Practical Leakage-Resilient Symmetric Cryptography

Sebastian Faust[1⋆], Krzysztof Pietrzak[2⋆⋆], and Joachim Schipper[2]

[1] Århus University
[2] IST Austria

Abstract. Leakage resilient cryptography attempts to incorporate side-channel leakage into the black-box security model and designs cryptographic schemes that are provably secure within it. Informally, a scheme is *leakage-resilient* if it remains secure even if an adversary learns a bounded amount of arbitrary information about the schemes internal state. Unfortunately, most leakage resilient schemes are unnecessarily complicated in order to achieve strong provable security guarantees. As advocated by Yu et al. [CCS'10], this mostly is an artefact of the security proof and in practice much simpler construction may already suffice to protect against *realistic* side-channel attacks. In this paper, we show that indeed for simpler constructions leakage-resilience can be obtained when we aim for relaxed security notions where the leakage-functions and/or the inputs to the primitive are chosen *non-adaptively*. For example, we show that a three round Feistel network instantiated with a leakage resilient PRF yields a leakage resilient PRP if the inputs are chosen non-adaptively (This complements the result of Dodis and Pietrzak [CRYPTO'10] who show that if a adaptive queries are allowed, a super-logarithmic number of rounds is necessary.) We also show that a minor variation of the classical GGM construction gives a leakage resilient PRF if both, the leakage-function and the inputs, are chosen non-adaptively.

1 Introduction

Traditional cryptographic security notions only consider adversaries who get black-box access to the primitive at hand. That is, an adversary can only observe the input/output behavior of the cryptosystem, but gets no other information about its inner workings. Unfortunately, such black-box security notions are often insufficient to guarantee real-world security of cryptosystems. This is due to information inadvertently emitting from the physical implementation of the

⋆ Sebastian Faust acknowledges support from the Danish National Research Foundation and The National Science Foundation of China (under the grant 61061130540) for the Sino-Danish Center for the Theory of Interactive Computation, within part of this work was performed; and from the CFEM research center, supported by the Danish Strategic Research Council.

⋆⋆ The 2nd and 3rd Author are Supported by the European Research Council/ERC Starting Grant 259668-PSPC.

cryptosystem, which can be exploited by *side-channel attacks*. In the last years a large body of theoretical work attempts to incorporate these side-channel attacks into the security model and to design new cryptographic schemes that provably protect against them. Despite important progress in this area, only very few works in the theory community consider how to protect symmetric primitives against leakage attacks. That is somewhat surprising as symmetric primitives such as pseudorandom number generators and block ciphers are the "working horses" of cryptography and are by far the most frequent target of side-channel attacks. Moreover, as frequently pointed out [22,23], many of the recent theoretical constructions are rather involved and use techniques which only seem to be required to enable the security proof, and do not necessarily contribute to the real-world security of the system. In this work, we show that *simpler* and *more natural* constructions of important *symmetric* primitives such as pseudorandom functions (PRFs) and pseudorandom permutations (PRPs) are provable leakage resilience if we aim for weaker security notions.

1.1 Modeling Leakage Resilience and Weaker Security Notions

As most previous works on leakage-resilient symmetric primitives [4,20,1,23], we follow Dziembowski and Pietrzak [4] who structure the computation into *time steps* and require that the leakage given to the adversary is some bounded amount of *arbitrary* polynomial-time computable information about the data/state that is used during this step. The latter restriction that the leakage function is only applied to the state touched in an invocation was suggested by [19] under the term "only computation leaks information". As the number of invocations of a scheme is usually unbounded, also the amount of leakage can become arbitrarily large.

On granular leakage resilience (gLR). Typically, a time step is one invocation of the scheme that leaks independently from the computation in the previous and next time step. This could for example be the computation of a signature [5], or the generation of a block of pseudorandom bits for stream-ciphers [4,20]. In this work, we will follow [1,22] and consider a more fine grained notion where the construction of some leakage resilient (LR) scheme CS requires several invocations of an underlying cryptographic primitive P,[3] and we require that each invocation of P leaks independently. We call this notion *granular leakage-resilience*, or gLR for short. We notice that in the literature on leakage resilience even more fine grained models have been considered [9,3]

As side-channel leakage is often a global phenomenon (e.g., in power analysis attacks the adversary measures the global power consumption of the device), the question arises whether such a locality restriction still suffices to model relevant leakages in practice. For certain important leakage classes, we can answer this question affirmatively. For instance, the prominent Hamming weight leakage

[3] Concretely, we will consider the cases where CS is a LR-PRF and P a wPRF, and the case where CS is a LR-PRP and P a LR-PRF.

function can be computed independently from the Hamming weight of the local states. A similar observation works for any *affine* leakage function.

Formally, we model granularity as follows. Let τ_i be the state that is used by the computation (keys, inputs, randomness) in time step i. Before each such step, the adversary can adaptively choose a leakage function f_i, and after this step has been processed, she learns $f_i(\tau_i)$.

On non-adaptive leakage resilience (naLR). Besides granularity, another natural relaxation of leakage resilience, which has been considered in e.g. [23,1,22], is to require that the adversary has to fix the leakage functions in advance before seeing any leakage or outputs. This notion is called non-adaptive leakage resilience, or naLR for short. In the leakage setting, a fully adaptive choice of the leakage function may be an overly powerful model to capture side-channel attacks, as in practice the leakage function is often fixed in advance by the device and the measurement equipment (for more discussion on this cf. [23,1,22]). Also, as discussed in [23], for stateless cryptographic schemes that do not allow to evolve the secret state, such as PRFs or PRPs, one simply cannot achieve security against adaptively chosen leakage functions: the adversary can just learn the state bit-by-bit by picking for each observation a different leakage function.[4]

1.2 Our contributions

In this work, we study various new and existing constructions of leakage-resilient pseudorandom objects. In a nutshell our results can be summarized as follows:

1. We revisit the work of Yu et al. [23] and show that the proof of the proposed (more natural) construction of a non-adaptive leakage-resilient (naLR) stream cipher has a subtle flaw. We propose a simple solution to this problem which unfortunately is impractical.
2. Inspired by the work of Dodis and Pietrzak [1], we show how to construct a nagLR non-adaptive PRF which is simpler and more natural and avoids the alternating structure used in [1].
3. We prove that a Feistel network with only 3 rounds, each instantiated with a non-adaptive leakage-resilient non-adaptive PRF, yields a non-adaptive leakage-resilient non-adaptive PRP. This completes a result of [1] who showed that a leakage-resilient PRP requires a superlogarithmic number of rounds instantiated with a leakage-resilient PRF.

We elaborate on these results further below.

[4] In this paper we not only differentiate between adaptive/non-adaptive leakage, but also adaptive/non-adaptive PRFs. In the latter non-adaptive means the adversary fixes all inputs in advance. We use the convention that non-adaptive leakage-resilient PRF means the leakage functions are chosen non-adaptively, whereas leakage-resilient non-adaptive PRF means the inputs to the PRF are chosen non-adaptively.

Section 2: Yu et al. [23] Revisited. The first leakage resilient symmetric primitive was the stream-cipher construction proposed by Dziembowski and Pietrzak in [4]. This construction has later been simplified in [20] using a weak PRF and is illustrated in Figure 1.

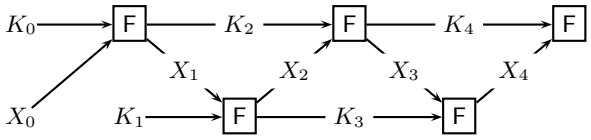

Fig. 1. Construction SC_{ALT} of a leakage resilient stream-cipher from any (weak) PRF F [20]. The initial secret key is X_0, K_0, K_1, the output is $X_0, X_1, \ldots$.

The constructions from [4,20] use an alternating structure (cf. Figure 1) and requires the secret state to hold *two* secret keys K_i, K_{i+1} for the underlying weak PRF F (a weak PRF is only guaranteed to be random on random inputs). The alternating structure enforces independence between inputs X_i and keys K_i, but seems mostly motivated by the security proof rather than contributing much to the scheme's real-world security.

