Leakage-Resilient Pseudorandom Functions and Side-Channel Attacks on Feistel Networks

Yevgeniy Dodis and Krzysztof Pietrzak

New York University and CWI Amsterdam

Abstract. A cryptographic primitive is leakage-resilient, if it remains secure even if an adversary can learn a bounded amount of arbitrary information about the computation with every invocation. As a consequence, the physical implementation of a leakage-resilient primitive is secure against every side-channel as long as the amount of information leaked per invocation is bounded.
In this paper we prove positive and negative results about the feasibility of constructing leakage-resilient pseudorandom functions and permutations (i.e. block-ciphers). Our results are three fold:
1. We construct (from any standard PRF) a PRF which satisfies a relaxed notion of leakage-resilience where (1) the leakage function is fixed (and not adaptively chosen with each query.) and (2) the computation is split into several steps which leak individually (a "step" will be the invocation of the underlying PRF.)
2. We prove that a Feistel network with a super-logarithmic number of rounds, each instantiated with a leakage-resilient PRF, is a leakage resilient PRP. This reduction also holds for the non-adaptive notion just discussed, we thus get a block-cipher which is leakage-resilient (against non-adaptive leakage).
3. We propose generic side-channel attacks against Feistel networks. The attacks are generic in the sense that they work for any round functions (e.g. uniformly random functions) and only require some simple leakage from the inputs to the round functions. For example we show how to invert an r round Feistel network over $2n$ bits making $4 \cdot (n+1)^{r-2}$ forward queries, if with each query we are also given as leakage the Hamming weight of the inputs to the r round functions. This complements the result from the previous item showing that a super-constant number of rounds is necessary.

1 Introduction

Traditional cryptographic security definitions only give the adversary black-box access to the primitive at hand. For example, a function $\mathsf{F} : \Sigma^k \times \Sigma^m \to \Sigma^n$ ($\Sigma \stackrel{\mathsf{def}}{=} \{0,1\}$) is pseudorandom if no efficient adversary given oracle access to a function $\mathcal{O} : \Sigma^m \to \Sigma^n$ can tell whether the oracle is a uniformly random function or instantiated with $\mathsf{F}(K,.)$ for a random key $K \in \Sigma^k$.

Unfortunately, this model does not capture many attacks in the real-world where adversaries can attack concrete *implementations* of cryptosystems which potentially leak information about their internal secret state during computation. Attacks exploiting such leakage are called side-channel attacks. Popular side-channels that have been exploited for cryptanalytic attacks include running-time [28], electromagnetic radiation [39, 20] or power consumption [30].

Countermeasures. Side-channel attacks are a very real threat for systems used in practice. Not surprisingly, much research has concentrated on developing countermeasures against such attacks. This research is mostly done by practitioners (i.e., the cryptographic hardware community) who are also active in finding and exploiting new side-channels, [37] gives an overview of this research. The countermeasures proposed are usually ad-hoc, in the sense that they aim to protect against some particular, known attack, and are backed up by heuristic security arguments. This is fundamentally different from the provable security approach taken by modern cryptography, where one requires that a scheme is *proven* secure against a class of resource bounded (e.g. polynomial time) adversaries and not only particular attacks. This situation is very unsatisfying; after all, what is a provably secure cryptosystem good for, if ultimately its security hinges on an ad-hoc side-channel countermeasure? Nonetheless, until recently there was almost no input from the theory community on side-channel countermeasures as it was believed that this is a practical problem, and theory can only be of limited use in this context. Fortunately, recent results indicate that this view was much too pessimistic. In an early influential paper, Micali and Reyzin [35] propose the "physically observable cryptography" framework which adapts the concept of cryptographic *reductions* to the context of side-channel attacks. Only very recently direct constructions of cryptographic schemes were proposed which are provably secure against general classes of side-channel attacks. We discuss one such model (leakage-resilience) in detail in Section 1, and some other models in the related work section 1.

Leakage-Resilient PRFs. A cryptographic primitive is *leakage-resilient* if it remains secure even if the adversary can – with each invocation – learn a bounded amount of arbitrary information about the computation. This notion was introduced in [17], and is formally modelled by allowing the adversary to choose (besides the regular input, if there is any) a leakage function g with bounded range Σ^λ for some leakage parameter λ.[1] After the invocation the adversary gets $g(\tau)$ where τ is all data accessed by the primitive during this invocation (that is, the part of the secret state that was accessed and – if the primitive is probabilistic – any random coins used). One can take a more "fine-grained" view and split one invocation into $t > 1$ sequential steps. Then the adversary is

[1] The basic idea to consider adversaries who can learn any (sufficiently compressing) function $g(S)$ about the secret state S goes back to Maurer's bounded storage model [32, 15, 42]. The bounded retrieval model [14, 8] adapts this to the computational setting.

allowed to learn a bounded amount of information $g_1(\tau_1), \ldots, g_t(\tau_t)$ about every step, where τ_i contains absolutely all information that is accessed in the i-th step.

As a consequence, the physical implementation of a leakage-resilient cryptosystem will remain secure in the presence of any side-channel attack, as long as the information exploited by this attack can be modelled by adaptively chosen leakage functions as just described. A sufficient (but not necessary) condition on the side-channel is to require that (1) the amount of information leaked per invocation (or, in the fine-grained approach, per step) is at most λ bits and (2) "only computation leaks information", which means that parts of the memory which are not accessed during an invocation (or step) will not leak.

Remark 1 (On "Only computation leaks information"). "Only computation leaks information" is an assumption about the physical properties of cryptodevices, and was originally put forward as one the "axioms" in the physically observable cryptography framework of Micali and Reyzin [35]. As just mentioned, devices adhering to this axiom are captured by the model of leakage resilience, but this is only a sufficient condition and by no means necessary. For example, [38] explains why the mathematical model of leakage-resilience also captures certain physical attacks which explicitly violate this axiom, like "cold-boot attacks" [22] or when considering memory that is subject to static leakage.

Limitations of Current Techniques. The only leakage-resilient primitives that were constructed so far *in the standard model* are stream-ciphers [17, 38] and signature schemes [19]. A leakage-resilient public-key encryption scheme has been constructed, but only in the idealised generic group model [27]. A central open problem is this line of research is the construction of pseudorandom functions (PRFs) and permutations (PRPs, or equivalently, block-ciphers). Block-ciphers are the work horses of crypto. Not surprisingly, they are also a favourite target of side-channel cryptanalysts.

In this work we consider the problem of constructing leakage-resilient PRFs and PRPs. The techniques used in the construction of leakage-resilient stream-ciphers and signature schemes crucially rely on *key evolution*. For example, in a stream-cipher the key evolves naturally, while for signatures one can sample a fresh public/secret key pair with each signature query and sign the new key with an old key. Unfortunately it is not clear how to evolve the key of a PRF/PRP. The same difficulty arises with public-key encryption, so the leakage-resilient PKE scheme from [27] does not rely on evolution, but rather on sharing the secret key. The sharing is rerandomized after each invocation. In order to decrypt using the shares of the secret key without actually reconstructing it, one exploits a homomorphic group property. Thus, even aside from the reliance on idealised generic groups [27], this technique is not an option to construct leakage-resilient PRFs/PRPs if we do not want to use inefficient techniques and assumptions (like DDH) that are used in public-key cryptography.

Our PRF Results. As leakage-resilient PRFs seem out of reach with our current techniques, we will consider a relaxed notion of leakage-resilience, where the

leakage function is not adaptively chosen by the adversary before each invocation, but is fixed. This notion still captures all side-channel attacks where the adversary will always measure (almost) the same leakage if she performs exactly the same computation. This for example captures timing and to some extent power-analysis attacks[2], but not probing attacks (where different wires can be probed on different invocations on the same input.) We construct a PRF which is secure under this relaxed notion from any standard PRF. The construction, as illustrated in Figure 1, can be seen as a hybrid of the GGM construction [21] (which constructs a PRF from any PRG) and the leakage-resilient stream cipher from [38].

Related Work. The idea to only consider non-adaptive leakage functions and that this could be useful in the context of the GGM construction goes back at least to Micali and Reyzin [35].[3] A similar point for a particular leakage function (power analysis) was made by Kocher [29]. The idea to consider leakage-resilience but to fix the leakage function is due to Standaert et al. [41]. They suggest that the GGM construction is secure in this setting if the PRG is modelled as a uniformly random function and the leakage function is fixed.[4]

Side-Channel Attacks on Feistel. A pseudorandom permutation (PRP) $\mathsf{F} : \Sigma^k \times \Sigma^n \rightarrow \Sigma^n$ is defined like a PRF, except that one requires that for every key $K \in \Sigma^k$, $\mathsf{F}(K, .)$ is a permutation. A super PRP (sPRP) satisfies a stronger notion where the adversary can also make inverse queries. The additional structural properties of permutations are often useful as they allow for better efficiency and/or security. Block-ciphers, which are strong PRPs, are the "work horses" of cryptography and a favourite target of side-channel cryptanalysts.

