# Leakage-Resilient Storage*

Francesco Davì[1], Stefan Dziembowski[1], and Daniele Venturi[2]

[1] Department of Computer Science,
University of Rome "La Sapienza",
via Salaria 113,
00198 Rome, Italy
davi@di.uniroma1.it
stefan@dziembowski.net
[2] INFOCOM Department,
University of Rome "La Sapienza",
via Eudossiana 18,
00186 Rome, Italy
venturi@infocom.uniroma1.it

**Abstract.** We study a problem of secure data storage on hardware that may leak information. We introduce a new primitive, that we call *leakage-resilient storage* (LRS), which is an (unkeyed) scheme for encoding messages, and can be viewed as a generalization of the *All-Or-Nothing Transform* (AONT, Rivest 1997). The standard definition of AONT requires that it should be hard to reconstruct a message $m$ if not all the bits of its encoding $\mathsf{Encode}(m)$ are known. LRS is defined more generally, with respect to a class $\Gamma$ of functions. The security definition of LRS requires that it should be hard to reconstruct $m$ even if some values $g_1(\mathsf{Encode}(m)), \ldots, g_t(\mathsf{Encode}(m))$ are known (where $g_1, \ldots, g_t \in \Gamma$), as long as the total length of $g_1(\mathsf{Encode}(m)), \ldots, g_t(\mathsf{Encode}(m))$ is smaller than some parameter $c$.

We construct an LRS scheme that is secure with respect to $\Gamma$ being a set of functions that can depend only on some restricted part of the memory. More precisely: we assume that the memory is divided in 2 parts, and the functions in $\Gamma$ can be just applied to one of these parts. We also construct a scheme that is secure if the cardinality of $\Gamma$ is restricted (but still it can be exponential in the length of the encoding). This construction implies security in the case when the set $\Gamma$ consists of functions that are computable by Boolean circuits of a small size.

We also discuss the connection between the problem of constructing leakage-resilient storage and a theory of the compressibility of NP-instances.

**Keywords:** leakage resilient cryptography, secure storage.

## 1 Introduction

Some of the most devastating attacks on cryptographic devices are those that break the actual physical implementation of the scheme, not its mathematical abstraction.

* The European Research Council has provided financial support under the EuropeanCommunity's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement no CNTM-207908.

J.A. Garay and R. De Prisco (Eds.): SCN 2010, LNCS 6280, pp. 121–137, 2010.
© Springer-Verlag Berlin Heidelberg 2010

These, so-called *side-channel attacks*, are based on the fact that the adversary may obtain some information about the internal data of the device by observing its running-time [27], electromagnetic radiation [35,19], power consumption [28], or even sound that the device is emitting [40] (see [34,32] for more examples of such attacks).

### 1.1 Memory Leakages — Previous Work

Over the last couple of years there has been a growing interest in the design of schemes that already on the abstract level guarantee that their physical implementation is secure against a large well-defined class of side-channel attacks (the pioneering paper in this area was [30]). The main idea is to augment the standard security definition by allowing the adversary to learn the value of a chosen by him *leakage function* $g$ on the internal data $\tau$ used by the cryptographic scheme. The results in this area can be categorized according to the class of leakage functions $g$ that the model covers. Some papers consider very restricted classes (e.g. in [24] the model assumes that the adversary can simply read-off some wires that represent the computation), while other ones consider more general leakages—e.g. [1] allow the adversary to choose any function $g$ that is *input-shrinking* (i.e. such that $|g(\tau)| \ll |\tau|$).

Another popular paradigm is to assume that *only computation leaks information*, i.e. the memory cells that do not take part in the computation (in a given time period) do not leak any information. The first paper to state this assumption is [30] (where it is stated as "Axiom 1", page 283), and the other papers that use it are [17,33]. The schemes of [17,33] are actually secure even if the total amount of information that leaks is greater than the memory size (this is possible since the memory contents is evolving during the computation). The other approach [1,31,25,12,11] is to assume that the memory may simply leak information, independently on the computation performed.

It may be questioned if the "only computation leaks information" paradigm is really relevant to the attack that the adversary can perform in real-life. In many situations memory may actually leak information, even if it is unaccessed. First of all, in modern computer systems it is hard to guarantee that a given part of memory really never gets accessed (for example the memory may be refreshed or moved to cache, etc.). Some practical attacks on unaccessed memory were also demonstrated in [38]. More recently a class of *cold boot attacks* relying on the data remanence property was presented in [20].

A natural question to ask is whether there exist methods for storing data securely in the memory that may leak information. This is the main subject of this paper.

**A relation to the Bounded-Retrieval Model.** The idea to reason about the partial key leakages by modeling them as input-shrinking functions originates from the *Bounded-Retrieval Model* (BRM) [14,9,15,6,16,2] (that in turn was inspired by the Bounded-Storage Model of Maurer [29]). Originally BRM was proposed as a method for protecting against computer viruses that may steal large amounts of data from the PCs: the main idea of the BRM is to construct schemes where the secret key $\tau$ is large and to assume that the adversary can retrieve the value of some input-shrinking function $g$ of $\tau$ . The main differences between this setting and the models for the side-channel attack come from the fact that the keys in the BRM are huge and hence: (1) one has to design scheme where the honest user does not need to frequently process the entire $\tau$ , and (2)

one can allow that some part of $\tau$ leaks each time the scheme is used. Nevertheless in [14] it was observed that BRM can be used to model the side-channel attacks.

### 1.2 Our Contribution

In this paper we introduce a new primitive, that we call *leakage-resilient storage*, which can be viewed as a secure storage scheme in the model where the physical memory may leak some side-channel information. A scheme like this consists of two poly-time algorithms Encode and Decode, where the *encoding* algorithm Encode takes as input a message $m$ and produces as output a string $\tau \stackrel{\mathsf{def}}{=} \mathsf{Encode}(m)$, and the *decoding* algorithm Decode is such that we always have $\mathsf{Decode}(\mathsf{Encode}(m)) = m$ (observe that these algorithms do not take as input any secret key).

Informally speaking, in the security definition we allow the adversary to *adaptively* choose a sequence of leakage functions $g_1, \ldots, g_t$, and learn the values of

$$g_1(\tau), \ldots, g_t(\tau).$$

We require that the adversary, after learning these values, should gain essentially no additional information on $m$ (this is formalized using a standard indistinguishability game, see Sect. 2 for details). We assume that the $g_i$'s are elements of some fixed set $\Gamma$ (that will be a parameter in the definition). Obviously, the larger $\Gamma$, the stronger is our definition, and we should aim at defining $\Gamma$ in such a way that it covers all the attacks the adversary can launch in real-life. All the $\Gamma$'s that we consider in this paper contain at least the set of functions that read-off the individual bits of $\tau$, hence we need to require that

$$\sum_{i=1}^{t} |g_i(\tau)| < |\tau| \tag{1}$$

(as otherwise the functions $g_i$ could be chosen in such a way that $(g_1(\tau), \ldots, g_t(\tau)) = \tau$). This is essentially the input-shrinking property that, as discussed above, was already used in the literature.

LRS can also be viewed as a generalization of the All-Or-Nothing Transform (AONT) introduced in [37]. More precisely: AONT is a special case of LRS, where the leakage functions are projections of the individual bits.

Obviously, if we go to the extreme and simply allow the adversary to choose *any* (poly-time) functions $g_i$ that satisfy (1) then there is no hope for any security, since the adversary could always choose $g_1$ in such a way that it simply calculates $\mathsf{Decode}(\tau)$ and outputs some information about $m$ (say: its first bit). Therefore $\Gamma$ cannot contain the Decode function, and hence, we need to restrict $\Gamma$ in some way.

