

Provable Security for Physical Cryptography ^{*}

Krzysztof Pietrzak

CWI Amsterdam, The Netherlands

Abstract. The modern approach to cryptography is provable security, where one defines a meaningful formal security model and proves that schemes are secure in this model. An exception is the design of countermeasures against cryptographic side-channel attacks, which even today is mostly based on heuristic arguments, which only try to prevent particular attacks.

It was long believed that side-channels are a practical problem where theoretical cryptography was only of limited use, but recent results indicate that this view is too pessimistic, and in fact, it is possible to extend the realm of provable security also to side-channel attacks. This survey is a personal and incomplete view on the current state of this exciting and fast moving field.

1 Modern Cryptography

For most of history, cryptography was the art of “secret communication”. The designers of encryption schemes were only guided by experience and intuition. Not surprisingly, pretty much all proposed schemes turned out to be insecure. It became evident that the only hope to get secure cryptosystems is by means of provable security, that is

1. to provide a precise and meaningful model capturing what it means to be “secure”.
2. to design systems which can be proven secure in this model.

Provable security dates back at least to Shannon’s proof that the *one-time pad* hides all information about the encrypted message [Sha49], but only with the rise of public-key cryptography [DH76,RSA78,Ell70,Coc73,Wil74], which requires constructions with a rich mathematical structure that can also be exploited by cryptanalysts, did provable security really take off.

Modern cryptographic security definitions usually consider a “security game”, which models how a potential adversary can attack the system. Classical examples are the definitions of CPA/CCA secure public-key encryption schemes [GM84,RS92], unforgeability for signatures schemes [GMR88] or pseudorandomness [Yao82,BM84]. More recent notions are security against key-dependent message attacks [CL01,BRS03,HK07,HU08] or security against selective openings [DNRS99,BHY09].

Proving security of a system then equates to showing that no (efficient) adversary can win the security game. Unfortunately, often one cannot hope to prove such a strong statement (as it e.g. would imply $\mathcal{P} \neq \mathcal{NP}$). In this cases one shows that the existence of an adversary who can win the game would imply that some problem generally believed to be hard is actually easy. Public-key cryptosystems can be based on many well studied assumptions, like the hardness of factoring [Rab79,HK09], or the shortest vector problem in lattices

^{*} This survey accompanies a talk with the same title given at the WEWORC’09 workshop. First posted online September 27, 2009. Last update March 9, 2010

[GGH97,Reg05]. Symmetric cryptography (aka. secret-key cryptography) can be based on even much weaker assumptions, e.g. block-ciphers can be built from any one-way function [HILL99,GGM84,LR88], but for efficiency reasons, in practice block-ciphers like DES or AES are actually constructed from scratch and not via reductions ([NR97] is a notable exception).

1.1 Why Black-Box Isn't Enough

What basically all modern security notions have in common, is that the cryptographic algorithm is modelled as a “black-box”, where an adversary can only observe the input/output behavior of the cryptographic algorithm as specified by the security game. Unfortunately, such models do not capture many real world scenarios where an adversary can attack an actual *implementation* of a cryptosystem which potentially leaks information to the adversary that cannot be learned from black-box access alone.

In the mid-90s Kocher demonstrated that the secret-key of the popular RSA cryptosystem can be recovered by simply measuring the time a cryptodevice needs to perform a decryption [Koc96a]. Such attacks, where an adversary exploits leakage of information from a cryptodevice during execution, are called “side-channel” attacks (as opposed to standard cryptanalytic attacks, where the adversary only exploits the “main-channel” – i.e. the legitimate input/output behavior– of the device.)

Light-weight crypto devices like smart-cards or RFID chips are particularly susceptible to side-channel attacks, and although [Koc96b] was by no means the first side-channel attack, it came quite as a surprise to the cryptographic community how easily such devices could be broken. Since [Koc96b], many more ingenious side-channel attacks have been published, for example by measuring the power-consumption [KJJ99] or the electromagnetic radiation [QS01,GMO01] of a cryptodevice. Some attacks go beyond simply measuring some physical properties of a device. Cold-boot attacks [HSH⁺08] exploit the fact that memory retains its content for several seconds or even minutes even after being ripped from a laptop. In a probing attack [AKA96], one measures the contents carried by some wires of the circuit which performs a computation (unlike the other attacks, probing attacks require rather elaborate equipment). A particularly intriguing class of attacks are “cache attacks” [OST06,RTSS09] which exploit leakage of information between different processes that run on the same CPU. Such leakage does occur due to the structure of memory caches on modern CPUs.

Side-channel attacks are a very real threat for systems used in practice. A recent example is the complete break of the KeeLoq cipher which is used as anti-theft system in millions of cars [EKM⁺08]. Not surprisingly, much research has concentrated on developing countermeasures against such attacks.

This research is mostly done by practitioners (i.e., the cryptographic hardware community) who are also active in finding and exploiting new side-channels. It was long believed that theory can only be of limited use to prevent side-channel attacks. But recent results indicate that this view was much too pessimistic, and in fact it is possible to extend the realm of provable security also to side-channel attacks as we will see in this survey.

We will only discuss countermeasures against *passive* attacks, where an adversary only observes leakage from a cryptodevice. In contrast, in an active attack [BDL97,BS91] the adversary does actively tamper with the device, for example by cutting wires in the circuit or by heating or overclocking it in order to introduce random errors to the computation or memory content. Currently, there are only very few results on provable security against active attacks [GLM⁺04,IPSW06,DPW09], but this is likely to change in the near future.

2 Side-Channel Attacks and Countermeasures

Countermeasures against side-channel attacks – as outlined above – can be on the hardware or algorithmic level.

- On the hardware level, the aim is to construct physical devices which reduce the amount of leakage, for example by shielding the circuit (to avoid electromagnetic radiation) or by inserting transistors (to flatten the power consumption curve).
- On the algorithmic level, the aim is to design cryptosystem which remain secure even if some information about the secret internal state is leaked. This is usually done by some kind of internal randomisation (called masking or blinding, cf. [oEE] for a list of relevant papers) in order to avoid the occurrence of predictable intermediate results.

As argued in § 1.2 of [FRR⁺10], due to the holographic bound conjecture – which asserts that the information contained in a volume of space is already encoded on the boundary to this region – in theory everything that goes on in a cryptodevice can be learned by measuring its surroundings. Fortunately, in practice we can still hope to get secure systems as (1) a real-world adversary will not even get close to a perfect measurement of that boundary, and (2) even if what the adversary measures contains all of the information about the secret state of the cryptodevice, it might still be computationally hard to extract any useful information (i.e. cryptographic keys) from it.