Yu et al. [23] advocate that already much simpler constructions will be secure against most practically relevant side-channel attacks. They propose a more natural construction SC_{SEQ} from any wPRF F as illustrated in Figure 2. The secret state of this scheme consists of only a *single* secret key K_i for F, and two fixed *public* random values p_0, p_1 which are used alternately as inputs to F. This scheme is not leakage resilience if the leakage functions can be chosen adaptively, which is easily seen by the so-called "precomputation" attack: as we know p_0, p_1, in the i^{th} round we can choose a leakage function f_i which (using its input K_{i-1}) computes a future key K_t (for some $t > i$) and leaks some bits about it. As we can do this for any $i < t$, we can (for a sufficiently large t) learn the entire K_t.

It is claimed in [23] that the construction from Figure 2 is a naLR stream cipher. Note that the precomputation attack becomes infeasible if one must choose the leakage functions f_i before seeing p_0, p_1, as now $f_i(K_{i-1}, p_{i-1 \bmod 2})$ cannot compute the future key $K_{i+1} = \mathsf{F}(K_i, p_{i \bmod 2})$. Unfortunately, as we discuss in Section 2, the main technical lemma used in their proof has a subtle flaw, and thus the security proof is incorrect. Currently, we do not know if the construction is actually insecure, or if the proof can be salvaged. Our counterexample showing that their main lemma is flawed does not lead to an actual attack on the naLR security of the cipher.

The proof in [23] uses a lemma from [20] which states that the output $\mathsf{F}(K, X)$ of a weak PRF F is pseudorandom, even if K, X only have high pseudoentropy and are *independent*. The flaw in their proof roots from the fact that the input p_0 is reused every second round (cf. Figure 2), and thus already in the 3rd round, where one computes $K_3 \leftarrow \mathsf{F}(K_2, p_0)$, the key K_2 is not independent from p_0, which means one cannot apply the lemma from [20] directly.

This dependence problem disappears if one uses fresh public random inputs $p_0, p_1, p_2, \ldots$ for every round instead of alternating the two values p_0 and p_1, we will denote this construction by SC^+_{SEQ}. Of course SC^+_{SEQ} is pretty much useless in practice as its description size (i.e. the public inputs $p_0, p_1, \ldots$) is linear in the length of the output it can generate. Nonetheless, the observation that SC^+_{SEQ} is naLR will be useful for constructing nagLR non-adaptive PRFs as discussed below.

Section 3: Leakage-Resilient PRFs. Dodis and Pietrzak [1] construct a nagLR PRF. Their basic idea is to use the leakage resilient stream-cipher from [20] in a tree-like construction (inspired by the classical GGM construction.). Their construction is rather involved, as the alternating structure of the stream-cipher must be preserved within the tree like structure of the GGM transformation.[5]

We propose a much simpler construction illustrated in Figure 3, which we get by using the naLR stream cipher SC^+_{SEQ} (discussed in the previous section) within a GGM-like tree-structure. One may expect that starting with naLR stream-cipher like SC^+_{SEQ} and use it within GGM, we obtain a naLR PRF. Surprisingly, we show that this intuition is wrong. In fact, our construction in Figure 3 can be completely broken even using only non-adaptive leakage.

Our attack exploits the fact that, even though the leakage-functions cannot be adaptively chosen, the inputs to the PRF can be chosen adaptively. In particular, the choice of the inputs can depend on the public values p_i. Intuitively, this allows us to commit to exponentially many leakage functions (one for each input to the PRF) at the beginning, and only later, when we learn the p_i's we can choose which leakage function to choose by choosing the appropriate input to the PRF adaptively.[6] On the positive side, we show that our construction $\Gamma^{\mathsf{F},m}$ is a nagLR non-adaptive PRF, that is, it is secure if *not only* the leakage-function, but *also* the inputs to $\Gamma^{\mathsf{F},m}$ are chosen non-adaptively. This, of course, is a strong assumption, but for some important applications, like the initialization of a stream cipher [22], such a non-adaptive PRF is sufficient (in fact, here even a weak PRF is sufficient). Also, we would like to mention that in practice many side-channel attacks, such as DPA attacks, work by measuring the power consumption of the device on *random* inputs. Our security analysis incorporates such important attacks where the adversary exploits leakages from random inputs to the cryptographic scheme. We emphasize that, of course, our construction is an adaptively secure PRF in the black-box sense.

Section 4: Leakage-Resilient PRPs. A classical result by Luby and Rackoff [16] shows that a three-round Feistel (cf. Figure 4) network, where each round

[5] Whereas the GGM construction is just a simple tree, the construction of [1] is a graph with tree-width 3.

[6] Let us mention that for the attack we require that the leakage functions are aware (i.e. get as input) which node in the tree they are leaking from. Modeling granular leakage like this makes our positive results stronger, but the attacks more artificial. We don't know if our construction can be broken with non-adaptive leakage where the leakage-function is oblivious about the node it is leaking from.

is instantiated with a secure PRF, is a secure PRP. Dodis and Pietrzak [1] show that three-round Feistel networks cannot be leakage resilient. More precisely, they show that every Feistel network with a constant number of rounds (using any perfectly leakage resilient round functions, e.g. a random oracle) can be broken using only very simple leakage (e.g., the Hamming-weight of the inputs to the round functions). On the positive side, they show that a Feistel network with a super-logarithmic number of rounds instantiated with $\mathcal{L}$-LR PRFs is a $\mathcal{L}$-gLR PRP for any class $\mathcal{L}$ of leakage functions. Here, $\mathcal{L}$ is some class of admissible leakage functions, which in our case will usually be all polynomial-time computable functions with range $\{0,1\}^\lambda$ for some $\lambda \in \mathbb{N}$.

The aforementioned attack requires that one can query the PRF adaptively. We show that this is inherent by proving that a 3-round Feistel instantiated with $\mathcal{L}$-LR PRFs yields a $\mathcal{L}$-gLR non-adaptive PRP. This again illustrates the power of non-adaptivity in the leakage setting.

1.3 More Related Work

We notice that an alternative way to construct symmetric leakage resilient primitives is by using techniques from leakage resilient circuit compilers. Leakage-resilient circuit compilers allow to transform any circuit, e.g., an implementation of the AES, into a transformed circuit that is protected against certain classes of leakage attacks. This line of research was initiated by Ishai et al. [14] who show security against probing attacks. This result was recently generalized to a setting where leakages can be described by an AC0 circuit [6]. The works that are most relevant to ours are recent leakage-resilient circuit compilers in the "only computation leaks" setting [15,9,3,10]. While on the positive side such compilers allow to provably protect any cryptographic scheme against certain classes of leakage, they typically make strong granularity assumptions and are inefficient.[7]

An approach exploiting parallelism to achieve practically efficient leakage-resilient block-ciphers was put forward by Medwed, Standaert and Joux in these proceedings [18].

1.4 Notation & Basic Definitions

In this section, we present some basic notation and definitions that will be used throughout this paper.

Strings & Sets. Concatenation of two strings x, y is denoted $x\|y$, or, if no confusion is possible, simply xy. For $X \in \{0,1\}^n$ we denote with $X[i]$ the i^{th} bit of X and with $X_{|i}$ the i bit prefix of X. $[a,b]$ denotes the interval $\{a, a+1, \ldots, b\}$, $[b]$ is short for $[1,b]$. For a set $\mathcal{X}$, $X \in_R \mathcal{X}$ denotes that X is assigned a value sampled uniformly at random from $\mathcal{X}$. For a distribution D, we denote $X \leftarrow D$ the random variable X sampled from the distribution D. To abbreviate notation, we often identify random variables with their distribution.

[7] Circuits that make use of techniques from [15,9,3,10] grow by a factor of n^2 compared to an unprotected circuit, where n as a statistical security parameter.

Functions. $\mathcal{R}_{m,n}$ denotes the set of all functions $\{0,1\}^m \to \{0,1\}^n$, $\mathcal{P}_n$ the set of all permutation over $\{0,1\}^n$.

