A Time-Success Ratio Analysis of wPRF-based Leakage-Resilient Stream Ciphers

Maciej Skórski
maciej.skorski@mimuw.edu.pl
Cryptology and Data Security Group, University of Warsaw

Abstract. Weak pseudorandom functions (wPRFs) found an important application as main building blocks for leakage-resilient ciphers (EUROCRYPT’09). Several security bounds, based on different techniques, were given to these stream ciphers. The security loss in these reduction-based proofs is always polynomial, but has not been studied in detail. The aim of this paper is twofold. First, we present a clear comparison of quantitatively different security bounds in the literature. Second, we revisit the current proof techniques and answer the natural question of how far we are from meaningful and provable security guarantees, when instantiating weak PRFs with standard primitives (block ciphers or hash functions). In particular, we demonstrate a flaw in the recent (TCC’14) analysis of the EUROCRYPT’09 stream cipher Our approach is a time-to-success ratio analysis, a universal measure introduced by Luby, which allow us to compare different security bounds.

Keywords. leakage-resilient cryptography, stream ciphers, side information, convex approximation

1 Introduction

1.1 Leakage-resilient cryptography

Leakage Resilience. Traditional security notions in cryptography consider adversaries who can interact with a primitive only in a black-box manner, observing its input/output behavior. Unfortunately, this assumption is unrealistic in practice. In fact, information might leak from cryptograms at the physical implementation layer. The attacks that capture information this way are called side-channel attacks, and include power consumption analysis [KJJ99], timing attacks [Koc96], fault injection attacks [BBKN12] or memory attacks [HSH+08]. Searching for countermeasures against side-channel attacks, one can try to prevent them modifying software or further secure hardware. However, these techniques are more ad-hoc than generic. A completely different viewpoint is to provide primitives which are provably secure against leakage. The research field following this paradigm is called leakage-resilient cryptography, and has become very popular in recent years. A lot of work and progress has been done in this topic so far, since the breakthrough paper on resilient stream ciphers [DP08], much more than we could mention here. We refer the reader to [ADW10] and [Mol10] for good surveys.
Modeling leakage. A number of ways to capture the leakage has been proposed. Very first works focused on strongly restricting the type of leakage. Here we very briefly discuss most important ones, referring interested readers to surveys.

- exposure resilient cryptography. In this line of work, the type of leakage is restricted so that adversaries learn subsets of the bits of the secret state key [CDH00, DSS01].

- continuous bounded computational leakage. Perhaps the most popular line of research restricting the leakage type, based on the “only computation leaks information” axiom introduced Micali and Reyzin [MR04]. In this modelling approach the overall execution of a cryptographic protocol is divided into time frames, and in every round leakage comes only from the parts of the internal state which are touched by computations. The amount of leakage is bounded in every round but unbounded overall. This model successfully captures side-channels attacks resulting from computation [Mol10], however memory attacks are more problematic as they are possible even if no computation is performed [HS08]. Nonetheless, leakage-resilient constructions under the “only computation leaks information” assumption, are of big interests [DP08, Pie09, DP10, FPS12, YS13], to mention only stream-ciphers related works. We also note that in specific cases, in particular for stream ciphers we will be interested in, the authors argue that their security models go beyond the “only computation leaks information” assumption and actually capture memory attacks (cf. [Pie09]).

- probing attacks. In this approach, initiated in [ISW03], adversaries can learn or influence the values at some wires, during the evaluation of a circuit.

- auxiliary inputs. The works [DKL09, DGTK+10] study a setting where adversaries can learn a function of the secret state, which is hard to invert. It allows leaking information larger than the size of the secret state and is believed to be most practical. However, it is also considered very challenging for proving security of constructions.

Being interested in leakage-resilient stream ciphers, we follow the related works and focus on continuous computational leakage through this paper (see Section 2 for a formal definition in the concrete setting).

1.2 Leakage-resilient stream ciphers.

What are stream ciphers? The purpose of stream ciphers is to efficiently encrypt data streams of arbitrary length. The most popular constructio mimics the one-time pad encryption, by deploying a generator which stretches the initial randomness into a keystream. Such a generator, when initialized with a secret state, recursively computes a sequence of output blocks where the security requirement is that the last part look random given the previous outputs.

Leakage-resilient design. The main concern in proving leakage-resilience is that the keystream generator must be secure against leakages, which appear in every
round (in the continuous leakage model). Such a generator could be deployed either with a pseudorandom generator and extractor [DP08], or a weak pseudorandom functions [Pie09,YSPY10,FPS12,YS13]. In any case, the idea is to refresh the secret state (key) in every round, to make compromising it possibly difficult. Below we briefly discuss some advantages of the second approach, and return to a more detailed discussion of the concrete designs in Section 3.

Why wPRFs-based design? Informally, pseudorandom functions look random on many adversarially chosen inputs (under a uniform secret key), whereas weak pseudorandom functions look random only on random inputs. Below we elaborate more on why weak pseudorandom functions are of special interests for leakage-resilient stream cipher constructions.

(a) From a high-level viewpoint, we have at least two very good reasons to build leakage-resilient stream ciphers using weak PRFs, as proposed in [Pie09]. First, this approach is simple and thus more efficient to implement and much easier to analyze than the original proposal [DP08], which combines a pseudorandom generator and an extractor. Second, and most important, it less vulnerable to side-channel attacks and more reliable from a practical viewpoint. This is because the construction can be instantiated with only one component - a weak PRF. Mounting an attack against one component is less likely, as opposed to the original construction [Sta10,MS11,MSJ12]. Moreover, this construction is more reliable from a practical viewpoint when instantiated with block ciphers understood as weak PRFs (like AES), because their security against side-channel attacks has been carefully analyzed.