PRPs seem to be much more complicated objects than PRFs, but in a classical paper, Luby and Rackoff [31] prove that a simple 3 (resp. 4) round Feistel

[2] If the power-analysis just leaks the number of wires set to 1, then this is captured, but if the power-analysis leaks the number of wires that "switch" from 0 to 1, then this is no longer possible.

[3] From [35]: *Our definitions allow for repeated computation to leak new information each time. However, the case can be made (e.g., due to proper hardware design) that some devices computing a given function f may leak the same information whenever f is evaluated at the same input x. This is actually implied by making the leakage function deterministic and independent of the adversary measurement. Fixed-leakage physically observable cryptography promises to be a very useful restriction of our general model (e.g., because, for memory efficiency, crucial cryptographic quantities are often reconstructed from small seeds, such as in the classical pseudorandom function of [21]).*

[4] The model considered is basically the random oracle model, but it is conceptually used in a different way. In the RO model, a uniformly random function is accessible to all parties, and security proofs only exploit the fact that a random oracle allows to efficiently access an exponential amount of true randomness. In contrast, in [41] the security proof exploits the fact that the adversarial leakage functions cannot *query* the random oracle.

network (cf. Definition 6) instantiated with PRFs, is a PRP (resp. super PRP). More recently, [7] prove that a six round Feistel network instantiated with random functions is *indifferentiable* [34] from a uniformly random permutation. These results suggest that a Feistel network with some small constant number of rounds instantiated with *leakage-resilient* PRFs, would yield a *leakage-resilient* PRP.

Unfortunately, strong notions as indifferentiability do not guarantee the security of reductions in the presence of leakage. In particular, we show very simple side-channel attacks against Feistel networks where the round functions can be *arbitrary*, and the only leakage is some (simple) function $g(.)$ of the inputs to the round functions. We identify a simple property of leakage-functions function $g(.)$ – which we call "reconstructible" (cf. Definition 7) – that is sufficient for our attack to work. This property is shared by many simple and natural leakage functions (like the Hamming weight or the identity function with very high noise). Thus our attacks are quite practical. We explain these attacks in detail in Section 3 (which is self contained and can be read independently of the rest of this paper), here only giving the brief summary. We show that getting leakage from any reconstructible leakage function $g(.)$ is sufficient to allow the side-channel attacker to invert the Feistel network on any input using a number of *forward* queries which is exponential in the number of rounds (and, thus, in polynomial time for any fixed constant number of rounds). This breaks the security of any fixed-round Feistel network as a PRP.

For readers familiar with the notion of Indifferentiability [34, 6], it might seem that our attacks contradict the beautiful result of Coron et al. [7] showing that a six round Feistel network with random functions is indifferentiable from a random permutation. The reason this is not a contradiction is that the indifferentiability simulator $\mathcal{S}$ is allowed to make arbitrary *additional* forward/backward queries to the random permutation when trying to "fake" the six random round functions, as opposed to the queries made by the distinguisher (which the simulator does not even see). For example, for our attack making only forward queries, the simulator will be "smart enough" to figure out the backward query we are "computing" using our forward queries, and will make such a query in advance to avoid any inconsistencies. Translated to the setting of leakage, the indifferentiability framework will imply the following much weaker notion of security than the one we are aiming for: after making q block-cipher queries and observing the leakage, all but specially chosen $poly(q)$ input/outputs of the block cipher will "look random". In contrast, we will ensure that *every* un-queried input/output pair will "look random".

We also mention that [12] defined a notion of "honest but curious indifferentiability". As observed by [12, 7] this notion is *incomparable* to standard indifferentiability. On one hand, it is stronger because the simulator $\mathcal{S}$ is not allowed to make any queries to $\mathbf{P}$ or $\mathbf{P}^{-1}$ (but only sees the queries made by the distinguisher). But it is also weaker, as the distinguisher is not allowed to query intermediate round functions, but only the entire Feistel network (or its simulation) together with all the inputs/outputs of the internal round functions.

This notion is much closer to the setting of side-channel attacks, except with side-channels we allow a much richer class of leakage functions (e.g., those that depend on the key). In fact, the side-channel attacks we propose generalize (and strengthen) a lower bound from [12] which basically corresponds to our attack for the special case where the leakage contains the entire inputs to the round functions.

Leakage-Resilient PRPs. In light of the results discussed in the previous section, the best we can hope for is that an r-round Feistel network Ψ_r, instantiated with leakage-resilient PRFs, is secure against adversaries who make at most an exponential (in r) number of queries. In Section 4 we show (again using techniques from [12]) that this is indeed the case: the r-round Feistel network is a secure leakage-resilient super PRP as long as the number of queries is bounded by $q \leq 1.38^{r/2-1}$.

We notice that the leakage-resilient sPRP, as just described, is secure in an attack scenario where the adversary with every query to Ψ_r gets to see all the inputs[5] to the r round functions and *also* leakage from every round function (as computed by any leakage function for which the underlying leakage-resilient PRF is secure). Also, the reductions works for other notions of leakage-resilience, in particular for the original notion of leakage-resilience where the leakage-function is chosen adaptively. Thus, although our current PRF constructions only give us "non-adaptive-leakage" sPRPs, future advances in leakage-resilient PRFs would immediately translate to stronger leakage-resilient sPRPs.

In contrast, when proposing attacks, we want to consider a setting where the adversary is as limited as possible. As explained in the previous section, the side-channel attacks we propose against Feistel require a very limited setting where the only leakage the adversary gets is some simple function (e.g. Hamming weight) of the inputs to the round functions. The attack works no matter what the round functions are, they can be leakage-proof PRFs or even uniformly random functions.

More Related Work. We shortly discuss some work on *provable* side-channel security not already covered in the introduction. The more practical work on this topic is too extensive to cover here, [37] gives an overview of this research.

Private Circuits. Ishai et al. [25, 24] consider a model model where the adversary can choose some wires in the circuit on which the cryptographic algorithm is run, and then learns the values carried by those wires during the computation (This can be seen as a generalisation of exposure resilient cryptography [13], where the adversary was restricted to learn some bits of the *input*.) They were the first to *prove* how to implement *any* algorithm secure against an interesting side-channel, i.e. probing attacks. This work uses techniques from general multiparty

[5] The outputs of the round functions can be computed from the input: the output of the ith round functions is the XOR of the inputs of rounds $i-1$ and $i+1$.

computation (MPC).[6] Recently Faust et al. [18] extended this result to significantly more general classes of leakage, in particular, they give a construction (also based on general MPC) which remains secure given leakage computed by any function from a low complexity class like AC_0. The main drawback of those constructions is that the amount of leakage that can be tolerated is very small: to tolerate t bits leakage, the circuits must be blown up by a factor of at least t. Moreover the construction from [18] requires (albeit very simple) completely leakage proof components.

(Continuous) Memory Attacks. A cryptographic scheme is secure against memory attacks, if it remains secure even if a bounded amount of information about the secret key is given to the adversary. In this model [1, 36, 4] construct public-key encryption schemes and [26, 2] construct signature schemes, identification schemes and key exchange protocols.[7] Unlike leakage-resilience, here the leakage function gets the *entire* secret state as input, and not only what was accessed. On the downside – unlike leakage-resilience or private circuits – memory attacks are a "one-shot" game where the total amount of leakage cannot be larger than the length of the secret key. Very recently [10, 5] extended the model of memory attacks to the continuous setting. In their model the secret key gets periodically updated (using local randomness and without changing the public key), and a bounded amount about of information about the secret key can leak in-between every two updates. The update phases can also leak, but only a logarithmic amount. In this model, [10] construct identification, signature and authenticated key agreement schemes, [5] construct signatures and PKE.

Auxiliary Input. [11] introduce the notion of security against auxiliary input, where one requires the scheme to be secure even if the adversary is given some leakage $g(K)$ about the secret key as long as $g(.)$ is uninvertible. That is, K cannot be inverted given $g(K)$ but with very small probability. In this model private-key [11] and public-key [9] encryption schemes have been constructed.

Notation & Basic Definitions.

- Σ^t denotes $\{0,1\}^t$, i.e. all bitstring of length t. $\Sigma^{\leq t} \stackrel{\mathsf{def}}{=} \bigcup_{i=0}^{t} \Sigma^t$ denotes all bitstrings of length at most t, including the empty string ε.

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

[7] Let us mention that PRFs and PRPs (i.e. the primitives considered in this paper) that are secure against memory attacks do not even exist. E.g. we can trivially distinguish $F(K, X)$ (here K is the key and X is any fixed input to the PRF $F(.,.)$) from uniform with advantage $1 - 2^{-\lambda}$ given as leakage the first λ bits of $F(K, X)$.