In practice we thus can reasonably assume that a cryptodevice can keep at least some secrets, but it’s unrealistic to assume the other extreme, i.e. that a “useful” device like a smart-card will leak no information at all. Thus to get secure devices, a combination of hardware and algorithmic countermeasures must be in place.

2.1 Provable Security against Side-Channel Attacks?

As already mentioned, most of the work on side-channel attacks and countermeasures is done by practitioners, i.e. the cryptographic hardware community, the CHES workshop is their major venue. The “side-channel cryptanalysis lounge” gives a good overview [oEE] on this field. To some extent this research is a cat and mouse game: new side-channels attacks are found, and subsequently countermeasures are proposed. Those are usually ad-hoc, in the sense that they aim at preventing some particular known type of attack and they often come without any formal security proof.

Research on side-channel security is quite different from the provable-security approach followed by modern cryptography. For example, in many works on side-channel countermeasures one encounters security arguments involving *simulations*. Because simulations can only show that some particular countermeasure is secure against some particular attack, they are meaningless in the context of provable security, where one has to quantify over all (time and/or space bounded) adversaries.

Clearly, this situation cannot be satisfying from a cryptographic point of view. What are our beautiful provably secure cryptosystems good for, when ultimately their security relies on some ad-hoc countermeasures against side-channel attacks? Despite this, until recently the theory community did not give much attention to this problem. One reason was the perception that side-channels are a practical problem, and theory can only be of limited use to prevent them.

It is not obvious what “provable security” should mean in the context of side-channel attacks. We can’t really hope to *prove* that a physical implementation like a smart card

does not leak its entire internal state, thus, we will always *assume* that the leakage from the physical device can be modelled by some class of leakage functions \mathcal{F} , and then *prove* that under this assumption, the implementation is secure. Several models have been proposed, which differ in how powerful the class \mathcal{F} is (we will consider $\mathcal{P}, \mathcal{AC}_0$, uninvertible functions and more particular classes) and how the leakage functions is applied (i.e. continuous or not, with restrictions on the range and/or domain of the functions).

A main difference to the traditional, more applied approach to side-channel countermeasures is that one only restricts the class of leakage functions, but not in which ways an adversary can exploit this leakage, by e.g. only considering template attacks [SMY09,PSP⁺08].¹ This distinction is crucial, as there is no way to argue why an adversary should restrict herself to running some particular algorithm on the measured leakage. In contrast, limiting the class of leakage functions is meaningful (and to some extent necessary), as the adversary is limited in what leakage she can learn by the physical properties of the cryptodevice and her measurement equipment.

What is gained by using provable security as just described? After all, in order to get a secure implementation of a scheme, one still has to construct hardware whose potential leakage is captured by the class \mathcal{F} for which this scheme can be proven secure.

Security: The main advantage is the security guarantee we can give for the *implementation* of the scheme. For example consider the case where \mathcal{F} contains all efficient functions of bounded range as in the model of leakage-resilience discussed in Section 2.7. It is unlikely that the implementation of a leakage-resilient cryptosystem will turn out be insecure due to a side-channel that was not foreseen by the designer of the device as, no matter what kind of new side-channel attack will be discovered, it can only threaten the implementation if it has a very high “leakage capacity”, that is, the attack must exploit a significant amount of information leaked with every invocation. This contrasts ad-hoc schemes which potentially can be broken by a side-channel attack that exploits only very few (even less than one) bits of information leaked with every invocation.

Modularity: Another advantage is the “modularity” of the approach: cryptographers can design schemes which are secure in some precisely defined leakage-model without having to care about any aspects of physical side-channel attacks. On the other hand, engineers can construct hardware whose leakage is captured by the leakage-model without having to understand anything about the actual schemes that will be implemented on it. In particular, once such hardware is in place, it can be used to securely implement any scheme proven secure in the leakage-model considered.

2.2 Physically Observable Cryptography.

Micali and Reyzin [MR04] already proposed the elegant and influential framework of “physically observable cryptography”. Unlike the other works we discuss below, the focus of [MR04] is not to construct primitives that are secure against some class of leakage functions \mathcal{F} from scratch, but rather on side-channel security preserving *reductions*. They observe that standard cryptographic reductions will in general fail in the presence of leakage, but in some cases one can still get meaningful results. In particular, they show that the Blum-Micali

¹ Sometimes template attacks are referred to as “Bayesian adversaries”, which is somewhat misleading as those are not adversaries in a cryptographic sense (i.e. only resource bounded), but refer to a very specific attack.

generator [BM84] is an unpredictable bit-generator (i.e. it outputs bits x_1, x_2, \dots such that x_{i+1} looks random after x_1, \dots, x_i has been computed), assuming the underlying one-way permutation $f(\cdot)$ can be *implemented* with a very strong security guarantee: for a random z , the output $f(z)$ looks uniformly random given all the leakage that resulted from the computation of $f(z)$.

The framework of Micali and Reyzin is based on five axioms which they assume leakage from physical devices to adhere. We will discuss their first, and somewhat controversial “only computation leaks information” axiom later.

2.3 Private Circuits

Ishai et al. [ISW03,IPSW06] consider a model where the adversary can choose some wires on a circuit on which the cryptographic algorithm is computed, and then learns the values carried by those wires during the computation. Moreover they consider *continuous leakage*, that is the adversary can make such a measurement on *every* invocation of the circuit.

What makes their work exceptional is that they were the first to *prove* how to implement *any* algorithm secure against an interesting side-channel (i.e. probing attacks).² Recently, Faust et al. [FRR⁺10] show that surprisingly, such a general compiler can even be constructed for leakage functions that get all the values carried by all the wires as input, as long as the functions is from a very low complexity class like \mathcal{AC}_0 . This work is particularly interesting as it seems to give the first cryptographic application of unconditional lower bounds for constant depth circuits [Hås86].

The drawback of those general compilers is that the amount of leakage that can be tolerated is very small: in [ISW03], to tolerate t bits leakage, the circuits must be blown up by a factor of at least t . Although this limits t to rather small values, it is still very meaningful with respect to the attacks considered, i.e. probing attacks, which become very impractical once one has to measure several wires simultaneously. The construction from [FRR⁺10] additionally requires (albeit very simple) completely leakage-proof modules.³

2.4 Protecting Storage

In some side-channel attacks, most notably cold-boot attacks [HSH⁺08], the adversary learns just a subset of the bits stored in memory. This attack can be successful even if this fraction is rather small,⁴ in particular, [HS09] show how to recover an RSA key given just a 0.27 fraction of the key bits. Such attacks can be used to e.g. break disc encryption schemes, where the encryption key is at many times on the memory in the clear (and not password protected as on the hard disk).