(b) From a technical viewpoint, weak pseudorandom functions are primitives very pleasurable to deal with in the context of leakage. As opposed to (strong) pseudorandom functions they can be shown to remain secure with weak keys (that is when keys are not uniform but have some entropy deficiency), which is the key ingredient of the cipher resilience proof. Security with weak keys can be proven either by a computational variant of the Dense Model Theorem [Pie09] or by a recent techniques involving the square-security notion [DY13].

Security. The stream cipher is considered secure, if for a sequence of its outputs, the last round output block looks pseudorandom, given the outputs from previous rounds. See Section 2 for a formal definition.

1.3 Reductions Quality Issues.

The security of leakage-resilient stream ciphers is always proven by a reduction to underlaying more standard components, as pseudorandom generators, extractors, pseudorandom functions, whose security is generally well understood. Proving these bounds is challenging and still we can only prove quite poor bounds, unless we impose strong idealistic assumptions. Below we elaborate more on this topic.
(a) **Significant security losses in the standard model.** Reduction proofs yield quite weak bounds, and this is common for all related works. For leakage-resilient stream ciphers we have to lose a constant fraction of the security compared to its original level, even if the leakage is just one bit!

(b) **No provable security with standard building bricks.** When we aim for the (provable) security level recommended nowadays, which is at least 80 bits, we need to start with primitives (like block ciphers) whose security is bigger than 400 bits, given current knowledge. This is a direct consequence of the issue with weak reductions we mentioned above.

(c) **Different bounds are hard to compare.** Depending on the technique, different bounds are obtained. Formulas offer security against different adversarial profiles - running time, success probability, leakage length.

### 1.4 Problem and results, informally

Motivated in studying the quality of reductions, we state our problem as a series of questions. We briefly answer them here, announcing our results informally, and discuss in the next section in more detail.

- **Q1:** How tight are reduction-based security proofs for leakage-resilient stream ciphers?

  We revisit the best known bounds and analyze the tightness of reductions using time-success ratios. We discuss these tools in more detail in the next section.

  **A1:** All results loses more than 75% of the original security (measured in bits), paying for the resilience feature. This holds even for one bit of leakage per computation!

The second issue we address is how far we really are from having provable security for constructions instantiated from practically used components.

- **Q2:** Can we instantiate a leakage-resilient stream cipher, provable secure in the standard model, with a standard (128 or 256-bit) block cipher as a weak PRF?

  The most serious attempt to achieve meaningful security using standard 256-bit block ciphers is due to Pietrzak and Jetchev [JP14]. They improved and simplified bounds for the EUROCRYPT’09 stream cipher. However, as we will explain later, the better of the two claimed bounds doesn’t apply because of a flaw in the proof [Pie].

  **A2:** No, given the current state of art. The recent analysis from TCC’14 which gives an affirmative answer, contains a flaw. We will discuss it in Section 4.3.

Because of the lack of a positive answer above, it is natural to ask how strong our starting primitive needs to be, given current proof techniques. We believe that it is of interests to know how far we are with provable secure bounds from the idealized bounds, especially that this approach seems to be relatively rarely taken.
Q3: What a weak PRF do we need to achieve the recommended security level of 80 bits, given the known techniques?

Using our time-success ratio analysis we given an answer

A3: At least with 512 bits of security (and assuming small leakage). We propose to instantiate with SHA512 as a weak PRF.

1.5 Results and techniques in details.

Flaws in the recent analysis of the EUROCRYPT'09 stream cipher. Pietrzak and Jetchev came up with an elegant idea to simplify the security proof of the EUROCRYPT'09 stream cipher built from a weak PRF. To this end, they prove a theorem about simulating auxiliars inputs, giving two alternative proofs [JP14]. One of them would imply good security in the standard model, with AES used as the weak PRF (for the first time). Unfortunately, as we point out in Section 4.3 in this paper, the proof of this stronger bound is wrong. For this reason, only the second much weaker bound applies so we cannot prove meaningful security instantiating the stream cipher with a standard 256 block cipher, like AES.

An improved simulator for auxiliary inputs and better security for the EUROCRYPT'09 stream cipher. We don’t know how to fix the issue with the flawed analysis in [JP14]. However, we improve the alternative proof of the simulating lemma by a significant factor, which gives a better analysis of the stream cipher than the [JP14]. Our proof might of independent interest because of the proof technique, which utilizes a variant of the Baron-Maurey approximation theorem. We refer the reader to Theorem 3 in Section 4.3 for more details.

A framework to compare different reductions. Bellare,Rogaway [BR96] were first who emphasized the importance of studying the tightness of security proofs in practical applications. Following the approach proposed by Luby [LM94], based on time-success ratio (see Section 2.2), we provide a general tool for determining the security of every stream cipher reducing to a weak PRF. Technically, by constrained optimization we determine the time-success ratio of a stream cipher from the security of its main building component. This approach is used in different area of provable security (cf. [BL13] and many similar works), but to our knowledge has never been taken with respect to leakage-resilient stream ciphers (in particular in all the works we cite).