- $[a, b]$ denotes the interval $\{a, a+1, \ldots, b\}$, $[b]$ is short for $[1, b]$.
- Sequential composition of functions is denoted with $g \circ f(x) \stackrel{\mathsf{def}}{=} g(f(x))$.
- Concatenation of two strings x, y is denoted $x \| y$, or, if no confusion is possible, simply xy.
- $w_H(x)$ denotes the number of 1's (i.e. Hamming weight) in x.
- $\mathbf{R}_{m,n}$ denotes a uniformly random function $\Sigma^m \to \Sigma^v$, $\mathbf{P}_n$ a uniformly random permutation over Σ^n.
- For $X \in \Sigma^n$ we denote with $X_{|i}$ the i bit prefix of X.
- $\mathsf{pre}(X) = \bigcup_{i=0}^{n} X_{|i}$ denotes the set of all prefixes of X, including the empty string $\varepsilon = X_{|0}$ and the entire $X = X_{|n}$.
- We sometimes write X^q to denote a sequence $X_1, \ldots, X_q$ of values.
- For a set $\mathcal{X}$, $X \xleftarrow{*} \mathcal{X}$ denotes that X is assigned a value sampled uniformly at random from $\mathcal{X}$.
- We denote with $\delta^{\mathsf{D}}(X; Y)$ the advantage of a circuit D in distinguishing the random variables X, Y, i.e.: $\delta^{\mathsf{D}}(X; Y) \stackrel{\mathsf{def}}{=} |\Pr[\mathsf{D}(X) = 1] - \Pr[\mathsf{D}(Y) = 1]|$. 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.

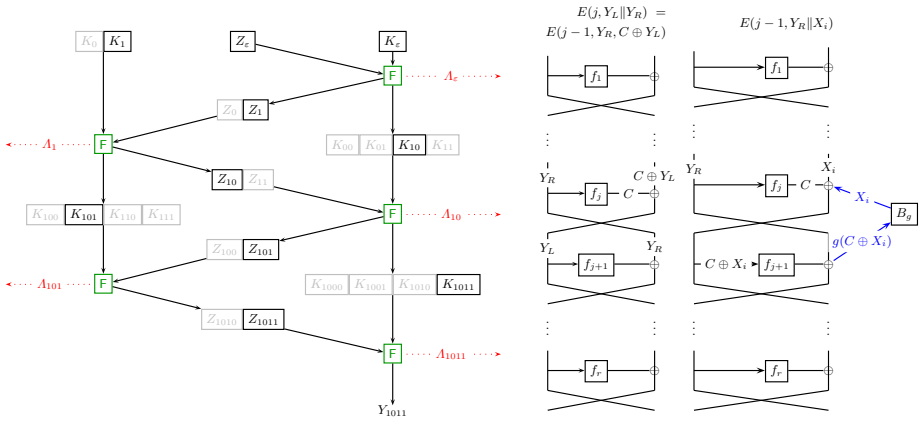

Fig. 1. Left: Illustration of the NALR-secure PRF $\Gamma^{\mathsf{F},m} : \Sigma^{3k+n} \times \Sigma^m \to \Sigma^{4k+2n}$ (here shown for $m = 4$ and input $1011 \in \Sigma^m$) from any standard (weak) PRF $\mathsf{F} : \Sigma^k \times \Sigma^n \to \Sigma^{4k+2n}$. We consider adversaries who with each such query X can get leakage Λ_I for every $I \in \mathsf{pre}(X)$ which is defined as $\Lambda_I \stackrel{\mathsf{def}}{=} g(K_I, Z_I, I)$, where g is any function of bounded size s and range λ. And moreover all the $Z_I, I \in \mathsf{pre}(X)$.
Right: Illustration of the second Claim from the proof of Theorem 2.

2 Leakage-Resilient PRFs

Figure 1 (left) illustrates our construction of a PRF $\mathsf{F} : \Sigma^k \times \Sigma^m \to \Sigma^n$ for which we will show that it satisfies a relaxed notion of leakage-resilience where the leakage function is a priori fixed (and not adaptively by the adversary with every query). Recall the standard definitions of (weak) PRFs.

Definition 1 (PRF/weak PRF). $\mathsf{F} : \Sigma^\kappa \times \Sigma^m \to \Sigma^n$ *is an* $(\epsilon_{\mathsf{prf}}, s_{\mathsf{prf}}, q_{\mathsf{prf}})$*-secure pseudorandom function (PRF) if no adversary of size* s_{prf} *can distinguish* F *(instantiated with a random key) from a uniformly random function, i.e. for any* $\mathcal{A}$ *of size* s_{prf} *making* q_{prf} *oracle queries we have*

$$\Pr_K[\mathcal{A}^{\mathsf{F}(K,.)} \to 1] - \Pr_{\mathbf{R}_{m,n}}[\mathcal{A}^{\mathbf{R}_{m,n}(.)} \to 1] \leq \epsilon_{\mathsf{prf}}$$

F *as above is a* $(\epsilon_{\mathsf{prf}}, s_{\mathsf{prf}}, q_{\mathsf{prf}})$*-secure* <u>weak</u> *PRF if the above only holds for randomly (and not adversarially) chosen inputs, i.e. for* $K \xleftarrow{*} \Sigma^\kappa$ *and*

$$\textit{for } i = 1, \ldots, q_{\mathsf{prf}}: \quad X_i \xleftarrow{*} \Sigma^m \qquad Y_i \leftarrow \mathsf{F}(K, X_i) \qquad R_i \leftarrow \mathbf{R}_{m,n}(X_i)$$

$$\textit{we have} \qquad \Pr[\mathcal{A}(X^{q_{\mathsf{prf}}}, Y^{q_{\mathsf{prf}}}) = 1] - \Pr[\mathcal{A}(X^{q_{\mathsf{prf}}}, R^{q_{\mathsf{prf}}}) = 1] \leq \epsilon_{\mathsf{prf}}$$

Definition 2 below specifies what we mean by a PRF F being leakage-resilient w.r.t. to a class of leakage functions $\mathcal{L}$. Informally, we consider an adversary $\mathcal{A}$ with access to two oracles. Initially, we sample a key $K \xleftarrow{*} \Sigma^k$. The first oracle then takes as input some $X \in \Sigma^m$ and outputs the output of the PRF $Y \leftarrow \mathsf{F}(K, X)$ on this input and the leakage $\Lambda \leftarrow g(K, X)$ (where g is any function from the class $\mathcal{L}$). The second oracle is either a uniformly random function $\mathbf{R}_{m,n}$, or the PRF $\mathsf{F}(K, .)$ (using the same key as K the first oracle.). We require that no efficient $\mathcal{A}$ can distinguish these two cases. Of course we have to require that $\mathcal{A}$ never queries the two oracles on the same input X, as otherwise distinguishing becomes trivial.

The practical implication of this definition is as follows. Consider an adversary who can launch a side-channel attack against $\mathsf{F}(K, .)$, where for every query $\mathsf{F}(K, X)$ made she can measure some leakage $\Lambda(K, X)$. If F is $\mathcal{L}$ resilient, and the leakage $\Lambda(K, X)$ can be modelled as $\Lambda(K, X) = g(K, X)$ for some $g \in \mathcal{L}$, then for all inputs X' on which $\mathsf{F}(K, .)$ has not yet been queried, the output $\mathsf{F}(K, X')$ will be indistinguishable from random.

Definition 2 ($\mathcal{L}$-resilient PRF/PRP/sPRP). $\mathsf{F} : \Sigma^\kappa \times \Sigma^m \to \Sigma^n$ *is a* $(\epsilon_{\mathsf{prf}}, s_{\mathsf{prf}}, q_{\mathsf{prf}})$*-secure* $\mathcal{L}$*-resilient pseudorandom function if for every adversary* $\mathcal{A}$ *of size* s_{prf} *and every* $g \in \mathcal{L}$

$$\Pr_K[\mathcal{A}^{\mathsf{F}^g(K,.),\mathsf{F}(K,.)} \to 1] - \Pr_{K,\mathbf{R}_{m,n}}[\mathcal{A}^{\mathsf{F}^g(K,.),\mathbf{R}_{m,n}(.)} \to 1] \leq \epsilon_{\mathsf{prf}} \qquad (1)$$

Here $\mathcal{A}$ *can make a total of* q_{prf} *queries (arbitrarily scheduled) to his two oracles, but the queries to the first and second oracle must be disjoint. The first oracle* $\mathsf{F}^g(K, .)$ *takes as input* $X \in \Sigma^m$ *and outputs* $\mathsf{F}(K, X), g(K, X)$*.*

$\mathcal{L}$*-resilient pseudorandom permutations (PRP) are defined similarly, except that now for every* K*,* $\mathsf{F}(K, .)$ *has to be a permutation and the random function* $\mathbf{R}_{m,n}$ *in eq.(1) is replaced with a random permutation* $\mathbf{P}_m$*. A* $\mathcal{L}$*-resilient* **super** *PRP (sPRP) is defined the same way, except that now we additionally allow the adversary to make inverse queries. Here* $\mathcal{A}$ *is also not allowed to make an inverse (forward) query* Y *to one oracle, if* Y *has been received as output to a forward (inverse) query from the other oracle.*

Definition 3 (NARL security). *We say that a PRF* F *(same for PRP,sPRP) is* non-adaptive leakage-resilient *if the computation of* $\mathsf{F}(K, X)$ *can be split into* $t \geq 1$ *steps, and* F *is* $\mathcal{L}$*-resilient w.r.t. to a class* $\mathcal{L}$ *which can leak, for every of the* t *steps, arbitrary* λ *bits of information about all the data that is accessed in this step.*

Below we define our construction $\Gamma^{\mathsf{F},m}$ of a function as illustrated is Figure 1 for which we will prove that it is NARL secure if instantiated with any standard weak PRF F. This construction can be seen as a hybrid of the GGM construction [21] and the leakage-resilient stream-cipher from [38].

