This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2021.3109302, IEEE Transactions on Information Theory

Systematic Security Analysis of Stream Encryption With Key Erasure

Yu Long Chen, Atul Luykx, Bart Mennink and Bart Preneel, Member, IEEE

***Abstract*—We consider a generalized construction of stream ciphers with forward security. The design framework is modular: it is built from a so-called layer function that updates the key and (optionally) the nonce and generates a new pseudorandom output stream. We analyze the generalized construction for four different instantiations: two possible layer functions that are in turn instantiated with either a block cipher or a pseudorandom function. We prove that each of these instantiations gives a stream cipher that is pseudorandom and forward secure in the multi-user setting with a very tight bound. A comprehensive analysis shows that the two block cipher based instantiations achieve very similar bounds. For the pseudorandom function based instantiations there is no clear winner: either layer can be beneficial over the other one, depending on the choice of parameters. By instantiating the pseudorandom function with a generic construction such as the sum of permutations, we obtain a highly efficient and competitive stream cipher based on an n-bit block cipher that is secure beyond the $2^{n/2}$ birthday bound.**

***Index Terms*—stream encryption; key erasure; pseudorandomness; forward security; beyond birthday bound**

I. Introduction

STREAM ciphers use a short secret key to generate a long pseudorandom string. This string, a "key stream", can be used to encrypt data by adding it to the plaintext. A simple example of a stream cipher is the counter mode (CTR) of encryption using an n-bit block cipher E:

$$E_K(N) \parallel E_K(N+1) \parallel \dots ,$$

where K denotes the secret key and N denotes the nonce. Bellare et al. [1] proved that its key stream is indistinguishable from random as long as (i) the block cipher cannot be broken by a computationally bounded adversary, (ii) no two block cipher calls are made for the same input, and (iii) the total amount of generated blocks σ satisfies $\binom{\sigma}{2} \ll 2^n$.

For some applications, this security guarantee is sufficient. In other applications, one might want stronger security. One example property is that of forward security [2]: even if a secret key is leaked, none of the previous pseudorandom strings should be endangered. Another property is that of multi-user security, where an adversary has access to multiple independent instances simultaneously. Further, if a block cipher with block size smaller than 128 is used, one would like to use a mode that is secure beyond the birthday bound, i.e., secure beyond an attack complexity of 2^{64}.

One way to achieve forward security is key erasure, a mechanism that consists of interlacing the generation of relatively short key streams with updates of the key. An example of this occurs in the stream generation of NIST's CTR_DRBG [3]. Here, one evaluates CTR mode encryption to generate a key stream of certain length, and then makes sufficient evaluations of E_K for new inputs to derive a new nonce and key. A comparable technique was employed in Bernstein's key erasure random number generator [4], with a main difference that the nonce does not get updated.

A. Our Contribution

We consider the generalized stream cipher construction $\mathcal{G}$ in Section III. It maintains a state consisting of a key K and a nonce N, and internally evaluates a layer function to generate a pseudorandom stream S and to update the internal state. The construction matches the generalized construction of Bellare and Yee [2].

We consider two possible layer functions that we deem most relevant, $\mathbf{layer}_1$ of Figure 2a and $\mathbf{layer}_2$ of Figure 2b. Here, we denote by $\sigma \in \mathbb{N}$ the number of pseudorandom blocks generated in one evaluation of the layer function, and Π is an underlying primitive that is used to instantiate the layer function. In our work, this primitive is either a block cipher with key size $2n$ and block size n, or a pseudorandom function with key size $2n$ and block size n. (The choice for primitives with $2n$-bit keys is inspired by our original applications, see Section I-B. Our results straightforwardly simplify to constructions from n-bit keyed primitives.)

We next prove tight security of the stream cipher construction $\mathcal{G}$ based on either layer and instantiated with either a block cipher or a pseudorandom function in Section V. The analyses are performed in the security formalism of Bellare and Yee [2]. These analyses yield tight bounds with respect to the number of key stream blocks generated in a layer function (denoted σ), the number of users, the number of (re-)initializations of the stream cipher construction, the number of streams generated per initialization, and the running time of the distinguisher. The pseudorandom function based constructions achieve beyond birthday bound security, the block cipher based versions are birthday bound secure. In this work, a stream cipher is called beyond birthday bound secure, if the distinguisher's advantage to break the construction remains negligible when the total number of blocks evaluated while interacting with the given oracle is significantly more than $2^{n/2}$.