A clear security loss formula. We abstract the “typical form” for the loss in most reductions from a stream cipher to the underlying weak PRF. Namely the time/advantage pairs, describing adversarial resources and success probability, for the original primitive \((s, \epsilon)\) and for the cipher \((s', \epsilon')\) are related as \(\epsilon' = \epsilon^A\) and \(s' = s \cdot \epsilon^C - \epsilon^{-B}\) for some explicit constants \(A, B, C\) in the exponents. Extending slightly this model to capture leakage-depended factors, we actually cover all related works. We solve the related optimization program and show
how explicitly the time-success ratio degradation depends on these constants (see Section 4.2). It turns out that remains is a fraction of roughly $\frac{A}{B+C+1}$ of the original security (measured in bits). For all known constructions, this is smaller than 25%.

A survey of known results. We present the time-success ratio analysis of wPRF-based leakage-resilient stream ciphers. The lack of such results is perhaps partially because of complicated formulas, and partially because in folklore these bounds are considered mainly of theoretical interests. Yet, we believe that comparing these bounds is interesting, in particular with respect to the “dream bounds” corresponding to the flawed analysis in [JP14], which - if can be proven - gives a much better security level than other techniques. For more details, see Section 5.

2 Preliminaries

2.1 Leakage resilient cryptography

We start with the definition of weak pseudorandom functions, which are computationally indistinguishable from random functions, when queried on random inputs and fed with uniform secret key.

Definition 1 (Weak pseudorandom functions). A function $F : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^m$ is an $(\epsilon, s, q)$-secure weak PRF if its outputs on $q$ random inputs are indistinguishable from random by any distinguisher of size $s$, that is

$$\left| \Pr \left[ D((X_i)_{i=1}^q, F((K, X_i)_{i=1}^q) = 1 \right] - \Pr \left[ D((X_i)_{i=1}^q, (R_i)_{i=1}^q) = 1 \right] \right| \leq \epsilon$$

where the probability is over the choice of the random $X_i \leftarrow \{0,1\}^n$, the choice of a random key $K \leftarrow \{0,1\}^k$ and $R_i \leftarrow \{0,1\}^m$ conditioned on $R_i = R_j$ if $X_i = X_j$ for some $j < i$.

Stream ciphers generate a keystream in a recursive manner. The security requires the output stream should be indistinguishable from uniform.

Definition 2 (Stream ciphers). A stream-cipher $SC : \{0,1\}^k \rightarrow \{0,1\}^k \times \{0,1\}^n$ is a function that need to be initialized with a secret state $S_0 \in \{0,1\}^k$ and produces a sequence of output blocks $X_1, X_2, \ldots$ computed as

$$(S_i, X_i) := SC(S_{i-1}).$$

A stream cipher $SC$ is $(s, \epsilon, q)$-secure if for all $1 \leq i \leq q$, the random variable $X_i$ is $(s, \epsilon)$-pseudorandom given $X_1, \ldots, X_{i-1}$ (the probability is also over the choice of the initial random key $S_0$).

\[1\] We note that in a more standard notion the entire stream $X_1, \ldots, X_q$ is indistinguishable from random. This is implied by the notion above by a standard hybrid argument, with a loss of a multiplicative factor of $q$ in the distinguishing advantage.
Now we define the security of leakage resilient stream ciphers, which follow the “only computation leaks” assumption.

**Definition 3 (Leakage-resilient stream ciphers).** A leakage-resilient stream-cipher is \((\varepsilon, s, q, \lambda)\)-secure if it is \((\varepsilon, s, q)\)-secure as defined above, but where the distinguisher in the \(j\)-th round gets \(\lambda\) bits of arbitrary deceptively chosen leakage about the secret state accessed during this round. More precisely, before \((S_j, X_j) := SC(S_{j1})\) is computed, the distinguisher can choose any leakage function \(f_j\) with range \(\{0, 1\}^\lambda\), and then not only get \(X_j\), but also \(\Lambda_j := f_j(\hat{S}_{j1})\), where \(\hat{S}_{j1}\) denotes the part of the secret state that was modified (i.e., read and/or overwritten) in the computation \(SC(S_{j1})\).

### 2.2 Time-Success Ratio

The running time (circuit size) \(s\) and success probability \(\varepsilon\) of attacks (practical and theoretical) against a particular primitive or protocol may vary. For this reason Luby \[LM94\] introduced the time-success ratio \(\frac{s}{\varepsilon}\) as a universal measure of security. This model widely used to analyze security, cf. \[BL13\] and related works.

**Definition 4 (Security by Time-Success Ratio \[LM94\]).** A primitive \(P\) is said to be \(2^k\)-secure if for every adversary with time resources (circuit size in the nonuniform model) \(s\), the success probability in breaking \(P\) (advantage) is at most \(\varepsilon < s \cdot 2^{-k}\). We also say that the time-success ratio of \(P\) is \(2^k\), or that it has \(k\) bits of security.

For example, AES with a 256-bit random key is believed to have 256 bits of security as a weak PRF\(^2\).

### 3 Leakage-Resilient Stream Ciphers Design

In this section we briefly discuss the known constructions of leakage-resilient stream ciphers in the standard model (without random-oracle assumptions)

#### 3.1 The very first idea (FOCS’08)

The first construction of leakage-resilient stream cipher was proposed by Dziembowski and Pietrzak in \[DP08\]. It has the characteristic alternating structure which allows for proving security against adaptively chosen leakage.