² Formally, Ishai et al. prove the following: let $t \geq 0$ be some constant and let $[X]$ denote a $(t+1)$ out of $(t+1)$ secret sharing of the value X . They construct a general compiler, which turns every circuit $G(\cdot)$ into a circuit $G_t(\cdot)$ (of size $O(t|G|)$) such that $[G(X)] = G_t([X])$ for all inputs X , and moreover one does not learn any information on $G(X)$ even when given the value carried by any t wires in the circuit $G_t(\cdot)$ while evaluating the input $[X]$. This transformation uses multiparty-computation, which is quite different from all other approaches we discuss here.

³ It is an interesting open problem if already the construction from [ISW03] is secure against leakage from low complexity classes.

⁴ How large this fraction is depends on the physical properties of the memory and on how fast this attack can be launched (i.e. how long the memory is without power.)

Assuming the fraction of bits learned by the adversary is less than 1, one can to some extent protect against such attacks by keeping a (randomized) encoding $f(s)$ on the memory (whenever s is not used), where the encoding guarantees that s remains secret, even if a subset of the bits of $f(s)$ is leaked. All-or-nothing transforms [Riv97], t -resilient functions [CGH⁺85] and exposure-resilient functions⁵ [CDH⁺00,DSS01] achieve exactly this.

Recently Davì and Dziembowski [DD09] consider the general problem of “leakage-resilient storage”, in particular, they show a probabilistic encoding Enc such that leakage $f(c)$ of a codeword $c = \text{Enc}(s)$ contains almost no information about s assuming only that (1) the range of f is bounded and (2) f can be computed by circuits of small size.

The solutions discussed above can only protect against a cold-boot attack if one can be sure that the secret(s) is encoded at the time-point when the memory is removed. Next we discuss particular cryptosystems which remain secure in the much more hostile setting where the adversary can leak any bounded-range function of the key. Note that in this setting it is unavoidable that information about a key is leaked, thus the best one can hope for is that the particular cryptosystem using this key remains secure even after leakage.

2.5 Security Against Memory Attacks

Akavia et al. [AGV09] define the security-notion of “security against memory attacks”. The idea is to consider a standard security notion, but give the adversary some extra power: she can initially chose any efficient leakage function $f : \{0, 1\}^n \rightarrow \{0, 1\}^{n \cdot c}$ and gets the leakage $f(sk)$. Here n is the length of the secret key sk and c is some parameter. Clearly, one must require $c < 1$ as otherwise f could just output the entire sk . Also stronger notions have been considered [KV09,NS09], where the leakage function does not only get the secret key as input, but the entire randomness which was used (by some key-generation algorithm) to sample it and/or additional state information like the randomness used by a signature scheme.

Akavia et al. [AGV09] show that the public-key encryption scheme of Regev [Reg05] and the identity based scheme of Gentry et al. [GPV08] are secure against memory attacks under basically the same hardness assumptions (about lattices) as the original schemes. Naor and Segev [NS09] show that any hash-proof system [CS03,KD04,KPSY09], initially introduced by Cramer-Shoup to construct chosen-ciphertext secure public-key encryptions schemes, can be used in a straight forward way to get public-key encryption secure against memory attacks. They achieve CPA/CCA1 and CCA2 security for leakage $c = 1 - \epsilon$, $c = 1/4 - \epsilon$ and $c = 1/6 - \epsilon$ for any $\epsilon > 0$ respectively (where c is the parameter as in the first paragraph of this section.)

Dodis et al. [ADW09] and Katz and Vaikuntanathan [KV09] construct signature schemes which are secure against memory attacks. In particular, they show that many signature schemes (like Okamoto or Schnorr) derived via the Fiat-Shamir [FS87] transformation are secure against memory-attacks. This schemes can leak even a fraction $c = 1 - \epsilon$, but due to the Fiat-Shamir transform, the above scheme can only be proven secure in the random oracle model. [KV09] also propose a scheme in the *standard* model, but which is inefficient as it uses NIZK proofs, and an efficient *one-time* signature scheme in the standard model.

⁵ A functions $f(\cdot)$ is t -resilient if any t bits of $f(s)$ are statistically independent of s . Exposure resilient functions are defined similarly, but using a relaxed security notion (statistical instead of perfect independence), which allows to leak larger t , i.e. t can be a $1 - \epsilon$ fraction of all bits, as opposed to less than $1/2$ which is the best one can get for t -resilient functions as proven in [CGH⁺85].

History recap on bounded storage/retrieval/leakage models. Ideally, a cryptographic scheme should be secure against *any* adversary, that is secure in an information theoretical sense. Unfortunately, there are inherent limitations to what can be achieved information theoretically. Already Shannon proved that the one-time pad is basically optimal: to encrypt a message of length n , the sender and receiver must share a key of length at least n .

To overcome this bound, one can put some reasonable assumptions about the power of an adversary. Usually, one assumes that the adversary is *time* bounded (e.g. can be modelled as a Turing machine running in polynomial time.) Maurer [Mau90] proposes a completely different approach, the “bounded storage model”, where one assumes only a bound on the *storage* of the adversary. Informally, one assumes a (huge) secret R is available for some short time to all parties, but the adversary can only save the output $f(R)$ of a (not necessarily efficient) compressing function f , say, $|f(R)| = |R|/2$.⁶

Subsequently, the bounded-retrieval model (BRM) was proposed independently by Dziembowski and Di Crescenzo et al. [Dzi06,DLW06]. Here the users have large (2GB, say) secret keys, which are subject to large amount (1GB, say) of adversarial leakage (unlike in the bounded memory model, there is no other “short” secret the adversary has no access to). The BRM model for example captures a setting where we want to protect cryptographic keys even on a computer which is infected by malware (a virus or Trojan) that can perform any computation on the computer, but can only communicate a bounded amount of information (like 1GB) back to the bad guys. In this model symmetric authentication schemes [Dzi06,DLW06,CDD⁺07], password authentication [DLW06] and secret-sharing [DP07] were constructed. Recently the first *public-key* primitives in the BRM model were constructed by Alwen et al. [ADW09].

The setting of memory attacks discussed before is basically the BRM model, but ignoring all the issues that additionally come up when using huge keys while requiring the computations of the honest parties to be efficient.