Definition 4 (Construction Γ^{F}). *For a functions* $\mathsf{F} : \Sigma^k \times \Sigma^n \to \Sigma^{4k+2n}$*, we denote with* Γ^{F} *a function* $\Sigma^{3k+n} \times \Sigma^m \to \Sigma^{4k+2n}$ *defined as follows (cf. Figure 1). The secret key* K *consists of the four values* $Z_\varepsilon \in \Sigma^n, K_\varepsilon, K_0, K_1 \in \Sigma^k$*. The output on input* $X \in \Sigma^m$ *is* $Y_X \leftarrow \mathsf{F}(K_X, Z_X)$ *where* Z_I, K_I *for* $I \in \mathsf{pre}(X)$ *are recursively defined as*

$$(Z_{I0}, Z_{I1}, K_{I00}, K_{I01}, K_{I10}, K_{I11}) \leftarrow \mathsf{F}(K_I, Z_I)$$

Figure 1 illustrates this construction for $m = 4$ *on input* $X = 1011$*.*

Theorem 1 below states that Γ^{F} is NARL secure. Or more precisely, $\mathcal{L}$-resilient, where $\mathcal{L}$ contains all functions that leak λ bits of arbitrary information about every invocation of F. How large λ can be depends on the security of F. Roughly, if F cannot be broken with advantage 2^{-w}, then we can leak $\lambda = w/6$ bits with each of the n invocations of F. (and thus $nw/6$ bits in total.)

NARL security requires that the leakage in each of the $m + 1$ steps (i.e.the invocations of the underlying F) can depend on absolutely all data that is accessed during this step. For step i ($0 \leq i \leq m$) this means Z_I, K_I, where $I = X_{|i}$ is the i bit prefix of the input X, but also the last two bits of I itself, as this bits specify which part of the state[8] must be accessed in this step. We will even give the entire I as input to the leakage function.

Theorem 1. *If* F *is a weak PRF,* $\Gamma^{\mathsf{F},m}$ *is a NARL super-PRP, where each invocation of the underlying* F *is considered a step as in Def. 3. If the PRF cannot be distinguished from random with advantage more than* ϵ_{prf}*, then we can tolerate leakage of* $\lambda = \log(\epsilon_{\mathsf{prf}}^{-1})/6$ *bits per step. The precise quantitative statement is given below.*

Assume $\mathsf{F} : \Sigma^k \times \Sigma^n \to \Sigma^{4k+2n}$ *is a* $(\epsilon_{\mathsf{prf}}, s_{\mathsf{prf}}, n/\epsilon_{\mathsf{prf}}^2)$ *secure weak PRF (where* $\epsilon_{\mathsf{prf}} \geq n \cdot 2^{-n/3}$ *and* $n \geq 20$*) and let* $\lambda = \log(\epsilon_{\mathsf{prf}}^{-1})/6$*. Then* $\Gamma^{\mathsf{F},m} : \Sigma^{3k+n} \times \Sigma^m \to \Sigma^{4k+2n}$ *is a* $(\epsilon'_{\mathsf{prf}}, s'_{\mathsf{prf}}, q'_{\mathsf{prf}})$ *secure* $\mathcal{L}_{s,\lambda}$*-resilient PRF for any* q'_{prf} *and*

$$s'_{\mathsf{prf}} = s_{\mathsf{prf}}\epsilon_{\mathsf{prf}}^2/2^{\lambda+2}(n+k)^3 - s \cdot m \cdot q'_{\mathsf{prf}} \qquad \epsilon'_{\mathsf{prf}} = 8 \cdot q'^2_{\mathsf{prf}} \cdot m \cdot \epsilon_{\mathsf{prf}}^{1/12}$$

[8] Let I_d denote I where the last d bits deleted. Then before step I the state is $Z_{I_10}Z_{I_11}, K_{I_200}, K_{I_201}, K_{I_210}, K_{I_211}$.

where the class $\mathcal{L}_{s,\lambda}$ contains all functions $\mathcal{L}_g$ indexed by a function $g : \Sigma^{k+n+m} \to \Sigma^\lambda$ of size at most s defined as (with K_I, Z_I as in Definition 4)

$$\mathcal{L}_g(K, X) = \{\Lambda_I, Z_I \quad : \quad I \in \mathsf{pre}(X)\} \qquad \Lambda_I \stackrel{\mathsf{def}}{=} g(K_I, Z_I, I)$$

Recall that a random variable X has min-entropy k, denoted $\mathsf{H}_\infty(X) = k$, if $\Pr[X = x] \leq 2^{-k}$ for any x in the support. In the proof, we will extensively use a computational version of this notion called HILL-pseudoentropy [23, 3].

Definition 5 (HILL-pseudoentropy[23, 3]). *We say X has* HILL *pseudoentropy k, denoted by $\mathsf{H}^{\mathsf{HILL}}_{\epsilon,s}(X) \geq k$, if there exists a distribution Y with min-entropy $\mathsf{H}_\infty(Y) = k$ where $\delta_s(X; Y) \leq \epsilon$.*

Proof (of Theorem 1). Our construction $I^{\mathsf{F},m}$ is inspired by the construction of the leakage-resilient stream-cipher from [38], and also the proof is very similar. We will use several technical results from [38, 17] which for space reasons are moved to Appendix A.

It will be convenient to consider an adversary which is stronger than what is actually required in the proof. We consider an adversary $\mathcal{A}$ who can adaptively "explore" the tree structure underlying the $I^{\mathsf{F},m}$ construction. This is modeled by giving her access to two oracles $\mathcal{O}_K(.)$ and $\mathcal{O}^b_K(.)$. These are initialised with a random key K (as used in $I^{\mathsf{F},m}$), a random bit b and a uniformly random function $\mathbf{R}$. The $\mathcal{O}^b_K$ oracle takes inputs from Σ^m and outputs either random outputs (if $b = 1$) or the output of $I^{\mathsf{F},m}$ (if $b = 0$). The $\mathcal{O}_K$ oracle allows to "explore" the tree structure of $I^{\mathsf{F},m}$.

$$\mathcal{O}_K(I) \to \begin{cases} Z_{I0}, Z_{I1}, \Lambda_I & \text{if } I \in \Sigma^{\leq m-1} \\ Y_I, \Lambda_I & \text{if } I \in \Sigma^m \end{cases} \qquad \mathcal{O}^b_K(I) \to \begin{cases} Y_I & \text{if } b = 0 \\ \mathbf{R}(I) & \text{if } b = 1 \end{cases}$$

We put the additional restriction on the order in which queries can be made: $\mathcal{A}$ can only make a query I to $\mathcal{O}_K$ or $\mathcal{O}^b_K$, if the $|I| - 1$ bit prefix of I has already been queried (the first query can only be ε). Wlog. we assume that $\mathcal{A}$ never makes the same query twice. $\mathcal{A}$ can never make the same query $I \in \Sigma^m$ to both oracles (which would trivially allow to distinguish the cases $b = 0$ and $b = 1$.)