#### 3.2 A construction based on a wPRF (EUROCRYPT’09)

On Figure 1 below we present a simplified version of this cipher \[Pie09\] based on a weak pseudorandom function (wPRF). A weak pseudorandom function is a primitive which “looks” like a random function when queried on random inputs, see Section 2 for a formal definition.

\(^2\) We consider the security of AES256 as a weak PRF, and not a standard PRF, because of non-uniform attacks which show that no PRF with a \(k\) bit key can have \(s/\varepsilon \approx 2^k\) security \[DTT09\], at least unless we additionally require \(\varepsilon \gg 2^{-k/2}\).
Fig. 1: The EUROCRYPT’09 stream cipher (adaptive leakage). $F$ denotes a weak PRF. By $K_i$ and $x_i$ we denote, respectively, values of the secret state and keystream bits. Leakages are denoted in gray with $L_i$.

3.3 Saving key randomness (CSS’10, CHESS’12)

A slightly different approach is proposed in [YSPY10]. The authors argue that side-channel attacks in practice are mounted against a specific target, and require specific measurements equipment; thus adaptive security is somewhat an overkill. The second observation is that the cipher in [Pie09] seems to waste lots of randomness, because the security in best case is only comparable to the length of one secret key, whereas the cipher is initialized with two random keys (denoted with $K_0, K_1$ on Figure 1). They remove the alternating structure and use only one key and two alternating public random values, aiming at (weaker) non-adaptive security. Unfortunately, the proof that these two alternating public values are enough were wrong, as pointed out in [FPS12]. However one gets provable non-adaptive security, assuming that every round uses fresh randomness [FPS12]. Such a big amount of randomness makes the cipher impractical, but the authors show how to reduce it further. Summing up, one gets only non-adaptive security but saves secret randomness replacing the “wasted” key by a public string. The scheme is illustrated in Figure 2 below.

3.4 Saving public randomness (CT-RSA’13)

The problem with large public randomness, required for the last cipher, was addressed in [YSPY10]. The public values, required in the previous construction, are generated on-the-fly from a single public value, by running a strong PRF in counter mode on it. For an illustration, see Figure 3 below. The result is only conditional and holds in the hypothetical world minicrypt, where one-way functions exist, but there is no public-key cryptography. Still, it may be a good clue on what we should aim for, when we want provable security in the standard model.
Fig. 2: The CSS’10/CHESS’12 stream cipher. $F$ denotes a weak PRF. By $K_i$ and $x_i$ we denote, respectively, the values of secret state and keystream bits. Leakages are denoted in gray with $L_i$. The cipher requires public independent random values $p_i$.

Fig. 3: The CTR-RSA’13 stream cipher (nonadaptive leakage, minicrypt). $F$ is a weak PRF and $G$ is a strong PRF. By $K_i$ and $x_i$ we denote, respectively, the values of secret state and keystream bits. Leakages are denoted in gray with $L_i$. The function $F$ generating the keystream is rerandomized using values $p_i$, produced by $G$ in counter mode from the single public seed $s$. 
4 Results

4.1 The time-success ratio under reductions

We consider first a very abstract setting, where a primitive $P'$ is built from $P$. Assume, that the security of $P'$ reduces to the security of $P$ in the following quantitative way:

**R:** If $P$ is secure against an adversary $(s, \epsilon)$, then $P'$ is secure against any adversary $(s', \epsilon')$, where

$$s' = p(s, \epsilon), \quad \epsilon' = q(s, \epsilon)$$

for some functions $p(\cdot), q(\cdot)$.

In the simplest case, the functions $p(\cdot)$ and $q(\cdot)$ are algebraic functions of original parameters, like $\epsilon' = \epsilon^{1/2}$ or $s' = s\epsilon^2$ (the second case appears particularly often as a result of the Chernoff Bounds). In leakage-resilient cryptography these formulas are more complicated and typically involve some additional parameters, like the leakage length or the number of queries. The natural question here is how the security, understood as in Definition 4, of the two primitives $P$ and $P'$ are related to each other. Before we give the answer (the proof appears in Appendix A).

**Theorem 1 (The time-success ratio as min-max optimization).** Let security of $P'$ reduces to security of $P$ as in Equation (1). If $P$ has $k$ bits of security then $P'$ has $k'$ bits of security where $k'$ is the maximal value such that the following program

$$\begin{align*}
\text{minimize} & \quad \frac{s'}{\epsilon'} \\
\text{maximize} & \quad \frac{s}{\epsilon} \\
\text{s.t.} & \quad \frac{s'}{\epsilon'} \leq 2^{k'}, \; 1 \leq s', \; 0 \leq \epsilon' \\
& \quad \frac{s}{\epsilon} \leq 2^k, \; 1 \leq s, \; 0 \leq \epsilon \\
& \quad s' \leq p(s, \epsilon), \; \epsilon' \geq q(s, \epsilon)
\end{align*}$$

has a positive finite value.

**Remark 1.** If we cannot find a pair $(s, \epsilon)$ corresponding to $(s', \epsilon')$ then the feasible set in Equation (2) is empty, so that the value of the program becomes $-\infty$.