2.6 Auxiliary Input

The absolutely minimal assumption one must make on a leakage function $f(\cdot)$ in order to hope for any security is that it is hard to invert, that is, given $f(sk)$ it is hard to find sk .⁷ Leakage functions as considered in memory attacks where $|f(sk)|$ is significantly shorter than $|sk|$ are unpredictable in an information theoretic sense: even a computationally unbounded adversary cannot find sk given $f(sk)$ with probability $\geq 2^{|f(sk)|-|sk|}$. Dodis, Kalai and Lovett [DKL09] construct a symmetric encryption scheme which remains secure given leakage $f(sk)$ for any function which is *exponentially* hard to invert, that is no polynomial time adversary can compute sk given $f(sk)$ with advantage more than $2^{-\epsilon \cdot sk}$ for some $\epsilon > 0$ (but sk can be information theoretically determined given $f(sk)$). The ultimate goal in this line of research is to construct cryptosystems which remain secure for all unpredictable functions which cannot be inverted by efficient adversaries with non-negligible (and not only exponentially small)

⁶ The honest parties, using some short shared secret key k can (very efficiently, only looking at small parts of R) extract a long key $K = ext(k, R)$, $|R| \gg |K| \gg |k|$, such that K is close to uniformly random given the adversary's view $f(R)$ (and then e.g. use K as a one-time pad to encrypt messages which are much longer than k). This model was further investigated in [AR99,DM02], yielding ultimately to the notion of locally computable extractors [Vad04].

⁷ Note that an uninvertible function is not necessarily a one-way function, where one requires that given $f(sk)$ it is hard to find *any* sk' such that $f(sk) = f(sk')$. E.g. the constant function is not one way, but uninvertible.

advantage. Besides capturing a larger class of functions (than memory attacks), another major advantage of this “security against auxiliary input” notion is the fact that it composes (cf. [DKL09] for a discussion). More recently Dodis et al. construct *public-key* cryptosystems secure against auxiliary input [DGK⁺10], also relaxing the required hardness of inverting the leakage function from exponential $2^{-sk-\epsilon}$ to sub-exponential 2^{-sk^ϵ} .

2.7 Leakage-Resilient Cryptography

Security against memory attacks only implies security as long as the *total* amount of leakage is no larger than the length of the secret key. This is sufficient for attacks where an adversary can only make one measurement like in cold-boot attacks. Bit in many, if not most, practically relevant side-channel attacks the adversary can measure a bounded amount of information on each invocation of the cryptosystem. The leakages from several invocations are then combined to break the system, which usually means recovering the secret key.

We already discussed private circuits, which provide a general compiler that makes any scheme secure against continuous leakage, but the current solutions are restricted to settings where on each invocation the values carried by a few particular wires [ISW03], or a few bits computed by a constant depth circuit [FRR⁺10] are leaked. Ideally, we want cryptosystems which can tolerate *lots of general leakage* (i.e. a constant fraction of the key length) with *every invocation*. Leakage-resilience, introduced in [DP08], is such a model.

Leakage-Resilience Model. Informally, a cryptosystem is *leakage-resilient* if it remains secure, even if the adversary is not only given the usual black-box access to the system, but additionally, with every invocation, can learn some $\lambda \in \mathbb{N}$ bits of arbitrary information about the entire computation.

This is modelled by letting the adversary adaptively choose any efficient “leakage-function” f before each query, and after the invocation she gets – besides the usual output – also the output of f . We put restrictions on the domain and range of f :

bounded range: The range of f is $\{0, 1\}^\lambda$ for some parameter $\lambda \in \mathbb{N}$.

bounded domain: f gets as input all the data that is accessed during this invocation.

More precisely, the bounded domain restriction means that f gets as input only a subset $S^+ \subseteq S$ of the entire secret state S that was actually accessed during this invocation. If the cryptosystem is probabilistic, then the leakage function also gets all the random coins used. We don’t have to explicitly give f the input to the cryptosystem (if there is any), as it is adversarially chosen together with f and thus can be “hard-coded” into f .

On bounded range and domain. A mathematical model of side-channel leakage is only good if it captures (and thus implies security against) leakage that occurs in practice.

Bounded range. It’s not clear how to determine how much information actual hardware like a smart-card does leak. Actually, in most side-channel attacks the adversary measures large amounts of data, e.g. an entire power-consumption curve. So at a glance this assumption might seem unreasonable, but this is a bit overly pessimistic.

Even though side-channel leakage may contain lots of data, only a small fraction can actually be exploited. On the other hand, the model of leakage-resilience allows only for the leakage of a small number λ of bits, but this leakage is “worst case” in the sense that the adversary may choose the leakage-function which outputs the most useful

information. Below we discuss two ways in which this observation can be made precise. The first shows that side-channel attacks used in practice are captured by leakage-resilience as they only exploit few bits of information from each invocation. The second is a relaxation of bounded leakage which can reasonably be assumed to be satisfied in practice.

Side-Channel Attacks Exploit Few Bits. Many side-channel attacks first measure large amounts of leakage A_1, A_2, \dots from every invocation, like a power consumption curve. Then, in a first step, each leakage A_i is preprocessed in order to extract some “useful” information A'_i (this A'_i could e.g. be a list of the most likely subkeys.) The attack then proceeds by trying to recover the secret key from A'_1, A'_2, \dots . Such attacks are covered by leakage-resilience whenever the amount of extracted data $|A'_i|$ is at most the amount of leakage λ allowed per invocation.

Relaxing Bounded Range. By inspecting the proofs of the leakage-resilient constructions [DP08, Pie09, KP09] one sees that a restriction on the leakage-functions is required which is weaker than restricting the range to λ bits: all one needs that the leakage $f(S^+)$ does not decrease the so called⁸ HILL-pseudoentropy [HILL99, BSW03] the adversary has about the active state S^+ by more than λ bits. The same is true for some constructions proven secure against memory attacks as discussed in Section 2.5, in particular [NS09].⁹ For digital signatures (leakage-resilient [FKPR10] or secure against memory attacks [KV09]) such a relaxation is not possible, as an unbounded leakage function could simply output a valid signature. Such leakage trivially breaks unforgeability, but might not decrease pseudoentropy at all.