Note that the assumption that $\Gamma$ is a restricted class of functions is actually very realistic. In practice, the leakage functions need to be computationally "simple": while it is plausible that the adversary can read-off the individual bits, or learn their sum, it seems very improbable that an attack based on measuring power consumption or electromagnetic radiation can directly give information about some more complicated functions of the secret bits.

In this paper we consider two natural choices of such $\Gamma$'s and show LRS schemes secure in these settings relying on *deterministic extractors* [43,4,7,8,5]. In Sect. 3.1 we

describe a construction where each leakage function can depend only on some restricted part of the memory: either because it consists of two separate blocks, or because it is infeasible for the adversary to choose a function that depends on the memory cells that are physically far from each other. In Sect. 3.2 we construct a scheme that is secure if the cardinality of $\Gamma$ is restricted (but still it can be exponential in $|\tau|$). This construction implies security in the case when the set $\Gamma$ consists of functions that are computable by Boolean circuits of a small size. Our construction is an adaptation of the technique already used (in a different context) in [41,3].

The idea to model the leakages as functions from a small complexity class appeared already in [17], and was recently used in an independent work by Faust et al. [18] (we discuss the relationship between our work and [18] in Sect. 5. We also discuss (in Sect. 4) the connection between the problem of constructing leakage-resilient storage and a theory of compressibility of NP-instances [23].

### 1.3 Preliminaries

Let $U_n$ be a random variable distributed uniformly over $\{0,1\}^n$. Given two random variables $X_0, X_1$ with values in $\mathcal{X}$, their *statistical distance* is defined as

$$\Delta(X_0; X_1) \stackrel{\mathsf{def}}{=} \frac{1}{2} \sum_{x \in \mathcal{X}} |\mathbb{P}[X_0 = x] - \mathbb{P}[X_1 = x]|.$$

If $X$ assumes values in $\{0,1\}^n$, then we let $d(X) \stackrel{\mathsf{def}}{=} \Delta(X, U_n)$ be the statistical distance[1] between $X$ and the uniform distribution over $\{0,1\}^n$. If $d(X) \leq \epsilon$ we say that $X$ is $\epsilon$-close to uniform. We also define $\Delta(X_0; X_1|Y) \stackrel{\mathsf{def}}{=} \Delta(X_0, Y; X_1, Y)$ and $d(X|Y) \stackrel{\mathsf{def}}{=} \Delta(X, Y; U_n, Y)$.

The following was proven in [16].

**Lemma 1 ([16]).** *Let $A, B$ be random variables where $A \in \mathcal{A}$. Then $\mathbb{P}[B = A] \leq d(A|B) + 1/|\mathcal{A}|$.*

Given a random variable $X \in \mathcal{X}$, the *min-entropy* of $X$ is $\mathbf{H}_\infty(X) \stackrel{\mathsf{def}}{=} -\log \max_{x \in \mathcal{X}} \mathbb{P}[X = x]$.

We will use the following lemma whose proof appears in Appendix A.

**Lemma 2.** *For every random variables $X, Y$ and an event $\mathcal{E}$ we have*

$$d(X|Y = y \wedge \mathcal{E}) + \mathbb{P}\left[\overline{\mathcal{E}}\right] \geq d(X|Y). \tag{2}$$

The proofs of the following lemmata appear in the full version of [16].

**Lemma 3.** *Let $A, B$ be two random variables and let $\phi$ be any function. Then $d(A|B) \geq d(A|\phi(B))$.*

[1] We will overload the symbols $\Delta(\cdot)$ and $d(\cdot)$ and sometimes apply them to the probability distributions instead of the random variables.

**Lemma 4.** *Let $A, B$ be independent random variables and consider a sequence $V_1, \ldots, V_i$ of random variables, where for some function $\phi$, $V_i = \phi_i(C_i) = \phi(V_1, \ldots, V_{i-1}, C_i)$, with each $C_i$ being either $A$ or $B$. Then $A$ and $B$ are independent conditioned on $V_1, \ldots, V_i$, i.e. $I(A; B|V_1, \ldots, V_i) = 0$, where $I$ denotes the Shannon's information*[2].

We will also use the following standard fact whose proof appears in Appendix B.

**Lemma 5.** *Let $X$ be a random variable uniformly distributed over $\{0,1\}^n$, and let $W$ be a random variable that is independent on $X$. Let $f : \{0,1\}^* \to \{0,1\}^c$. Then for every $k \in \mathbb{N}$ we have*

$$\mathbb{P}_{y:=f(X,W)} \left[ \mathbf{H}_\infty(X|f(X,W) = y) \leq k \right] \leq 2^{k+c-n}. \tag{3}$$

A family $\{h_s\}_{s \in \mathcal{S}}$ of functions $h_s : \mathcal{X} \to \mathcal{Y}$ is called a *collection of $\ell$-wise independent hash functions* if for every set $\{x_1, \ldots, x_\ell\} \subseteq \mathcal{X}$ of $\ell$ elements, and a uniformly random $S \in \mathcal{S}$ we have that $(h_S(x_1), \ldots, h_S(x_\ell))$ is distributed uniformly over $\mathcal{Y}^\ell$. Several constructions of such functions exist in the literature. For example if $GF(2^n)$ is the field with $2^n$ elements, and for $s = (s_0, \ldots, s_\ell) \in GF(2^n)^{\ell+1}$ and every $n' \leq n$ we define

$$h_s(x) = \left( \sum_{i=0}^{\ell} s_i x^i \right)_{1 \ldots n'}$$

(where $z_{1 \ldots n'}$ denotes the set of $n'$ first bits of $z$) then $\{h_s\}$ is a collection of $\ell$-wise independent hash functions.

We will also use the following lemma (proven in [3]):

**Lemma 6 ([3]).** *Let $Y$ be an $n$-bit random variable with $\mathbf{H}_\infty(Y) \geq k$. Let $H = \{h_s\}_{s \in \mathcal{S}}$ be a collection of $\ell$-wise independent hash functions $h_s : \{0,1\}^n \to \{0,1\}^\alpha$ (for $\ell \geq 2$). For at least $1 - 2^{-u}$ fraction of $s \in \mathcal{S}$, we have $d(h_s(Y)) \leq \epsilon$ for*

$$u = \frac{\ell}{2}(k - \alpha - 2\log(1/\epsilon) - \log \ell + 2) - \alpha - 2. \tag{4}$$

## 2 The Definition

Formally, a *leakage-resilient storage (LRS)* scheme is a pair $\Phi \stackrel{\mathsf{def}}{=} (\mathsf{Encode}, \mathsf{Decode})$, where

- $\mathsf{Encode}$ is a randomized, efficiently computable function $\mathsf{Encode} : \{0,1\}^\alpha \to \{0,1\}^\beta$, and
- $\mathsf{Decode}$ is a deterministic, efficiently computable function $\mathsf{Decode} : \{0,1\}^\beta \to \{0,1\}^\alpha$.

Security of such a scheme is defined as follows. Consider the following game between an adversary $\mathcal{A}$ and an oracle $\mathcal{O}$ (a similar game was used to define security of the Forward-Secure Storage (FSS) [15], the main difference being that (1) FSS had a secret key and (2) the FSS game had just one round)

[2] In [16] this lemma is stated in terms of a Markov chain.