A q'_{prf}-query adversary $\mathcal{A}'$ who breaks the $\mathcal{L}_{s,\lambda}$ security of $I^{\mathsf{F},m}$ with advantage ϵ can be turned into an adversary $\mathcal{A}$ of almost the same size who has advantage ϵ in distinguishing the cases $b = 0$ and $b = 1$ in the experiment just described: A query X to $I^{\mathsf{F},m}(X)$ can be simulated by making the queries $\mathsf{pre}(X)$ to O_K. A query X to the second oracle can be simulated the same way, except that the query X is forwarded to $O^b_K(.)$. This $\mathcal{A}$ makes at most $(m-1)q'_{\mathsf{prf}}$ and q'_{prf} queries to the first and second oracle respectively. Thus it remains to upper bound

$$\Pr_K[\mathcal{A}^{\mathcal{O}_K(.),\mathcal{O}^0_K(.)} \to 1 - \Pr_{K,\mathbf{R}}[\mathcal{A}^{\mathcal{O}_K(.),\mathcal{O}^1_K(.)} \to 1]$$

This means we must show that the outputs of the oracle $\mathcal{O}^0_K : I \to \mathsf{F}(K_I, Z_I)$ are pseudorandom even given access to $\mathcal{O}_K$, and thus cannot be distinguished

from the uniformly random outputs of $\mathcal{O}_K^1 : I \to \mathbf{R}(I)$. Let $\mathtt{view}_i$ denote the view of $\mathcal{A}$ after the ith query, the initial view is $\mathtt{view}_0 = \{Z_\varepsilon\}$. We say that $I \in \Sigma^{\leq m}$ is a "potential query" if $\mathcal{A}$ did not yet make the query I but all the its prefixes $\mathsf{pre}(I) \setminus I$. The following facts hold (with high probability) after the ith query and for any potential query I. (We ignore the precise bounds on HILL pseudoentropy, writing only $\mathsf{H}^{\mathsf{HILL}}$ to denote $\mathsf{H}^{\mathsf{HILL}}_{\epsilon,s}$ for "small" ϵ and "large" s.)

1. K_I and Z_I are independent given the view $\mathtt{view}_i$ of $\mathcal{A}$.
2. $\mathsf{H}^{\mathsf{HILL}}(K_I|\mathtt{view}_i) = k - 2\lambda$ and $\mathsf{H}^{\mathsf{HILL}}(Z_I|\mathtt{view}_i \setminus Z_I) = k - 2\lambda$.
3. If K_I, Z_I satisfy fact 1 & 2 then
   (a) $\mathsf{F}(K_I, Z_I)$ is pseudorandom given $\mathtt{view}_i$.
   (b) $\mathsf{H}^{\mathsf{HILL}}(\mathsf{F}(K_I, Z_I)|\Lambda_I, \mathtt{view}_i) = |\mathsf{F}(K_I, Z_I)| - 2\lambda$.

Note that fact 3.(a) implies that a query I to $\mathcal{O}_K^0$ will result in a pseudorandom value $\mathsf{F}(K_I, Z_I)$. As just described, this establishes the theorem. The lemmata below are given in Appendix A.

Fact 1 follows from Lemma 3 (originally from [16], also given as Lemma 5 in [38]). The only reason we add $Z_{I0}Z_{I1}$ to the output of $\mathcal{O}_K(I)$ (and not only the leakage Λ_I) is so we can apply this lemma.

Fact 3.(a) follows from Fact 2 using Lemmata 4 and 5, which state that the output $\mathsf{F}(K, Z)$ of a weak PRF is pseudorandom as long as K and Z are independent and have sufficiently high pseudoentropy.

Fact 3.(b) follows from Fact 3.(a) and Theorem 2 from [17] (or, independently [40]), which states that a pseudorandom value like $\mathsf{F}(K, Z)$ has high pseudoentropy, even if a bounded amount of information about the seed (in our case K, Z) is leaked. The precise quantitative statement of Fact 3.(b) is given as Lemma 6 (which is Lemma 6 from [38]).

Finally, Fact 2 holds by induction over the queries that $\mathcal{A}$ makes using Fact 3.(b). To see this, note that Fact 2 holds initially for $i = 0$ as $K_0, K_1, K_\varepsilon, Z_\varepsilon$ are independently and uniformly sampled. Now assume it holds after the ith query, and $\mathcal{A}$ makes the query I (where $|I| < m$), then by Fact 3.(b) the newly computed values $Z_{I0}, Z_{I1}, K_{I00}, \ldots, K_{I11} \leftarrow \mathsf{F}(K_I, Z_I)$ will also satisfy Fact 2.

So far we have only established the qualitative statement that $I^{\mathsf{F},m}$ is a NARL secure PRP but said nothing about the exact security as claimed in the proof. The HILL-pseudoentropy in the facts above must be quantified, e.g. in fact 2. above $\mathsf{H}^{\mathsf{HILL}}(K_I|\mathtt{view}_i) = k - 2\lambda$ can be expressed as $\mathsf{H}^{\mathsf{HILL}}_{\epsilon,s}(K_I|\mathtt{view}_i) = k - 2\lambda$ for some ϵ, s. One then has to do some bookkeeping bounding how this parameters get worse (i.e. how s decreases and ϵ increases) during the run of the experiment. As this is not very instructive we omit this calculations. The bounds we get here are exactly the same bounds that are proven for the leakage-resilient stream-cipher in [38] (when using the same F and the number of invocations to the underlying F is the same). In fact, minor adaptions of the proof from [38] give us the claimed bounds. The only difference is that the advantage ϵ'_{prf} in this paper is a factor q'_{prf} larger, the reason is that our $\mathcal{A}$ can make q'_{prf} "challenge queries" to the $\mathcal{O}_K^b$ oracle, whereas in [38] only one challenge query is considered. □

3 Side-Channel Attacks on Feistel

In this section we put forward generic side-channel attacks on Feistel networks. As Feistel networks (and minor variations thereof) are the only generic constructions of PRPs from PRFs known, this indicates that constructing leakage-resilient PRPs from leakage-resilient PRFs might be significantly harder than constructing PRPs from PRFs in the normal (non-leakage) setting. Below we first define the Feistel network.

Definition 6 (Feistel, μ). *For a function $f : \Sigma^n \rightarrow \Sigma^n$, we denote with $\Psi[f]$ the permutation over Σ^{2n} defined as $\Psi[f](x_L, x_R) \stackrel{\mathsf{def}}{=} f(x_L) \oplus x_R \| x_L$. $\Psi[f_1, \ldots, f_r]$ denotes $\Psi[f_r] \circ \ldots \circ \Psi[f_1]$.*

We define μ as $(R_0, \ldots, R_{r+1}) \stackrel{\mathsf{def}}{=} \mu(\Psi[f_1, \ldots, f_r], R_1 \| R_0)$ where for $i \geq 1$: $R_i \stackrel{\mathsf{def}}{=} R_{i-1} \oplus f_{i-1}(R_{i-1})$, so R_i is the input to the ith round function on input $X = R_1 \| R_0$.

In a classical paper, Luby and Rackoff prove that the advantage of any q-query distinguisher in distinguishing $\Psi_3 \stackrel{\mathsf{def}}{=} \Psi[f_1, \ldots, f_3]$ from a uniformly random permutation over Σ^{2n} is upper bounded by[9] $q^2/2^n$ if the $f_i : \Sigma^n \rightarrow \Sigma^n$ are uniformly random functions.[10] This in particular implies that no adversary who can query Ψ_3 in forward direction can invert Ψ_3 on a random $Y \in \Sigma^{2n}$, unless she makes $q = \Theta(2^{n/2})$ queries.

We consider a setting where the adversary not only can make queries to some Feistel network $\Psi_r \stackrel{\mathsf{def}}{=} \Psi[f_1, \ldots, f_r]$, but with each query X, besides the output $Y \leftarrow \Psi_r(X)$, also gets some "leakage" about the intermediate values.

We will consider different leakage functions $g : \Sigma^n \rightarrow \Sigma^*$, our attack will work for any functions which allow "reconstruction" as defined below

Definition 7 (reconstructible). *A function $g : \Sigma^n \rightarrow \Sigma^*$ is (k, δ) reconstructible, if there exists an efficient algorithm B_g such that $\Pr[C' = C] \geq \delta$ in the experiment below:*

1. *Sample a random challenge $C \stackrel{*}{\leftarrow} \Sigma^n$.*
2. *B_g can adaptively make k queries $X_1, \ldots, X_k$ to an oracle which on input X_i outputs $g(C \oplus X_i)$.*
3. *B_g outputs C'.*

If g is probabilistic, then it is (k, δ) reconstructible if there exits a single B_g such that the expectation (over the randomness of g) of the probability $\mathsf{E}[\Pr[C' = C]]$ is at least δ. Two examples of reconstructible functions are given below.

[9] With one round more, the same result holds even if the distinguisher is allowed to make inversion queries.

[10] This then implies that $\Psi[f_1, \ldots, f_3]$ is a pseudorandom permutation if the f_i's are pseudorandom functions. In fact, Luby-Rackoff proved this latter result directly, but as advocated e.g. in [33], the detour via uniformly random objects is cleaner and easier.