Distance. With $\delta^{\mathsf{D}}(X;Y)$ we denote the advantage of a circuit D in distinguishing the random variables X, Y, i.e.: $\delta^{\mathsf{D}}(X;Y) \stackrel{\mathsf{def}}{=} |\Pr[\mathsf{D}(X)=1] - \Pr[\mathsf{D}(Y)=1]|$. $\Delta(X;Y) \stackrel{\mathsf{def}}{=} \max_{\mathsf{D}} \delta^{\mathsf{D}}(X;Y)$ denotes the statistical distance of X and Y. With $\delta_s(X;Y)$ we denote $max_{\mathsf{D}}\delta^{\mathsf{D}}(X;Y)$ where the maximum is over all circuits D of size s.

Entropies. We recall some basic definitions for different types of entropy.

Definition 1. *A random variable Z has* min-entropy *k, denoted $H_\infty(Z) = k$, if for all z in the range of Z we have $\Pr[Z=z] \leq 2^{-k}$.*

A "computational" version of min-entropy called HILL-pseudoentropy was introduced in [12].

Definition 2. *We say X has* HILL pseudoentropy *k, denoted by $H^{\mathrm{HILL}}_{\epsilon,s}(X) \geq k$, if there exists a distribution Y with min-entropy $H_\infty(Y) = k$ where $\delta_s(X;Y) \leq \epsilon$.*

Dodis et al. [2], and Hsiao et al. [13] extended the above notions to analyze what happens to the min-entropy (resp. HILL-pseudoentropy) of a random variable X given a possibly correlated random variable Z.

Definition 3. *Let (X,Z) be a pair of random variables. The* average min-entropy *of X conditioned on Z is defined as*

$$\widetilde{H}_\infty(X|Z) = -\log \sum_{z \in Z} \Pr[Z=z] 2^{-H_\infty(X|Z=y)}$$

A computational version was given in [13] and is formally defined as follows:

Definition 4. *Let (X,Z) be a pair of random variables. X has* conditional HILL pseudoentropy *at least k conditioned on Z, denoted $\widetilde{H}^{\mathrm{HILL}}_{\epsilon,s}(X|Z) \geq k$ if there exists a collection of distributions Y_z for each $z \in Z$, giving rise to a joint distribution (Y,Z), such that $\widetilde{H}_\infty(Y|Z) \geq k$ and $\delta_s((X,Z);(Y,Z)) \leq \epsilon$.*

Pseudorandomness. Pseudorandomness is a fundamental and extremely useful cryptographic concept. Informally, an object (such as a bit-string, function or permutation) is pseudorandom if (1) it can be efficiently implemented using a small amount of randomness and (2) it cannot be distinguished from the corresponding uniformly random object by any efficient algorithm. A basic building block to generate pseudorandomness that will be used a basic building block in our constructions is a weak pseudorandom function (weak PRF). In contrast to standard PRFs, the notion of a weak PRF is weaker, as its output only has to be pseudorandom for random inputs. We recall the definition of (weak) PRFs/PRPs below.

Definition 5. *A function* $\mathsf{F} : \{0,1\}^k \times \{0,1\}^m \to \{0,1\}^n$ *is an* (ϵ, s, q)-pseudorandom function (PRF) *if no adversary* $\mathcal{A}$ *of size* s *can distinguish* $\mathsf{F}(K, \cdot)$ *(instantiated with a random key* K*) from a random function* $\mathsf{R} \leftarrow \mathcal{R}_{m,n}$*. More precisely, for any* $\mathcal{A}$ *of size* s *that can make up to* q *queries to its oracle, we have*

$$|\Pr[K \leftarrow \{0,1\}^k : \mathcal{A}^{\mathsf{F}(K,\cdot)} \to 1] - \Pr[\mathsf{R} \leftarrow \mathcal{R}_{m,n} : \mathcal{A}^{\mathsf{R}(\cdot)} \to 1]| \leq \epsilon. \qquad (1)$$

A non-adaptive *PRF is defined similarly, except that we only consider non-adaptive adversaries who must choose the queries* $X_1, \ldots, X_q$ *before seeing any outputs. A* weak *PRF is defined similarly, except that the inputs* $X_1, \ldots, X_q$ *are chosen uniformly at random and not chosen by* $\mathcal{A}$*.*

A (non-adaptive/weak) pseudorandom permutation (PRP) is defined analogously, except that we require $\mathsf{F}(K, .)$ *to be a permutation for every* K*.*

2 Stream Ciphers

2.1 Yu et al. [23] Revisited

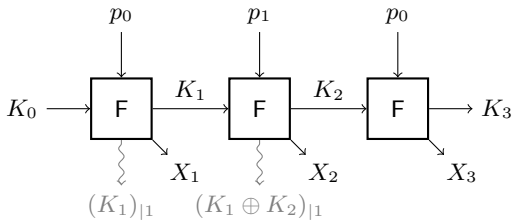

Fig. 2. The stream cipher construction SC_{SEQ} from a weak PRF F from [23]. K_0 is the secret initial key, p_0, p_1 are public random values and $X_1, X_2, \ldots$ is the output. The leakage leading to our counterexample to Lemma 3 from [23] is shown in gray.

A stream cipher is a function $\mathsf{SC} : \{0,1\}^k \to \{0,1\}^k \times \{0,1\}^n$ that, for every key K_0, defines a sequence $X_1, X_2, \ldots$ of outputs which are recursively defined as

$$(K_{i+1}, X_{i+1}) = \mathsf{SC}(K_i)$$

The security notion for stream ciphers requires that for a random initial secret key $K_0 \in_R \{0,1\}^k$, the outputs $X_1, X_2, \ldots, X_\ell$ are pseudorandom.

A stream cipher is leakage-resilient [4] if, for any ℓ, the outputs $X_\ell, X_{\ell+1}, \ldots$ are pseudorandom given $X_0, X_1, \ldots, X_{\ell-1}$ and a bounded amount of adaptively chosen leakage $\Lambda_0, \Lambda_1, \ldots, \Lambda_{\ell-1}$. This leakage is computed as follows: for any $i = 0, 1, \ldots, \ell - 2$, before $(K_{i+1}, X_{i+1}) \leftarrow \mathsf{SC}(K_i)$ is computed, an adversary chooses a leakage function f_i with range $\{0,1\}^\lambda$ (the parameter $\lambda \in \mathbb{N}$ bounds the amount of leakage we allow per round), and then gets $\Lambda_i = f_i(K_i')$ where

$K'_i \subseteq K_i$ is the part of the secret state which is accessed during the evaluation of $\mathsf{SC}(K_i)$.

Yu, Standaert, Pereira and Yung [23] propose a construction, SC_{SEQ}, illustrated in Figure 2. As outlined in the introduction this construction is vulnerable to the precomputation attack if the leakage functions can be chosen adaptively depending on the public values. In [23] it is claimed that it satisfies a relaxed notion of leakage-resilience where the leakage functions $f_1, f_2, \ldots$ are chosen non-adaptively.

The construction is initialized with a secret key $K_0 \in_R \{0,1\}^k$ for a wPRF $\mathsf{F} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^m$ and two *public* random values $p_0, p_1 \in_R \{0,1\}^n$ (although these values are public, it will be crucial that the adversary chooses the leakage functions *before* seeing these values.) The output is recursively computed as

$$(K_{i+1}, X_{i+1}) \leftarrow \mathsf{F}(K_i, p_{i \bmod 2})$$

The proofs in [20,23] use a lemma which states that the output of a weak PRF on a random input is pseudorandom even if the key is not uniform, but only has high min-entropy.

Proposition 1 (wPRF with non-uniform keys, Lemma 2 from [20]). *Let $\mathsf{F} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^m$ be a (ϵ, s, q) secure weak PRF, $X \in_R \{0,1\}^n$ be uniform and $K \in \{0,1\}^k$ be any random variable which is independent of X and has min-entropy $H_\infty(K) \geq k - \lambda$ for some $\lambda \in \mathbb{N}$, then*

$$(X, \mathsf{F}(K, X)) \text{ is pseudorandom.} \tag{2}$$

Quantitatively, $(X, \mathsf{F}(K, X))$ cannot be distinguished by adversaries of size $\approx s\epsilon^2$ with advantage $\approx \epsilon 2^\lambda$, so we have a loss of [8] ϵ^2 in circuit size and 2^λ in distinguishing advantage. The reduction makes $O(\lambda/\epsilon^2)$ queries, so q has to be at least that large.

