On the Impossibility of Cryptography with Tamperable Randomness

Per Austrin[1,⋆], Kai-Min Chung[2,⋆⋆], Mohammad Mahmoody[3,⋆⋆⋆], Rafael Pass[4,†], and Karn Seth[4]

[1] KTH Royal Institute of Technology austrin@kth.se
[2] Academica Sinica kmchung@iis.sinica.edu.tw
[3] University of Virginia mohammad@cs.virginia.edu
[4] Cornell {rafael,karn}@cs.cornell.edu

Abstract. We initiate a study of the security of cryptographic primitives in the presence of efficient tampering attacks to the randomness of honest parties. More precisely, we consider p-tampering attackers that may *efficiently* tamper with each bit of the honest parties' random tape with probability p, but have to do so in an "online" fashion. Our main result is a strong negative result: We show that any secure encryption scheme, bit commitment scheme, or zero-knowledge protocol can be "broken" with probability p by a p-tampering attacker. The core of this result is a new Fourier analytic technique for biasing the output of bounded-value functions, which may be of independent interest.
We also show that this result cannot be extended to primitives such as signature schemes and identification protocols: assuming the existence of one-way functions, such primitives can be made resilient to $(1/\mathrm{poly}(n))$-tampering attacks where n is the security parameter.[5]

Keywords Tampering, Randomness, Encryption.

1 Introduction

A traditional assumption in cryptography is that the only way for an attacker to gather or control information is by receiving and sending messages to honest

⋆ Work done while at University of Toronto, funded by NSERC
⋆⋆ Supported in part by NSF Award CNS-1217821.
⋆⋆⋆ Research done while in part supported by NSF Award CNS-1217821.
† Pass is supported in part by a Alfred P. Sloan Fellowship, Microsoft New Faculty Fellowship, NSF Award CNS-1217821, NSF CAREER Award CCF-0746990, NSF Award CCF-1214844, AFOSR YIP Award FA9550-10-1-0093, and DARPA and AFRL under contract FA8750-11-2- 0211. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the US Government.
[5] The full version of the paper is available at [4].

parties. In particular, it is assumed that the attacker may not access the *internal states* of honest parties. However, such assumptions on the attacker—which we refer to as *physical assumptions*—are quite strong (and even unrealistic). In real-life, an attacker may through a "physical attack" learn some "leakage" about the honest parties' internal states and may even tamper with their internal states. For instance, a computer virus may (e.g., using a, so-called, buffer overflow attack [2, 18, 32]) be able to bias the randomness of an infected computer. Understanding to what extents the traditional physical assumptions can be relaxed, to capture such realistic attacks, is of fundamental importance.

Indeed, in recent years *leakage-resilient cryptography*—that is, the design of cryptographic schemes and protocols that remain secure even when the attacker may receive (partial) leakage about the internal state of the honest parties—has received significant attention (see e.g., [1, 7, 10, 12–14, 21, 25, 30, 31, 34]).

In this work, we focus on understanding the power of *tampering attacks*—that is, attacks where the adversary may partially modify (i.e., tamper with) the internal state of honest parties. Early results in the 1990's already demonstrate that tampering attacks may be very powerful: by just slightly tampering with the computation of specific implementations of some cryptographic schemes (e.g., natural implementations of RSA encryption [33]), Boneh, DeMillo and Lipton [6] demonstrated that the security of these schemes can be broken completely.

Previous works on tamper-resilient cryptography consider tampering of computation [3,5,6,16,24,30] and tampering with the memory of an honest party who holds a secret (e.g., a signing or a decryption algorithm) [9,15,19,26,29,30]. This line of research culminated in strong *compilers* turning any polynomial-size circuit C into a new "tamper-resilient" polynomial-size circuit C'; tamper-resilience here means that having "grey-box" access to C' (i.e., having black-box access while tampering with the computation of C') yields no more "knowledge" than simply having black-box access to C. These works, thus, show how to keep a secret hidden from a tampering attacker. Our focus here is somewhat different. In analogy with recent work of leakage-resilient security, we aim to investigate to what extent a tampering attacker may violate the *security* of a cryptographic protocol by tampering with the internal state of honest parities.

For concreteness, let us focus on the security of public-key encryption schemes (but as we shall see shortly, our investigation applies to many more cryptographic tasks such as zero-knowledge proofs and secure computation). Roughly speaking, we require a tamper-resilient encryption scheme to guarantee that ciphertexts conceal the encrypted messages, even if the internal computation of the sender (of the ciphertext) has been tampered with.[6] As first observation note that if the attacker may completely change the computation of the sender, he could simply

[6] Let us remark that the simulation property of tamper-resilient compilers do not necessarily guarantee that if the sender algorithm is compiled into a "tamper-resilient" version, then the encryption scheme is tamper-resilient. This is due to the fact that the simulation property of those compilers only guarantee that an attacker cannot learn more from tampering with the sender strategy than it could have with black-box access to it. But in the case of encryption schemes, it is actually the *input* to the algorithm (i.e., the message to be encrypted) that we wish to hide (as opposed

make the sender send the message in the clear. Thus, to hope for any reasonable notion of tamper-resilient security we need to restrict the attacker's ability to tamper with the computation.

Tampering with Randomness. Among various computational resources, randomness might be one of the hardest to protect against tampering. This is due to the fact that randomness is usually *generated* (perhaps based on some "physical" resources available to the system) and any malicious attacker who is able to change the bits along their generation can mount a tampering attack against the randomness. Indeed given the breakthrough results of [22, 27, 28] it is becoming even more clear that randomness is one of the most vulnerable aspects of a cryptographic system. Thus, a very basic question is to what extent we can protect our systems against tampering with randomness. In this work we initiate a formal study of this question by considering tampering attacks to the randomness of the honest players; namely we study the following basic question:

> *Can security of cryptographic primitives be preserved under tampering attacks to the randomness of honest parties?*

Note that we need to restrict the tampering ability of the attacker, for otherwise the adversary can effectively make the scheme deterministic by always fixing the randomness of the honest parties to all zeros. But it is well-known that deterministic encryption schemes cannot be semantically secure. Therefore, here we initiate study of the power of *weak* types of tampering attacks to the randomness of the honest parties.

General Model: The Tampering Virus. We envision the adversary as consisting of two separate entities: **(1)** a *classical attacker* who interacts with the honest parties only by sending/receiving messages to/from them (without any side-channels), and **(2)** a *tampering circuit* (a.k.a. the "virus") who observes the internal state of the honest parties and may *only* tamper with their randomness (but may not communicate with the outside world, and in particular with the classical attacker). The tampering circuit only gets to tamper with a *small fraction* of the random bits, and in an *efficient* manner. Note that this model excludes a scenario in which the virus (even efficiently) samples the *whole* randomness, regardless of the original randomness sampled by the system, because in this cases all of the sampled tampered bits might be different from the system's original random seed. However, here we study weak attackers who only tamper with a *small fraction* of the random bits. In fact, previous works on *resettable cryptography* [8] can be interpreted as achieving tamper resilience against adversaries who tamper with *all* of the randomness of the honest parties by *resampling* the randomness of the honest parties and executing them again and again. This is incomparable to our model, since our adversary does not have control over

to some secret held by the algorithm). See the full version for a more detailed comparison with previous work.

the honest parties' execution (to run them again), but is more powerful in that it could change the value of some of the random bits.