4.2 The time-success ratio under algebraic transformations

In the most typical case we can solve Equation (2) explicitly, as shown by Theorem 2 below.
Theorem 2 (Time-success ratio for algebraic transformations). Let $a, b, c$ and $A, B, C$ be positive constants. Suppose that $P'$ is secure against adversaries $(s', \epsilon')$, whenever $P$ is secure against adversaries $(s, \epsilon)$, where

$$
\begin{align*}
s' &= s \cdot ce^C - be^{-B} \\
\epsilon' &= a e^A.
\end{align*}
$$

(3)

In addition, suppose that the following condition is satisfied

$$A \leq C + 1.
$$

(4)

Then the following is true: if $P$ is $2^k$-secure, then $P'$ is $2^{k'}$-secure where

$$k' = \begin{cases} 
\frac{A}{B+C+1}k + \frac{A}{B+C+1}(\log c - \log b) - \log a, & b \geq 1 \\
\frac{4}{a+1}k + \frac{4}{a+1}\log c - \log a, & b = 0
\end{cases}
$$

(5)

The proof is elementary though not immediate. It appears in Appendix B.

Remark 2 (On the technical condition (4)). This condition is satisfied in almost all applications, at in the reduction proof typically $\epsilon'$ cannot be better (meaning higher exponent) than $\epsilon$. Thus, quite often we have $A \leq 1$.

4.3 An error in the recent EUROCRYPT’09 stream cipher analysis (TCC’13) and our improvement

Simulating auxiliary inputs. In [JP14] there is the following theorem (here we state the correct version [Pie]):

Lemma 1 (Simulating auxiliary inputs). For any random variable $X \in \{0,1\}^n$, any correlated $Z \in \{0,1\}^\lambda$ and every choice of parameters $(\epsilon, s)$ there is a randomized function $\text{Sim} : \{0,1\}^n \rightarrow \{0,1\}^\lambda$ of complexity $O(s \cdot 2^{4\lambda}\epsilon^{-4})$ such that $Z$ and $\text{Sim}(X)$ are $(\epsilon, s)$-indistinguishable given $X$.

This theorem is the core of the improved analysis of the EUROCRYPT’09 stream cipher. Using it, as described in [JP14], one proves the resilience of the cipher if the underlying weak PRF is $(s, \epsilon)$-secure against two queries on random inputs.

More on the flaws. In the claimed better bound $O(s \cdot 2^{3\lambda}\epsilon^{-2})$ there is a mistake on page 18 (eprint version), when the authors enforce a signed measure to be a probability measure by a mass shifting argument. The number $M$ defined there is in fact a function of $x$ and is hard to compute. The original proof assumes that this is a constant independent of $x$. In the alternative bound $O(s \cdot 2^{3\lambda}\epsilon^{-2})$ a fixable flaw is a missing factor of $2^\lambda$ in the complexity (page 16 in the eprint version), which is because what is constructed in the proof is only a probability mass function, not yet a sampler [Pie].
Our improvement. We don’t know how to reduce the exponent in $\epsilon$. However, we can improve the constant in the exponent of $\lambda$, from 4 to 2. This is significant for the application to the cipher, as we improve its security by a factor of $2^{\Theta(\lambda)}$, which is typically of order $\Theta(\epsilon^{-1})$ (see [JP14,Pie09]).

**Theorem 3 (Better simulating auxiliary inputs).** For every distribution $X,Z$ on $X \times \{0,1\}^\lambda$ and every $\epsilon, s$ there exists a “simulator” $h : X \rightarrow \{0,1\}^\lambda$ such that (a) the distributions of $(X,h(X))$ and $(X,Z)$ are $(s,\epsilon)$-indistinguishable and (b) $h$ is of complexity $s_h = O(\epsilon \cdot 2^{2\lambda \epsilon^{-4}})$.

## 5 Survey of security bounds

In Table 1 below we present the comparison of different bounds for leakage-resilient stream ciphers built from weak PRFs. We assume that the number of blocks $q$ is constant. Without loosing generality, we assume that the time-advantage ratio for our PRF is constant, that is $s/\epsilon \approx 2^k$ where $k$ is the key length. This corresponds to the assumption that the best attack is a brute-force search\(^3\). This assumption is reasonable, for example best block ciphers like AES are believed to have such security as PRFs. The security level is computed from Theorem 2 by putting the bounds from the related works (we omit computations).

| Cipher | Analysis | Proof techniques | Security level | Comments |
|--------|----------|------------------|----------------|----------|
| (1) [Pie09] | Pseudoentropy chain rules | $k' \ll \frac{1}{8}k$ | large number of blocks |
| (1) [JP14] | Aux. Inputs Simulator (corr.) | $k' \approx \frac{1}{3} - \frac{4}{3} \lambda$ |
| (1) [VZ13] | Aux. Inputs Simulator | $k' \approx \frac{1}{3} - \frac{4}{3} \lambda$ |
| (1) **This work** | Aux. Inputs Simulator (impr.) | $k' \approx \frac{1}{3} - \frac{4}{3} \lambda$ | unproven (the flaw) |
| (1) Dream bound | Aux. Inputs Simulator (impr.) | $k' \approx \frac{1}{3} - \frac{4}{3} \lambda$ | large public seed |
| (2) [FPS12] | Pseudoentropy chain rules | $k' \approx \frac{1}{3} - \frac{4}{3} \lambda$ |
| (3) [YS13] | Square-friendly apps. | $k' \approx \frac{1}{3} - \frac{4}{3} \lambda$ | only in minicrypt |

Table 1: Different bounds for wPRF-based leakage-resilient stream ciphers. $k$ is the length of the secret key for the wPRF. The value $k'$ is the security level for the cipher, understood in terms of time-success ratio. The numbers denote: (1) The EUROCRYPT’09 cipher, (2) The CSS’10/CHESS’12 cipher, (3) The CT-RSA’13 cipher.