Bounded domain. The *bounded domain* restriction is a very mild restriction. Unlike for bounded range, it’s nontrivial to even imagine a remotely realistic side-channel attack which would break a scheme by not adhering to it. This restriction (on how leakage functions are mathematically modelled) is implied by the “only computation leaks information” axiom (which states something about physical properties of devices) of [MR04]. But it also covers other practical attacks which do not satisfy this axiom. For example note that an adversary can learn any linear function $f(S)$ of the *entire* state S (which is split in, say, two parts S_1, S_2 that are accessed individually) by specifying leakage functions f_1, f_2 such that $f_1(a) + f_2(b) = f(a, b)$ (the adversary can ask to learn $f_1(S_1)$ and $f_2(S_2)$ as S_1 and S_2 are accessed respectively, and then compute $f(S)$ locally.) This simple observation already shows that claims made in the literature arguing that the bounded range & domain restrictions do not cover attacks like “cold-boot attacks” [HSH⁺08] or static leakage (as claimed in [SPY⁺09]) are not well-founded.¹⁰ As argued by Dziembowski,¹¹ this restriction not only covers all linear function $f(a, b) = f_1(a) + f_2(b)$, but

⁸ HILL-pseudoentropy is a computational analogue of min-entropy. As for min-entropy, λ bits of information cannot decrease it (in expectation) by more than λ bits.

⁹ The only reason that [DP08] defines leakage-resilience by bounding the range instead putting a bound on the degradation of pseudoentropy is that the latter is quite technical and not very intuitive.

¹⁰ In the above argument we implicitly assumed that ultimately the entire secret state will be touched, although this seems obvious (after all, why would one save a secret state if it’s not supposed to be ever read), the tokens used in the construction of one-time programs [GKR08] are an example where exactly this happens. For such primitives obeying the “only computation leaks information” axiom in its original physical sense is necessary.

¹¹ at the “Provable security against physical attacks workshop”, February 2010, Leiden.

in fact any function $f(a, b)$ which has a communication complexity at most λ . A good candidate for an actual leakage function that does invalidate this assumption¹² is the inner product $f(a, b) = \sum_i a_i \cdot b_i \bmod 2$ which has linear communication complexity.

Currently, the only symmetric (aka. secret-key) leakage-resilient primitives are stream-ciphers [DP08, Pie09]. The only asymmetric (aka. public-key) leakage-resilient primitives are digital signatures [FKPR10], and public-key encryption [KP09], the latter only in the idealized generic group model.

Stream Ciphers. A stream-cipher is a function $\text{SC} : \{0, 1\}^k \rightarrow \{0, 1\}^k \times \{0, 1\}^n$, which is instantiated with random secret-key $K_0 \in \{0, 1\}^k$, and then generates a stream X_1, X_2, \dots of output blocks $X_i \in \{0, 1\}^n$ recursively as

$$(K_i, X_i) \leftarrow \text{SC}(K_{i-1})$$

The standard security definition requires the X_i to be pseudorandom, that is, no efficient adversary can distinguish the X_i 's from uniformly random strings. By definition, the state K_0, K_1, \dots of a stream-cipher “evolves”. Because of this property it is potentially easier to achieve provable security against side-channel attacks, and in fact, stream-ciphers have been investigated in several works. As discussed in Section 2.2, Micali-Reyzin investigate the security of the Blum-Micali generator.

Kocher [Koc05] discusses a very simple construction, where one computes the state iteratively using a hash-function H (SHA256 is proposed) as $(K_i, X_i) \leftarrow H(K_{i-1})$, and the X_i 's are used as session-keys in some protocol. Although it is only very informally argued why such a construction should be secure, already this simple construction looks quite reasonable,¹³ at least it's not obvious how existing side-channel attacks could break it. Unfortunately, it seems impossible to actually prove anything about this construction. Standaert et al. [SPY⁺09] prove this simple iterating construction is leakage-resilient in a idealized model where H is a uniformly random function that can be queried by the adversary, but not by the leakage-functions.¹⁴

Leakage-Resilient Secret-Key Cryptography. The first leakage-resilient primitive, a stream-cipher, was constructed in [DP08]. In some sense, this construction is a computational version of “alternating extraction” from [DP07] using a result on “dense subsets of pseudorandom sets” which was independently, and in a more general context (cf. Section 5 of [Tre09]), discovered by Reingold et al. [RTTV08].¹⁵ Subsequently, a simplified construction shown in Figure 1 was given in [Pie09]. It can be instantiated with any pseudorandom function, whereas [DP08] used extractors.

¹² and thus might be used to construct an actual real world counterexample where the security of an implementation gets broken because the bounded domain restriction is invalidated.

¹³ Actually, Kocher considers a slightly different construction $K_i \leftarrow H(K_{i-1})$ and the same K_i is used as a session key and to update the state. This seems like a bad idea for several reasons, e.g. all $K_j, j \geq i$ will be compromised if a single session key K_i is compromised.

¹⁴ This is basically the random-oracle model [BR93], except that in the RO model one usually assumes that all parties have access to the RO, and security proofs only exploit the fact that ROs provide a means to efficiently access an exponential amount of uniform randomness to all parties.

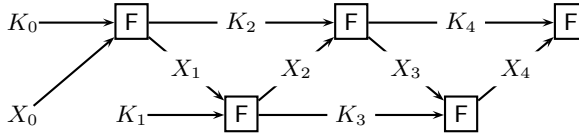


Fig. 1. Illustration of the first four rounds of the leakage-resilient stream-cipher SC^F from a any weak pseudorandom function F from [Pie09]. The initial key is $[K_0, K_1, X_0]$, the output are the blocks X_1, X_2, X_3, \dots

Leakage-Resilient Public-Key Cryptography. Faust et al. [FKPR10] construct leakage-resilient signatures. The basic idea is to use any signature scheme which is secure against memory attacks (as discussed in §2.5), and use it in a tree based mode [NY89]. In particular, using the one-time signature schemes secure against memory attacks (as constructed in [KV09]) one gets leakage-resilient signatures under the minimal assumption that one-way functions exist, and where the amount of leakage on each invocation can be even a constant fraction of the state that is accessed. Due to the tree based structure, the length of a signature is logarithmic in the number of messages that can be signed. Constructing leakage-resilient signatures of constant length is an interesting open problem.

Constructing a leakage-resilient public-key encryption (PKE) scheme is more challenging. Intuitively, the reason is that it is not clear how to “evolve” the secret key of a PKE scheme, as it always must correspond to some fixed public-key (for signatures, this can be done by sampling a new secret/public key pair and sign it using the old key). In [KP09] a different approach to leakage-resilience is taken which is not based on evolving, but rather sharing, the secret key.¹⁶ A leakage-resilient version of the popular ElGamal [ElG86] PKE scheme is shown, but unlike the constructions discussed above, which can be proven in the standard model, the security proof of this scheme is only given in the idealized generic-group model [Sho97, MW98]. Constructing a leakage-resilient PKE scheme in the standard model is still open.