Hamming-weight: The Hamming-weight function $g : \Sigma^n \to \Sigma^{\lceil \log n \rceil}$, $g(X) \stackrel{\mathsf{def}}{=} w_H(X)$ is $(n, 1)$ reconstructible: For $i \in [n]$ let B ask for $\Lambda_i = g(X \oplus e_i)$, where $e_i = 0^{i-1}10^{n-i-1}$ for $i = 1, \ldots, n$. Note that Λ_i can only take two values, $w_H(X) - 1$ or $w_H(X) + 1$, which is the case if the ith bit of X is 1 and 0 respectively.[11]

Noise: For some $\gamma > 0$ consider the probabilistic function $g_\gamma : \Sigma^n \to \Sigma^n$ which flips every bit of its input with probability $1/2 - \gamma$ (and each bit of every input is flipped independently.) For any k, g_γ is $(k, 1 - n \cdot e^{-2 \cdot k \cdot \gamma^2})$ reconstructible: B_{g_γ} uses any sequence $X_1, \ldots, X_k$ of distinct inputs, and guesses that the ith bit of C is 0 iff the majority of the ith bits in $g_\gamma(C \oplus X_1), \ldots, g_\gamma(C \oplus X_k)$ is 0. By the Chernoff bound, the probability that the ith bit is guessed wrong is at most $e^{-2 \cdot k \cdot \gamma^2}$, taking the union bound over all n bits we get the bound as claimed.

Theorem 2. *For some $r \geq 3$ and any $f_1, \ldots, f_r : \Sigma^n \to \Sigma^n$, consider the r round Feistel network $\Psi_r = \Psi[f_1, \ldots, f_r]$ and some leakage function $g : \Sigma^n \to \Sigma^*$ which is (k, δ) reconstructible. Then there exists an attacker $\mathcal{A}$ which can invert Ψ_r on any value Y with probability $\delta^{(k+1)^{r-2}}$, where $\mathcal{A}$ makes $4(k+1)^{r-2}$ forward queries to Ψ_r, and with each query X learns the output $\Psi_r(X)$ and leakage $g(R_1), \ldots, g(R_{r-1})$ about the inputs to the round functions $(R_0, \ldots, R_{r+1}) \leftarrow \mu(\Psi_r, X)$. The running time of $\mathcal{A}$ is $O((k+1)^{r-3}|B_g|)$ where $|B_g|$ is the running time of B_g as in Definition 7.*

In the theorem we only consider the case $r \geq 3$, for $r = 0, 1$ or 2 one can trivially invert with probability 1 making $0, 1$ or 4 forward queries respectively. This theorem generalizes Theorem 3.1 from [12], who consider the case where the adversary gets all the R_i's. (or equivalently, where g is $(1, 1)$ reconstructible.)

Remark 2. Note that we don't have to leak $g(R_i)$ for $i \in \{0, 1, r, r+1\}$ as for those i the entire R_i is already contained in the input or output. The above theorem can also be proven (with worse bounds: $(k+1)^r$ queries and probability $\delta^{(k+1)^r}$) in a weaker setting where the adversary does not even get to see the output $\Psi_r(X) = R_r \| R_{r+1}$, but instead gets the leakage $g(R_r), g(R_{r+1})$.

Remark 3. The success probability $\delta^{(k+1)^{r-2}}$ drops very fast in k and r. This is not an issue for leakage functions where $\delta = 1$ like Hamming weight. But this also is good enough for noisy leakage, where we get a success probability of $(1 - n \cdot e^{-2 \cdot k \cdot \gamma^2})^{(k+1)^{r-2}} \geq (1 - n \cdot e^{-2 \cdot k \cdot \gamma^2} \cdot (k+1)^{r-2})$ which approaches 1 exponentially fast in k.

Proof (of Theorem 2). The proof by induction on the number of rounds r. For $j \in [r]$ let $\Psi_j \stackrel{\mathsf{def}}{=} \Psi[f_1, \ldots, f_j]$ denote the first j rounds of Ψ_r. For any $j, 1 \leq j \leq r$, we let $E(j, Y_j) \stackrel{\mathsf{def}}{=} \Psi_j^{-1}(Y_j)$, that is, the input Z such that the intermediate value after j rounds in the computation $\Psi_r(Z)$ is Y_j. It will be convenient to define

[11] If all Λ_i are the same then $X = 1^n$ or 0^n, which is the case can be deduced from Λ_1 (which is $n - 1$ or 1 in those cases).

$E'(j, Y_j) = \{Z, \Psi_r(Z), g(R_1), \ldots, g(R_r)\}$ where $(R_0, \ldots, R_{r+1}) \leftarrow \mu(\Psi_r, Z)$. We show that

Claim. $E'(1, Y_L \| Y_R)$ can be computed (with probability δ) making $k+1$ forward queries to Ψ_r.

Proof (of Claim). As $Z \stackrel{\mathsf{def}}{=} E(1, Y_L \| Y_R)$ is $Y_R \| f_1(Y_R) \oplus Y_L$, to get Z it is sufficient to learn $C \stackrel{\mathsf{def}}{=} f_1(Y_R)$. To get $E'(1, Y_L \| Y_R)$ we then make one more Ψ_r query Z. Let B_g be as in Definition 7, we will use it to reconstruct C as follows: For every query X_i asked by B_g, we make the query $Y_R \| X_i$ to Ψ_r. The answer will contain the leakage $\Lambda_2 = g(C \oplus X_i)$, which is exactly what B_g expects as answer to his query X_i. Thus after k queries we learn C with probability δ. □

Claim. For $j \in [2, r-2]$, $E'(j, Y_L \| Y_R)$ can be computed (with probability δ) making $k+1$ queries to $E'(j-1, .)$.

Proof (of Claim). The proof of this claim is illustrated in Figure 1. The idea is similar as in the previous claim; We will use B_g to reconstruct $C \stackrel{\mathsf{def}}{=} f_j(Y_R)$ (as explained below) and then we get $E'(j, Y_L \| Y_R) = E'(j-1, Y_R \| C \oplus Y_L)$ with one more $E'(j-1, .)$ query.

To reconstruct $C = f_j(Y_R)$, for every query X_i made by B_g, we ask for $E'(j-1, Y_R \| X_i)$ which includes the leakage $\Lambda_{j+1} = g(C \oplus X_i))$ as expected by B_g. Thus after k queries $X_1, \ldots, X_k$, B_g outputs $C = f_j(Y_L)$ with probability δ.

Claim. For $j \in \{r-1, r\}$, $E'(j, Y_L \| Y_R)$ can be computed making 2 queries to $E'(j-1, .)$.

Proof (of Claim). We ask for $E'(j-1, 0^n \| Y_L) = \{Z, \Psi_r(Z), \ldots\}$, here $\Psi_r(Z)$ contains $f_j(Y_L)$ in the clear (it's the left part of $\Psi_r(Z)$ for $j = r-1$ and right part for $j = r$). Make one more $E'(j-1, .)$ query to get $E'(j, Y_L \| Y_R) = E'(j-1, Y_R \| f_j(Y_L) \oplus Y_L)$. □

Let us for now assume that $\delta = 1$ (i.e. B_g always reconstructs correctly) and let $T_{j,r}$ denote the number of forward queries to Ψ_r one has to make in order to compute $E'(j, .)$. By the above claims

1. $T_{1,r} = k + 1$
2. $T_{i,r} = (k+1) T_{i-1,r}$ for $i \in [2, r-2]$.
3. $T_{i,r} = 2 \cdot T_{i-1,r}$ for $i = r-1$ or $i = r$.

For $i \leq r-2$, the relations 1. and 2. are satisfied by

$$T_{i,r} \leq (k+1)^i$$

So $T_{r-2,l} = (k+1)^{r-2}$, with 3. this gives

$$T_{r,r} = 4(k+1)^{r-2}$$

As claimed in the theorem. We just have to verify the success probability, the error $\delta^{(k+1)^{r-2}}$ comes up as follows: by the first claim, we can compute $E(1, .)$ with probability δ. For $E(j, .)$ $(1 < j \leq r-1)$ we need $k+1$ invocations of $E(j-1, .)$, thus the error exponentiates with $k+1$. For $j = r-1$ and $j = r$ no extra error is introduced. □

4 Leakage-Resilient PRPs

Theorem 3 below states that an r round Feistel network, instantiated with $\mathcal{L}$-resilient PRFs, is a $\mathcal{L}'$-resilient super PRP. Here $\mathcal{L}'$ contains all leakage functions which for every round round $i \in [r]$ leak $g_i(K_i, R_i)$ where $g_i \in \mathcal{L}$ is an admissible leakage function for the leakage-resilient PRF used in the round functions. Moreover the round function inputs R_i are leaked entirely. Thus, if the PRF is NALR secure, so is the super PRP. The number of queries a distinguisher can make is exponential in r, thus for super-logarithmic r we get security against any polynomial distinguisher.