It seems that the best cipher (in the standard model) is the EUROCRYPT’09 cipher. It provides the adaptive security in the standard model and loses about $\frac{5}{6}$ of its original security (the best analysis is due to Vadhan, the second best is this paper). The CSS’10/CHESS’12 loses about $\frac{4}{5}$ of its original security but requires large public randomness.

\(^3\) This is not the case of asymetric primitives: consider e.g. RSA, here given our current understanding of the hardness of factoring, $\epsilon$ goes from basically 0 to 1 as the running time $s$ reaches the time required to run the best factoring algorithms.
References

ADW10. Jol Alwen, Yevgeniy Dodis, and Daniel Wichs, Survey: Leakage resilience and the bounded retrieval model, Information Theoretic Security (Kaoru Kurosawa, ed.), Lecture Notes in Computer Science, vol. 5973, Springer Berlin Heidelberg, 2010, pp. 1–18 (English).

BBKN12. Alessandro Barenghi, Luca Breveglieri, Israel Koren, and David Naccache, Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures, Proceedings of the IEEE, 2012, pp. 3056–3076.

BL13. Ahto Buldas and Risto Laanooja, Security proofs for hash tree timestamping using hash functions with small output size, Information Security and Privacy (Colin Boyd and Leonie Simpson, eds.), Lecture Notes in Computer Science, vol. 7959, Springer Berlin Heidelberg, 2013, pp. 235–250 (English).

BR96. Mihir Bellare and Phillip Rogaway, The exact security of digital signatures-how to sign with rsa and rabin, Proceedings of the 15th Annual International Conference on Theory and Application of Cryptographic Techniques (Berlin, Heidelberg), EUROCRYPT'96, Springer-Verlag, 1996, pp. 399–416.

BSW03. Boaz Barak, Ronen Shaltiel, and Avi Wigderson, Computational analogues of entropy, In 11th International Conference on Random Structures and Algorithms, 2003, pp. 200–215.

CDH+00. Ran Canetti, Yevgeniy Dodis, Shai Halevi, Eyal Kushilevitz, and Amit Sahai, Exposure-resilient functions and all-or-nothing transforms, Advances in Cryptology EUROCRYPT 2000 (Bart Preneel, ed.), Lecture Notes in Computer Science, vol. 1807, Springer Berlin Heidelberg, 2000, pp. 453–469 (English).

DGTK+10. Yevgeniy Dodis, Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan, Public-key encryption schemes with auxiliary inputs, Proceedings of the 7th International Conference on Theory of Cryptography (Berlin, Heidelberg), TCC’10, Springer-Verlag, 2010, pp. 361–381.

DKL09. Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett, On cryptography with auxiliary input, Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing (New York, NY, USA), STOC ’09, ACM, 2009, pp. 621–630.

DP08. Stefan Dziembowski and Krzysztof Pietrzak, Leakage-resilient cryptography, Proceedings of the 2008 49th Annual IEEE Symposium on Foundations of Computer Science (Washington, DC, USA), FOCS ’08, IEEE Computer Society, 2008, pp. 293–302.

DP10. Yevgeniy Dodis and Krzysztof Pietrzak, Leakage-resilient pseudorandom functions and side-channel attacks on feistel networks, Advances in Cryptology CRYPTO 2010 (Tal Rabin, ed.), Lecture Notes in Computer Science, vol. 6223, Springer Berlin Heidelberg, 2010, pp. 21–40 (English).

DSS01. Yevgeniy Dodis, Amit Sahai, and Adam Smith, On perfect and adaptive security in exposure-resilient cryptography, Advances in Cryptology EUROCRYPT 2001 ( Birgit Pfitzmann, ed.), Lecture Notes in Computer Science, vol. 2045, Springer Berlin Heidelberg, 2001, pp. 301–324 (English).

DTT09. Anindya De, Luca Trevisan, and Madhur Tulsiani, Non-uniform attacks against one-way functions and prgs, Electronic Colloquium on Computational Complexity (ECCC) 16 (2009), 113.
A Proof of Theorem 1

We notice that we are looking for the biggest value $k'$ such that for every $(s', \epsilon')$ satisfying $s' \geq 1$, $\epsilon' \geq 0$, $2^{k'} \geq s'/\epsilon'$ there exist some values $(s, \epsilon)$ such that $s \geq 1$, $\epsilon \geq 0$, $s' \leq p(s, \epsilon)$, $\epsilon' \geq q(s, \epsilon)$ and $2^k \geq s/\epsilon$. For given values $(s', \epsilon')$ we can choose $(s, \epsilon)$ so that the ratio $s'/\epsilon'$ is possibly maximal, provided that the constraint $s/\epsilon \leq 2^k$ is satisfied. Taking into account the quantifiers every and some we get the following min-max characterization.