The proofs are, in fact, performed in a layer-wise adaptive model, where the adversary does not query the construction on input of a nonce and a requested number of bits to get a certain output stream, but instead it has access to the initialization and next interfaces of $\mathcal{G}$ and can query those adaptively. In addition, along the way, we formalize a lifting lemma, that

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2021.3109302, IEEE Transactions on Information Theory

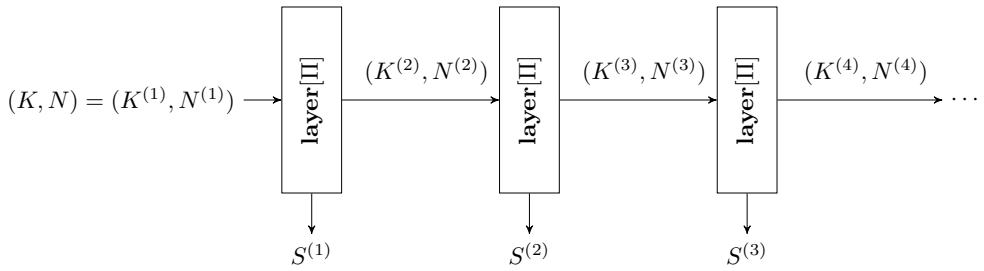
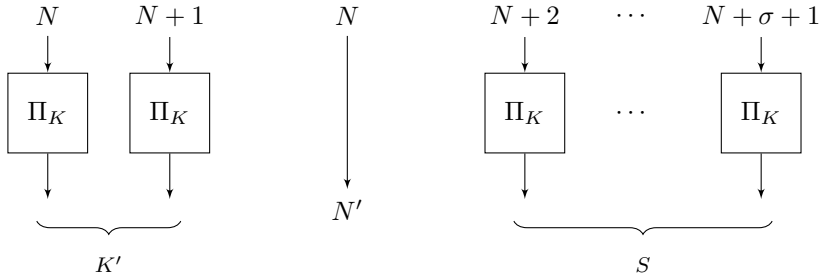
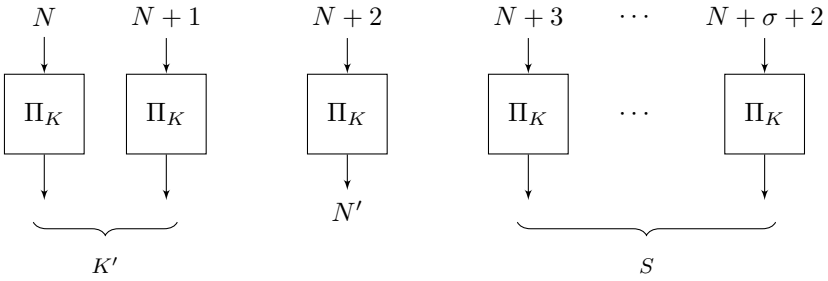


Figure 1: Generalized stream cipher construction $\mathcal{G}$ of Section III. Here, $\Pi \in \mathrm{Func}(3n, n)$ is a cryptographic primitive, K is the master key, N is the initial nonce, and $S^{(i)}$ are output streams.



(a) $\mathbf{layer}_1$



(b) $\mathbf{layer}_2$

Figure 2: The layer functions $\mathbf{layer}_1$ of Algorithm 3 and $\mathbf{layer}_2$ of Algorithm 4. Here, $\Pi \in \mathrm{Func}(3n, n)$ is a cryptographic primitive, S is the σn-bit stream for $\sigma \in \mathbb{N}$, and (K', N') is set as new state.

formalizes security of cascaded usage of a standard model primitive (Section II-C). This lemma has been implicit in earlier PRNG security proofs [4]–[6].