Online Tampering. Let $0 < p < 1$ be the parameter describing the "power" of adversary (which defines the fraction of tampered bits). It still remains to clarify how the tampering is done over these bits. The first naive model would allow the adversary to tamper with a p fraction of the bits *after* all the bits are sampled by the system (and, thus, are known to the virus as well). However, this is not realistic since the sequence of random bits used by the system could always be sampled in an *online* manner; namely, the system could sample the i-th random bit whenever it needs it and might use it "on the fly". Therefore, a tampering adversary also needs to tamper with them one-by-one, efficiently, and in an on-line fashion.

More precisely, we consider a so-called *p-tampering attack*, where the adversary gets to tamper with the random tape of the honest players as follows. The randomness of honest parties is generated bit-by-bit, and for each generated bit x_i the efficient tampering circuit gets to tamper with it with independent probability p having only knowledge of previously generated random bits $x_1, x_2, \ldots, x_{i-1}$ (but not the value of the random bits tossed in the future). Roughly speaking, requiring security with respect to p-tampering attacks amounts to requiring that security holds even if the honest players' randomness comes from a *computationally efficient* analog of a Santha-Vazirani (SV) source [35]. Recall that a random variable $X = (X_1, \ldots, X_n)$ over bit strings is an SV source with parameter δ if for every $i \in [n]$ and every $(x_1, \ldots, x_i) \in \{0,1\}^i$, it holds that $\delta \leq \mathsf{P}[X_i = x_i | X_1 = x_1, \ldots, X_{i-1} = x_{i-1}] \leq 1 - \delta$. It is easy to see that the random variable resulting from performing a p-tampering attack on a uniform n-bit string is an SV source with parameter $(1-p)/2$; in fact, any SV source is equivalent to performing a *computationally unbounded* p-tampering attack on a uniform n-bit string.

The main focus of this work is on the following question:

Can security be achieved under p-tampering attacks?

2 Our Results and Techniques

Our main result is a strong negative answer to the question above for a variety of basic cryptographic primitives. A p-tampering attacker can break all of the following with advantage $\Omega(p)$: **(1)** the security of any CPA-secure (public-key or private-key) encryption scheme, **(2)** the zero-knowledge property of any efficient-prover proof (or argument) system for nontrivial languages, **(3)** the hiding property of any commitment scheme, and **(4)** the security of any protocol for computing a "nontrivial" finite function. More formally, we prove the following theorems.

Theorem 1 (Impossibility of Encryption). *Let Π be any CPA-secure public-key encryption scheme. Then a p-tampering attacker can break the security of Π*

with advantage $\Omega(p)$. Moreover, the attacker only tampers with the random bits of the encryption *(not the key-generation)* without *knowing the message.*

A similar impossibility result holds for private-key encryption schemes in which the tampering adversary can also tamper with the randomness of the key-generation phase.[7]

Theorem 2 (Impossibility of Zero-Knowledge). *Let (P, V) be an efficient prover proof/argument system for a language $L \in \mathsf{NP}$ such that the view of any p-tampering verifier can be simulated by an efficient simulator with indistinguishability gap $o(p)$, then the language L is in* BPP.

Theorem 3 (Impossibility of Commitments). *Let (S, R) be a bit-commitment scheme. Then, either an efficient malicious sender can break the biding with advantage $\Omega(p)$ (without tampering), or an efficient malicious p-tampering receiver can break the hiding with advantage $\Omega(p)$.*

Following [19] we consider two-party functions $f\colon \mathcal{D}_1 \times \mathcal{D}_2 \mapsto \mathcal{R}$ where only one player gets the output. A function f is called trivial in this context, if there is a deterministic single-message (i.e., only one player speaks) protocol for computing f that is information theoretically secure.

Theorem 4 (Impossibility of Secure Computation). *The security of any protocol for computing a two-party non-trivial function can be broken with advantage $\Omega(p)$ through a p-tampering attack.*

Relation to Subliminal Channels. Cryptographic research on "subliminal channels" [36] and the related field of "kleptography" [37] study whether a cryptographic scheme can be "misused" for a purpose other than the original purpose it was designed for (e.g., by putting an undetectable trapdoor in the systems). The existence of subliminal channels between "outside" and "inside" adversaries could be a huge security concern in certain scenarios such as voting schemes [17]. Our Theorem 1 (and its private-key variant) show that *any* efficient encryption scheme always has a subliminal channel between an outsider adversary and an insider virus who is (only) able to tamper with the randomness of the encryption device no matter how the encryption algorithm tries to "detect" a virus who is signaling a bit of information to the adversary.

Tampering with Randomness vs. Imperfect Randomness. Our negative results are closely related to the impossibility result of Dodis et al. [11] on the "impossibility of cryptography with imperfect randomness", where the security of cryptographic primitives are analyzed assuming that the honest parties only have access to randomness coming from an SV source (as opposed to the randomness being perfectly uniform). [11] present several strong impossibility results for secure realizability of cryptography primitives in a setting where players only have

[7] Note that this is necessary, because the one-time pad encryption is deterministic during its encryption phase.

access to such imperfect randomness. The SV sources they consider for their impossibility results, however, may not be efficiently computable.

The key-difference between tamper-resilient security in our setting and security with imperfect randomness is that we restrict to randomness sources that are efficiently sampled through an (online) p-tampering attack; thus achieving tamper-resilient security becomes easier than resilience to imperfect randomness. Note that even if one can *efficiently* sample from the sources employed by [11], that still does not solve our main question, because by sampling fresh randomness for the system the adversary is indeed tampering with *all* of the random seed. As we discussed above, however, in such scenario the adversary can always fix the randomness to zero and so we are essentially down to the deterministic case. Another, perhaps less important difference, is that for primitives with simulation-based security, we allow the simulator to depend on the p-tampering attacker, whereas in [11] the simulator must work for any randomness source; this further makes achieving tamper-resilient security easier than resilience to imperfect randomness.

Positive Results. We complement the above negative results by demonstrating some initial positive results: Assuming the existence of one-way functions, for any $p = n^{-\alpha}$, where $\alpha > 0$ is a constant and n is the security parameter, every implementation of signature schemes, identification protocols, and witness hiding protocols can be made resilient against p-tampering attackers. We also present a relaxed notion of semantic security for encryption schemes that can be achieved under $n^{-\alpha}$-tampering attacks. We show that for these primitives, security holds even if the randomness source "min-entropy loss" of at most $O(\log n)$. We next show how to use PRGs to ensure that a tampering attacker will only be able to decrease the overall (pseudo) min-entropy by $O(\log n)$. The above mentioned primitives already imply the existence of one-way functions [23], thus preventing against $n^{-\alpha}$-tampering attacks can be achieved for these primitives unconditionally. Finally, we present positive results for tamper-resilient key-agreement and secure multi-party computation in the presence of (at least) two honest players. For further details see the full version [4] of the paper.