The other main ingredient of the proof is a theorem from [4],[9] which states that a pseudorandom value $Z \in \{0,1\}^k$ has whp. HILL pseudoentropy almost $k - \lambda$ given any λ bits of auxiliary information A about Z. In our case, Z will be $(X, \mathsf{F}(K, X))$ as in eq.(2) and A will be leakage $f(X, X) \in \{0,1\}^\lambda$. Concretely, we get

Proposition 2. *For F, X, K as in Proposition 1 and f any leakage function with range $\{0,1\}^\lambda$*

$$\Pr[H^{\mathrm{HILL}}_{\epsilon',s'}(X, \mathsf{F}(K, X) \mid f(K, X)) \geq n + m - 2\lambda] \geq 1 - 2^{-\lambda} \tag{3}$$

[8] Let us note that there is a typo in the conference version of [20] (the t^2 in eq.(3) shoud be t), suggesting that the loss in circuit size is only ϵ, not ϵ^2.

[9] A more general "dense model theorem" was independently given in [21], cf. [7] for a good overview

where $s' \approx s\epsilon^4 2^{4\lambda}$ *and* $\epsilon' = \epsilon 2^{2\lambda}$*, so setting, say* $\lambda = \log(\epsilon^{-1}/4)$,[10] *we get* $s' \approx s\epsilon^5, \epsilon' = \sqrt{\epsilon}$.[11]

Before we turn to the problem with the security proof in [23], let us consider a slightly different construction which we will call SC^+_{SEQ}. This construction is defined like SC_{SEQ}, except that we use a fresh random input p_i (for $i = 0, \ldots, L-1$) in *every* round, i.e. $(K_{i+1}, X_{i+1}) \leftarrow \mathsf{SC}(K_i, p_i)$. Of course this is not a practical construction as we can output at most L blocks (where L denotes the number of the public p_i values.) But it illustrates the proof idea, and we will use this construction as a starting point to construct leakage-resilient PRFs in the next section.

Theorem 1. *The construction* SC^+_{SEQ} *is a* naLR *stream cipher. The amount* λ *of leakage tolerated per round depends on* F *as explained in Footnote 10.*

Proof. By the definition of a leakage-resilient stream cipher, we have to consider the following random experiment: an adversary $\mathcal{A}$ chooses some $L' \in [L]$ and leakage functions $f_1, \ldots, f_{L'} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^\lambda$. Then we sample $K_0 \in_R \{0,1\}^k$, $p_0, \ldots, p_{L-1} \in_R \{0,1\}^n$ and flip a coin $b \leftarrow \{0,1\}$.

The adversary gets the public values $p_0, \ldots, p_{L-1}$, the outputs $X_1, \ldots, X_{L'}$ and leakage $\Lambda_1, \ldots, \Lambda_{L'}$, where $\Lambda_i = f_i(K_{i-1}, p_{i-1})$.

If $b = 0$ the adversary gets a random $Z \in_R \{0,1\}^{(L-L')m'}$, if $b = 1$ she gets the remaining outputs $X_{L'+1}, \ldots, X_L$. We must prove, that she cannot guess b with probability much better than $1/2$.

We will prove that $K_{L'}$ is indistinguishable from a $\tilde{K}_{L'}$ which has $k - \lambda$ bits of average min-entropy given the view $\mathsf{view}_{L'}$ of the adversary after L' rounds, where view_i denotes the view of the adversary after the ith round, i.e.[12]

$$\mathsf{view}_i = \{p_0, \ldots, p_{i-1}, X_1, \ldots, X_i, \Lambda_1, \ldots, \Lambda_i\}$$

This will prove the theorem, as by eq.(3) and the fact that the $p_{L'+1}, \ldots, p_L$ are all chosen uniformly at random the remaining outputs $X_{L'+1}, \ldots, X_L$ will be pseudorandom. To see that $K_{L'}$ has high conditional pseudoentropy given $\mathsf{view}_{L'}$ we proceed in rounds, showing that for any $j \leq L'$, if K_{j-1} has high conditional pseudoentropy given view_{j-1}, then K_j has high pseudoentropy given view_j. For $j = 1$ this follows directly from eq.(3) as $(K_1, X_1) \leftarrow \mathsf{F}(K_0, p_0)$, where K_0 and p_0 are uniform.

[10] I.e. the leakage bound $\lambda = \log(\epsilon^{-1})/4$ is a function of the distinguishing advantage of the best distinguisher for the weak PRF F: If F is secure against polynomial-size distinguishers (i.e. $\epsilon = \omega(\log k)$), λ is superlogarithmic in the security parameter k. If F is exponentially hard, λ can be linear in k.

[11] Due to the very loose reductions in [20,7], these bounds will not imply any practical security guarantees if instantiated with a standard block cipher where k is typically something like 128 or 256. To get practical bounds, one would have to make idealized assumptions like assuming F is a random orcale [23].

[12] Note that we only include $p_0, \ldots, p_{i-1}$ into view_i, but in the actual security experiment the adversary gets to see all the $p_0, \ldots, p_L$ right away. We can do so as we only consider non-adaptive adversaries and the p_i's are chosen uniformly at random.

After the first round whp. K_1 has conditional pseudoentropy $k - 2\lambda$ given view_1. Thus, there exists a $\tilde{K}_1$ with average min-entropy $\widetilde{H}_\infty(\tilde{K}_1|\mathsf{view}_1) = k-2\lambda$ that is indistinguishable from K_1 (given view_1). Because of this, in the above experiment we can replace K_1 with $\tilde{K}_1$ and the probability that $\mathcal{A}$ will finally guess b correctly can only change by a negligible amount (otherwise $\mathcal{A}$ would constitute a distinguisher for K_1 and $\tilde{K}_1$.) We proceed as above for L' rounds (replacing K_i with $\tilde{K}_i$ for all $i = 1, \ldots, L'$) concluding that $K_{L'}$ is indistinguishable from a $\tilde{K}_{L'}$ where $\widetilde{H}_\infty(\tilde{K}_{L'}|\mathsf{view}_{L'}) = k - 2\lambda$. As the p_i are independent of K_i, we get by Proposition 1 the claimed statement. □

Let us go back to the construction SC_{SEQ} from [23], where we alternate between two inputs p_0, p_1 instead of using a fresh p_i for every round. Towards proving that this construction is a naLR stream-cipher, we can proceed as in the proof of Theorem 1 for the first two rounds arguing that K_1 and K_2 are indistinguishable from $\tilde{K}_1, \tilde{K}_2$ satisfying $H_\infty(\tilde{K}_i|\mathsf{view}_i) = k - 2\lambda$, but the 3rd step becomes more difficult.

The reason is that (in our adapted experiment, where K_i got replaced with $\tilde{K}_i$ for $i = 1, 2$) we compute $K_3 \leftarrow \mathsf{F}(\tilde{K}_2, p_0)$; but p_0 is clearly *not* random (and independent) given the view of $\mathcal{A}$, as p_0 was already used in the first round. Thus we cannot just apply Proposition 1 eq. (2) to conclude that the next key K_3 to be computed has high conditional pseudoentropy.

The authors of [23] are well aware of this problem. In order to "enforce" independence between $\tilde{K}_2$ and p_0, they put forward a lemma which claims these values become independent when given the leakage from the previous round (for clarity, we only state their lemma for the case of K_3)

Lemma 1 (Lemma 3 [23]). *$\tilde{K}_2$ and $\{p_0, p_1, X_1, X_2, \Lambda_1\}$ are independent given $\{p_1, \Lambda_2\}$.*

Although this approach looks promising, unfortunately, it turns out that this lemma is wrong (already for $\lambda = 1$) as can be seen by a simple counterexample illustrated in Figure 2: choose leakage functions f_1, f_2 that output the first bits $\Lambda_1 = K_1[1]$ and $\Lambda_2 = K_1[1] \oplus K_2[1]$ of K_1 and $K_1 \oplus K_2$ respectively.