Whereas for block ciphers well-analyzed primitives are in abundance, and most applications use the well-established AES [7], this is not the case for dedicated pseudorandom functions [8]. It is therefore more interesting to instantiate the pseudorandom function based stream ciphers with a modular pseudorandom function that internally uses a block cipher. For this, we consider the sum of permutations [9]–[14], or more detailed the $XORP_w$ construction ($w \geq 1$) used in CENC [15]–[17]. In Section VI we consider the instantiation of $\mathcal{G}$ with the pseudorandom function $XORP_w$. The resulting construction is internally based on a block cipher, and achieves a much higher level of security than the naive instantiation using a block cipher.

It turns out that in these standard model analyses, the differences between $\mathcal{G}$ instantiated with $\mathbf{layer}_1$ and instantiated with $\mathbf{layer}_2$ are rather small. To precisely qualify the differences between the two layer functions, we next consider security of the stream cipher construction $\mathcal{G}$ in the ideal model in Section VII, both if based on an ideal cipher and on a random function. Although this ideal model approach might be debatable in light of the existence of our standard model results, ideal model analyses *do* give more detailed and more fine-grained bounds on the actual security levels. As we discuss in detail in Section VII-D1, the derived bounds for the two instantiations (with $\mathbf{layer}_1$ and $\mathbf{layer}_2$) with an ideal cipher E are comparable. This should not come as a surprise as block collisions dominate the security. For the instantiations with a random function, more surprising results turn up: for certain parameter choices, $\mathbf{layer}_1$ performs better, whereas for other parameter choices, $\mathbf{layer}_2$ performs better. We refer to Section VII-D2, and particularly Figure 5, for a clean visualization of this phenomenon.

B. Application

The two instantiations of our generalized stream cipher with a block cipher E generalize existing schemes in the literature:

- $\mathcal{G}$ based on $\mathbf{layer}_1$ instantiated with a block cipher E. This stream cipher is a variant of a key erasure random number generator recently introduced by Bernstein [4], where Bernstein's construction takes nonce 0 and can only be initialized once. More detailed: Bernstein's construction starts from a 256-bit key K, and uses it to generate blocks $B_0 = E_K(N), B_1 = E_K(N+1), \ldots, B_{\sigma+1} = E_K(N+\sigma+1)$ where $N = 0^n$ and E is the AES block cipher. After these block are generated, immediately overwrite the key K with the first two blocks B_0, B_1, and use the other blocks $B_2, \ldots, B_{\sigma+1}$ as the stream cipher output. Then start the next query with the new key. Bernstein already presented an analysis of a slight variant of his random number generator in [18]. Unfortunately, [18] only contains a proof sketch. The random number generator considered in [18] is also slightly different: the subkeys are generated in the first layer in the form $K_1 = E_K(0)\|E_K(1)$, $K_2 = E_K(2)\|E_K(3)$, etc. This conceptually simplifies the reasoning a bit, in that the lifting result of Section II-C is redundant. However, our main contribution is a generalized proof of the construction of [4], which is slightly different as the one of [18]. In the example of Bernstein, $\sigma = 46$. In this case, the computational load of refreshing the key and nonce is reasonably small compared to the amount of pseudorandom data that is produced. However, if we put $\sigma = 1$ (like in NIST's CTR_DRBG), the cost of refreshing the key and nonce becomes relatively high.
- $\mathcal{G}$ based on $\mathbf{layer}_2$ instantiated with a block cipher E. If we set $\sigma = 1$, this stream cipher corresponds to the encryption part of NIST's CTR_DRBG [3] that was already analyzed by Shrimpton and Terashima [5], [19]. Our scheme directly generalizes this result by introducing an arbitrary σ and by taking multi-user security into account.

Finally, $\mathcal{G}$ based on $\mathbf{layer}_1$ instantiated with a pseudorandom function F is a variant of AES-STREAM suggested and implemented by Denis [20]. This random number generator combines above-mentioned Bernstein's key erasure random number generator [4] with the FastPRF pseudorandom function of Mennink and Neves [8]. So far, the mode behind Denis' AES-STREAM has never been formally analyzed. We are not aware of any existing encryption mode that corresponds to $\mathbf{layer}_2$ based on a pseudorandom function, but as mentioned above and discussed in Section VII-D2, it turns out to improve over $\mathbf{layer}_1$ based on a random function for certain parameter choices.