1. The adversary chooses a pair of messages $m_0, m_1 \in \{0,1\}^\alpha$ and sends them to $\mathcal{O}$.
2. $\mathcal{O}$ chooses a random bit $b \in \{0,1\}$ and sets $\tau \stackrel{\mathsf{def}}{=} \mathsf{Encode}(m_b)$.
3. The following is executed $t$ times, for $i = 1, \ldots, t$:
   (a) $\mathcal{A}$ selects a function $g_i : \{0,1\}^\beta \to \{0,1\}^{c_i} \in \Gamma$, and sends it to $\mathcal{O}$,
   (b) $\mathcal{O}$ sends $g_i(\tau)$ to $\mathcal{A}$. We say that $\mathcal{A}$ *retrieved* $c_i$ *bits from* $\tau$.
4. The adversary outputs $b'$. We say that he *won the game* if $b = b'$.

Such an adversary is called a $(\Gamma, c, t)$-adversary if $\sum_{i=1}^{t} c_i \leq c$. We say that $\Phi$ is $(\Gamma, c, t, \epsilon)$-*secure* if no $(\Gamma, c, t)$-adversary wins the game with probability greater than $\frac{1}{2} + \epsilon$.[3] We will drop $t$ and say that $\Phi$ is $(\Gamma, c, \epsilon)$-*secure* if the parameter $t$ does not matter, i.e. if no $(\Gamma, c, t)$-adversary wins the game with probability greater than $\frac{1}{2} + \epsilon$, for any $t$. Unless explicitly stated otherwise, we will assume that the adversary is computationally-unbounded. In this case we assume that the adversary is deterministic. This can be done without loss of generality, since the unbounded adversary can always compute the optimal randomness. For an adversary $\mathcal{A}$ as above, let $\mathsf{view}_\mathcal{A}$ denote the vector of values that the adversary $\mathcal{A}$ retrieves from $\tau$, i.e. $\mathsf{view}_\mathcal{A} \stackrel{\mathsf{def}}{=} (g_1(\tau), \ldots, g_t(\tau))$. Note that $|\mathsf{view}_\mathcal{A}| \leq c$.

As argued in the introduction, LRS can be viewed as a generalization of the All-Or-Nothing Transform (AONT) introduced in [37] (see also e.g. [5] for a formal definition). In our framework AONT is simply a $(\Gamma_\downarrow, c, \epsilon)$-secure LRS, $\Gamma_\downarrow$ being a set of functions $g_i$ that leak some bits of the memory, i.e. the functions that have a form $g_i(\tau_1, \ldots, \tau_\beta) = \tau_i$, where $\epsilon$ is equal to 0 if we consider perfectly-secure AONT, or is some negligible value if we consider statistically-secure AONT.

### 2.1 A Weaker Definition

In our schemes, the encoding $\tau$ of a string $m \in \{0,1\}^\alpha$ is composed of two parts: (1) the randomness $\tau_{rand}$ used to encode the message and (2) the result of the encoding process, i. e. some value $f(\tau_{rand})$ xored with the message $m$ (where $f$ is some publicly-known function). More generally, one can assume that $m$ is a member of some group $(\mathbb{G}, +)$ and $f$ has a type $\{0,1\}^* \to \mathbb{G}$. In this case the encoding of a message $m$ is $(\tau_{rand}, f(\tau_{rand}) + m)$.

For the sake of the security proofs in this paper, we will consider a game that we call a *weak attack* in which $f(\tau_{rand}) + m$ is hidden from the adversary, and the $g_i$'s are applied only to $\tau_{rand}$. The adversary in this game will be called a *weak adversary* and denoted $\mathcal{A}_{weak}$, and we will say that the LRS scheme is *weakly* $(\Gamma, c, t, \epsilon)$-*secure* if $d(f(\tau_{rand})|\mathsf{view}_{\mathcal{A}_{weak}}) \leq \epsilon$, for any $\mathcal{A}_{weak}$, where $\tau_{rand}$ is distributed uniformly over $\{0,1\}^n$. We will say that $\Gamma$ is *robust* if $\Gamma$ is closed on the operation of fixing the second part of the input, i.e. if for every $g \in \Gamma$ and every $z \in \mathbb{G}$ we have that $g'(x) := g(x, z)$ is also a member of $\Gamma$. The following lemma shows that a weakly-secure scheme is also secure according to the general definition.

**Lemma 7.** *Let $\Gamma$ be an arbitrary robust set as above. For any $c, t$ and $\epsilon$, if an encoding scheme is* weakly $(\Gamma, c, t, \epsilon)$-secure *then it is also $(\Gamma, c, t, \epsilon \cdot 2^\alpha)$-secure.*

[3] We say that $\Phi$ is *non-adaptively* $(\Gamma, c, t, \epsilon)$-*secure* if the adversary wins the game with probability at most $\frac{1}{2} + \epsilon$, with the restriction that his choice of the functions $g_i$ is non-adaptive (i.e. he has to choose all the $g_i$'s in advance).

*Proof.* Take some adversary $\mathcal{A}$ that wins the game described in Sect. 2 with some probability $0.5 + \delta$. We construct a *weak* adversary $\mathcal{A}_{weak}$ such that

$$d(f(\tau_{rand})|Out_{\mathcal{A}_{weak}}) = \delta \cdot 2^{-\alpha}, \tag{5}$$

where $Out_{\mathcal{A}_{weak}}$ is some value that is a function of $\mathsf{view}_{\mathcal{A}_{weak}}$ (we will think of it as an output of the adversary $\mathcal{A}_{weak}$ at the end of the execution). Therefore, by Lemma 3, we will have that $d(f(\tau_{rand})|\mathsf{view}_{\mathcal{A}_{weak}}) \geq \delta \cdot 2^{-\alpha}$. After showing this we will be done, by setting $\delta := \epsilon \cdot 2^{\alpha}$. The adversary $\mathcal{A}_{weak}$ works by simulating $\mathcal{A}$. First, it chooses a random string $z \in \{0,1\}^{\alpha}$ and it starts $\mathcal{A}$. Let $m_0, m_1$ be the messages that $\mathcal{A}$ outputs. Then, $\mathcal{A}_{weak}$ handles the requests issued by $\mathcal{A}$ in the following way. Recall that each request of $\mathcal{A}$ is a function $g_i : \{0,1\}^n \times \{0,1\}^{\alpha} \to \{0,1\}^{c_i}$ that should be applied to $\tau$. Each time such a request is issued, the adversary $\mathcal{A}_{weak}$ constructs a request $g'_i$ defined for every $\tau_{rand}$ as follows:

$$g'_i(\tau_{rand}) := g_i(\tau_{rand}, z).$$

(By the robustness of $\Gamma$ we have that if $g_i \in \Gamma$ then also $g'_i \in \Gamma$.) When the interaction is over and $\mathcal{A}$ outputs $b'$, the adversary $\mathcal{A}_{weak}$ outputs $Out_{\mathcal{A}_{weak}} := z - m_{b'}$. By Lemma 1 we have

$$\mathbb{P}\left[Out_{\mathcal{A}_{weak}} = f(\tau_{rand})\right] \leq 2^{-\alpha} + d(f(\tau_{rand})|Out_{\mathcal{A}_{weak}}). \tag{6}$$

Now suppose that for some $i \in \{0,1\}$ the following event $\mathcal{E}_i$ occurred: $z = m_i + f(\tau_{rand})$. In this case $\mathcal{A}_{weak}$ simply simulated the execution of $\mathcal{A}$ against the oracle $\mathcal{O}$ with $b = i$. Since $z$ is chosen uniformly hence $\mathbb{P}[\mathcal{E}_0] = \mathbb{P}[\mathcal{E}_1] = 2^{-\alpha}$. Therefore the probability that $b' = b(= i)$ is equal to $0.5 + \delta$. Moreover, in this case (i.e. when $\mathcal{E}_0 \cup \mathcal{E}_1$ occurred and $b' = b$) we get that $Out_{\mathcal{A}_{weak}} = m_i + f(\tau_{rand}) - m_{b'}$, and therefore $Out_{\mathcal{A}_{weak}} = f(\tau_{rand})$. Hence we have

$$\begin{aligned}\mathbb{P}\left[Out_{\mathcal{A}_{weak}} = f(\tau_{rand})\right] &\geq \mathbb{P}\left[b = i \mid \mathcal{E}_0 \cup \mathcal{E}_1\right] \cdot \mathbb{P}\left[\mathcal{E}_0 \cup \mathcal{E}_1\right] \\ &= (0.5 + \delta) \cdot 2^{-\alpha+1} \\ &= 2^{-\alpha} + \delta \cdot 2^{-\alpha+1}.\end{aligned}$$