First, we observe that in our adapted experiment where we replace the K_i's (having only conditional pseudoentropy) with $\tilde{K}_i$'s (having min-entropy), Λ_1, Λ_2 will be the first bits of $\tilde{K}_1$ and $\tilde{K}_1 \oplus \tilde{K}_2$. To see this, just note that, e.g., K_1 and $\tilde{K}_1$ are indistinguishable given $\Lambda_1 = K_1[1]$, this can only be the case if K_1 and $\tilde{K}_1$ agree on the first bit. To see why the lemma is flawed, we first observe that if $\{p_1, \Lambda_2 = \tilde{K}_1[1] \oplus \tilde{K}_2[1]\}$ is known, then given $\Lambda_1 = \tilde{K}_1[1]$ we can compute $\tilde{K}_2[1] = \tilde{K}_1[1] \oplus \tilde{K}_1[1] \oplus \tilde{K}_2[1]$. Now, the lemma claims $\tilde{K}_2$ is independent of $\{p_0, p_1, X_1, X_2, \tilde{K}_1[1]\}$ given $\{p_1, \tilde{K}_1[1] \oplus \tilde{K}_2[1]\}$, which by our observation means that $\tilde{K}_2[1]$ is already determined (i.e., has no entropy) given $\{p_1, \tilde{K}_1[1] \oplus \tilde{K}_2[1]\}$, but this is not true as shown by the claim below.

Claim. If $\tilde{K}_2[1]$ has no entropy given $\{p_1, \tilde{K}_1[1] \oplus \tilde{K}_2[1]\}$ then F is not a wPRF.

Proof. We will show that if F is a secure weak PRF, then $\{p_1, \tilde{K}_1[1], \tilde{K}_2[1]\}$ is close to being uniform (which implies the claim.) The value p_1 is uniform by

definition. To see that $\{p_1, \tilde{K}_1[1]\}$ is uniform recall that $K_1[1] = \tilde{K}_1[1]$, and $K_1 = \mathsf{F}(K_0, p_0)$. Clearly, every individual bit of K_1 (in particular $K_1[1]$) must be close to uniform as otherwise we could distinguish K_1 from uniform (and thus break the security of F) by just outputting this bit. Similarly, $K_2 = \mathsf{F}(\tilde{K}_1, p_1)$ is pseudorandom given $\{p_1, \tilde{K}_1[1]\}$, and thus $\tilde{K}_2[1] = K_2[1]$ is close to uniform given $\{p_1, \tilde{K}_1[1]\}$.

3 Leakage-Resilient PRFs

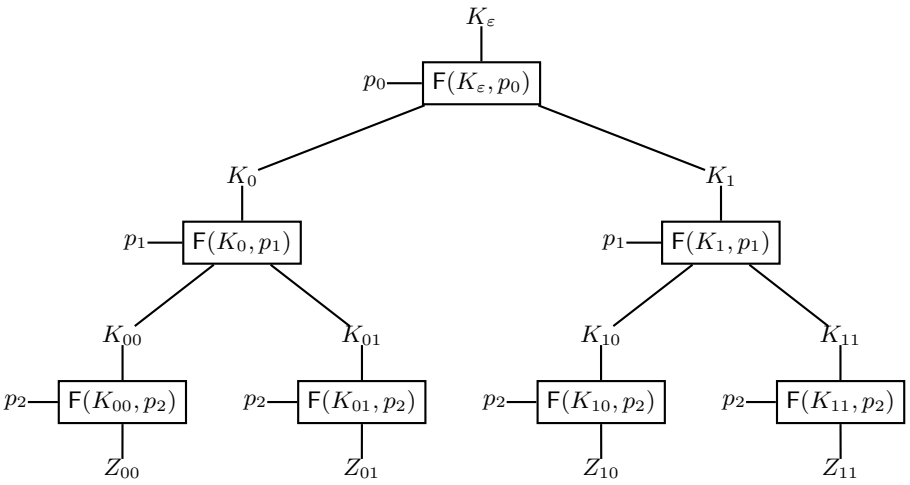

Fig. 3. Illustration of $I^{\mathsf{F},m}$ for $m = 2$. p_0, p_1 and p_2 are the random public values and $K_\varepsilon = K$ is the initial random key of the PRF. The output of the PRF for each $X \in \{0,1\}^2$ is represented by Z_X, i.e., the leaves of the tree.

In [1] Dodis and Pietrzak construct a nagLR PRF from any wPRF F. Informally, a PRF is leakage resilient if its outputs on all "fresh" inputs are pseudorandom, even if the adversary can query the PRF, and besides the outputs also gets the leakage from these computations.[13] In this section we propose a much simpler construction than the one from [1], which is a nagLR non-adaptive PRF, that is, it remains secure if not only the leakage function, but also the inputs are chosen non-adaptively.

We define a naLR PRF by considering an adversary $\mathcal{A}$ who has access to two oracles: the challenge and the leakage oracle. The first is as in Definition 5, i.e.,

[13] Let us remark that this is not the only meaningful notion of leakage-resilience for PRF. Instead of requiring that only fresh outputs look pseudorandom, we could ask for a simulator that can efficiently fake leakage. A notion along this lines in a somewhat different context and for nog-continious leakage (called "seed incompressibility") has been considered in [11].

either it is the pseudorandom function $\mathsf{F}(K,\cdot)$, or a random function $\mathsf{R} \leftarrow \mathcal{R}$. The latter oracle $\mathsf{F}^f(K,\cdot)$ can be queried on some input $X \in \{0,1\}^m$ and returns $\mathsf{F}(K,X)$ together with the leakage $f(K,X)$, where f is the leakage function non-adaptively chosen at the beginning of the experiment.[14] Of course, the queries to the two oracles must be disjoint.

Definition 6. *[$\mathcal{L}$-naLR (non-adaptive) PRF] A function $\mathsf{F} : \{0,1\}^k \times \{0,1\}^m \to \{0,1\}^n$ is a (ϵ, s, q)-secure $\mathcal{L}$-naLR PRF if for any $\mathcal{A}$ of size s that can make up to q disjoint queries to its two oracles, and for any leakage function $f \in \mathcal{L}$, we have*

$$\left| \Pr_{K \leftarrow \{0,1\}^k}[\mathcal{A}^{\mathsf{F}(K,\cdot),\mathsf{F}^f(K,\cdot)} = 1] - \Pr_{\mathsf{R} \leftarrow \mathcal{R}_{m,n}\ K \leftarrow \{0,1\}^k}[\mathcal{A}^{\mathsf{R}(\cdot),\mathsf{F}^f(K,\cdot)} = 1] \right| \quad \leq \quad \epsilon.$$

We will mostly omit the parameters ϵ, s and q and say that F is a $\mathcal{L}$-naLR PRF *if ϵ is some negligible function in k and s, q are superpolynomial in k.*

A $\mathcal{L}$-naLR non-adaptive *PRF is defined equivalently, except that $\mathcal{A}$ must choose the q PRF input queries non-adaptively.*

Recall that naLR security denotes $\mathcal{L}$-naLR security where $\mathcal{L}$ is the class of all efficiently computable functions with range $\{0,1\}^\lambda$ for some $\lambda \in \mathbb{N}$. In this section, we will only consider this special case, but we gave the general definition as it will be used in the next section. As outlined in the introduction, stateless (cf. Footnote 10) na-LR naPRFs don't exist, and thus following [22,1], we consider a "granular" nagLR-security notion, informally discussed in Section 1.1. Our construction $\Gamma^{\mathsf{F},m}$, illustrated in Figure 3, is inspired by the classical GGM construction of a PRF from a PRG [8]. On input $X \in \{0,1\}^m$ computes its output Z_X by invoking a wPRF F $m+1$ times sequentially. The inputs to the $m+1$ invocations are fixed random public values $p_0, \ldots, p_m$. The ith bit of the input $X[i]$ determines which half of the output of F in the ith invocation is used as a key for the $(i+1)$th invocation.

Let us define the PRF $\Gamma^{\mathsf{F},m} : \{0,1\}^{k+(m+1)\ell} \times \{0,1\}^m \to \{0,1\}^n$, which uses a wPRF $\mathsf{F} : \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^{2k}$ as main building block. The secret key is $K \leftarrow \{0,1\}^k$ and moreover we sample $m+1$ random public values $p = p_0, \ldots, p_m \leftarrow \{0,1\}^\ell$. Below we define how the output Z_X is computed by $\Gamma^{\mathsf{F},m}(K,p,X)$ in pseudocode. We explicitly state which bit of the input is read as this will determine the inputs that the leakage functions will get. With $\mathsf{F}_0(K,X)$ and $\mathsf{F}_1(K,X)$ we denote the function computing $\mathsf{F}(K,X)$ but only outputting the left and right half of the output, respectively.