Theorem 3. *An r round Feistel network instantiated with NARL secure PRFs is a NARL secure super PRP for q-query distinguishers satisfying $q \leq 1.38^{r/2-1}$.*

More precisely, let $\mathsf{F} : \Sigma^k \times \Sigma^n \to \Sigma^n$ be a $(\epsilon_{\mathsf{prf}}, s_{\mathsf{prf}}, q)$-secure $\mathcal{L}$-resilient PRF and $\Psi_r = \Psi[f_1, \ldots, f_r]$ denote an r round Feistel network instantiated with $f_i = \mathsf{F}(K_i, .)$. Then Ψ_r (whose key is $K \stackrel{\mathsf{def}}{=} \{K_1, \ldots, K_r\}$) is a (ϵ, s, q) $\mathcal{L}'$-resilient super-PRP for

$$q \leq 1.38^{r/2-1} \qquad s = s_{\mathsf{prf}} - |F| \cdot q \cdot r \qquad \epsilon = (2 + q \cdot r) \cdot \epsilon_{\mathsf{prf}} + \frac{q^6 r^6}{5! \cdot 2^n} + \frac{q^2}{2^n}$$

Where $\mathcal{L}'$ contains, for every $g_1, \ldots, g_r \in \mathcal{L}$, the function g' defined as

$$g'(K, X) = \{g_1(K_1, R_1), \ldots, g_r(K_r, R_r), R_0, \ldots, R_{r+1}\}$$

with $(R_0, \ldots, R_{r+1}) \leftarrow \mu(\Psi_r, X)$.

We will prove this theorem using a combinatorial lemma from [12]. Consider an adversary $\mathcal{A}$ making q queries (forward or inverse) to $\Psi_r = \Psi[f_1, \ldots, f_r]$. Let $R[i, j]$ denote the input to the jth round function on the ith query. We say $R[i, j+1]$ (resp. $R[i, j-1]$) is "freshly generated" if the ith query is a forward (resp. inverse) query where $R[i, j]$ is fresh in the sense that $R[i, j] \neq R[k, j]$ for all $k < j$ (and thus f_j has not been invoked on $R[i, j]$ before). We say that for this sequence of queries the 5-XOR condition holds, if some freshly generated value can be expressed as the bitwise XOR of 5 previously computed round function inputs. In [12] the following Lemma is proven

Lemma 1 (Lemma 4.1 from [12]). Let Ψ_r be any r round Feistel network. For any $s \leq r/2$, if after making $q \leq 1.38^{s/2}$ forward/inverse queries to Ψ_r the 5-XOR condition does *not* hold, then there is no collision on the input to the jth round function for any $j \in [s, r-s]$.

Next we show that it is hard to provoke the 5-XOR condition in Ψ_r.

Lemma 2. Assume an adversary $\mathcal{A}$ of size s can satisfy the 5-XOR condition with probability ϵ making q queries to $\Psi_r(K, .)$ as in Theorem 3 (with each query X also getting the leakage $g'(K, X)$ for some $g' \in \mathcal{L}'$.) Then F is *not* a $(s_{\mathsf{prf}}, \epsilon_{\mathsf{prf}}, q)$-secure $\mathcal{L}$-resilient PRF where $s_{\mathsf{prf}} = s + |F| \cdot q \cdot r$ and $\epsilon_{\mathsf{prf}} = \frac{\epsilon}{q \cdot r} - \frac{q^5 r^5}{5! \cdot 2^n}$.

Proof. We define an adversary $\mathcal{A}'$ (which will use $\mathcal{A}$ as a black-box) against the $\mathcal{L}$-resilience of F. As in Definition 2, $\mathcal{A}'$ has access to $\mathsf{F}^g(K,.)$ (Where $g \in \mathcal{L}$ and $\mathsf{F}^g(K,X) \stackrel{\mathsf{def}}{=} [\mathsf{F}(K,X), g(K,X)].$)[12] and $\mathcal{O}(.)$, and has to guess whether $\mathcal{O}(.)$ is a random function or $\mathsf{F}(K,.)$.

$\mathcal{A}'$ first guesses a random query i and round j ($1 \leq i \leq q, 1 \leq j \leq r$). Then it simulates an attack of $\mathcal{A}$ on Ψ_r, where for the first i queries it uses its first oracle $\mathsf{F}^g(K,.)$ as the function for the jth round, and samples the round keys for the other $r-1$ rounds at random.

On the ith query, if the input to the jth round function is not fresh or the 5-XOR conditions already holds, $\mathcal{A}'$ outputs 0 and stops. Otherwise it uses its second oracle $\mathcal{O}(.)$ to compute the output, which gives a "freshly generated" value R. If this value can be expressed as the XOR of 5 previous round values, $\mathcal{A}'$ outputs 1 and 0 otherwise.

Assume $\mathcal{O}(.)$ is a uniformly random function, then the probability that $\mathcal{A}'$ outputs 1 is at most $q^5r^5/(5!\cdot 2^n)$ as the output of $\mathcal{O}(.)$ is uniformly random, and there are at most $q^5r^5/5!$ possible values (i.e. each subset of 5 queries specifies one possibility) which will trigger the 5-XOR condition.

Now assume the other case, where $\mathcal{O}(.)$ is $\mathsf{F}(K,.)$. If $\mathcal{A}$ will provoke the 5-XOR condition (which holds with prob. ϵ), and $\mathcal{A}'$ guessed which fresh query will satisfy this condition for the first time (with happens with prob $1/(q\cdot r)$), then $\mathcal{A}'$ will output 1. Thus in this case $\mathcal{A}'$ outputs 1 with prob. $\epsilon/(q\cdot r)$.

By definition, the gap $\epsilon/q\cdot r - q^5r^5/(5!\cdot 2^n)$ between those two probabilities is $\mathcal{A}'$ advantage in breaking the $\mathcal{L}$-resilience of F. □

Proof (of Theorem 3). Consider an adversary $\mathcal{A}$ of size s against the $\mathcal{L}'$-resilience of Ψ_r as specified in Definition 2. This $\mathcal{A}$ has access to two oracles, the first being $\Psi_r^{g'}(K,.) : X \rightarrow [\Psi_r(K,X), g'(K,X)]$ and the second being either $\Psi_r(K,.)$ or a uniformly random permutation $\mathbf{P}_n(.)$ (we call this the real and random experiment). By Lemma 2, in the real experiment the inputs to the functions in round $w \stackrel{\mathsf{def}}{=} \lfloor r/2 \rfloor$ and $w+1$ will be distinct with probability at least $1-\epsilon'$ where $\epsilon' = q\cdot r\cdot \epsilon_{\mathsf{prf}} + q^6r^6/(5!\cdot 2^n)$. Conditioned on this, the output of the right oracle in the real experiment is pseudorandom and thus cannot be distinguished from the output of the right oracle $\mathbf{P}_n(.)$ in the random experiment but with probability $2\cdot\epsilon_{\mathsf{prf}} + q^2/2^n$, here the $2\epsilon_{\mathsf{prf}}$ accounts for the output only being *pseudo*random, and the $q^2/2^n$ accounts for the fact that even if those values were uniform, the distribution would still be slightly off from what the oracle $\mathbf{P}_n$ in the random experiment outputs (we omit the details here.) Thus, $\mathcal{A}$ cannot distinguish the two experiments better than with probability $\epsilon' + 2\cdot\epsilon_{\mathsf{prf}} + q^2/2^n$. □

[12] The following reduction also works for the original notion of leakage-resilience where the leakage-function can be adaptively chosen. For this one must consider the oracle $\mathsf{F}^{\mathcal{L}}$ (instead F^g) defined as $\mathsf{F}^{\mathcal{L}}(K,X,g) \stackrel{\mathsf{def}}{=} [\mathsf{F}(K,X), g(K,X)]$ (where $g \in \mathcal{L}$). Thus, although our current PRF constructions only give us "non-adaptive-leakage" sPRPs, future advances in leakage-resilient PRFs would immediately translate to stronger leakage-resilient sPRPs.