Combining it with (6) we get (5). □

## 3 The Implementations

In this section we consider two types of leakage functions $\Gamma$, and show LRS schemes secure against these $\Gamma$'s relying on deterministic extractors [43,4,7,8,5]. In Sect. 3.1 we describe a construction where each leakage function can depend only on some restricted part of the memory: either because it consists of two separate blocks, or because it is infeasible for the adversary to choose a function that depends from the memory cells that are physically far from each other. In Sect. 3.2 we construct a scheme that is secure if the cardinality of the set of functions that the adversary can choose is restricted.

### 3.1 Memory Divided into Two Parts

Suppose that the encoding is stored on some physical storage device that consists of two separate chips, i.e. the memory $\mathcal{M}$ is divided into two parts $\mathcal{M}_0$ and $\mathcal{M}_1$, and each leakage function can be applied to one of the $\mathcal{M}_i$'s separately. In other words, the only restriction is that the adversary cannot choose leakage functions that depend simultaneously on both $\mathcal{M}_0$ and $\mathcal{M}_1$. More precisely, take some $\beta' < \beta$ and let $\tau = (\tau^0, \tau^1)$ where $\tau^0 \stackrel{\mathsf{def}}{=} (\tau_1, \ldots, \tau_{\beta'})$, and $\tau^1 \stackrel{\mathsf{def}}{=} (\tau_{\beta'+1}, \ldots, \tau_\beta)$. Let $\Gamma_2$ be the set of all functions $g_i$ that "depend only on $\tau^0$ or $\tau^1$", i. e. they have a form

$$g_i(\tau) = g'_i(\tau^0),$$

or

$$g_i(\tau) = g'_i(\tau^1)$$

(for some $g'_i$). Of course, $\tau^0$ and $\tau^1$ do not need to be stored on two separate memory chips, and it is enough that it is simply impossible for the adversary to compute any function of $\tau^0$ and $\tau^1$ jointly. This may happen for example if $\tau^0$ and $\tau^1$ are stored on one chip, but are physically far from each other. Observe also that the class $\Gamma_2$ includes all the functions $g(\cdot)$ that have communication complexity $c$ (where $c$ is the bound on the total amount of bits that the adversary can retrieve). This includes for example the function that computes sum of the bits in $(\tau^0, \tau^1)$ (as long as $c$ is at least logarithmic in the length of $(\tau^0, \tau^1)$).

**The construction.** One may observe that this model is very similar to the one of the two-party Intrusion-Resilient Secret Sharing (IRSS) of Dziembowski and Pietrzak (see [16], Sect. 2.1). The main difference is that the scheme of [16] has an additional property that the decoding function needs to access only small part of the encoded message. Since we do not need this property here, we can use in our construction a standard tool called *two source extractors* [7]. A function $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^\alpha$ is a $(k_0, k_1, \epsilon)$-two source extractor if it has the following property: for every two independent random variables $R_0$ and $R_1$, such that $\mathbf{H}_\infty(R_0) \geq k_0$ and $\mathbf{H}_\infty(R_1) \geq k_1$ we have that $d(\mathsf{Ext}(R_0, R_1)) \leq \epsilon$. Let $\Phi_2 \stackrel{\mathsf{def}}{=} (\mathsf{Encode}_2, \mathsf{Decode}_2)$. To encode a message $m \in \{0,1\}^\alpha$, we pick two $n$-bit strings $R_0$ and $R_1$ uniformly at random and we set

$$\tau = \mathsf{Encode}_2(m) = (\tau_{rand}, m^*) \stackrel{\mathsf{def}}{=} (R_0, R_1, \mathsf{Ext}(R_0, R_1) \oplus m)$$

and we store $R_0$ in the first part of the memory ($\mathcal{M}_0$), and $(R_1, \mathsf{Ext}(R_0, R_1) \oplus m)$ in the second part ($\mathcal{M}_1$). To decode it suffices to evaluate

$$\mathsf{Decode}_2(R_0, R_1, m^*) \stackrel{\mathsf{def}}{=} m^* \oplus \mathsf{Ext}(R_0, R_1).$$

We have the following lemma.

**Lemma 8.** *If* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^\alpha$ *is a* $(k, k, \epsilon)$*-two source extractor then* $\Phi_2$ *is* $(\Gamma_2, c, 2^\alpha \cdot \epsilon + 2^{1+\alpha+k+c-n})$*-secure.*

*Proof.* First we show that $\Phi_2$ is weakly secure against the adversary $\mathcal{A}_{weak}$ outlined in Section 2.1 (with $\tau_{rand} = (R_0, R_1)$) and then we use Lemma 7. Let $\mathcal{A}_{weak}$ be an adversary that can apply the leakage functions $g_i$ only to $\tau_{rand}$ and denote with $\mathsf{view}_{\mathcal{A}_{weak}} = (g_1(\tau_{rand}), \ldots, g_t(\tau_{rand}))$ the view of the adversary after $t$ queries to the oracle $\mathcal{O}$. We can now apply Lemma 4 (with $A = R_0$, $B = R_1$, $\phi_i = g_i$ and $V_i = g_i(\tau_{rand})$[4]) and conclude that $R_0$ and $R_1$ are independent given $\mathsf{view}_{\mathcal{A}_{weak}}$, i.e. $I(R_0; R_1|\mathsf{view}_{\mathcal{A}_{weak}}) = 0$. Moreover by Lemma 5 we know that for each $i \in \{0,1\}$

$$\mathbb{P}_{y:=\mathsf{view}_{\mathcal{A}_{weak}}}[\mathbf{H}_\infty(R_i|\mathsf{view}_{\mathcal{A}_{weak}} = y) \leq k] \leq 2^{k+c-n}.$$

Thus with probability at least $1 - 2^{1+k+c-n}$ it happens that $y = \mathsf{view}_{\mathcal{A}_{weak}}$ is such that for both $i \in \{0,1\}$ we have $\mathbf{H}_\infty(R_i|\mathsf{view}_{\mathcal{A}_{weak}} = y) \geq k$. Let $\mathcal{V}$ denote the corresponding event. We clearly have that

$$d(\mathsf{Ext}(R_0, R_1)|\mathsf{view}_{\mathcal{A}_{weak}} = y \wedge \mathcal{V}) \leq \epsilon.$$

Hence, by Lemma 2 we get that $d(\mathsf{Ext}(R_0, R_1)|\mathsf{view}_{\mathcal{A}_{weak}}) \leq \epsilon + \mathbb{P}[\overline{\mathcal{V}}] = \epsilon + 2^{1+k+c-n}$. Combining it with Lemma 7 we get that $\Phi_2$ is $(\Gamma_2, c, 2^\alpha \cdot \epsilon + 2^{1+\alpha+k+c-n})$-secure. □