PRF $\Gamma^{\mathsf{F},m}(K,(p_0,\ldots,p_m),X)$, where $X \in \{0,1\}^m$ and $K \leftarrow \{0,1\}^k$:
Set $i := 0$ and $K_\varepsilon := K$
Repeat:
$i := i + 1$.

[14] Note that we allow the adversary to only choose one leakage function. On could also consider a stronger non-adaptive notion where the adversary can initially choose a different leakage function for every query to be made.

Read the input bit $X[i]$.
Compute $K_{X_{|i}} := \mathsf{F}_{X[i]}(K_{X_{|i-1}}, p_{i-1})$.
Until $i = m$
Compute $Z_X := \mathsf{F}(K_X, p_m)$.
Output Z_X.

We think of the above computation as being performed in $m + 1$ time steps. Each of the m loops, and the final computation of Z_X, is a time step. Thus, the nagLR non-adaptive PRF security notion allows the adversary to initially choose a leakage function $f : \{0,1\}^\ell \times \{0,1\}^k \times \{0,1\} \to \{0,1\}^\lambda$ and inputs to the two oracles. For every input X to the $\mathsf{F}^f(K, .)$ oracle, the adversary gets $F(K, X)$ and leakage

$$f(p_0, K_\varepsilon, X[1]), f(p_1, K_{X_{|1}}, X[2]), \ldots, f(p_{m-1}, K_{X_{|m-1}}, X[m]), f(p_m, K_X, 0) \tag{4}$$

As in [1], we can actually handle somewhat stronger leakage functions which not only get the bit $X[i]$ of X touched in the ith time step, but all the bits $X_{|i}$ of X touched so far, i.e.

$$f(p_0, K_\varepsilon, X_{|1}), f(p_1, K_{X_{|1}}, X_{|2}), \ldots, f(p_{m-1}, K_{X_{|m-1}}, X_{|m-1}), f(p_m, K_X, X) \tag{5}$$

The interpretation here is, that the leakage function f knows exactly at which node of the tree it is. We are now ready to prove our main theorem in this section

Theorem 2. *If $\mathsf{F} : \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^{2k}$ is a weak PRF, then $\Gamma^{\mathsf{F},m}$ is a* nagLR *non-adaptive PRF. The amount of leakage λ per time step (i.e., for each invocation of F) depends on the security of F (cf. Footnote 10)*

Proof. Let $\mathcal{A}$ be an adversary which initially chooses a leakage function f (as described above), q distinct inputs $x_1, \ldots, x_q$ and some q_0, meaning that the first q_0 queries will be leakage queries, and the last $q_1 := q - q_0$ queries are challenge queries. We sample a random key $K \leftarrow \{0,1\}^k$ and a coin $b \leftarrow \{0,1\}$ determining if we're in the real or random experiment (note that we do not yet sample the $p_0, \ldots, p_m \leftarrow \{0,1\}^\ell$.)

We now evaluate all q queries simultaneously, going down the tree as illustrated in Figure 3 layer by layer, sampling the random p_i's as we go down (this parallel evaluation is only possible as the queries are chosen non-adaptively.) It will be convenient to give the adversary the leakage for all q queries (even though the last q_1 challenge queries are not supposed to leak at all), except for the very last layer. In the last layer, we evaluate the first q_0 queries, and give the adversary this outputs together with the leakage. If $b = 0$ (which means we're in the real experiment) the adversary gets the outputs, and random values otherwise.

Below we formally describe how the leakage is computed. As just mentioned, we give the adversary more power than required for nagLR-security. Concretely, in item 2 below, she gets leakage from internal nodes on all queries, not just he leakage queries. Set $i := 0$, sample a random $K = K_\varepsilon$, and then the outputs and leakage are computed layer by layer as follows:

1. sample a random p_i and give it to the adversary.
2. compute $K_{I\|0}\|K_{I\|1} := \mathsf{F}(K_I, p_i)$ and leakage $\Lambda_I := f(K_I, p_i, I)$ for all i bit prefixes of $x_1, \ldots, x_q$. Give all the computed leakage to the adversary.
3. If $i < m-1$ then set $i := i+1$ and go back to step 1, otherwise go to next step (at this point we have computed K_{x_i} for all queries x_i.)
4. sample a random p_m and give it to the adversary.
5. Compute the final outptus $Z_{x_i} := \mathsf{F}(K_{x_i}, p_m) = \Gamma^{\mathsf{F},m}(K, x_i)$ and leakage $\Lambda_{x_i} := f(X_{x_1}, p_m, x_i)$ for $i = 1, \ldots, q_0$. Give this outptus and leakage to the adversary.
6. If $b = 0$, for $i = q_0+1, \ldots, q$, compute $Z_{x_i} := \mathsf{F}(K_{x_i}, p_m)$, otherwise, if $b = 1$, sample random $Z_{x_i} \leftarrow \{0,1\}^{2n}$. Give this values to the adversary.

We denote by view_0 the view of the adversary in the above experiment if $b = 0$, and with view_m if $b = 1$. To prove the theorem we must show that view_0 and view_m are computationally indistinguishable. We will consider hybrid views $\mathsf{view}_1, \ldots, \mathsf{view}_{m-1}$, and show that for every $i = 1, \ldots, m$, view_{i-1} and view_i indistinguishable.

Consider the computation $K_0\|K_1 = \mathsf{F}(K_\varepsilon, p_0), \Lambda_\varepsilon = f(K_\varepsilon, p_0)$ in the first layer. As K_ε has min-entropy $n - 2\lambda$ (in fact, in this first layer, this key is even uniform) and p_0 is uniform, by Proposition 1 $K_0\|K_1$ is pseudorandom given p_0, and by Proposition 2 $K_0\|K_1$ has (whp.) HILL pseudoentropy $2n - 2\lambda$ when additionally given Λ_ε. The first hybrid view_1 is dervied from the hybrid view_0 by replacing this $K_0\|K_1$ with a random variable $\tilde{K}_0\|\tilde{K}_1$ which has min-entorpy $2n - 2\lambda$ given p_0, Λ_ε. By the definition of HILL pseudoentropy, such a $\tilde{K}_0\|\tilde{K}_1$ exists, where view_0 and view_1 are computationally indistinguishable. Thus, in view_1, the inputs $\tilde{K}_0$ and $\tilde{K}_1$ to the first layer (which are outptus from the zero layer) have min-entropy $n - 2\lambda$, and by Proposition 1, each outputs of this layer will have pseudoentropy $n - 2\lambda$ given the entire view of the adversary. The hybrid view_2 is derived from view_1 by replacing this outputs which have min-entropy $n - 2\lambda$, and so on, until we get the hybrid view_{m-1} which is indistinguishable from view_0. In view_{m-1}, the inputs $\tilde{K}_{x_i}$ to the last layer has min-entropy $n - 2\lambda$. We choose p_m uniformly at random, and it follows by Proposition 1, that the "challange" outputs $Z_{x_i} := \mathsf{F}(\tilde{K}_{x_i}, p_m)$ are pseudorandom, and thus indistinguishable from view_m which is derived from view_{m-1} by replacing all challange outputs by uniformly random values (as in the case $b = 1$.) □

3.1 An Adaptive Attack Against Our Construction $\Gamma^{\mathsf{F},m}$

In Theorem 2 we showed that $\Gamma^{\mathsf{F},m}$ is a nagLR non-adaptive PRF. As discussed in Section 1.2, it trivially is not a naLR non-adaptive PRF or gLR non-adaptive PRF, i.e. the non-adaptivity and granularity for the leakage are necessary. It is a natural question whether it is a nagLR PRF like the (much more sophisticated) construction from [1]).