C. Related Work

Bellare and Yee [2] already considered generic security of $\mathcal{G}$ of Figure 1, for instance instantiated with a pseudorandom function F. However, our description changes in the way key and nonce are updated, and in addition, we endeavor *tight* bounds that allow us to draw fine-grained conclusions on the security level of the scheme for a given use case. We remark that tight security bounds are important: security is typically dependent on many parameters (such as the number of evaluations, the amount of generated data per evaluation, the number of (re-)initializations of the system, etc.), and all parameters contribute to a security bound. To maintain a sufficient level of security, the advantage of breaking a scheme should be $\ll 1$. Looser bounds lead to unnecessary usage restrictions.

Shrimpton and Terashima [5], [19] considered Intel's hardware RNG in the context of robustness. The encryption part of that RNG matches our scheme for instantiation $\mathbf{layer}_2$ with $\sigma = 1$. They particularly prove that CTR_DRBG PRNG is *not* forward secure, seemingly contradicting our results. However, the security models of forward security are different. The main difference is at the implementation level of the scheme: the model of Shrimpton and Terashima allows to obtain a pseudorandom stream and subsequently the secret state used to generate this stream. In our model, the secret state is always updated *directly after* a pseudorandom stream is generated, and it is not possible to obtain the secret state of a stream that is already generated. Forward security in [5] is also achievable, as long as one is willing to make some reasonable assumptions about the adversary's limitations. Hence, the practical value of our model is that it shows that if we handle the secret state well, forward security is easily achievable. In a recent work, Woodage and Shumow [6] perform a detailed analysis of the other two PRNGs in NIST's SP 800-90A [3]: HMAC_DRBG and HASH_DRBG (of course omitting the retracted Dual_EC_DRBG design).

Bertoni et al. [21] proposed a way to design a PRNG from a sponge. Gaži and Tessaro [22] expanded their scheme to a robust PRNG with input, and Hutchington [23] presented Reverie, an improved and likewise robust version of the sponge-based PRNG.

Hamann et al. [24] proposed the LIZARD stream cipher, which defines another closely related line of research about stream ciphers with provable beyond birthday bound security. This research focuses more on hardware-based lightweight ciphers and uses similar information-theoretic methods as the one used in this work. More analysis of LIZARD was recently presented by Banik et al. [25] and Hamann and Krause [26].

II. Preliminaries

For $n \in \mathbb{N}$, denote by $\{0,1\}^n$ the set of all bit strings of length n. For a finite set S, denote by $s \xleftarrow{\$} \mathsf{S}$ that s gets sampled uniformly at random from S. For an algorithm $\mathcal{D}$ and an oracle $\mathcal{O}$, denote by $\mathcal{O}.X$ the global state variable X across the oracle $\mathcal{O}$, denote by $\mathcal{D}^{\mathcal{O}}$ the evaluation of $\mathcal{D}$ with oracle access to $\mathcal{O}$. We denote the advantage of $\mathcal{D}$ in distinguishing two oracles $\mathcal{O}$ and $\mathcal{P}$ by

$$\Delta_{\mathcal{D}}(\mathcal{O}\,;\,\mathcal{P}) = \left|\Pr\left[\mathcal{D}^{\mathcal{O}} = 1\right] - \Pr\left[\mathcal{D}^{\mathcal{P}} = 1\right]\right|. \qquad (1)$$

A. PRP and PRF

For $k, m, n \in \mathbb{N}$, denote by $\mathrm{Func}(m,n)$ the set of all functions with domain $\{0,1\}^m$ and range $\{0,1\}^n$, by $\mathrm{Perm}(n)$

the set of all permutations on $\{0,1\}^n$, and by $\mathrm{Perm}(k,n)$ the set of all families of permutations on $\{0,1\}^n$ indexed by keys from $\{0,1\}^k$.

A block cipher is a function $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ such that for fixed key $K \in \{0,1\}^k$, $E_K(\cdot) = E(K,\cdot)$ is a permutation on $\{0,1\}^n$. Its security is measured by considering a distinguisher $\mathcal{D}$ that is given access to either E_K for secret key $K \xleftarrow{\$} \{0,1\}^k$, or a random permutation $\pi \xleftarrow{\$} \mathrm{Perm}(n)$, and it is defined as:

$$\mathbf{Adv}^{\mathrm{prp}}_E(\mathcal{D}) = \Delta_{\mathcal{D}}\left(E_K\,;\,\pi\right). \tag{2}$$

For $q,t \in \mathbb{N}$, we denote

$$\mathbf{Adv}^{\mathrm{prp}}_E(q,t) = \max_{\mathcal{D}} \mathbf{Adv}^{\mathrm{prp}}_E(\mathcal{D})$$

as the maximum advantage over any distinguisher that can make at most q queries and that runs in time at most t.