2.1 Our Techniques

Our main technical contribution is to develop new methods for biasing Boolean, and more generally, bounded-value functions, using a p-tampering attack.

Biasing Bounded-Value Functions Our first (negative) result uses elementary Fourier analysis to prove an efficient version of the Santha-Vazirani theorem: Any balanced (or almost balanced) efficiently computable Boolean function f can be biased by $\Omega(p)$ through an efficient p-tampering attack.

Specifically, let U_n denote the uniform distribution over $\{0,1\}^n$ and let $U_n^{\mathsf{Tam},p}$ denote the distribution obtained after performing a p-tampering attack on U_n using a tampering algorithm Tam; more precisely, let $U_n^{\mathsf{Tam},p} = (X_1, \ldots, X_n)$

where with probability $1-p$, X_i is a uniform random bit, and with probability p, $X_i = \mathsf{Tam}(1^n, X_1, \ldots, X_{i-1})$.

Theorem 5 (Biasing Boolean Functions: Warm-up). *There exists an oracle machine* Tam *with input parameters* n *and* $\varepsilon < 1$ *that runs in time* $\mathrm{poly}(n/\varepsilon)$ *and for every* $n \in N$ *and* $\varepsilon \in (0,1)$*, every Boolean function* $f : \{0,1\}^n \to \{-1,1\}$*, and every* $p < 1$*, for* $\mu = \mathsf{E}[f(U_n)]$ *it holds that*

$$\mathsf{E}[f(U_n^{\mathsf{Tam}^f,p})] \geq \mu + p \cdot (1 - |\mu| - \varepsilon).$$

The tampering algorithm Tam is extremely simple and natural; it just greedily picks the bit that maximizes the bias at every step. More precisely, $\mathsf{Tam}^f(x_1, \ldots, x_{i-1})$ estimates the value of

$$\mathsf{E}_{U_{n-i}}[f(x_1, \ldots, x_{i-1}, b, U_{n-i})]$$

for both of $b = 0$ and $b = 1$ by sampling, and sets x_i to the bit b with larger estimated expectation.

Theorem 5 suffices for our impossibility result for tamper-resilient zero-knowledge. For all our remaining impossibility results, however, we need a more general version that also deals with bounded value functions $f : \{0,1\}^n \to [-1,1]$. Our main technical theorem provides such a result.

Theorem 6 (Main Technical Theorem: Biasing Bounded-Value Functions). *There exists an efficient oracle machine* Tam *such that for every* $n \in N$*, every bounded-value function* $f : \{0,1\}^n \to [-1,1]$*, and every* $p < 1$*,*

$$\mathsf{E}[f(U_n^{\mathsf{Tam}^f,p})] \geq \mathsf{E}[f(U_n)] + \frac{p \cdot \mathrm{Var}[f(U_n)]}{5}.$$

Note that in Theorem 6 the dependence on the variance of f is necessary because f may be the constant function $f(x) = 0$, whereas for the case of balanced Boolean functions this clearly cannot happen. Let us also point out that we have not tried to optimize the constant $1/5$, and indeed it seems that a more careful analysis could be used to bring it down since for small p the constant gets close to 1.

The greedy algorithm does not work in the non-Boolean case anymore. The problem, roughly speaking, is that a greedy strategy will locally try to increase the expectation, but that might lead to choosing a wrong path. As a "counter-example" consider a function f such that: conditioned on $x_1 = 0$ f is a constant function ε, but conditioned on $x_1 = 1$, f is a Boolean function with average $-\varepsilon$. For such f, the greedy algorithm will set $x_1 = 0$ and achieves bias at most ε, while by choosing $x_1 = 1$ more bias could be achieved. To circumvent this problem we use a "mildly greedy" strategy: we take only one sample of $f(\cdot)$ by choosing $x'_i, x'_{i+1}, \ldots, x'_n$ at random ($x_1, \ldots, x_{i-1}$ are already fixed). Then, we keep the sampled x'_i with probability proportional to how much the output of f is close to our "desired value", and flip the value of x'_i otherwise.

More precisely, $\mathsf{Tam}(1^n, x_1, \ldots, x_{i-1})$ proceeds as follows:

– Samples $(x'_i, x'_{i+1}, \ldots, x'_n) \leftarrow U_{n-i+1}$ and compute $y = f(x_1, \ldots, x_{i-1}, x'_i, \ldots, x'_n)$.
– Sample $\mathsf{Tam}(1^n, x_1, \ldots, x_{i-1})$ from a Boolean random variable with average $y \cdot x'_i$ (i.e. output x'_i with probability $\frac{1+y}{2}$, and $-x'_i$ with probability $\frac{1-y}{2}$).

Note that our mildly greedy strategy is even easier to implement than the greedy one: to tamper with each bit, it queries f *only once*.

Impossibility Results for Tamper-Resilient Cryptography We employ the biasing algorithms of Theorems 5 and 6 to obtain our negative results using the following blue-print: We first prove a *computational version* of the "splitting lemma" of [11] (Lemma 7 below which follows from Corollary 3.2 in [11]). Then we will use the same arguments as those of [11] to derive our impossibility results.

Lemma 7 ([11]). *Let f_0 and f_1 be two efficient functions from $\{0,1\}^m$ to $\{0,1\}^{\mathrm{poly}(m)}$ such that $\Pr_{x \leftarrow U_m}[f_0(x) \neq f_1(x)] \geq 1/\mathrm{poly}(n)$. Then there an Santha-Vazirani source of randomness X with parameter $1/2 - 1/\mathrm{poly}(n)$ such that $f_0(X)$ is computationally distinguishable from $f_1(X)$.*

We use our Theorem 6 to prove the following computational version of Lemma 7 which allows one to distinguish the functions f_0, f_1 by tampering with their random input.

Lemma 8 (Computational Splitting Lemma). *Let f_0 and f_1 be two efficient functions from $\{0,1\}^m$ to $\{0,1\}^{\mathrm{poly}(m)}$ and $\Pr_{x \leftarrow U_m}[f_0(x) \neq f_1(x)] \geq \varepsilon > 1/\mathrm{poly}(m)$. Then one can efficiently find a $\mathrm{poly}(m)$-size function f and a $\mathrm{poly}(m)$-size tampering circuit Tam such that*

$$\Pr[f(f_1(U_n^{\mathsf{Tam},p})) = 1] \geq \Pr[f(f_0(U_n^{\mathsf{Tam},p})) = 1] + \Omega(\varepsilon \cdot p).$$

Proof Outline. We derive Lemma 8 from Theorem 6 as follows. We use Theorem 6 to bias the *difference* function $g_f(x) = f(f_1(x)) - f(f_0(x))$ (with domain $\{-1, 0, +1\}$) towards 1 by a tampering circuit Tam. It is easy to see that if f is Boolean, doing this is equivalent to the goal of Lemma 8. We show that if one samples f from a family of pairwise independent Boolean functions, then the resulting function $g_f(\cdot)$ has sufficient variance as needed by Theorem 6.