We answer this question negatively and show a simple attack against $\Gamma^{\mathsf{F},m}$. The attack allows the adversary to learn leakage that reveals the first λ bits of $\Gamma^{\mathsf{F},m}(K, X)$ for an input X that has not yet been queried. Clearly, this breaks

the security of the PRF as required by Definition 6. Suppose $m = \ell + 1$, then the attack works as follows:

1. Define $f(p_{m-1}, K_I, I)$ to be the first λ bits of $\mathsf{F}(\mathsf{F}_0(K_I, p_{m-1}), I)$.
2. Learn the public values $p_0, \ldots, p_m \in \{0,1\}^\ell$.
3. Query the leakage oracle for $p_m||1$ and obtain $\Gamma^{\mathsf{F},m}(K, p_m||1)$ and, from the leakage, the first λ bits of

$$\mathsf{F}(\mathsf{F}_0(K_{p_m}, p_{m-1}), p_m) = \Gamma^{\mathsf{F},m}(K, p_m||0).$$

Thus, for a leakage query $p_m||1$ the attack reveals the first λ bits of $\Gamma^{\mathsf{F},m}(K, p_m||0)$. We emphasize that this attack is rather artificial and most likely will not affect the real-world security of our construction. However, it illustrates that any attempt to prove the security of $\Gamma^{\mathsf{F},m}$ in an adaptive setting must fail (indeed, this attack works even if we assign a different public value to every node – details are omitted in this extended abstract).

Let us emphasize that this attack requires that for each execution of the weak PRF F the corresponding leakage function is "aware" of its current position in the tree, that is, we need a leakage function as in eq.(5) and not eq.(4). Although for our positive result considering a stronger leakage model only strengthens the result, for an attack we would like the model to be as weak as possible and stick with leakage functions that only get whatever is touched, and nothing beyond that, as required for nagLR PRFs. We do not know if such an attack exists against $\Gamma^{\mathsf{F},m}$.

4 Leakage-Resilient PRPs

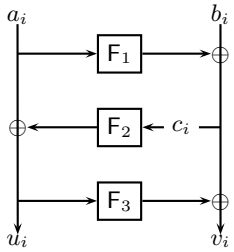

Fig. 4. 3-round Feistel Network $\Phi_{\mathsf{F}_1,\mathsf{F}_2,\mathsf{F}_3} : \{0,1\}^{2n} \to \{0,1\}^{2n}$ with round functions $\mathsf{F}_i : \{0,1\}^n \to \{0,1\}^n$.

In the previous section we gave a simple construction of a PRF which is secure against non-adaptive leakage if queried on non-adaptively chosen inputs. In practice, one usually doesn't use pseudorandom functions, but rather pseudorandom permutations (PRPs). In particular, block ciphers, the work horses of cryptography, are assumed to be PRPs. Block ciphers are also the main targets of side-channel cryptanalysts, thus coming up with leakage-resilient PRPs is a particularly worthwhile task.

In the standard setting (i.e. without leakage), Luby and Rackoff [16] famously showed that one can construct a PRP from a PRF by using a three-round Feistel network as illustrated in Figure 4. With one round more one even gets a strong PRP, i.e. an object that is indistinguishable from a uniformly random permutation even if one can query it from both sides.

To prove that a 3-round Feistel using PRFs as round functions is a PRP one proceeds in two steps.[15] First one shows that a 3-round Feistel instantiated with *uniformly random functions* is indistinguishable from a *uniformly random permutation* (this step is completely information theoretic). In the second step one then observes that a 3-round Feistel instantiated with URFs (uniformly random functions) is indistinguishable from a 3-round Feistel using PRFs. This second step follows by a simple hybrid argument where we replace the pseudorandom round functions with uniformly random functions one by one. A restricted case of the statement claiming only non-adaptive security and using only random functions as round functions, is given by the proposition below.

Proposition 3 (3-Round Feistel is Non-Adaptively Secure PRP). *For any $n, q \in \mathbb{N}$ and $x_1, \ldots, x_q \in \{0,1\}^{2n}$ consider the distributions:*

- *Sample $\mathsf{P} \in_R \mathbf{P}_{2n}$ and, for $i \in [q]$, set $y_i = \mathsf{P}(x_i)$.*
- *Sample $\mathsf{F}_1, \mathsf{F}_2, \mathsf{F}_3 \in_R \mathbf{R}_n$ and for $i \in [q]$ set $z_i = \Phi_{\mathsf{F}_1,\mathsf{F}_2,\mathsf{F}_3}(x_i)$ (as in Figure 4)*

then

$$\Delta([y_1, \ldots, y_q], [z_1, \ldots, z_q]) \leq \frac{q^2}{2^n}$$

Proof (sketch). Consider the values $c_i = \mathsf{F}_1(a_i) \oplus b_i$ for $i = 1, \ldots, q$ (where $x_i = a_i \| b_i$, cf. Figure 4.) As F_1 is a URF, these c_i's will contain a collision with probability at most $q(q-1)/2^{n+1}$. Assuming they are all distinct, the $u_i = \mathsf{F}_2(c_i) \oplus a_i$'s are uniformly random as F_2 is a URF. As they are uniformly random, they also will contain a collision with with probability at most $q(q-1)/2^{n+1}$. This implies the values $z_i = u_i \| v_i$ are $2 \cdot q(q-1)/2^{n+1} = q(q-1)/2^n$ close to uniform over $\{0,1\}^{2n}$. The uniform distribution over q elements over $\{0,1\}^{2n}$ is $q(q-1)/2^{2n}$ close to the distribution of the $y_1, \ldots, y_q$ (which is uniform, but without repetition.) Thus, as statistical distance obeys the triangle inequality, the z_i's are $q(q-1)/2^n + q(q-1)/2^{2n} \leq q^2/2^n$ close to the y_i's. □

Proposition 3 also holds if the inputs x_i are chosen adaptively, but the proof for this case is significantly more delicate. The proof of Proposition 3 above uses the fact that the inputs $c_1, \ldots, c_q$ to the second round function (cf. Figure 4) are all distinct (with high probability). The adaptive case also goes along these lines, but here one has to argue that the c_i's are also "hidden", as an adaptive adversary who could "guess" the c_i values could compute inputs to the Feistel network where the outputs partially collide.

As shown in [1], it is already sufficient to get some simple leakage (e.g. the Hamming Weight) of the c_i values to launch such an attack. This attack can

[15] This proof template follows [17]; the original proof of Luby and Rackoff [16] is "direct", but also more complicated.

be adapted to work on Feistel networks with any number r of rounds, but its complexity (i.e. number of adaptive queries) grows exponentially in r. Still, this implies that a constant-round Feistel network, instantiated with leakage-resilient PRFs, can be broken in polynomial time, and thus is *not* a leakage-resilient PRP.

The queries to the Feistel network made in the [1] attack are adaptive, and here we show that this is indeed crucial. By Theorem 3 below, a 3-round Feistel is a *non-adaptively* secure leakage-resilient PRP if instantiated with leakage-resilient PRFs. The notion of leakage-resilience achieved by the PRP is inherited from the underlying PRF. If the round functions are $\mathcal{L}$-naLR PRFs, then we get a $\mathcal{L}$-nagLR PRP.

More formally, we initially choose a bit $b \in \{0,1\}$ and three keys k_1, k_2, k_3 for F which defines the round functions $\mathsf{F}_i(.) = \mathsf{F}(k_i, .)$ for $i = \{1,2,3\}$, and if $b = 1$ a random permutation $\mathsf{P} \in_R \mathbf{P}_{2n}$ (using lazy sampling.) The adversary can initially choose three leakage functions $f_1, f_2, f_3 \in \mathcal{L}$, distinct inputs $a_i \| b_i$ for $i = 1, \ldots, q$ and some q_0 which specifies that the first q_0 inputs are leakage queries, and the last $q_1 := q - q_0$ are challange queries (as we consider non-adaptive queries, we can wlog. assume the queries are ordered like this.) She then gets, for every $i \leq q_0$, the outputs $u_i \| v_i = \Phi_{\mathsf{F}_1,\mathsf{F}_2,\mathsf{F}_3}(a_i \| b_i)$ and the leakage $f_1(k_1, a_i), f_2(k_2, c_i)$ and $f_3(k_3, u_i)$ (so, each round of the Feistel network is considered a time-step which leaks independently.) For the queries $i > q_0$ she gets the regular output $\Phi_{\mathsf{F}_1,\mathsf{F}_2,\mathsf{F}_3}(a_i \| b_i)$ if $b = 0$ and the random $\mathsf{P}(a_i \| b_i)$ otherwise. Note that besides the evaluation of the round functions F_i, one also has to compute three XORs. It would be cheating to assume that this XORs are leakage free. We go to the other extreme, and assume the XORs leak completely by giving the adversary the entire c_i value for every leakage query. This c_i together with the known values a_i, b_i, u_i, v_i specifies all the inputs/outputs to the three XOR computations (e.g. the first XOR takes as inputs b_i and $b_i \oplus c_i$.)