A pseudorandom function (PRF) is a function $F : \{0,1\}^k \times \{0,1\}^m \to \{0,1\}^n$ such that for fixed key $K \in \{0,1\}^k$, $F_K(\cdot) = F(K,\cdot)$ is a function that maps $\{0,1\}^m$ to $\{0,1\}^n$. Its security is measured by considering a distinguisher $\mathcal{D}$ that is given access to either F_K for secret key $K \xleftarrow{\$} \{0,1\}^k$, or a random function $\phi \xleftarrow{\$} \mathrm{Func}(m,n)$, and it is defined as:

$$\mathbf{Adv}^{\mathrm{prf}}_F(\mathcal{D}) = \Delta_{\mathcal{D}}\left(F_K\,;\,\phi\right). \tag{3}$$

For $q,t \in \mathbb{N}$, we denote

$$\mathbf{Adv}^{\mathrm{prf}}_F(q,t) = \max_{\mathcal{D}} \mathbf{Adv}^{\mathrm{prf}}_F(\mathcal{D})$$

as the maximum advantage over any distinguisher that can make at most q queries and that runs in time at most t.

Here and throughout, we assume that offline evaluations of E and F always take *1 unit of time*.

B. PRP-PRF Switch

A well-known result states that a PRP behaves like a PRF up to the birthday bound in the number of construction queries.

Lemma II.1 (PRP-PRF switch [27])**.** *Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher. Then,*

$$\mathbf{Adv}^{\mathrm{prf}}_E(q,t) \le \binom{q}{2}/2^n + \mathbf{Adv}^{\mathrm{prp}}_E(q,t)\,.$$

C. Lifting Lemma

We pose a useful standard model lifting lemma. The lemma will be used to replace the standard primitive (in our proofs, a block cipher or a pseudorandom function) by a random permutation or function. The lemma follows by a simple hybrid argument, and it has been observed in several papers. Notably, the argument is implicit in the PRNG security proofs of Shrimpton and Terashima [5] and Woodage and Shumow [6]. Bernstein describes the idea in the context of his key erasure random number generator [4]. We are not aware of any published formal argument of this lifting lemma.

Lemma II.2 (lifting lemma)**.** *Let $f : \mathsf{K} \to \mathsf{X}$, $g : \mathsf{K} \to \mathsf{Y}$, and $h : \mathsf{Y} \to \mathsf{Z}$ be functions. Let $R_{\mathsf{K}} \xleftarrow{\$} \mathsf{K}$, $R_{\mathsf{X}} \xleftarrow{\$} \mathsf{X}$, $R_{\mathsf{Y}} \xleftarrow{\$} \mathsf{Y}$, and $R_{\mathsf{Z}} \xleftarrow{\$} \mathsf{Z}$. Then, for any distinguisher $\mathcal{D}$ running in time t,*

$$\begin{aligned}&\Delta_{\mathcal{D}}\left(f(R_{\mathsf{K}}), h(g(R_{\mathsf{K}}))\,;\,R_{\mathsf{X}}, R_{\mathsf{Z}}\right)\\ &\le \Delta_{\mathcal{D}'}\left(f(R_{\mathsf{K}}), g(R_{\mathsf{K}})\,;\,R_{\mathsf{X}}, R_{\mathsf{Y}}\right) + \Delta_{\mathcal{D}}\left(h(R_{\mathsf{Y}})\,;\,R_{\mathsf{Z}}\right),\end{aligned} \tag{4}$$

where $\mathcal{D}'$ is some distinguisher that runs in time t plus the time to evaluate h.