We use our Lemma 8 similar to the way Lemma 7 is employed in [11] to derive our impossibility results for tamper resilient: encryption schemes, commitments, and two-party secure function evaluation protocols. For all these primitives an adversary uses Lemma 8 to generate a tampering circuit Tam that later on lets him distinguish the corresponding challenges (generated using the tampered randomness) and break the security.

Zero-Knowledge. Zero-knowledge proofs in the setting of [11] require a *universal* simulator that simultaneously handles a large class of imperfect randomness sources. We can also use our Lemma 8 to rule out such tamper-resilient zero-knowledge proofs. In the computational setting, however, it is the malicious

verifier who generates the bad source of randomness, and so we shall allow the simulator to depend on the tampering circuit as well. Thus, the simulator in our setting has more power. This prevents us from applying Lemma 8 directly.

We proceed in using the following high level outline. In the first step, we generalize a result by Goldreich and Oren [20] showing that non-trivial zero-knowledge protocols cannot have deterministic provers. Our generalization to [20] shows that non-trivial zero-knowledge protocols require having prover messages with min-entropy $\omega(\log n)$. This means that the verifier can apply a (seeded) randomness extractor to the transcript and obtain one almost unbiased bit. In a second step, we show how to use (the proof of) Theorem 5 to tamper with the prover's randomness so as to signal bits of the witness to the verifier.

This outline, however, oversimplifies: is it not the case that every non-trivial zero-knowledge protocol requires the prover messages to have min-entropy $\omega(\log n)$; in fact, for some "easy" instances, the prover may not communicate at all. Rather, we demonstrate that an "instance-based" version of the min-entropy extension of the Goldreich and Oren [20] theorem holds, and using it we can prove that *either* the prover's messages have high min-entropy (and thus the witness can be leaked to the verifier), or the instance can be decided "trivially". It follows that in either case, we can correctly decide the instance and thus the language must be trivial.

3 Biasing Functions via Online Tampering

In this section we study how much the output of a bounded function can be biased through a tampering attack, and we will formally prove Theorem 5 and Theorem 6. For the full proofs of the applications of these two theorems (sketched in previous section) we refer the reader to the full version of the paper [4].

First we formally define an online tampering process and a tampering source of randomness (as a result of an online tampering attack performed on a uniform source of randomness).

Definition 9. *A distribution $X = (X_1, \ldots, X_n)$ over $\{-1, 1\}^n$ is an (efficient) p-tampering source if there exists an (efficient) tampering algorithm* Tam *such that X can be generated in an online fashion as follows: For $i = 1, \ldots, n$,*

$$X_i = \begin{cases} \mathsf{Tam}(1^n, X_1, \ldots, X_{i-1}) & \text{with probability } p, \\ U_1^i & \text{with probability } 1 - p, \end{cases}$$

where U_1^i denotes a uniformly random bit over $\{-1, 1\}$. In other words, with probability p, Tam *gets to tamper the next bit with the knowledge of the previous bits (after the tampering)*[8]*. The tampering algorithm* Tam *might also receive an*

[8] In a stronger variant of tampering attacks, the attacker might be completely stateful and memorize the original values of the previous bits before and after tampering and also the places where the tampering took place, and use this extra information in its future tampering. Using the weaker stateless attacker of Definition 9 only makes our negative results stronger. Our *positive* results hold even against stateful attackers.

auxiliary input and use it in its tampering strategy.[9] *We use* $U_n^{\mathsf{Tam},p}$ *to denote the p-tampered source obtained by the above tampering process with tampering algorithm* Tam.

Note that in the definition above, the tampering algorithm Tam might be completely oblivious to the parameter p. By referring to Tam as a *p-tampering* algorithm, we emphasize on the fact that Tam's algorithm might depend on p.

Remark 10. Every p-tampering source is also a Santha-Vazirani source [35] with parameter $\delta = (1-p)/2$. In fact, it is not hard to see that without the efficiency consideration, the two notions are equivalent.

3.1 Preliminaries: Calculating the Effect of a Single Variable

Recall that the Fourier coefficients of any function $f : \{-1,1\}^n \to [-1,1]$ are indexed by the subsets S of $[n]$ and are defined as $\hat{f}(S) := \mathsf{E}_x[f(x)\chi_S(x)]$, where $\chi_S(x) := \prod_{i\in S} x_i$. Note that the Fourier coefficient of the empty set $\hat{f}(\emptyset)$ is simply the expectation $\mathsf{E}[f(U_n)]$.

For every prefix $x_{\le i} = (x_1, \ldots, x_i)$, let $f_{x_{\le i}} : \{-1,1\}^{n-i} \to [-1,1]$ be the restriction of f on $x_{\le i}$, i.e., $f_{x_{\le i}}(x_{i+1}, \ldots, x_n) := f(x_1, \ldots, x_n)$. We note that the variables of $f_{x_{\le i}}$ are named $(x_{i+1}, \ldots, x_n)$ and thus the Fourier coefficients of $f_{x_{\le i}}$ are $\hat{f}_{x_{\le i}}(S)$'s with $S \subseteq \{i+1, \ldots, n\}$. The following basic identity can be proved by straightforward calculation.

$$\hat{f}_{x_1}(\emptyset) = \hat{f}(\emptyset) + \hat{f}(\{1\}) \cdot x_1. \tag{1}$$

Recall that $\hat{f}(\emptyset)$ and $\hat{f}_{x_1}(\emptyset)$ are simply expectations. One interpretation of the above identity is that $\pm\hat{f}(\{1\})$ is the change of expectation when we set $x_1 = \pm 1$. This is thus useful for analyzing the bias introduced as the result of a tampering attack.

Using the above identity with a simple induction, we can express $f(x)$ as a sum of Fourier coefficients of restrictions of f. Namely, $f(x)$ equals to the expectation $\hat{f}(\emptyset)$ plus the changes in expectation when we set x_i bit by bit.

Lemma 11. $f(x) = \hat{f}(\emptyset) + \sum_{i=1}^{n} \hat{f}_{x_{\le i-1}}(\{i\}) \cdot x_i$ *for every* $x \in \{-1,1\}^n$.

Proof. By expanding $\hat{f}_{x_{\le j}}(\emptyset) = \hat{f}_{x_{\le j-1}}(\emptyset) + \hat{f}_{x_{\le j-1}}(\{j\}) \cdot x_j$, (implied by Equation (1)) and a simple induction on j it follows that:

$$f(x) = \hat{f}_{x_{\le j}}(\emptyset) + \sum_{i=j+1}^{n} \hat{f}_{x_{\le i-1}}(\{i\}) \cdot x_i,$$

which proves the lemma.