B Proof of Theorem 2

Proof. Consider the program in Theorem 1. In our setting we have

$$p(s, \epsilon) = s \cdot c \epsilon^C - b \epsilon^B$$
$$q(s, \epsilon) = a \epsilon^A$$

The constraint $s' \geq 1$ is equivalent to

$$s \geq c^{-1}(1 + b \epsilon^{-B}) \epsilon^{-C}$$  \hspace{1cm} (6)

Thus, the all constraints on $s$ can be written as

$$c^{-1}(1 + b \epsilon^{-B}) \epsilon^{-C} \leq s, \quad s \leq 2^k \epsilon, \quad s' \leq p(s, \epsilon).$$

By definition $p(\cdot)$ is increasing in $s$. Therefore we can assume that

$$\frac{s}{\epsilon} = 2^k.$$  \hspace{1cm} (7)

The constraint $\epsilon' \geq 0$ simply reduces to $\epsilon \geq 0$. Thus, the all constraints where $\epsilon$ is involved are

$$0 \leq \epsilon, \quad s \leq 2^k \epsilon, \quad q(s, \epsilon) \leq \epsilon'$$
Since \( q(\cdot) \) is increasing, we can assume that \( \epsilon' = q(s, \epsilon) \), or in other words that
\[
\epsilon' = a\epsilon^A.
\] (8)

Given Equations (7) and (8) the maximum part of the optimization is eliminated. Our task reduces to minimizing the following expression
\[
\frac{s'}{\epsilon'} = s \cdot \frac{c}{a} \epsilon^{C-A} - \frac{b}{a} \epsilon^{-B-A}
= \epsilon^{-A} \left( \frac{c}{a} \cdot 2^{k} \epsilon^{C+1} - \frac{b}{a} \epsilon^{-B} \right),
\]
over \((s', \epsilon')\) or equivalently over \(s, \epsilon\) (given the equalities (7) and (8)), provided that Equation (6) is satisfied. Thus we obtain the following problem in one variable
\[
\text{minimize } \frac{a^{-1} \epsilon^{-A} \left( 2^{k} \epsilon^{C+1} - b \epsilon^{-B} \right)}{\epsilon}
\text{ s.t. } 2^{k} \epsilon^{C+1} - b \epsilon^{-B} \geq 1.
\] (9)

Now everything depends on the behavior of the objective function
\[
f(u) = \frac{2^{k} \epsilon^{C+1-A} - b \epsilon^{-B-A}}{a^{-1} \epsilon^{-A}}
\]
However the condition (4) implies that \( f(u) \) is increasing. Thus, it attains its minimum on the boundary point, which is given by
\[
2^{k} \epsilon^{C+1} - b \epsilon^{-B} = 1.
\] (10)
The objective function evaluated at this point gives us
\[
f(\epsilon) = a^{-1} \epsilon^{-A}
\] (11)
Note that from Equation (10) it follows that
\[
2^{k} \epsilon^{B+C+1} = b + \epsilon^{B}
\]
If \( b \geq 1 \) we obtain \( \epsilon^{B+C+1} \approx 2^{-k} \epsilon^{B-1} \) (up to a multiplicative factor of at most 2). If \( b = 0 \) then \( \epsilon^{C+1} \approx 2^{-k} \epsilon^{-1} \).

C Proof of Theorem 3

Proof (of ??). In the first step we show how to construct a simulator \( h = h^D \) for one circuit \( D \) of size \( s \).

Claim 1 (A perfect simulator for any any fixed real-valued distinguisher). For any \([0,1]\)-valued \( D \) of size \( s \) there exists a function \( h^D : \mathcal{X} \to \{0,1\}^m \) of complexity \( O(s \cdot 2^m) \) such that \( \mathbb{E} D(X, Z) = \mathbb{E} D(X, h(X)) \).
Proof (of Claim 1). Let $h_0^+$ and $h_0^-$ be functions such that

$$D(x, h_0^+(x)) = \max_z D(x, z), \quad D(x, h_0^-(x)) = \min_z D(x, z)$$

Both functions can be computed by enumerating over all $z \in \{0, 1\}^m$, using $2 \cdot 2^m$ calls to $D$. For any $X, Z$ we have

$$\mathbb{E}_{x \leftarrow X} D(x, h_0^-(x)) \leq \mathbb{E} D(X, Z) \leq \mathbb{E}_{x \leftarrow X} D(x, h_0^+(x))$$

Therefore there exists a number $\gamma_D \in [0, 1]$ such that

$$\mathbb{E} D(X, Z) = \gamma_D \mathbb{E}_{x \leftarrow X} D(x, h_0^-(x)) + (1 - \gamma_D) \mathbb{E}_{x \leftarrow X} D(x, h_0^+(x)).$$

We define $h(x) = h_D(x)$ as follows: sample $r \in [0, 1]$; if $r \leq b$ then we output $h_0^-(x)$ else we output $h_0^+(x)$.

Now we apply the min-max theorem in a standard way to change the order of quantifiers.

Claim 2 (One simulator for all distinguishers). There exists a distribution $\overline{F}$ on functions $h$ of complexity $O(s \cdot 2^m)$ such that $|\mathbb{E} D(X, Z) - \mathbb{E}_{h \sim \overline{F}} D(X, h(X))| \leq \epsilon$ for all $D$ of size $s \epsilon^2$.

Proof. By a standard application of the min-max theorem combined with the Chernoff Bound (see [BSW03] for essentially the same technique) we get that there is a distribution $\overline{F}$ such that for all $D$ of size $s \epsilon^2$ we have $\mathbb{E} D(X, Z) - \mathbb{E}_{h \sim \overline{F}} D(X, h(X)) \leq \epsilon$. Since this holds for $D$ and $D^c$ for any $D$ of size $s$, the result follows.