Proof. The proof follows by a simple hybrid argument:

$$\begin{aligned}&\Delta_{\mathcal{D}}\left(f(R_{\mathsf{K}}), h(g(R_{\mathsf{K}}))\,;\,R_{\mathsf{X}}, R_{\mathsf{Z}}\right)\\ &\quad\le \Delta_{\mathcal{D}}\left(f(R_{\mathsf{K}}), h(g(R_{\mathsf{K}}))\,;\,R_{\mathsf{X}}, h(R_{\mathsf{Y}})\right)\\ &\quad\quad+ \Delta_{\mathcal{D}}\left(R_{\mathsf{X}}, h(R_{\mathsf{Y}})\,;\,R_{\mathsf{X}}, R_{\mathsf{Z}}\right)\\ &\quad\le \Delta_{\mathcal{D}'}\left(f(R_{\mathsf{K}}), g(R_{\mathsf{K}})\,;\,R_{\mathsf{X}}, R_{\mathsf{Y}}\right) + \Delta_{\mathcal{D}}\left(h(R_{\mathsf{Y}})\,;\,R_{\mathsf{Z}}\right),\end{aligned}$$

where, in order for the reduction to go through, $\mathcal{D}'$ has to evaluate h in order to simulate the oracle of $\mathcal{D}$. □

III. Generalized Stream Cipher Construction

Let $k,n,\sigma \in \mathbb{N}$. A *layer function* takes as input a k-bit key and an n-bit nonce and returns a new key, new nonce, and a pseudorandom string of σn-bits. More formally, let $\mathbf{layer} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^k \times \{0,1\}^n \times \{0,1\}^{\sigma n}$ be a function such that

$$(K', N', S) \leftarrow \mathbf{layer}(K, N)\,.$$

We consider a generalized stream cipher $\mathcal{G}[\mathbf{layer}]_K$ that is a *stateful* ensemble of algorithms $(\mathcal{G}.\mathbf{init}, \mathcal{G}.\mathbf{next})$ as described in Algorithms 1 and 2. It is instantiated with **layer**, keyed with a master key $K \in \{0,1\}^k$, and maintains a state $(\mathcal{G}.K, \mathcal{G}.N)$. Here, $\mathcal{G}.K$ is the current session key which is updated in every iteration, and $\mathcal{G}.N$ is the current session nonce which may or may not have been updated depending on the used layer function. A user may call $\mathcal{G}.\mathbf{init}$ on input of a nonce to initialize the stream cipher, and per initialization it may call $\mathcal{G}.\mathbf{next}$ a restricted number of times to generate σ n-bit blocks per call. The user cannot evaluate $\mathcal{G}.\mathbf{next}$ without having evaluated $\mathcal{G}.\mathbf{init}$.

Algorithm 1 $\mathcal{G}.\mathbf{init}$

Input: $(K,N) \in \{0,1\}^k \times \{0,1\}^n$
Output: $\varnothing$
1: $(\mathcal{G}.K, \mathcal{G}.N) \leftarrow (K,N)$
2: **return**

Algorithm 2 $\mathcal{G}.\mathbf{next}$

Input: $\varnothing$
Output: $S \in \{0,1\}^{\sigma n}$
1: $(K,N,S) \leftarrow \mathbf{layer}(\mathcal{G}.K, \mathcal{G}.N)$
2: $(\mathcal{G}.K, \mathcal{G}.N) \leftarrow (K,N)$
3: **return** S

The stream cipher $\mathcal{G}$ simply specifies the two interfaces to the core function **layer**. In Sections III-A and III-B, we will discuss two prominent layer functions that we will consider in our work.

A. Layer Function 1

Let $\Pi \in \mathrm{Func}(k+n,n)$ be a function, with $k = 2n$, and consider layer function $\mathbf{layer}_1[\Pi]$ of Algorithm 3. The function is depicted in Figure 2a. Every evaluation generates

a new key K and a σn-bit stream S. The nonce N remains unchanged. The resulting scheme generalizes the random number generator of Bernstein [4], as explained in Section I-B.