[9] The auxiliary input could, e.g., be the information that the tampering algorithm receives about the secret state of the tampered party; this information might not be available at the time the tampering circuit is generated by the adversary.

As a corollary, the above lemma implies that the sum of Fourier coefficients (of restrictions of f) in absolute value is at least $|f(x)| - |\hat{f}(\emptyset)|$.

Corollary 12. *For every $x \in \{-1,1\}^n$, it holds that $\sum_{i=1}^{n} \left|\hat{f}_{x_{\leq i-1}}(\{i\})\right| \geq |f(x)| - |\hat{f}(\emptyset)|$.*

Proof. By triangle inequality we have

$$\sum_{i=1}^{n} \left|\hat{f}_{x_{\leq i-1}}(\{i\})\right| = \sum_{i=1}^{n} \left|\hat{f}_{x_{\leq i-1}}(\{i\}) \cdot x_i\right| \geq \left|\sum_{i=1}^{n} \hat{f}_{x_{\leq i-1}}(\{i\}) \cdot x_i\right|$$
$$= |f(x) - \hat{f}(\emptyset)| \geq |f(x)| - |\hat{f}(\emptyset)|$$

where the second equality uses Lemma 11.

3.2 The Boolean Case

A seminal result by Santha and Vazirani [35] shows that for every balanced Boolean function f (e.g., a candidate "extractor"), there exists a p-tampering source X that biases the output of f by at least p. We now present a strengthening of this result that additionally shows that if the function f is efficiently computable, then the source X could be an efficient p-tampering one (and only needs to use f as a black box). In the language of extractors, our result thus proves a strong impossibility result for deterministic randomness extraction from "efficient" Santha-Vazirani sources. Our proof of the generalized result is quite different (and in our eyes simpler) than classic proofs of the Santha-Vazirani theorem and may be of independent interest.

In fact, we present two different proofs. The first one achieves optimal bias p for balanced f, whereas the second uses an extremely simple "*lazy greedy*" tampering algorithm that makes only a *single* query to f and achieves bias $p/3$ for balanced f.

Theorem 13 (Theorem 5 restated). *There exists an oracle machine* Tam *with input parameters n and $\varepsilon < 1$ that runs in time* $\mathrm{poly}(n/\varepsilon)$ *and for every $n \in N$ and $\varepsilon \in (0,1)$, every Boolean function $f : \{0,1\}^n \to \{-1,1\}$, and every $p < 1$, for $\mu = \mathsf{E}[f(U_n)]$ it holds that*

$$\mathsf{E}[f(U_n^{\mathsf{Tam}^f,p})] \geq \mu + p \cdot (1 - |\mu| - \varepsilon).$$

Proof (Proof of Theorem 5). Let us first present a proof with an inefficient tampering algorithm achieving bias $p \cdot (1 - |\mu|)$; next, we show how to make it efficient while not loosing much in bias. On input $x_{\leq i-1} = (x_1, \ldots, x_{i-1})$, Tam sets $x_i = \mathrm{sgn}(\hat{f}_{x_{\leq i-1}}(\{i\}))$. By Equation (1), $\hat{f}_{x_{\leq i-1}}(\{i\})$ corresponds to the change in expectation of $f_{x_{\leq i-1}}$ when setting the value of x_i. This amounts to

greedily choosing the x_i that increases the expectation. Let $X = U_n^{\mathsf{Tam},p}$. By applying Lemma 11 and the linearity of expectations, we have

$$\begin{aligned}\mathsf{E}[f(X)] &= \hat{f}(\emptyset) + \sum_{i=1}^{n} \mathsf{E}_X\left[\hat{f}_{X_{\leq i-1}}(\{i\}) \cdot X_i\right] \\ &= \hat{f}(\emptyset) + \sum_{i=1}^{n} \mathsf{E}_{X_{\leq i-1}}\left[\hat{f}_{X_{\leq i-1}}(\{i\}) \cdot \mathsf{E}[X_i | X_{\leq i-1}]\right].\end{aligned}$$

Since Tam tampers with the i'th bit with independent probability p, therefore

$$\mathsf{E}[X_i | X_{\leq i-1}] = p \cdot \operatorname{sgn}(\hat{f}_{X_{\leq i-1}}(\{i\})$$

and so it holds that

$$\begin{aligned}\mathsf{E}[f(X)] = \hat{f}(\emptyset) + p \cdot \sum_{i=1}^{n} \mathsf{E}_X\left[\left|\hat{f}_{X_{\leq i-1}}(\{i\})\right|\right] &= \hat{f}(\emptyset) + p \cdot \mathsf{E}_X\left[\sum_{i=1}^{n}\left|\hat{f}_{X_{\leq i-1}}(\{i\})\right|\right] \\ &\geq \hat{f}(\emptyset) + p \cdot (1 - \hat{f}(\emptyset))\end{aligned}$$

where the last inequality follows by Corollary 12.

Note that the above tampering algorithm Tam in general is not efficient since computing $\hat{f}_{x_{\leq i-1}}(\{i\})$ exactly may be hard. However, we show that Tam may approximate $\hat{f}_{x_{\leq i-1}}(\{i\})$ using $M = \Theta(\frac{n^2}{\varepsilon^2} \cdot \log \frac{n}{\varepsilon})$ samples, and set x_i according to the sign of the approximation of $\hat{f}_{x_{\leq i-1}}(\{i\})$, while still inducing essentially the same bias. This clearly can be done efficiently given oracle access to f. As before, let $X = U_n^{\mathsf{Tam}^f,p}$ denote the corresponding p-tampering source. To lower bound $\mathsf{E}[f(X)]$, we note that the only difference from the previous case is that $\mathsf{Tam}(1^n, x_{\leq i-1})$ is no longer always outputting $\operatorname{sgn}(\hat{f}_{x_{\leq i-1}}(\{i\}))$. Nevertheless, we claim that for every $x_{<i}$ it holds that

$$\hat{f}_{x_{\leq i-1}}(\{i\}) \cdot \mathsf{E}[X_i | X_{\leq i-1} = x_{\leq i-1}] \geq p \cdot \left(|\hat{f}_{x_{\leq i-1}}(\{i\})| - \varepsilon/n\right)$$

since either (i) $|\hat{f}_{x_{\leq i-1}}(\{i\})| \geq \varepsilon/2n$ in which case (by a Chernoff bound) Tam outputs $\operatorname{sgn}(\hat{f}_{x_{\leq i-1}}(\{i\}))$ with probability at least $1-\varepsilon/2n$, or (ii) $|\hat{f}_{x_{\leq i-1}}(\{i\})| < \varepsilon/2n$ in which case the inequality holds no matter what Tam outputs since $|\mathsf{E}[X_i | X_{\leq i-1} = x_{\leq i-1}]| \leq p$. A lower bound on $\mathsf{E}[f(X)]$ then follows by the same analysis as before:

$$\mathsf{E}[f(X)] \geq \hat{f}(\emptyset) + p \cdot \sum_{i=1}^{n} \mathsf{E}_X\left[\left|\hat{f}_{X_{\leq i-1}}(\{i\})\right| - \varepsilon/n\right] \geq \mu + p \cdot (1 - |\hat{f}(\emptyset)| - \varepsilon).$$

Before presenting the second proof, we state the following lemma, which follows similarly to lemma 11, but instead it relies on a squared version of Equation (1). See the full version for a proof.