**Instantiations.** Several constructions [7,39,42,10,36] of a two-source extractor exist in the literature, and can be used in our scheme. Let $\mathbb{F}$ be a finite field and denote with $\mathsf{Ext}_{\mathsf{Had}} : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ the inner product in $\mathbb{F}$, denoted $\mathsf{Ext}_{\mathsf{Had}}(x, y) = \langle x, y \rangle$. As shown in [36], for any $\delta > 0$, the function $\mathsf{Ext}_{\mathsf{Had}}$ is a $(k_{\mathsf{Had}}, k_{\mathsf{Had}}, \epsilon_{\mathsf{Had}})$-two source extractor, for $k_{\mathsf{Had}} > (1/2 + \delta)n \log |\mathbb{F}|$ and $\epsilon_{\mathsf{Had}} = |\mathbb{F}|^{(n+1)/2} 2^{-k_{\mathsf{Had}}}$ (this generalizes previous results of Chor and Goldreich [7] and Vazirani [42]). Plugging it into the construction described above we get the following LRS scheme $\Phi_{\mathsf{Had}} = (\mathsf{Encode}_{\mathsf{Had}}, \mathsf{Decode}_{\mathsf{Had}})$ for encoding messages $m \in \mathbb{F}$:

$$\begin{aligned} \mathsf{Encode}_{\mathsf{Had}}(m) &= (r_0, r_1, \langle r_0, r_1 \rangle + m) \\ \mathsf{Decode}_{\mathsf{Had}}(r_0, r_1, m^*) &= m^* - \langle r_0, r_1 \rangle. \end{aligned} \tag{7}$$

Observe that above, instead of using the xor we used the group operation in $\mathbb{F}$. This is ok, since, as explained in Sect. 2.1, one can transform a weakly-secure scheme into a standard one by using any group operation (not necessarily xor). Using Lemma 8 we get that $\Phi_{\mathsf{Had}}$ is $(\Gamma_2, c, |\mathbb{F}| \cdot \epsilon_{\mathsf{Had}} + |\mathbb{F}|^{1-n} 2^{1+k_{\mathsf{Had}}+c})$.

### 3.2 Functions That Have Small Descriptions

The second case that we consider is when the only restriction on $\Gamma$ is that it is a small set of robust functions: $|\Gamma| = 2^v$, where $v$ is some parameter (that can be for example quadratic in $\beta$). One way to look at this family is to fix some method to describe the leakage functions as binary strings, and observe that the set of functions whose description has length $v$ has exactly size $2^v$.

A natural example of such a $\Gamma$ is a set of *functions computable by Boolean circuits of a fixed size* (see e.g. [44] for an introduction to the complexity of Boolean circuits).

[4] Clearly $\phi_i$ depends only on the values $V_i$ that the adversary retrieved in the previous rounds.

Recall that the *size* of a Boolean circuit is the number $\rho$ of its gates. Each gate $G$ can be connected with two other gates $(G_1, G_2)$ (and we can assume that $G$ is an AND gate if $G_1 \neq G_2$, and it is a NOT gate otherwise). Hence, for each gate we can have at most $(\rho-1)(\rho-1) < \rho^2$ choices. Therefore there are at most $(\rho^2)^\rho = \rho^{2\rho}$ circuits of size $\rho$. Thus the circuits of size $\rho$ can be described using $2\rho \log_2 \rho$ bits.

Several natural functions can be computed by Boolean circuits of a small size (see Sect. 3 of [44]). For example every symmetric function[5] can be computed by a circuit of a linear size (in its input).

Let $\Gamma_v$ be any robust set of functions such that $|\Gamma_v| = 2^v$. We will now construct a $(\Gamma_v, c, t, \epsilon)$-secure LRS. Let $H = \{h_s : \{0,1\}^n \to \{0,1\}^\alpha\}_{s\in\mathcal{S}}$ be a collection of $\ell$-wise independent hash functions. The scheme is parameterized by a value $s \in \mathcal{S}$. For any $s \in \mathcal{S}$ let $\Phi_s \stackrel{\mathsf{def}}{=} (\mathsf{Encode}_s, \mathsf{Decode}_s)$, being

$$\mathsf{Encode}_s(m) = (R, h_s(R) \oplus m),$$

where $R \in \{0,1\}^n$ is random. Let

$$\mathsf{Decode}_s(R, d) = h_s(R) \oplus d.$$

We point out that also the above construction can be interpreted in terms of deterministic extractors. Indeed, as shown in [41] (and in [3]), $\ell$-wise independent hash functions are, with high probability, deterministic extractors for sources (with some min-entropy) that can be generated by an efficient sampling algorithm or circuit of a small size.[6] Stated in other words, an $\ell$-wise independent hash function can be viewed as a function $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^\alpha$ with the following property: for every source $R \in \{0,1\}^n$ with min-entropy $k$ which is samplable by a circuit of a small size, $\mathsf{Ext}(R)$ is close to uniform with high probability. The same construction was also used by Dodis et al. [13] in the context of AONT. Both [41] and [13] consider only the non-adaptive case. Here we show that this scheme is secure in the context of leakage-resilient storage. The following lemma states that with a good probability (over the choice of $s \in \mathcal{S}$) the scheme $\Phi_s$ is secure.

**Lemma 9.** *Fix an arbitrary robust set $\Gamma_v$ such that $|\Gamma_v| = 2^v$. For a randomly chosen $s$ with probability at least $1-\xi$ we have that $\Phi_s$ is $(\Gamma_v, c, t, 2^\alpha \cdot \epsilon + 2^{\alpha+k+c-n})$-secure, for any $c, k, t, v, \ell, \epsilon$ and $\xi$ such that*

$$\xi = 2^{tv - \frac{\ell}{2}(k-\alpha-2\log(1/\epsilon)-\log\ell+2)+\alpha+2}. \tag{8}$$

In the lemma above $k$ is a parameter, that in the proof will correspond to the min-entropy of $R$ conditioned on the view of the adversary. Observe that we have a trade-off between $2^\alpha \cdot \epsilon + 2^{\alpha+k+c-n}$ and $\xi$ (larger $k$ increases the first term, and decreases the second). The proof of this lemma is more involved and we present it in Sect. 3.2.1. Let us first discuss this lemma for more concrete values of the parameters.

[5] A function is *symmetric* if its output does not depend on the permutation of the input bits. For example every function that just depends on the sum of the input bits is symmetric. See Sect. 3.4 of [44].

[6] The approach used in [41] is orthogonal to the one used in [7]: in the latter setting, distributions can be arbitrarily complex, but they have to satisfy a strong independence requirement; in the former setting distributions have to be samplable but can involve arbitrary dependencies.

**Corollary 1.** *Fix an arbitrary robust set $\Gamma_v$ such that $|\Gamma_v| = 2^v$. For a randomly chosen $s$ with probability at least $1 - \xi$ we have that $\Phi_s$ is $(\Gamma_v, c, t, 2^{-\lambda})$-secure, for any $c, t, v, \ell, \lambda$ and $\xi$ such that*

$$\xi = 2^{tv - \frac{\ell}{2}(n-c-3\lambda-4\alpha-\log \ell - 1)+\alpha+2}. \tag{9}$$

*Proof.* Set $\epsilon := 2^{-\alpha-\lambda-1}$ and let $k := n - \lambda - 1 - \alpha - c$. Take $\Phi_s$ from Lemma 9. We have that

$$2^{\alpha} \cdot 2^{-\alpha-\lambda-1} + 2^{\alpha+k+c-n} \leq 2^{-\lambda-1} + 2^{-\lambda-1} \leq 2^{-\lambda},$$

and

$$\xi = 2^{tv - \frac{\ell}{2}((n-\lambda-1-\alpha-c)-\alpha-2(\alpha+\lambda+1)-\log \ell+2)+\alpha+2}$$

which is equal to (9). □

*Concrete values* If we want to have security against circuits of size $\chi n$ (for some constant $\chi > 1$) then the size of $\Gamma$ is equal to $2^{2\chi n \log(\chi n)}$. If we apply it $t = \omega n$ times (for some constant $\omega < 1$) then $tv = 2\chi\omega n^2 \log(\chi n)$. To be more precise set $\lambda := 24$ and $\alpha := 128$, and $n := 1024$. If we set $\chi := 10$, $\omega := 3/25$ then we can allow the adversary to retrieve at most 180 bits by setting $\ell = 278323$. With these settings we get $\xi \leq 4 \cdot 10^{-12}$.
If we consider a non-adaptive scenario, in which the adversary chooses a single leakage function (i.e. $t = 1$) and retrieves at most $c$ bits[7], then we obtain a better value for $\ell$: for $\lambda := 24$, $\alpha := 128$, $n := 1024$ and $\chi := 10$, we can allow the adversary to retrieve at most 180 bits by setting $\ell = 2203$. With these settings we get $\xi \leq 2 \cdot 10^{-28}$.