More Open Problems. For now, we only have leakage-resilient realizations of some particular primitives, but no general compiler like for private circuits §2.3. Private circuits use techniques from general multiparty computation [GMW87, BGW88], to apply this ideas to leakage-resilience, we would need a secret-sharing scheme which (1) is secure even if a bounded amount of information about *every* share is leaked and (2) the scheme has enough structure, like multiplicity, to allow for MPC. The first condition is satisfied by intrusion-resilient secret-sharing [DP07], the second e.g. by Shamir’s scheme [Sha79], but we don’t know a scheme which has both properties simultaneously.

Whereas in [SPY⁺09] the security proof exploits the fact that the leakage functions cannot *query* the RO.

¹⁵ Recall that a pseudorandom generator $PRG : \{0, 1\}^n \rightarrow \{0, 1\}^m, m > n$ is a length increasing function where for a random seed X , the output $PRG(X)$ is indistinguishable from random by any efficient distinguisher. The result shown in [DP08, RTTV08] state that if X has only high entropy (but is not necessarily uniform), then $PRG(X)$ has high “pseudoentropy”. That means that a pseudorandom generator, which is designed to generate lots of pseudorandomness from a short uniform seed, also does generate pseudoentropy from a short seed with high entropy.

¹⁶ The general idea to secret-share a secret key in order to counter side-channel attacks is well known in the side-channel community under the name “blinding”.

Even if such a general compiler is feasible, it is unlikely to give highly efficient and thus practical constructions. Pseudorandom functions and permutations (i.e. block-ciphers) are the “work-horses” of cryptography, and thus achieving leakage resilience for them is of particular interest. As observed already by Micali and Reyzin [MR04], most standard cryptographic reductions do not preserve security in the presence of side-channel attacks. This very much applies to the notion of leakage-resilience. In particular, the GGM construction [GGM86] of a PRF from a PRG, does not preserve leakage-resilience. Standaert et al. [SPY⁺09] show that GGM is a leakage-resilient PRF in an idealized model where the PRG is a random function, the leakage-functions may not query this random function (so far it’s the same restrictions as discussed in Footnote 14) and moreover the leakage-function is fixed and not adaptively chosen by the adversary before every query. Recently Dodis and Pietrzak [DP10] give a construction of a PRF which can be seen as a hybrid of GGM and the leakage-resilient stream-cipher from [Pie09]. This construction also requires the leakage-function to be fixed, but otherwise is leakage-resilient in the standard model.

References

- [ADW09] Joël Alwen, Yevgeniy Dodis, and Daniel Wichs. Leakage-resilient public-key cryptography in the bounded-retrieval model. In Shai Halevi, editor, *CRYPTO 2009*, LNCS, pages 36–54, Santa Barbara, CA, USA, August 2009. Springer, Berlin, Germany.
- [AGV09] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 474–495. Springer, Berlin, Germany, March 15–17, 2009.
- [AKA96] Ross Anderson, Markus Kuhn, and England U. S. A. Tamper resistance - a cautionary note. In *In Proceedings of the Second Usenix Workshop on Electronic Commerce*, pages 1–11, 1996.
- [AR99] Yonatan Aumann and Michael O. Rabin. Information theoretically secure communication in the limited storage space model. In Michael J. Wiener, editor, *CRYPTO’99*, volume 1666 of *LNCS*, pages 65–79, Santa Barbara, CA, USA, August 15–19, 1999. Springer, Berlin, Germany.
- [BDL97] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the importance of checking cryptographic protocols for faults (extended abstract). In Walter Fumy, editor, *EUROCRYPT’97*, volume 1233 of *LNCS*, pages 37–51, Konstanz, Germany, May 11–15, 1997. Springer, Berlin, Germany.
- [BGW88] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for noncryptographic fault-tolerant distributed computations. In *20th ACM STOC*, pages 1–10, Chicago, Illinois, USA, May 2–4, 1988. ACM Press.
- [BHY09] Mihir Bellare, Dennis Hofheinz, and Scott Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 1–35, Cologne, Germany, April 26–30, 2009. Springer, Berlin, Germany.
- [BM84] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudorandom bits. *SIAM Journal on Computing*, 13(4):850–864, 1984.
- [BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93*, pages 62–73, Fairfax, Virginia, USA, November 3–5, 1993. ACM Press.
- [BRS03] John Black, Phillip Rogaway, and Thomas Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In Kaisa Nyberg and Howard M. Heys, editors, *SAC 2002*, volume 2595 of *LNCS*, pages 62–75, St. John’s, Newfoundland, Canada, August 15–16, 2003. Springer, Berlin, Germany.

- [BS91] Eli Biham and Adi Shamir. Differential cryptanalysis of DES-like cryptosystems. *Journal of Cryptology*, 4(1):3–72, 1991.
- [BSW03] Boaz Barak, Ronen Shaltiel, and Avi Wigderson. Computational analogues of entropy. In *RANDOM-APPROX*, pages 200–215, 2003.
- [CDD⁺07] 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 2007*, volume 4392 of *LNCS*, pages 479–498, Amsterdam, The Netherlands, February 21–24, 2007. Springer, Berlin, Germany.
- [CDH⁺00] Ran Canetti, Yevgeniy Dodis, Shai Halevi, Eyal Kushilevitz, and Amit Sahai. Exposure-resilient functions and all-or-nothing transforms. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 453–469, Bruges, Belgium, May 14–18, 2000. Springer, Berlin, Germany.
- [CGH⁺85] Benny Chor, Oded Goldreich, Johan Håstad, Joel Friedman, Steven Rudich, and Roman Smolensky. The bit extraction problem of t-resilient functions (preliminary version). In *FOCS*, 1985.
- [CL01] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118, Innsbruck, Austria, May 6–10, 2001. Springer, Berlin, Germany.
- [Coc73] C. C. Cocks. A note on non-secret encryption, 1973. Available from <http://www.cesg.gov.uk/site/publications/media/notense.pdf>.
- [CS03] Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003.
- [DD09] Francesco Davi and Stefan Dziembowski. Leakage-resilient storage. Cryptology ePrint Archive, Report 2009/399, 2009. <http://eprint.iacr.org/>.
- [DGK⁺10] Yevgeniy Dodis, Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In *TCC 2010*, LNCS, pages 361–381. Springer, Berlin, Germany, 2010.
- [DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.
- [DKL09] Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett. On cryptography with auxiliary input. In *41st ACM STOC*, pages 621–630. ACM Press, May 17–20, 2009.
- [DLW06] Giovanni Di Crescenzo, Richard J. Lipton, and Shabsi Walfish. Perfectly secure password protocols in the bounded retrieval model. In Shai Halevi and Tal Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 225–244, New York, NY, USA, March 4–7, 2006. Springer, Berlin, Germany.
- [DM02] Stefan Dziembowski and Ueli M. Maurer. Tight security proofs for the bounded-storage model. In *34th ACM STOC*, pages 341–350, Montréal, Québec, Canada, May 19–21, 2002. ACM Press.
- [DNRS99] Cynthia Dwork, Moni Naor, Omer Reingold, and Larry J. Stockmeyer. Magic functions. In *40th FOCS*, pages 523–534, New York, New York, USA, October 17–19, 1999. IEEE Computer Society Press.
- [DP07] Stefan Dziembowski and Krzysztof Pietrzak. Intrusion-resilient secret sharing. In *48th FOCS*, pages 227–237, Providence, USA, October 20–23, 2007. IEEE Computer Society Press.
- [DP08] Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *49th FOCS*, pages 293–302, Philadelphia, Pennsylvania, USA, October 25–28, 2008. IEEE Computer Society Press.
- [DP10] Yevgeniy Dodis and Krzysztof Pietrzak. Leakage-resilient pseudorandom functions and side-channel attacks on feistel, 2010. Manuscript.
- [DPW09] Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs. Non-malleable codes. Manuscript, 2009.