Lemma 14. *For every $x \in \{-1,1\}^n$,*

$$f(x)^2 = \hat{f}(\emptyset)^2 + \sum_{i=1}^{n} \left(\hat{f}_{x_{\leq i-1}}(\{i\})^2 + 2\hat{f}_{x_{\leq i-1}}(\emptyset) \cdot \hat{f}_{x_{\leq i-1}}(\{i\}) \cdot x_i \right).$$

We continue to present the second proof using a "*lazy greedy*" tampering algorithm that makes a *single* query to f and achieves bias $p/3$ for balanced f.

Theorem 15. *There exists an oracle machine* Tam *that makes a* single *query to its oracle such that for every $n \in N$, every Boolean function $f : \{0,1\}^n \to \{-1,1\}$, and every $p < 1$, for $\mu = \mathsf{E}[f(U_n)]$ it holds that*

$$\mathsf{E}[f(U_n^{\mathsf{Tam}^f,p})] \geq \mu + p \cdot (1-\mu^2)/3.$$

Proof. We consider a *lazy greedy tampering algorithm* LTam that on input $x_{\leq i-1} = (x_1, \ldots, x_{i-1})$, samples uniformly random $(x'_i, \ldots, x'_n) \leftarrow U_{n-i+1}$, queries $y = f(x_1, x_{i-1}, x'_i, \ldots, x'_n)$, and outputs $X_i = x'_i$ if $y = 1$ and $X_i = -x'_i$ if $y = -1$. Namely, LTam samples a random completion of $x_{\leq i-1}$ and output the sampled bit x'_i iff the sample evaluates to 1.

Interestingly, this simple lazy greedy LTam implicitly "plays the first Fourier coefficient" in expectation in the sense that $\mathsf{E}[\mathsf{Tam}(x_{\leq i-1})] = \hat{f}_{x_{\leq i-1}}(\{i\})$.

Claim. For every $x_{\leq i-1}$, $\mathsf{E}[\mathsf{LTam}(x_{\leq i-1})] = \hat{f}_{x_{\leq i-1}}(\{i\})$.

Proof. Let $X_i = \mathsf{LTam}(x_{\leq i-1})$. We have $\mathsf{E}[X_i] = \mathsf{E}_{x \geq i \leftarrow U_{n-i+1}}[f(x) \cdot x_i] = \hat{f}_{x_{\leq i-1}}(\{i\})$.

To analyze LTam, we derive two equalities analogous to that in the proof of Theorem 5, and the theorem follows by combining the two equalities. First, since LTam gets to tamper with bit i with independent probability p and $\mathsf{E}[\mathsf{LTam}(x_{\leq i-1})] = \hat{f}_{x_{\leq i-1}}(\{i\})$, by Lemma 3.3, we have that $\mathsf{E}[X_i | X_{\leq i-1}] = p \cdot \hat{f}_{X_{\leq i-1}}(\{i\})$. Thus,

$$\mathsf{E}[f(X)] = \hat{f}(\emptyset) + p \cdot \sum_{i=1}^{n} \mathsf{E}_X \left[\hat{f}_{X_{\leq i-1}}(\{i\})^2 \right]. \tag{2}$$

Similarly, by applying Lemma 14 and the linearity of expectations, we have

$$\begin{aligned} \mathsf{E}[f(X)^2] = \hat{f}(\emptyset)^2 &+ \sum_{i=1}^{n} \left(\mathsf{E}_X[\hat{f}_{X_{\leq i-1}}(\{i\})^2] \right) \\ &+ \sum_{i=1}^{n} \left(2\mathsf{E}_{X_{\leq i-1}}[\hat{f}_{X_{\leq i-1}}(\emptyset) \cdot \hat{f}_{X_{\leq i-1}}(\{i\}) \cdot \mathsf{E}[X_i | X_{\leq i-1}]] \right). \end{aligned}$$

Simplifying using the fact that f is Boolean, the trivial bound $|\hat{f}_{X_{\leq i-1}}(\emptyset)| \leq 1$, and $\mathsf{E}[X_i | X_{\leq i-1}] = p \cdot \hat{f}_{X_{\leq i-1}}(\{i\})$ gives

$$1 \leq \hat{f}(\emptyset)^2 + (1+2p) \cdot \sum_{i=1}^{n} \mathsf{E}_X[\hat{f}_{X_{\leq i-1}}(\{i\})^2]. \tag{3}$$

Plugging Equation (3) in Equation (2) yields

$$\mathsf{E}[f(X)] \geq \hat{f}(\emptyset) + \frac{p}{1+2p}\left(1 - \hat{f}(\emptyset)^2\right) \geq \mu + p \cdot \left(1 - \mu^2\right)/3,$$

which completes the proof of Theorem 15.

3.3 Tampering with Bounded-Value Functions—The General Case

We further consider the more general case of tampering non-Boolean, bounded-value functions. We present an efficient tampering algorithm that biases the expectation of the function by an amount linear in the variance of the function.

Theorem 16 (Theorem 6 restated). *There exists an efficient oracle machine* Tam *such that for every* $n \in N$*, every bounded-value function* $f : \{0,1\}^n \to [-1,1]$*, and every* $p < 1$*,*

$$\mathsf{E}[f(U_n^{\mathsf{Tam}^f,p})] \geq \mathsf{E}[f(U_n)] + \frac{p \cdot \mathrm{Var}[f(U_n)]}{5}.$$

We prove Theorem 5 using *lazy greedy* tampering algorithm again. As before, we let LTam take a single sample, and make decision based on the outcome of the sample, but since f is not Boolean, we make randomized decision based on the function value on the sample. Specifically, on input $x_{\leq i-1} = (x_1, \ldots, x_{i-1})$:

- LTam samples uniformly random $(x'_i, \ldots, x'_n) \leftarrow U_{n-i+1}$, and computes $y = f(x_1, x_{i-1}, x'_i, \ldots, x'_n)$.
- LTam outputs $X_i = x'_i$ with probability $(1+y)/2$, and $X_i = -x'_i$ with probability $(1-y)/2$. Note that X_i has expectation $\mathsf{E}[X_i] = y \cdot x'_i$.

The following claim says that LTam "implicitly plays the first Fourier coefficient" in expectation.

Claim. For every $x_{\leq i-1} \in \{-1,1\}^{i-1}$, $\mathsf{E}[\mathsf{LTam}(x_{\leq i-1})] = \hat{f}_{x_{\leq i-1}}(\{i\})$.