References

1. Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 474–495. Springer, March 2009.
2. Joël Alwen, Yevgeniy Dodis, and Daniel Wichs. Leakage-resilient public-key cryptography in the bounded-retrieval model. In Shai Halevi, editor, *CRYPTO 2009*, LNCS, pages 36–54. Springer, August 2009.
3. Boaz Barak, Ronen Shaltiel, and Avi Wigderson. Computational analogues of entropy. In *RANDOM-APPROX*, pages 200–215, 2003.
4. Zvika Brakerski and Shafi Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability (or: Quadratic residuosity strikes back). In *CRYPTO*, 2010.
5. Zvika Brakerski, Yael Tauman Kalai, Jonathan Katz, and Vinod Vaikuntanathan. Cryptography resilient to continual memory leakage. Cryptology ePrint Archive, Report 2010/278, 2010. http://eprint.iacr.org/.
6. Jean-Sébastien Coron, Yevgeniy Dodis, Cécile Malinaud, and Prashant Puniya. Merkle-Damgård revisited: How to construct a hash function. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 430–448. Springer, August 2005.
7. Jean-Sébastien Coron, Jacques Patarin, and Yannick Seurin. The random oracle model and the ideal cipher model are equivalent. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 1–20. Springer, August 2008.
8. Giovanni Di Crescenzo, Richard J. Lipton, and Shabsi Walfish. Perfectly secure password protocols in the bounded retrieval model. In Shai Halevi and Tal Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 225–244. Springer, March 2006.
9. Yevgeniy Dodis, Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In *TCC 2010*, LNCS, pages 361–381. Springer, 2010.
10. Yevgeniy Dodis, Kristiyan Haralambiev, Adriana Lopez-Alt, and Daniel Wichs. Cryptography against continuous memory attacks. Cryptology ePrint Archive, Report 2010/196, 2010. http://eprint.iacr.org/.
11. Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett. On cryptography with auxiliary input. In *STOC*, pages 621–630, 2009.
12. Yevgeniy Dodis and Prashant Puniya. Feistel networks made public, and applications. In Moni Naor, editor, *EUROCRYPT 2007*, volume 4515 of *LNCS*, pages 534–554. Springer, May 2007.
13. Yevgeniy Dodis, Amit Sahai, and Adam Smith. On perfect and adaptive security in exposure-resilient cryptography. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 301–324. Springer, May 2001.
14. Stefan Dziembowski. Intrusion-resilience via the bounded-storage model. In Shai Halevi and Tal Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 207–224. Springer, March 2006.
15. Stefan Dziembowski and Ueli M. Maurer. Tight security proofs for the bounded-storage model. In *34th ACM STOC*, pages 341–350. ACM Press, May 2002.
16. Stefan Dziembowski and Krzysztof Pietrzak. Intrusion-resilient secret sharing. In *FOCS*, pages 227–237, 2007.
17. Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *49th FOCS*, pages 293–302. IEEE Computer Society Press, October 2008.

18. Sebastian Faust, Tal Rabin annd Leonid Reyzin, Eran Tromer, and Vinod Vaikuntanathan. Protecting circuits from leakage: The computationally-bounded and noisy cases. In *EUROCRYPT*, 2010.
19. Sebastian Faust, Eike Kiltz, Krzysztof Pietrzak, and Guy N. Rothblum. Leakage-resilient signatures. In *TCC 2010*, LNCS, pages 343–360. Springer, 2010.
20. Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic analysis: Concrete results. In Çetin Kaya Koç, David Naccache, and Christof Paar, editors, *CHES 2001*, volume 2162 of *LNCS*, pages 251–261. Springer, May 2001.
21. Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33:792–807, 1986.
22. J. Alex Halderman, Seth D. Schoen, Nadia Heninger, William Clarkson, William Paul, Joseph A. Calandrino, Ariel J. Feldman, Jacob Appelbaum, and Edward W. Felten. Lest we remember: Cold boot attacks on encryption keys. In *USENIX Security Symposium*, pages 45–60, 2008.
23. Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.
24. Yuval Ishai, Manoj Prabhakaran, Amit Sahai, and David Wagner. Private circuits II: Keeping secrets in tamperable circuits. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 308–327. Springer, May / June 2006.
25. Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 463–481. Springer, August 2003.
26. Jonathan Katz and Vinod Vaikuntanathan. Signature schemes with bounded leakage resilience. In *ASIACRYPT 2009*, LNCS, pages 703–720. Springer, December 2009.
27. Eike Kiltz and Krzysztof Pietrzak. How to secure elgamal against side-channel attacks. Manuscript, 2009.
28. Paul C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In Neal Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 104–113. Springer, August 1996.
29. Paul C. Kocher. Design and validation strategies for obtaining assurance in countermeasures to power analysis and related attacks. In *Proceedings of the NIST Physical Security Workshop*, 2005.
30. Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 388–397. Springer, August 1999.
31. Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, 17(2), 1988.
32. Ueli M. Maurer. A provably-secure strongly-randomized cipher. In Ivan Damgård, editor, *EUROCRYPT'90*, volume 473 of *LNCS*, pages 361–373. Springer, May 1990.
33. Ueli M. Maurer. Indistinguishability of random systems. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 110–132. Springer, April / May 2002.
34. Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 21–39. Springer, February 2004.
35. Silvio Micali and Leonid Reyzin. Physically observable cryptography (extended abstract). In *TCC*, pages 278–296, 2004.

36. Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In Shai Halevi, editor, *CRYPTO 2009*, LNCS, pages 18–35. Springer, August 2009.
37. European Network of Excellence (ECRYPT). The side channel cryptanalysis lounge. http://www.crypto.ruhr-uni-bochum.de/en_sclounge.html. retrieved on 29.03.2008.
38. Krzysztof Pietrzak. A leakage-resilient mode of operation. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 462–482. Springer, April 2009.
39. Jean-Jacques Quisquater and David Samyde. Electromagnetic analysis (ema): Measures and counter-measures for smart cards. In *E-smart*, pages 200–210, 2001.
40. Omer Reingold, Luca Trevisan, Madhur Tulsiani, and Salil P. Vadhan. Dense subsets of pseudorandom sets. In *FOCS*, pages 76–85, 2008.
41. Francois-Xavier Standaert, Olivier Pereira, Yu Yu, Jean-Jacques Quisquater, Moti Yung, and Elisabeth Oswald. Leakage resilient cryptography in practice. Cryptology ePrint Archive, Report 2009/341, 2009. http://eprint.iacr.org/.
42. Salil P. Vadhan. Constructing locally computable extractors and cryptosystems in the bounded-storage model. *Journal of Cryptology*, 17(1):43–77, January 2004.

A Technical Lemmata

Lemma 3 ([16]). Let A_0, B_0 be independent and $\phi_1, \phi_2, \ldots$ be any sequence of functions. Let $A_1, A_2, \ldots$, $B_1, B_2, \ldots$ and $V_1, V_2, \ldots$ be defined as

$$\begin{aligned}((A_{i+1}, V_{i+1}), B_{i+1}) &:= (\phi_{i+1}(A_i, V_1, \ldots, V_i), B_i) \quad \text{if } i \text{ is even}\\ (A_{i+1}, (V_{i+1}, B_{i+1})) &:= (A_i, \phi_{i+1}(B_i, V_1, \ldots, V_i)) \quad \text{otherwise}\end{aligned}$$

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

Lemma 4 ([38]). For any $\alpha > 0$ and $t \in \mathbb{N}$: If $\mathsf{F} : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^m$ is a $(\epsilon_{\mathsf{prf}}, s_{\mathsf{prf}}, q_{\mathsf{prf}})$-secure wPRF (for uniform keys), then it is a $(\epsilon'_{\mathsf{prf}}, s'_{\mathsf{prf}}, q'_{\mathsf{prf}})$-secure wPRF even if the keys are only sampled from a distribution with min-entropy $\kappa - \alpha$ with

$$q_{\mathsf{prf}} \geq q'_{\mathsf{prf}} \cdot t \qquad s_{\mathsf{prf}} \geq s'_{\mathsf{prf}} \cdot t \qquad \epsilon_{\mathsf{prf}} \leq \epsilon'_{\mathsf{prf}}/2^{\alpha+1} - \frac{q^2_{\mathsf{prf}}}{2^{n+1}} - 2 \cdot \exp\left(-\frac{t \cdot \epsilon'^2_{\mathsf{prf}}}{8}\right)$$

Lemma 5 ([38]). Let $\beta > 0$, then if $\mathsf{F} : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^m$ is a $(\epsilon_{\mathsf{prf}}, s_{\mathsf{prf}}, 1)$-secure wPRF (for uniform inputs), it's also a $(\epsilon'_{\mathsf{prf}}, s'_{\mathsf{prf}}, 1)$-secure wPRF if the input is chosen from a distribution with min-entropy $m - \beta$, where for any $t \in \mathbb{N}$

$$s_{\mathsf{prf}} \geq s'_{\mathsf{prf}} \cdot 2t \qquad \epsilon_{\mathsf{prf}} \leq \epsilon'_{\mathsf{prf}}/2^{\beta+1} - 2 \cdot \exp\left(-\frac{2 \cdot t \cdot \epsilon'^2_{\mathsf{prf}}}{64}\right)$$

Lemma 6 ([38]). Let $\mathsf{F} : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^m$ be a $(\epsilon_{\mathsf{prf}}, s_{\mathsf{prf}}, n/\epsilon^2_{\mathsf{prf}})$-secure wPRF. Let $K \in \{0,1\}^\kappa$ and $X \in \{0,1\}^n$ be independent where $H_\infty(K) = \kappa - 2\lambda$ and $H_\infty(X) = n - 2\lambda$ and let $f : \{0,1\}^{\kappa+n} \to \{0,1\}^\lambda$ be any leakage function, then for $\lambda \leq \log(\epsilon^{-1}_{\mathsf{prf}})/6$

$$\Pr_{X,Y}[\mathsf{H}^{\mathsf{HILL}}_{\epsilon',s'}(\mathsf{F}(K,X)|X, f(K,X)) \geq m - 2\lambda] \geq 1 - 2^{-\lambda/2+1}$$

with $\epsilon' = 2^{-\lambda/2+2}$ and $s' = s_{\mathsf{prf}}/2^{\lambda+3}(n+\kappa)^3$.