*Practical considerations* The parameter $s$ can be public. Therefore if $\xi$ is negligible, then for the real-life applications $s$ can be just chosen once and for all by some trusted party. For example, one can assume that $s = H(0)||H(1)|| \cdots$, where $H$ is some hash function (this of course can be proven secure only in the random oracle model).

Alternatively, we could just assume that $s$ is chosen independently each time $\mathsf{Encode}_s$ is calculated, and becomes a part of the encoding. In other words we could define

$$\mathsf{Encode}'(m) \stackrel{\mathsf{def}}{=} (s, \mathsf{Encode}_s(m)) \quad \text{and} \quad \mathsf{Decode}'(s, x) \stackrel{\mathsf{def}}{=} \mathsf{Decode}_s(x).$$

Of course, in this way the length $\beta$ of encoding gets larger, and hence if $\Gamma_v$ is a family of circuits whose size $\rho$ is some function of $\beta$, then $v$ becomes much larger.

### 3.2.1 Proof of Lemma 9

We first show that $\Phi_s$ is weakly secure. Suppose that the adversary $\mathcal{A}_{weak}$ performs a weak attack against $\Phi_s$. Let $R$ be distributed uniformly over $\{0,1\}^n$. Then we show that for any $\epsilon > 0$ and for at least $1 - \xi$ fraction of $s \in \mathcal{S}$ we have

$$d(h_s(R)|\mathsf{view}_{\mathcal{A}_{weak}}) \leq \epsilon + 2^{k+c-n},$$

[7] This is equivalent to consider an adversary who chooses $t > 1$ leakage functions in advance, with the same total number of retrieved bits. Note that this scenario is theoretically weaker than the adaptive one but it is useful from a practical point of view.

where $\xi$ is a function of $t, v, \ell, k, \alpha$ and $\epsilon$ as defined in (8). Consider some fixed adversary $\mathcal{A}_{weak}$. Let $Good_{\mathcal{A}_{weak}}$ denote the event that $\mathbf{H}_\infty(R|\mathsf{view}_{\mathcal{A}_{weak}} = y) \geq k$, where $y := \mathsf{view}_{\mathcal{A}_{weak}}$. By Lemma 5 we get that $\mathbb{P}\left[\overline{Good_{\mathcal{A}_{weak}}}\right] \leq 2^{k+c-n}$. On the other hand, we have

$$\mathbf{H}_\infty(R|\mathsf{view}_{\mathcal{A}_{weak}} = y, Good_{\mathcal{A}_{weak}}) \geq k.$$

Therefore, by Lemma 6 we get that

$$\mathbb{P}_s\left[d(h_s(R)|\mathsf{view}_{\mathcal{A}_{weak}} = y, Good_{\mathcal{A}_{weak}}) \geq \epsilon\right] \leq 2^{-u}, \tag{10}$$

where $\mathbb{P}_s$ means that the probability is taken over the choice of $s \in \mathcal{S}$, and $u$ is defined in (4). From Lemma 2 we get that (10) implies that

$$\mathbb{P}_s\left[d(h_s(R)|\mathsf{view}_{\mathcal{A}_{weak}}) \geq \epsilon + \mathbb{P}\left[\overline{Good_{\mathcal{A}_{weak}}}\right]\right] \leq 2^{-u},$$

which implies that

$$\mathbb{P}_s\left[d(h_s(R)|\mathsf{view}_{\mathcal{A}_{weak}}) \geq \epsilon'\right] \leq 2^{-u}, \tag{11}$$

where $\epsilon' := \epsilon + 2^{k+c-n}$. Of course (11) holds just for a fixed adversary and to complete the proof we need to give a bound on the value

$$\max_{\mathcal{A}_{weak}}\left(\mathbb{P}_s\left[d(h_s(R)|\mathsf{view}_{\mathcal{A}_{weak}}) \geq \epsilon'\right]\right). \tag{12}$$

We will do it by applying a union-bound (over all $\mathcal{A}_{weak}$) to (11). However, since that the total number of different adversaries $\mathcal{A}_{weak}$ is doubly-exponential in $c$,[8] we cannot do it in a straightforward way. Instead, we first observe that

$$\max_{\mathcal{A}_{weak}} \mathbb{P}_s\left[d(h_s(R)|\mathsf{view}_{\mathcal{A}_{weak}}) \geq \epsilon'\right] = \max_{g_1,\ldots,g_t} \mathbb{P}_s\left[d(h_s(R)|g_1(R),\ldots,g_t(R)) \geq \epsilon'\right]. \tag{13}$$

Since each $g_i \in \Gamma_v$, and $|\Gamma_v| = 2^v$ we get

$$\max_{\mathcal{A}_{weak}}\left(\mathbb{P}_s\left[d(h_s(R)|\mathsf{view}_{\mathcal{A}_{weak}}) \geq \epsilon'\right]\right) \leq (2^v)^t \cdot 2^{-u} = 2^{tv-u}. \tag{14}$$

This completes the proof, since now using Lemma 7 we are done. □

## 4 Connection with the Theory of Compressibility of NP-Instances

We believe that in general the idea to model the leakage as functions from some low complexity class is worth investigating further, as it may lead to new applications of the circuit lower bounds. Interestingly, this is probably the first scenario ever considered in cryptography in which the computing power of the adversary is smaller than the computing power of the users (during some part of the attack). A similar observation was already made in [17] (footnote 3, page 295).

It may also be worth exploring some interesting connections between this area and the theory of the compressibility of NP-instances of Harnik and Naor [23]. Informally,

[8] This is because after retrieving $c_i$ bits in the $i$th round the adversary can choose $2^v$ different functions $g_{i+1}$, hence in every round there are $2^{v \cdot 2^{c_i}}$.

an NP-language $L$ is *compressible* if every $x \in \{0,1\}^*$ can be "compressed" to a much shorter string $\mathsf{compress}(x)$ (where $g$ is some poly-time function, and $c = |\mathsf{compress}(x)| \ll |x|$) such that an infinitely powerful machine $M$ can determine if $x \in L$ just by looking at $\mathsf{compress}(x)$. Call this $(PTIME, \infty)$-$c$-compressibility. As a natural generalization of this concept, one can consider any $(\mathcal{P}_0, \mathcal{P}_1)$-compressibility (where $\mathcal{P}_0$ and $\mathcal{P}_1$ are some complexity classes): in this setting we would require that $g \in \mathcal{P}_0$, and the machine $M$ operates in $\mathcal{P}_1$.