Proof. Let $X_i = \mathsf{LTam}(x_{\leq i-1})$. We have:

$$\mathsf{E}[X_i] = \mathsf{E}_{x \geq i \leftarrow U_{n-i+1}}[f(x) \cdot x_i] = \hat{f}_{x_{\leq i-1}}(\{i\}).$$

Let $X = U_n^{\mathsf{LTam}^f,p}$. Also, let the mean $\mathsf{E}[f(U_n)] = \mu$, the second moment $\mathsf{E}[f(U_n)^2] = \nu$, and the variance $\mathrm{Var}[f] = \sigma^2$ be denoted so. The analyze of the lazy greedy algorithm LTam for the non-Boolean case is significantly more involved. We first follow an analogous step in the analysis of Boolean case to derive an inequality between $\mathsf{E}[f(X)]$ and $\mathsf{E}[f(X)^2]$, then rely on a potential function analysis to derive a second inequality relation between $\mathsf{E}[f(X)]$ and $\mathsf{E}[f(X)^2]$, and then derive a lower bound on $\mathsf{E}[f(X)]$ by combining the two. The two inequalities are stated in the following lemmas.

Lemma 17. $\mathsf{E}[f(X)] - \mu \geq \frac{p}{1+2p} \cdot \left(\mathsf{E}[f(X)^2] - \nu + \sigma^2\right).$

Lemma 18. $\mathsf{E}[f(X)] + \frac{\mathsf{E}[f(X)^2]}{2} + \frac{\mathsf{E}[f(X)^2]^2}{4} \geq \mu + \frac{\nu}{2} + \frac{\nu^2}{4}$.

We first use the above two lemmas to show that $\mathsf{E}[f(X)] - \mu \geq (p\sigma^2)/5$, which implies that $\mathsf{E}[f(X)] \geq \mu + p \cdot \mathrm{Var}[f]/5$, as desired. If $\mathsf{E}[f(X)^2] \geq \nu$, then Lemma 17 implies

$$\mathsf{E}[f(X)] - \mu \geq \frac{p}{1+2p} \cdot \sigma^2 \geq \frac{1}{5} \cdot p\sigma^2.$$

For the case that $\mathsf{E}[f(X)^2] \leq \nu$, let $\alpha \triangleq \nu - \mathsf{E}[f(X)^2] \geq 0$. Lemma 18 implies

$$\mathsf{E}[f(X)] - \mu \geq \frac{1}{2}(\nu - \mathsf{E}[f(X)^2]) + \frac{1}{4}(\nu^2 - \mathsf{E}[f(X)^2]^2) \geq \frac{\alpha}{2}$$

which together with Lemma 17 implies that

$$[f(X)] - \mu \geq \max\left\{ \frac{p}{1+2p} \cdot \left(\sigma^2 - \alpha\right), \frac{\alpha}{2} \right\} \geq \frac{p}{1+4p} \geq \frac{p\sigma^2}{5}.$$

Now we prove Lemmas 17 and 18. The proof of Lemma 17 is a generalization of the analysis for biasing Boolean functions.

Proof (Proof of Lemma 17). By applying Lemma 11 and the linearity of expectations, we have

$$\begin{aligned}\mathsf{E}[f(X)] &= \hat{f}(\emptyset) + \sum_{i=1}^{n} \mathsf{E}_X\left[\hat{f}_{X_{\leq i-1}}(\{i\}) \cdot X_i\right] \\ &= \hat{f}(\emptyset) + \sum_{i=1}^{n} \mathsf{E}_{X_{\leq i-1}}\left[\hat{f}_{X_{\leq i-1}}(\{i\}) \cdot \mathsf{E}[X_i | X_{\leq i-1}]\right].\end{aligned}$$

Since LTam gets to tamper with bit i with independent probability p, by Lemma 3.3 we have that $\mathsf{E}[X_i | X_{\leq i-1}] = p \cdot \hat{f}_{X_{\leq i-1}}(\{i\})$. Thus,

$$\mathsf{E}[f(X)] = \hat{f}(\emptyset) + p \cdot \sum_{i=1}^{n} \mathsf{E}_X\left[\hat{f}_{X_{\leq i-1}}(\{i\})^2\right]. \tag{4}$$

Similarly, by applying Lemma 14 and the linearity of expectations, we have

$$\begin{aligned}\mathsf{E}[f(X)^2] = \hat{f}(\emptyset)^2 &+ \sum_{i=1}^{n} \left(\mathsf{E}_X[\hat{f}_{X_{\leq i-1}}(\{i\})^2]\right) \\ &+ \sum_{i=1}^{n} \left(2\mathsf{E}_{X_{\leq i-1}}[\hat{f}_{X_{\leq i-1}}(\emptyset) \cdot \hat{f}_{X_{\leq i-1}}(\{i\}) \cdot \mathsf{E}[X_i | X_{\leq i-1}]]\right).\end{aligned}$$

Simplifying using the trivial bound $|\hat{f}_{X_{\leq i-1}}(\emptyset)| \leq 1$ and $\mathsf{E}[X_i | X_{\leq i-1}] = p \cdot \hat{f}_{X_{\leq i-1}}(\{i\})$ gives

$$\mathsf{E}[f(X)^2] \leq \hat{f}(\emptyset)^2 + (1+2p) \cdot \sum_{i=1}^{n} \mathsf{E}_X[\hat{f}_{X_{\leq i-1}}(\{i\})^2]. \tag{5}$$

The lemma follows by combining Equations (4) and (5):

$$\mathsf{E}[f(X)] \geq \hat{f}(\emptyset) + \frac{p}{1+2p}\left(\mathsf{E}[f(X)^2] - \hat{f}(\emptyset)^2\right) = \mu + \frac{p}{1+2p}\left(\mathsf{E}[f(X)^2] - \nu + \sigma^2\right)$$

where the last equality uses the fact that $\hat{f}(\emptyset)^2 = \mu^2 = \nu - \sigma^2$.

The proof of Lemma 18 is less trivial. Our key observation is the following useful property of the lazy greedy tampering algorithm LTam: consider the function f together with an arbitrary function $g : \{-1,1\}^n \to [-1,1]$ (ultimately, we shall set $g(x) = f(x)^2$, but in the discussion that follows, g can be completely unrelated to f). While intuitively we expect the expectation of f to be increasing after tampering, it is clearly possible that the tampering causes the expectation of g to decrease. Nevertheless, we show that for a properly defined potential function combining the expectations of f and g, the potential is guaranteed to be non-decreasing after tampering. Namely, we prove the following lemma whose proof can be found in the full version of the paper.

Lemma 19. *Let $g : \{-1,1\}^n \to [-1,1]$ be an arbitrary function. For every prefix $x_{\leq i} \in \{-1,1\}^i$, define a potential*

$$\Phi(x_{\leq i}) := \hat{f}_{x_{\leq i}}(\emptyset) + \frac{\hat{g}_{x_{\leq i}}(\emptyset)}{2} + \frac{\hat{g}_{x_{\leq i}}(\emptyset)^2}{4},$$

and let $\Phi := \Phi(x_{\leq 0})$. Then it holds that $\mathsf{E}[\Phi(X)] \geq \Phi$.