In the last step we approximate this possibly inefficient simulator in the statistical distance.

Claim 3 (One efficient simulator for all distinguishers). There exists a simulator $h$ of complexity $O(s \cdot 2^m \epsilon^{-2})$ such that $|\mathbb{E} D(X, Z) - \mathbb{E} D(X, h(X))| \leq 2\epsilon$ for all $D$ of size $s \epsilon^2$.

Proof (Proof of Claim 3). Let $h_0$ be the inefficient simulator guaranteed by Claim 2. We know that $h_0$ is of the following form

$$\mathbb{P}_{X, h_0(x)} = \mathbb{E}_{h \sim \overline{F}} \mathbb{P}_{X, h(x)} = \mathbb{E}_{x \sim X} \mathbb{E}_{h \sim \overline{F}} \mathbb{P}_{x, h(x)}$$  \hspace{1cm} (12)$$

Fix a number $t$ and sample $h_j \leftarrow \overline{h}$ for $j = 1, \ldots, t$. For a fixed choice of $h_1, \ldots, h_t$ we define the randomized function $\overline{h}(x)$ as follows: $\mathbb{P}_{\overline{h}(x)}(z) = t^{-1} \sum_{i=1}^{t} \mathbb{P}_{h_i(x)}(z)$ (it simply takes $i \leftarrow \{1, \ldots, t\}$ and outputs $h_i$). Below we assume that $x$ is
sampled according to $X$. Let us compute

$$
\mathbb{E}_{\{h_i\}_{i=1}^t} \mathbb{E}_x \left\| \mathbf{P}_{h}(\cdot) - \mathbf{P}_{\tilde{h}}(\cdot) \right\|_2^2 = t^{-2} \mathbb{E}_x \mathbb{E}_{\{h_i\}_{i=1}^t} \left\| \sum_{j=1}^t (\mathbf{P}_{h_j}(\cdot) - \mathbf{P}_{\tilde{h}}(\cdot)) \right\|_2^2
$$

$$
= t^{-2} \mathbb{E}_x \left( \sum_{j=1}^t \mathbb{E}_x \left\| \mathbf{P}_{h_j}(\cdot) - \mathbf{P}_{\tilde{h}}(\cdot) \right\|_2^2 \right)
$$

$$
= t^{-1} \left( \mathbb{E}_x \mathbb{E}_{\tilde{h}} \left\| \mathbf{P}_{h}(\cdot) \right\|_2^2 - \mathbb{E}_x \left\| \mathbf{P}_{\tilde{h}}(\cdot) \right\|_2^2 \right)
$$

Therefore for some choice of $h_1, \ldots, h_t$ we have

$$
\mathbb{E}_x \left\| \mathbf{P}_{\tilde{h}}(\cdot) - \mathbf{P}_{h}(\cdot) \right\|_2^2 \leq \frac{1}{t} \tag{13}
$$

Note that the simple probabilistic proof of Equation (13) resembles the proof of Maurey-Jones-Barron Theorem (see Lemma 1 in [?]) on approximating convex hulls in Hilbert spaces. Using the fact that $|D(\cdot, \cdot)| \leq 1$ and inequality between the first and the second norm

$$
\left| \mathbb{E} D(X, \tilde{h}(X)) - \mathbb{E} D(X, \bar{h}(X)) \right| = \mathbb{E}_x \left| \mathbb{E} D(x, h(x)) - \mathbb{E} D(x, \bar{h}(x)) \right|
$$

$$
\leq \mathbb{E}_x \left\| \mathbf{P}_{h}(\cdot) - \mathbf{P}_{\bar{h}}(\cdot) \right\|_1
$$

$$
\leq 2^{m/2} \cdot \left( \mathbb{E}_x \left\| \mathbf{P}_{\tilde{h}}(\cdot) - \mathbf{P}_{h}(\cdot) \right\|_2 \right)^{1/2} \tag{14}
$$

Combining Equation (13) and Equation (14) we get for some choices of $h_1, \ldots, h_t$

$$
\left| \mathbb{E} D(X, \tilde{h}(X)) - \mathbb{E} D(X, \bar{h}(X)) \right| \leq (2^m t^{-1})^{1/2} \tag{15}
$$

Setting $t = 2^m \epsilon^{-2}$ we finish the proof.

The result follows now directly from Claim 3, for real-valued circuits. Up to an error of $\delta = 2^{-\rho}$ in the advantage, we can approximate them by circuits taking values in the discrete set $\{2^{-\rho}, 2 \cdot 2^{-\rho}, \ldots, 1\}$. Any such a circuit $D$ we start our proof with, can be viewed as a combination of

(a) the coding vector $(D^{(i)})_{i=1}^\rho$ of $\rho$ circuits of size $s$, computing the first $\rho$ digits of the binary expansion of the output

(b) the decoding circuit of size $2\rho$ which uses additional $\rho$ random bits to read $(D^{(i)}(x, z))$, and to output 1 with probability $D(x, z)$ (in the $i$-th round it toss a coin and either halts and outputs $D^{i}(x, z)$ or it moves to the round $i + 1$; in round $n + 1$ the output is 0)

Now, the correct complexity for $h$ in Claim 1 and ?? is $O(s \cdot 2^m \rho)$, by the use of sorting network. Setting $\rho = \log(1/\epsilon)$ we see that we lose $O(s \cdot 2^m \log(\epsilon^{-1}))$ in the simulator complexity.