For simplicity in this section consider only the one-round LRS's i.e. $t = 1$ (cf. game in Sect. 2). Moreover, assume that the adversary is poly-time. Informally speaking what we are looking for, when constructing a $\Gamma$-secure LRS $\Phi = (\mathsf{Encode}, \mathsf{Decode})$ is a class of problems that are not $(\Gamma, PTIME)$-$c$-compressible on average. More precisely, consider the language $L$ of all valid encodings of some fixed message $M$. Of course, if this language is $(\Gamma, PTIME)$-$c$-compressible with some probability $\epsilon$ then $\Phi$ cannot be $(\Gamma, c, 1, \epsilon)$-secure (as otherwise the adversary could just choose $\mathsf{compress}$ to be his leakage function). We leave investigating these connections as a future research direction.

## 5 Comparison with [18]

In an independent work Faust et al. [18] consider a problem of leakage-resilient computation. In their work, that can be viewed as an extension of the "private circuits" paper of [24], they provide a formal definition of a circuit computation that is secure against a class of leakages $\mathcal{L}_{TR}$ (cf. Def. 1 of [18]), and for certain classes $\mathcal{L}_{TR}$, they construct (Theorem 1, [18]) a generic transformation that, given any circuit $C$ transforms it into another circuit $C'$ that is secure against the leakages in $\mathcal{L}_{TR}$.

The main ingredient of their construction is a *linear* encoding scheme that is secure against leakages in some class $\mathcal{L}$. *Linearity* of the encoding means that the decoding function can be expressed as $\mathsf{Decode}(x_1, \ldots, x_\beta) = r_1 x_1 + \cdots + r_\beta x_\beta$, where $r_1, \ldots, r_\beta$ are constants from some field. Their definition of an encoding scheme is very similar to ours: essentially their *p-adaptive* $(\mathcal{L}, \tau, \epsilon)$*-leakage-indistinguishable encoding* is the same as our $(\mathcal{L}, c, t, \epsilon)$-secure LRS scheme. The additional parameter $\tau$, that they use indicates the running time of the adversary (that we do not consider in our paper). On the other hand we use the parameter $c$, that indicates the total amount of bits retrieved from the encoding, which is absent in [18].

We note that while the work of [18] has an obvious advantage over ours by considering not only secure storage, but also computation, our schemes cover different (and possibly more realistic) classes of leakage functions. In particular, both of the approaches in our paper cover trivially the so-called *Hamming weight attacks* [26], where the adversary is allowed to learn a sum of the bits, while the approach of [18] does not cover them.

## Acknowledgments

The authors thank Yevgeniy Dodis for a helpful discussion.

## References


1. Akavia, A., Goldwasser, S., Vaikuntanathan, V.: Simultaneous hardcore bits and cryptography against memory attacks. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 474–495. Springer, Heidelberg (2009)
2. Alwen, J., Dodis, Y., Wichs, D.: Leakage-resilient public-key cryptography in the bounded-retrieval model. In: Halevi [21], pp. 36–54
3. Barak, B., Shaltiel, R., Tromer, E.: True random number generators secure in a changing environment. In: Walter, C.D., Koç, Ç.K., Paar, C. (eds.) CHES 2003. LNCS, vol. 2779, pp. 166–180. Springer, Heidelberg (2003)
4. Blum, M.: Independent unbiased coin flips from a correlated biased source: A finite state markov chain. In: 25th Annual Symposium on Foundations of Computer Science, 1984, pp. 425–433. IEEE Computer Society, Los Alamitos (1984)
5. Canetti, R., Dodis, Y., Halevi, S., Kushilevitz, E., Sahai, A.: Exposure-Resilient Functions and All-Or-Nothing Transforms. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 453–469. Springer, Heidelberg (2000)
6. Cash, D., Ding, Y.Z., Dodis, Y., Lee, W., Lipton, R.J., Walfish, S.: Intrusion-resilient key exchange in the bounded retrieval model. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 479–498. Springer, Heidelberg (2007)
7. Chor, B., Goldreich, O.: Unbiased bits from sources of weak randomness and probabilistic communication complexity. SIAM J. on Computing 17(2), 230–261 (1988)
8. Chor, B., Goldreich, O., Håstad, J., Friedman, J., Rudich, S., Smolensky, R.: The bit extraction problem of t-resilient functions (preliminary version). In: FOCS, pp. 396–407. IEEE, Los Alamitos (1985)
9. Di Crescenzo, G., Lipton, R.J., Walfish, S.: Perfectly secure password protocols in the bounded retrieval model. In: Halevi, Rabin (eds.) [22], pp. 225–244
10. Dodis, Y., Elbaz, A., Oliveira, R., Raz, R.: Improved randomness extraction from two independent sources. In: Jansen, K., Khanna, S., Rolim, J.D.P., Ron, D. (eds.) RANDOM 2004 and APPROX 2004. LNCS, vol. 3122, pp. 334–344. Springer, Heidelberg (2004)
11. Dodis, Y., Goldwasser, S., Kalai, Y.T., Peikert, C., Vaikuntanathan, V.: Public-key encryption schemes with auxiliary inputs. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 361–381. Springer, Heidelberg (2010)
12. Dodis, Y., Kalai, Y.T., Lovett, S.: On cryptography with auxiliary input. In: STOC 2009: Proceedings of the 41st annual ACM symposium on Theory of computing, pp. 621–630. ACM, New York (2009)
13. Dodis, Y., Sahai, A., Smith, A.: On perfect and adaptive security in exposure-resilient cryptography. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 301–324. Springer, Heidelberg (2001)
14. Dziembowski., S.: Intrusion-resilience via the bounded-storage model. In: Halevi, Rabin (eds.) [22], pp. 207–224
15. Dziembowski, S.: On forward-secure storage. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 251–270. Springer, Heidelberg (2006)
16. Dziembowski, S., Pietrzak, K.: Intrusion-resilient secret sharing. In: FOCS, pp. 227–237 (2007)
17. Dziembowski, S., Pietrzak, K.: Leakage-resilient cryptography. In: FOCS 2008: Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science, Washington, DC, USA. IEEE Computer Society, Los Alamitos (2008)
18. Faust, S., Rabin, T., Reyzin, L., Tromer, E., Vaikuntanathan, V.: Protecting circuits from leakage: The computationally-bounded and noisy cases. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 135–156. Springer, Heidelberg (2010)

