d(K^{\mathsf{nxt}}|V,R) \leq \epsilon \qquad \mathbf{H}_\infty(R|V) \geq n_{\mathsf{ext}}$$

*and let* $\mathsf{ext}$ *be an* $(\epsilon_{\mathsf{ext}}, n_{\mathsf{ext}})$*-extractor, then*

$$d(\mathsf{ext}(K^{\mathsf{nxt}},R)|V) \leq \epsilon + \epsilon_{\mathsf{ext}}$$

**Lemma 11** *Let $T, E, F$ be random variables where $T \to E \to F$ is a Markov chain, then $d(F|E,T) = d(F|E)$.*

**Proof** Let $U$ be independent and uniformly distributed over $\mathcal{F}$. We have

$$\begin{aligned}
& d(F|E,T) \\
= & \ \delta(F,E,T;U,E,T) \\
= & \ \tfrac{1}{2} \sum_{(f,e,t)\in\mathcal{F}\times\mathcal{E}\times\mathcal{T}} |\Pr[F=f,E=e,T=t] - \Pr[U=f,E=e,T=t]| \\
= & \ \tfrac{1}{2} \sum_{(f,e,t)\in\mathcal{F}\times\mathcal{E}\times\mathcal{T}} \Pr[E=e,T=t]\cdot|\Pr[F=f \mid E=e,T=t] - \Pr[U=f \mid E=e,T=t]| \\
= & \ \tfrac{1}{2} \sum_{(f,e,t)\in\mathcal{F}\times\mathcal{E}\times\mathcal{T}} \Pr[E=e,T=t]\cdot|\Pr[F=f \mid E=e] - \Pr[U=f \mid E=e]| \qquad (38) \\
= & \ \tfrac{1}{2} \sum_{(e,t)\in\mathcal{E}\times\mathcal{T}} \Pr[E=e]\cdot|\Pr[F=f \mid E=e] - \Pr[U=f \mid E=e]| \\
= & \ \delta(F,E;U,E), \\
= & \ d(F|E) \qquad (39)
\end{aligned}$$

where (38) comes from the Markov property. □

**Lemma 2** *Let $X_0, Y_0$ be independent random variables, and $\phi_1, \phi_2, \dots$ be any sequence of functions. Let $X_1, X_2, \dots, Y_1, Y_2, \dots$ and $V_1, V_2, \dots$ be defined as*

$$\begin{array}{rcll} ((X_{i+1}, V_{i+1}), Y_{i+1}) & := & (\phi_{i+1}(X_i, V_1, \dots, V_i), Y_i) & \text{if } i \text{ is even} \\ (X_{i+1}, (V_{i+1}, Y_{i+1})) & := & (X_i, \phi_{i+1}(Y_i, V_1, \dots, V_i)) & \text{otherwise} \end{array}$$

*Then $Y_i \to \{V_1, \dots, V_i\} \to X_i$ (and $X_i \to \{V_1, \dots, V_i\} \to Y_i$) is a Markov chain (or equivalently, $X_i$ and $Y_i$ are independent given the $V_1, \dots, V_i$)*

**Proof** Recall that $X \to Y \to Z$ is a Markov chain if and only if

$$H(Z|Y, X) = H(Z|Y)$$

Moreover $X \to Y \to Z$ is a Markov chain iff $Z \to Y \to X$ is a Markov chain. We will prove by induction over $i$ that

$$Y_i \to \{V_1, \dots, V_i\} \to X_i \text{ or equivalently } X_i \to \{V_1, \dots, V_i\} \to Y_i \tag{40}$$

is a Markov chain. This trivially holds for $i = 0$ as $X_0$ and $Y_0$ are independent. We now show (by contradiction) that if (40) holds for some $i$, then it also holds for $i + 1$: Assume (40) is wrong for $i + 1$, i.e.

$$X_{i+1} \to \{V_1, \dots, V_i, V_{i+1}\} \to Y_{i+1}$$

is not Markov. For even $i$ this implies (for odd $i$ replace the role of $X$ and $Y$)

$$H(Y_{i+1}|V_1, \dots, V_i, \underbrace{\phi_{i+1}(X_i, V_1, \dots, V_i)}_{(X_{i+1}, V_{i+1})}) < H(Y_{i+1}|V_1, \dots, V_i)$$

by Lemma 4

$$H(Y_{i+1}|V_1, \dots, v_i, X_i) \le H(Y_{i+1}|V_1, \dots, V_i, \phi_{i+1}(X_i, V_1, \dots, V_i))$$

the two equations above and the fact that $Y_i = Y_{i+1}$ for even $i$ implies

$$H(Y_i|V_1, \dots, V_i, X_i) < H(Y_i|V_1, \dots, V_i)$$

the last inequality means that

$$Y_i \to \{V_1, \dots, V_i\} \to X_i$$

is *not* Markov, but this contradicts the assumption that the induction hypothesis (40) holds for $i$. □