Theorem 3. *Let* $\mathsf{F} : \{0,1\}^\ell \times \{0,1\}^n \to \{0,1\}^n$ *be an* (q, ϵ, s)*-secure* $\mathcal{L}$-naLR *non-adaptive PRF. Then the three round Feistel network* $\Phi_{\mathsf{F}_1,\mathsf{F}_2,\mathsf{F}_3}$*, where each* $\mathsf{F}_i = \mathsf{F}(k_i, .)$ *is an independent instantiation of* F*, is a* (q, ϵ', s')*-secure* $\mathcal{L}$-nagLR *non-adaptive PRP where*

$$\epsilon' = 3\epsilon + q^2/2^n \quad s' = s - poly(q, n)$$

Proof. Let $x_1, \ldots, x_{q_0}$ and $x'_1, \ldots, x'_{q_1}$ (where $q_0 + q_1 = q$) denote the non-adaptively chosen leakage and challenge queries. Let $k_i \leftarrow \{0,1\}^\ell$ be randomly chosen keys for the round functions $\mathsf{F}_i(.) = \mathsf{F}(k_i, .)$. Let $f_1, f_2, f_3 \in \mathcal{L}$ denote the leakage functions chosen by the adversary. The adversary gets (with $x_i \stackrel{\mathsf{def}}{=} a_i \| b_i$ and c_i, u_i, v_i as in Figure 4)

$$y_i = \Phi_{\mathsf{F}_1,\mathsf{F}_2,\mathsf{F}_3}(x_i) \quad \Lambda_i \stackrel{\mathsf{def}}{=} \{f_1(k_1, a_i), f_2(k_2, c_i), f_3(k_3, u_i), c_i\}$$

We must prove that the outputs $y'_1, \ldots, y'_{q_1}$, where

$$y'_i = \Phi_{\mathsf{F}_1,\mathsf{F}_2,\mathsf{F}_3}(x'_i)$$

are pseudorandom given $y_1, \ldots, y_{q_0}$ and $\Lambda_1, \ldots, \Lambda_{q_0}$.

Claim. The c_i's corresponding to the q queries are distinct with probability at least $q(q-1)/2^{n+1} + \epsilon$.

Proof. To see this, let δ denote the probability that the c_i's collide; we can construct a non-adaptive q-query distinguisher for F with advantage $\delta - q(q-1)/2^{n+1}$ (note that as F is an ϵ-secure PRF this will imply that $\delta \leq q(q-1)/2^{n+1} + \epsilon$ as claimed.) This distinguisher simply queries its oracle (which is either a URF or $\mathsf{F}(k,.)$) on inputs $a_1, \ldots, a_q$, obtaining $z_1, \ldots, z_q$; the oracle outputs 1 if and only if any of the $z_i \oplus b_i$ collide. If the outputs come from a URF, this probability is $q(q-1)/2^{n+1}$, whereas if they come from $\mathsf{F}(k,.)$ this probability is δ by definition. This concludes the proof of the claim. □

Now assume all the c_i's are distinct. Conditioned on this, we can show by a similar argument that also all the $u_i = \mathsf{F}_2(k_2, c_i) \oplus a_i$ will be distinct with probability $q(q-1)/2^{n+1} + \epsilon$.

Assume the c_i's and u_i's are all distinct and recall that $v_i = \mathsf{F}_3(k_3, u_i) \oplus c_i$. Then it follows from the $\mathcal{L}$-naLR non-adaptive PRF security of F that the $y'_i = u_{q_0+i} \| v_{q_0+i}$ values for $i = 1, \ldots, q_1$ are pseudorandom given $y_1, \ldots, y_{q_0}$ and $\Lambda_1, \ldots, \Lambda_{q_0}$, as $\mathsf{F}_2(k_2, .)$ and $\mathsf{F}_3(k_3, .)$ are queried on distinct inputs in the first q_0 and the last q_1 queries. □

References

1. Y. Dodis and K. Pietrzak. Leakage-resilient pseudorandom functions and side-channel attacks on Feistel networks. In T. Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 21–40. Springer, Aug. 2010.
2. Y. Dodis, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 523–540. Springer, May 2004.
3. S. Dziembowski and S. Faust. Leakage-resilient circuits without computational assumptions. In *TCC*, pages 230–247, 2012.
4. S. Dziembowski and K. Pietrzak. Leakage-resilient cryptography. In *49th FOCS*, pages 293–302. IEEE Computer Society Press, Oct. 2008.
5. S. Faust, E. Kiltz, K. Pietrzak, and G. N. Rothblum. Leakage-resilient signatures. In D. Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 343–360. Springer, Feb. 2010.
6. S. Faust, T. Rabin, L. Reyzin, E. Tromer, and V. Vaikuntanathan. Protecting circuits from leakage: the computationally-bounded and noisy cases. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 135–156. Springer, May 2010.
7. B. Fuller and L. Reyzin. Computational entropy and information leakage. `http://www.cs.bu.edu/~reyzin/research.html`.
8. O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33:792–807, 1986.
9. S. Goldwasser and G. N. Rothblum. Securing computation against continuous leakage. In T. Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 59–79. Springer, Aug. 2010.

10. S. Goldwasser and G. N. Rothblum. How to compute in the presence of leakage. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:10, 2012.
11. S. Halevi, S. Myers, and C. Rackoff. On seed-incompressible functions. In R. Canetti, editor, *TCC 2008*, volume 4948 of *LNCS*, pages 19–36. Springer, Mar. 2008.
12. J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.
13. C.-Y. Hsiao, C.-J. Lu, and L. Reyzin. Conditional computational entropy, or toward separating pseudoentropy from compressibility. In M. Naor, editor, *EUROCRYPT 2007*, volume 4515 of *LNCS*, pages 169–186. Springer, May 2007.
14. Y. Ishai, A. Sahai, and D. Wagner. Private circuits: Securing hardware against probing attacks. In D. Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 463–481. Springer, Aug. 2003.
15. A. Juma and Y. Vahlis. Protecting cryptographic keys against continual leakage. In T. Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 41–58. Springer, Aug. 2010.
16. M. Luby and C. Rackoff. How to construct pseudo-random permutations from pseudo-random functions (abstract). In H. C. Williams, editor, *CRYPTO'85*, volume 218 of *LNCS*, page 447. Springer, Aug. 1986.
17. U. M. Maurer. Indistinguishability of random systems. In L. R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 110–132. Springer, Apr. / May 2002.
18. M. Medwed, F.-X. Standaert, and A. Joux. Towards super-exponential side-channel security with efficient leakage-resilient prfs. In *CHES*, 2012.
19. S. Micali and L. Reyzin. Physically observable cryptography (extended abstract). In M. Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 278–296. Springer, Feb. 2004.
20. K. Pietrzak. A leakage-resilient mode of operation. In A. Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 462–482. Springer, Apr. 2009.
21. O. Reingold, L. Trevisan, M. Tulsiani, and S. P. Vadhan. Dense subsets of pseudorandom sets. In *49th FOCS*, pages 76–85. IEEE Computer Society Press, Oct. 2008.
22. F.-X. Standaert, O. Pereira, Y. Yu, J.-J. Quisquater, M. Yung, and E. Oswald. Leakage resilient cryptography in practice. *Towards Hardware Intrinsic Security: Foundation and Practice*, pages 105– 139, 2010.
23. Y. Yu, F.-X. Standaert, O. Pereira, and M. Yung. Practical leakage-resilient pseudorandom generators. In *ACM CCS 10*, pages 141–151. ACM Press, 2010.