Lemma 18 now follows easily.

Proof (Proof of Lemma 18). By applying Lemma 19 with $g = f^2$ and noting that $\hat{g}(\emptyset) = \nu$, we have

$$\mathsf{E}[f(X)] + \frac{\mathsf{E}[f(X)^2]}{2} + \frac{\mathsf{E}[f(X)^2]^2}{4} \geq \mu + \frac{\nu}{2} + \frac{\nu^2}{4}.$$

References

1. Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In Omer Reingold, editor, *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings*, volume 5444 of *Lecture Notes in Computer Science*, pages 474–495. Springer, 2009.
2. Aleph One. Smashing the stack for fun and profit. *Phrack Magazine*, 7(49):File 14, 1996.
3. Ross Anderson and Markus Kuhn. Tamper resistance – a cautionary note. In *Proceedings of the Second USENIX Workshop on Electronic Commerce*, pages 1–11, November 1996.
4. Per Austrin, Kai-Min Chung, Mohammad Mahmoody, Rafael Pass, and Karn Seth. On the impossibility of cryptography with tamperable randomness. Cryptology ePrint Archive, Report 2013/194, 2013. http://eprint.iacr.org/.

5. Biham and Shamir. Differential fault analysis of secret key cryptosystems. In *CRYPTO: Proceedings of Crypto*, 1997.
6. Boneh, DeMillo, and Lipton. On the importance of checking cryptographic protocols for faults. In *EUROCRYPT: Advances in Cryptology: Proceedings of EUROCRYPT*, 1997.
7. Zvika Brakerski, Yael Tauman Kalai, Jonathan Katz, and Vinod Vaikuntanathan. Overcoming the hole in the bucket: Public-key cryptography resilient to continual memory leakage. In *FOCS*, pages 501–510. IEEE Computer Society, 2010.
8. Ran Canetti, Oded Goldreich, Shafi Goldwasser, and Silvio Micali. Resettable zero-knowledge (extended abstract). In *STOC*, pages 235–244, 2000.
9. Seung Geol Choi, Aggelos Kiayias, and Tal Malkin. BiTR: Built-in tamper resilience. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, volume 7073 of *Lecture Notes in Computer Science*, pages 740–758. Springer, 2011.
10. Dana Dachman-Soled and Yael Tauman Kalai. Securing circuits against constant-rate tampering. *IACR Cryptology ePrint Archive*, 2012:366, 2012. informal publication.
11. Dodis, Ong, Prabhakaran, and Sahai. On the (im)possibility of cryptography with imperfect randomness. In *FOCS: IEEE Symposium on Foundations of Computer Science (FOCS)*, 2004.
12. Yevgeniy Dodis, Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In Daniele Micciancio, editor, *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings*, volume 5978 of *Lecture Notes in Computer Science*, pages 361–381. Springer, 2010.
13. Yevgeniy Dodis, Kristiyan Haralambiev, Adriana López-Alt, and Daniel Wichs. Cryptography against continuous memory attacks. In *FOCS*, pages 511–520. IEEE Computer Society, 2010.
14. Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *FOCS*, pages 293–302. IEEE Computer Society, 2008.
15. Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs. Non-malleable codes. In Andrew Chi-Chih Yao, editor, *ICS*, pages 434–452. Tsinghua University Press, 2010.
16. Sebastian Faust, Krzysztof Pietrzak, and Daniele Venturi. Tamper-proof circuits: How to trade leakage for tamper-resilience. In *ICALP (1)*, pages 391–402, 2011.
17. Ariel J. Feldman and Josh Benaloh. On subliminal channels in encrypt-on-cast voting systems. In *Proceedings of the 2009 conference on Electronic voting technology/workshop on trustworthy elections*, EVT/WOTE'09, pages 12–12, Berkeley, CA, USA, 2009. USENIX Association.
18. Niklas Frykholm. Countermeasures against buffer overflow attacks. Technical report, RSA Data Security, Inc., pub-RSA:adr, November 2000.
19. Rosario Gennaro, Anna Lysyanskaya, Tal Malkin, Silvio Micali, and Tal Rabin. Algorithmic tamper-proof (atp) security: Theoretical foundations for security against hardware tampering. In Moni Naor, editor, *TCC*, volume 2951 of *Lecture Notes in Computer Science*, pages 258–277. Springer, 2004.
20. Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32, 1994.
21. Shafi Goldwasser and Guy Rothblum. How to compute in the presence of leakage. 2012.

22. Nadia Heninger, Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. Mining your Ps and Qs: Detection of widespread weak keys in network devices. In *Proceedings of the 21st USENIX Security Symposium*, August 2012.
23. Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 230–235, 1989.
24. Yuval Ishai, Manoj Prabhakaran, Amit Sahai, and David Wagner. Private circuits II: Keeping secrets in tamperable circuits. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 308–327. Springer, 2006.
25. Yael Kalai, Allison Lewko, and Anup Rao. Formulas resilient to short-circuit errors. 2012.
26. Yael Tauman Kalai, Bhavana Kanukurthi, and Amit Sahai. Cryptography with tamperable and leaky memory. In *CRYPTO*, pages 373–390, 2011.
27. Arjen K. Lenstra, James P. Hughes, Maxime Augier, Joppe W. Bos, Thorsten Kleinjung, and Christophe Wachter. Public keys. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO*, volume 7417 of *Lecture Notes in Computer Science*, pages 626–642. Springer, 2012.
28. Arjen K. Lenstra, James P. Hughes, Maxime Augier, Joppe W. Bos, Thorsten Kleinjung, and Christophe Wachter. Ron was wrong, whit is right. Cryptology ePrint Archive, Report 2012/064, 2012. http://eprint.iacr.org/.
29. Feng-Hao Liu and Anna Lysyanskaya. Algorithmic tamper-proof security under probing attacks. In *SCN*, pages 106–120, 2010.
30. Feng-Hao Liu and Anna Lysyanskaya. Tamper and leakage resilience in the split-state model. In *Crypto*, 2012.
31. Micali and Reyzin. Physically observable cryptography (extended abstract). In *Theory of Cryptography Conference (TCC), LNCS*, volume 1, 2004.
32. Jonathan D. Pincus and Brandon Baker. Beyond stack smashing: Recent advances in exploiting buffer overruns. *IEEE Security & Privacy*, 2(4):20–27, 2004.
33. Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, Feb 1978.
34. Guy N. Rothblum. How to compute under $\exists\rfloor^0$ leakage without secure hardware. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO*, volume 7417 of *Lecture Notes in Computer Science*, pages 552–569. Springer, 2012.
35. Miklos Santha and Umesh V. Vazirani. Generating quasi-random sequences from semi-random sources. *J. Comput. Syst. Sci.*, 33(1):75–87, 1986.
36. Gustavus J. Simmons. Subliminal channels; past and present. *ETT*, 5(4):15, July 1994.
37. Young and Yung. The dark side of 'black-box' cryptography, or: Should we trust capstone? In *CRYPTO: Proceedings of Crypto*, 1996.