- [DSS01] Yevgeniy Dodis, Amit Sahai, and Adam Smith. On perfect and adaptive security in exposure-resilient cryptography. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 301–324, Innsbruck, Austria, May 6–10, 2001. Springer, Berlin, Germany.
- [Dzi06] Stefan Dziembowski. Intrusion-resilience via the bounded-storage model. In Shai Halevi and Tal Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 207–224, New York, NY, USA, March 4–7, 2006. Springer, Berlin, Germany.
- [EKM⁺08] Thomas Eisenbarth, Timo Kasper, Amir Moradi, Christof Paar, Mahmoud Salmasizadeh, and Mohammad T. Manzuri Shalmani. On the power of power analysis in the real world: A complete break of the keeloqcode hopping scheme. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 203–220, Santa Barbara, CA, USA, August 17–21, 2008. Springer, Berlin, Germany.
- [ElG86] Taher ElGamal. On computing logarithms over finite fields. In Hugh C. Williams, editor, *CRYPTO'85*, volume 218 of *LNCS*, pages 396–402, Santa Barbara, CA, USA, August 18–22, 1986. Springer, Berlin, Germany.
- [Ell70] J.H. Ellis. The possibility of secure non-secret digital encryption, 1970. Available from <http://www.cesg.gov.uk/possnse.htm>.
- [FKPR10] Sebastian Faust, Eike Kiltz, Krzysztof Pietrzak, and Guy N. Rothblum. Leakage-resilient signatures. In *TCC 2010*, LNCS, pages 343–360. Springer, Berlin, Germany, 2010.
- [FRR⁺10] Sebastian Faust, Tal Rabin, Leonid Reyzin, Eran Tromer, and Vinod Vaikuntanathan. Protecting circuits from leakage: The computationally-bounded and noisy cases. In *EUROCRYPT 2010*, LNCS. Springer, Berlin, Germany, May 2010.
- [FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194, Santa Barbara, CA, USA, August 1987. Springer, Berlin, Germany.
- [GGH97] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In Burton S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 112–131, Santa Barbara, CA, USA, August 17–21, 1997. Springer, Berlin, Germany.
- [GGM84] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. In *25th FOCS*, pages 464–479, Singer Island, Florida, October 24–26, 1984. IEEE Computer Society Press.
- [GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33:792–807, 1986.
- [GKR08] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. One-time programs. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 39–56, Santa Barbara, CA, USA, August 17–21, 2008. Springer, Berlin, Germany.
- [GLM⁺04] Rosario Gennaro, Anna Lysyanskaya, Tal Malkin, Silvio Micali, and Tal Rabin. Algorithmic tamper-proof (atp) security: Theoretical foundations for security against hardware tampering. In *TCC*, pages 258–277, 2004.
- [GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.
- [GMO01] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic analysis: Concrete results. In Çetin Kaya Koç, David Naccache, and Christof Paar, editors, *CHES 2001*, volume 2162 of *LNCS*, pages 251–261, Paris, France, May 14–16, 2001. Springer, Berlin, Germany.
- [GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, April 1988.
- [GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game, or a completeness theorem for protocols with honest majority. In Alfred Aho, editor, *19th*

- ACM STOC*, pages 218–229, New York City,, New York, USA, May 25–27, 1987. ACM Press.
- [GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206, Victoria, British Columbia, Canada, May 17–20, 2008. ACM Press.
- [Hås86] Johan Håstad. Almost optimal lower bounds for small depth circuits. In *STOC*, pages 6–20, 1986.
- [HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.
- [HK07] Shai Halevi and Hugo Krawczyk. Security under key-dependent inputs. In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors, *ACM CCS 07*, pages 466–475, Alexandria, Virginia, USA, October 28–31, 2007. ACM Press.
- [HK09] Dennis Hofheinz and Eike Kiltz. Practical chosen ciphertext secure encryption from factoring. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 313–332, Cologne, Germany, April 26–30, 2009. Springer, Berlin, Germany.
- [HS09] Nadia Heninger and Hovav Shacham. Reconstructing RSA private keys from random key bits. In Shai Halevi, editor, *CRYPTO 2009*, LNCS, pages 1–17, Santa Barbara, CA, USA, August 2009. Springer, Berlin, Germany.
- [HSH⁺08] J. Alex Halderman, Seth D. Schoen, Nadia Heninger, William Clarkson, William Paul, Joseph A. Calandrino, Ariel J. Feldman, Jacob Appelbaum, and Edward W. Felten. Lest we remember: Cold boot attacks on encryption keys. In *USENIX Security Symposium*, pages 45–60, 2008.
- [HU08] Dennis Hofheinz and Dominique Unruh. Towards key-dependent message security in the standard model. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 108–126, Istanbul, Turkey, April 13–17, 2008. Springer, Berlin, Germany.
- [IPSW06] Yuval Ishai, Manoj Prabhakaran, Amit Sahai, and David Wagner. Private circuits II: Keeping secrets in tamperable circuits. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 308–327, St. Petersburg, Russia, May 28 – June 1, 2006. Springer, Berlin, Germany.
- [ISW03] Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 463–481, Santa Barbara, CA, USA, August 17–21, 2003. Springer, Berlin, Germany.
- [KD04] Kaoru Kurosawa and Yvo Desmedt. A new paradigm of hybrid encryption scheme. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 426–442, Santa Barbara, CA, USA, August 15–19, 2004. Springer, Berlin, Germany.
- [KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael J. Wiener, editor, *CRYPTO’99*, volume 1666 of *LNCS*, pages 388–397, Santa Barbara, CA, USA, August 15–19, 1999. Springer, Berlin, Germany.
- [Koc96a] Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *CRYPTO*, pages 104–113, 1996.
- [Koc96b] Paul C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In Neal Koblitz, editor, *CRYPTO’96*, volume 1109 of *LNCS*, pages 104–113, Santa Barbara, CA, USA, August 18–22, 1996. Springer, Berlin, Germany.
- [Koc05] Paul C. Kocher. Design and validation strategies for obtaining assurance in countermeasures to power analysis and related attacks. In *Proceedings of the NIST Physical Security Workshop*, 2005.
- [KP09] Eike Kiltz and Krzysztof Pietrzak. How to secure elgamal against side-channel attacks. Manuscript, 2009.
- [KPSY09] Eike Kiltz, Krzysztof Pietrzak, Martijn Stam, and Moti Yung. A new randomness extraction paradigm for hybrid encryption. In Antoine Joux, editor, *EUROCRYPT 2009*,