19. Gandolfi, K., Mourtel, C., Olivier, F.: Electromagnetic analysis: Concrete results. In: Koç, Ç.K., Naccache, D., Paar, C. (eds.) CHES 2001. LNCS, vol. 2162, pp. 251–261. Springer, Heidelberg (2001)
20. Halderman, A.J., Schoen, S.D., Heninger, N., Clarkson, W., Paul, W., Calandrino, J.A., Feldman, A.J., Appelbaum, J., Felten, E.W.: Lest we remember: cold-boot attacks on encryption keys. Commun. ACM 52(5), 91–98 (2009)
21. Halevi, S. (ed.): CRYPTO 2009. LNCS, vol. 5677. Springer, Heidelberg (2009)
22. Halevi, S., Rabin, T. (eds.): TCC 2006. LNCS, vol. 3876. Springer, Heidelberg (2006)
23. Harnik, D., Naor, M.: On the compressibility of NP instances and cryptographic applications. In: FOCS 2006: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science, Washington, DC, USA, pp. 719–728. IEEE Computer Society, Los Alamitos (2006)
24. Ishai, Y., Sahai, A., Wagner, D.: Private Circuits: Securing Hardware against Probing Attacks. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 463–481. Springer, Heidelberg (2003)
25. Katz, J., Vaikuntanathan, V.: Signature schemes with bounded leakage resilience. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 703–720. Springer, Heidelberg (2009)
26. Kelsey, J., Schneier, B., Wagner, D., Hall, C.: Side channel cryptanalysis of product ciphers. Journal of Computer Security 8, 141–158 (2000)
27. Kocher, P.C.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 104–113. Springer, Heidelberg (1996)
28. Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999)
29. Maurer, U.M.: Conditionally-perfect secrecy and a provably-secure randomized cipher. J. Cryptology 5(1), 53–66 (1992)
30. Micali, S., Reyzin, L.: Physically observable cryptography (extended abstract). In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 278–296. Springer, Heidelberg (2004)
31. Naor, M., Segev, G.: Public-key cryptosystems resilient to key leakage. In: Halevi (ed.) [21], pp. 18–35
32. European Network of Excellence (ECRYPT). The side channel cryptanalysis lounge, `http://www.crypto.ruhr-uni-bochum.de/en_sclounge.html` (retrieved on 29.03.2010)
33. Pietrzak, K.: A leakage-resilient mode of operation. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 462–482. Springer, Heidelberg (2010)
34. Quisquater, J.-J., Koene, F.: Side channel attacks: State of the art. In: [32] (October 2002)
35. Quisquater, J.-J., Samyde, D.: Electromagnetic analysis (EMA): Measures and countermeasures for smart cards. In: Attali, S., Jensen, T. (eds.) E-smart 2001. LNCS, vol. 2140, pp. 200–210. Springer, Heidelberg (2001)
36. Rao, A.: An exposition of Bourgain's 2-source extractor. Electronic Colloquium on Computational Complexity (ECCC), 14(034) (2007), `http://eccc.hpi-web.de/eccc-reports/2007/TR07-034/index.html`
37. Rivest, R.L.: All-or-nothing encryption and the package transform. In: Biham, E. (ed.) FSE 1997. LNCS, vol. 1267, pp. 210–218. Springer, Heidelberg (1997)
38. Samyde, D., Skorobogatov, S., Anderson, R., Quisquater, J.-J.: On a new way to read data from memory. In: SISW 2002: Proceedings of the First International IEEE Security in Storage Workshop, Washington, DC, USA, p. 65. IEEE Computer Society, Los Alamitos (2002)
39. Shaltiel, R.: How to get more mileage from randomness extractors. In: CCC 2006: Proceedings of the 21st Annual IEEE Conference on Computational Complexity, Washington, DC, USA, 2006, pp. 46–60. IEEE Computer Society, Los Alamitos (2006)

40. Shamir, A., Tromer, E.: Acoustic cryptanalysis. on nosy people and noisy machines, http://people.csail.mit.edu/tromer/acoustic/ (accessed on 29.03.2010)
41. Trevisan, L., Vadhan, S.P.: Extracting randomness from samplable distributions. In: FOCS, pp. 32–42 (2000)
42. Vazirani, U.V.: Strong communication complexity or generating quasirandom sequences form two communicating semi-random sources. Combinatorica 7(4), 375–392 (1987)
43. von Neumann, J.: Various techniques used in connection with random digits. J. Research Nat. Bur. Stand., Appl. Math. Series 12, 36–38 (1951)
44. Wegener, I.: The Complexity of Boolean Functions. John Wiley and Sons Ltd., B. G. Teubner (1987), http://eccc.hpi-web.de/static/books/The_Complexity_of_Boolean_Functions/

## A Proof of Lemma 2

Before showing this lemma let us first prove the following:

**Lemma 10.** *For every random variable $X$ and events $\mathcal{E}, \mathcal{H}$ we have*

$$d(X|\mathcal{H}) \leq d(X|\mathcal{H} \wedge \mathcal{E}) + \mathbb{P}\left[\overline{\mathcal{E}}|\mathcal{H}\right]. \tag{15}$$

*Proof.* It is enough to show that

$$\Delta(P_{X|\mathcal{H}}; P_{X|\mathcal{H}\wedge\mathcal{E}}) \leq \mathbb{P}\left[\overline{\mathcal{E}}|\mathcal{H}\right]. \tag{16}$$

After showing this we will be done, since from the triangle inequality we have

$$\overbrace{\Delta(P_{X|\mathcal{H}}; U_{\mathcal{X}})}^{=d(X|\mathcal{H})} \leq \overbrace{\Delta(P_{X|\mathcal{H}\wedge\mathcal{E}}\ ;\ U_{\mathcal{X}})}^{=d(X|\mathcal{H}\wedge\mathcal{E})} + \Delta(P_{X|\mathcal{H}}; P_{X|\mathcal{H}\wedge\mathcal{E}}),$$

where $U_{\mathcal{X}}$ denotes the uniform distribution over $\mathcal{X}$. Let $\mathcal{F}$ denote the set

$$\{x : \mathbb{P}[X = x|\mathcal{H}] > \mathbb{P}[X = x|\mathcal{H} \wedge \mathcal{E}]\}.$$

We have that the left-hand side of (16) is equal to

$$\sum_{x\in\mathcal{F}} \mathbb{P}[X = x|\mathcal{H}] - \overbrace{\mathbb{P}[X = x|\mathcal{H} \wedge \mathcal{E}]}^{=\frac{\mathbb{P}[X=x\wedge\mathcal{E}|\mathcal{H}]}{\mathbb{P}[\mathcal{E}|\mathcal{H}]} \geq \mathbb{P}[X=x\wedge\mathcal{E}|\mathcal{H}]}. \tag{17}$$

$$\leq \sum_{x\in\mathcal{F}} \mathbb{P}[X = x|\mathcal{H}] - \mathbb{P}[X = x \wedge \mathcal{E}|\mathcal{H}] \tag{18}$$

$$= \sum_{x\in\mathcal{F}} \mathbb{P}[X = x|\mathcal{H}] - \sum_{x\in\mathcal{F}} \mathbb{P}[X = x \wedge \mathcal{E}|\mathcal{H}] \tag{19}$$

$$= \mathbb{P}[X \in \mathcal{F}|\mathcal{H}] - \mathbb{P}[(X \in \mathcal{F}) \wedge \mathcal{E}|\mathcal{H}] \tag{20}$$

$$\leq \mathbb{P}\left[\overline{\mathcal{E}}|\mathcal{H}\right]. \tag{21}$$

□

*Proof (of Lemma 2).* The right-hand side of (2) is equal to

$$\sum_y d(X|Y=y)\cdot \mathbb{P}\left[Y=y\right], \tag{22}$$

and the left-hand side of (2) is equal to

$$\sum_y \left(d(X|(Y=y)\wedge\mathcal{E}) + \mathbb{P}\left[\overline{\mathcal{E}}|Y=y\right]\right)\cdot \mathbb{P}\left[Y=y\right]. \tag{23}$$

To finish the proof it suffices to show that for every $y$ we have

$$d(X|(Y=y)\wedge\mathcal{E}) + \mathbb{P}\left[\overline{\mathcal{E}}|Y=y\right] \geq d(X|Y=y).$$

This follows directly from Lemma 10, with $\mathcal{H}$ being the event that $Y=y$. □

## B Proof of Lemma 5

*Proof.* We prove that (3) holds for a fixed $w$. This clearly implies that (3) holds when $W$ is a random variable independent on $X$. Since $|f(X,w)|\leq c$, hence the number of all $y$'s is at most equal to $2^c$. Therefore the number of $x$'s for which there exists some $y$ such that

$$|x : f(x,w)=y| \leq 2^k \tag{24}$$

holds is at most $2^{c+k}$. Hence the probability that for a *random* $X$ we have that (24) holds is at most $2^{c+k-n}$. Since clearly if (24) does not hold then $\mathbf{H}_\infty(X|f(X,w)=y) > k$ we get that

$$\mathbb{P}_{y:=f(X,w)}(\mathbf{H}_\infty(X|f(X,w)=y)\leq k) \leq 2^{c+k-n}.$$

Thus we are done. □