- volume 5479 of *LNCS*, pages 590–609, Cologne, Germany, April 26–30, 2009. Springer, Berlin, Germany.
- [KV09] Jonathan Katz and Vinod Vaikuntanathan. Signature schemes with bounded leakage resilience. In *ASIACRYPT 2009*, LNCS, pages 703–720. Springer, Berlin, Germany, December 2009.
- [LR88] Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, 17(2), 1988.
- [Mau90] Ueli M. Maurer. A provably-secure strongly-randomized cipher. In Ivan Damgård, editor, *EUROCRYPT'90*, volume 473 of *LNCS*, pages 361–373, Aarhus, Denmark, May 21–24, 1990. Springer, Berlin, Germany.
- [MR04] Silvio Micali and Leonid Reyzin. Physically observable cryptography (extended abstract). In *TCC*, pages 278–296, 2004.
- [MW98] Ueli M. Maurer and Stefan Wolf. Lower bounds on generic algorithms in groups. In Kaisa Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*, pages 72–84, Espoo, Finland, May 31 – June 4, 1998. Springer, Berlin, Germany.
- [NR97] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *38th FOCS*, pages 458–467, Miami Beach, Florida, October 19–22, 1997. IEEE Computer Society Press.
- [NS09] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In Shai Halevi, editor, *CRYPTO 2009*, LNCS, pages 18–35, Santa Barbara, CA, USA, August 2009. Springer, Berlin, Germany.
- [NY89] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *21st ACM STOC*, pages 33–43, Seattle, Washington, USA, May 15–17, 1989. ACM Press.
- [oEE] 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.
- [OST06] Dag Arne Osvik, Adi Shamir, and Eran Tromer. Cache attacks and countermeasures: The case of AES. In David Pointcheval, editor, *CT-RSA 2006*, volume 3860 of *LNCS*, pages 1–20, San Jose, CA, USA, February 13–17, 2006. Springer, Berlin, Germany.
- [Pie09] Krzysztof Pietrzak. A leakage-resilient mode of operation. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 462–482, Cologne, Germany, April 26–30, 2009. Springer, Berlin, Germany.
- [PSP⁺08] Christophe Petit, François-Xavier Standaert, Olivier Pereira, Tal Malkin, and Moti Yung. A block cipher based pseudo random number generator secure against side-channel key recovery. In *ASIACCS*, pages 56–65, 2008.
- [QS01] Jean-Jacques Quisquater and David Samyde. Electromagnetic analysis (ema): Measures and counter-measures for smart cards. In *E-smart*, pages 200–210, 2001.
- [Rab79] M. O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical report, Cambridge, MA, USA, 1979.
- [Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93, Baltimore, Maryland, USA, May 22–24, 2005. ACM Press.
- [Riv97] Ronald L. Rivest. All-or-nothing encryption and the package transform. In Eli Biham, editor, *FSE'97*, volume 1267 of *LNCS*, pages 210–218, Haifa, Israel, January 20–22, 1997. Springer, Berlin, Germany.
- [RS92] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 433–444, Santa Barbara, CA, USA, August 11–15, 1992. Springer, Berlin, Germany.
- [RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signature and public-key cryptosystems. *Communications of the Association for Computing Machinery*, 21(2):120–126, 1978.

- [RTSS09] Thomas Ristenpart, Eran Tromer, Hovav Shacham, and Stefan Savage. Hey, you, get off of my cloud! Exploring information leakage in third-party compute clouds. In Somesh Jha and Angelos Keromytis, editors, *Proceedings of CCS 2009*. ACM Press, November 2009. To appear.
- [RTTV08] Omer Reingold, Luca Trevisan, Madhur Tulsiani, and Salil P. Vadhan. Dense subsets of pseudorandom sets. In *49th FOCS*, pages 76–85, Philadelphia, Pennsylvania, USA, October 25–28, 2008. IEEE Computer Society Press.
- [Sha49] Claude E. Shannon. Communication theory of secrecy systems. *Bell Systems Technical Journal*, 28(4):656–715, 1949.
- [Sha79] Adi Shamir. How to share a secret. *Communications of the Association for Computing Machinery*, 22(11):612–613, November 1979.
- [Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266, Konstanz, Germany, May 11–15, 1997. Springer, Berlin, Germany.
- [SMY09] François-Xavier Standaert, Tal Malkin, and Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 443–461, Cologne, Germany, April 26–30, 2009. Springer, Berlin, Germany.
- [SPY⁺09] Francois-Xavier Standaert, Olivier Pereira, Yu Yu, Jean-Jacques Quisquater, Moti Yung, and Elisabeth Oswald. Leakage resilient cryptography in practice. Cryptology ePrint Archive, Report 2009/341, 2009. <http://eprint.iacr.org/>.
- [Tre09] Luca Trevisan. Guest column: additive combinatorics and theoretical computer science. *SIGACT News*, 40(2):50–66, 2009.
- [Vad04] Salil P. Vadhan. Constructing locally computable extractors and cryptosystems in the bounded-storage model. *Journal of Cryptology*, 17(1):43–77, January 2004.
- [Wil74] M.J. Williamson. Non-secret encryption using a finite field, 1974. Available from <http://www.cesg.gov.uk/site/publications/media/secenc.pdf>.
- [Yao82] Andrew C. Yao. Theory and applications of trapdoor functions. In *23rd FOCS*, pages 80–91, Chicago, Illinois, November 3–5, 1982. IEEE Computer Society Press.