Algorithm 3 $\mathbf{layer}_1[\Pi]$

Input: $(K, N) \in \{0,1\}^k \times \{0,1\}^n$
Output: $(K', N', S) \in \{0,1\}^k \times \{0,1\}^n \times \{0,1\}^{\sigma n}$
1: $K' \leftarrow \Pi_K(N) \parallel \Pi_K(N+1)$
2: $N' \leftarrow N$
3: $S \leftarrow \Pi_K(N+2) \parallel \cdots \parallel \Pi_K(N+\sigma+1)$
4: **return** (K', N', S)

B. Layer Function 2

Let $\Pi \in \mathrm{Func}(k+n, n)$ be a function, with $k = 2n$, and consider layer function $\mathbf{layer}_2[\Pi]$ of Algorithm 4. The function is depicted in Figure 2b. Every evaluation generates a new key K, a nonce N, and a σn-bit stream S. The resulting scheme generalizes the stream generation of NIST's CTR_DRBG [3], as explained in Section I-B.

Algorithm 4 $\mathbf{layer}_2[\Pi]$

Input: $(K, N) \in \{0,1\}^k \times \{0,1\}^n$
Output: $(K', N', S) \in \{0,1\}^k \times \{0,1\}^n \times \{0,1\}^{\sigma n}$
1: $K' \leftarrow \Pi_K(N) \parallel \Pi_K(N+1)$
2: $N' \leftarrow \Pi_K(N+2)$
3: $S \leftarrow \Pi_K(N+3) \parallel \cdots \parallel \Pi_K(N+\sigma+2)$
4: **return** (K', N', S)

IV. Security Notions

The first security property we require of our stream ciphers is pseudorandomness, stating that the output streams are hard to distinguish from random. In addition, as we particularly consider key erasure, we require forward security. This property dictates that, once a key of a certain layer leaks, the security of earlier layers is not affected. We follow the security formalism of Bellare and Yee [2]. The notions are, by default, considered in the multi-user security setting, where a distinguisher has access to $\mu \in \mathbb{N}$ instances. For $\mu = 1$, our models collapse to single-user security.

A. Streaming Oracle

Ideally, the streams generated by $\mathcal{G}[\mathbf{layer}]$ are indistinguishable from random. This boils down to saying that the output of $\mathcal{G}[\mathbf{layer}]$ is indistinguishable from a random bit string of the same length. This suggests to describe security as a distinguishing game between the output of $\mathcal{G}[\mathbf{layer}]$ and some random bit string. However, in the context of forward security, the ideal world should still be somehow able to reveal an internal state that is reminiscent of that of $\mathcal{G}[\mathbf{layer}]$. We resolve this by defining a general oracle $\mathcal{O}b$ that gets as input a bit $b \in \{0,1\}$, and outputs its (pseudo-)random blocks either as $\mathcal{G}[\mathbf{layer}]$ (if $b = 0$) or uniform random (if $b = 1$). We denote by $\mathcal{O}b_{K_i}$ the oracle $\mathcal{O}b$ initialized with a key K_i.

The oracle $\mathcal{O}b$ is formally described in Algorithms 5, 6, and 7. It is easy to see that $\mathcal{O}b$ fulfills our goals: for $b = 0$, it matches $\mathcal{G}[\mathbf{layer}]$ of Algorithms 1 and 2, whereas for $b = 1$, it always outputs random strings. In case of a leak, it gives the state of $\mathcal{G}[\mathbf{layer}]$.

Algorithm 5 $\mathcal{O}b$.**init**

Input: $(K, N) \in \{0,1\}^k \times \{0,1\}^n$
Output: $\varnothing$
1: $(\mathcal{O}b.K, \mathcal{O}b.N) \leftarrow (K, N)$
2: **return**

Algorithm 6 $\mathcal{O}b$.**next**

Input: $\varnothing$
Output: $S \in \{0,1\}^{\sigma n}$
1: $(K, N, S_0) \leftarrow \mathbf{layer}(\mathcal{O}b.K, \mathcal{O}b.N)$
2: $(\mathcal{O}b.K, \mathcal{O}b.N) \leftarrow (K, N)$
3: $S_1 \xleftarrow{\$} \{0,1\}^{\sigma n}$
4: **return** S_b

Algorithm 7 $\mathcal{O}b$.**leak**

Input: $\varnothing$
Output: $(K, N) \in \{0,1\}^k \times \{0,1\}^n$
1: **return** $(\mathcal{O}b.K, \mathcal{O}b.N)$

B. Distinguishing Advantages

Pseudorandomness and forward security are comparable, the only difference is in the capabilities of the distinguisher. As a general notion, we define the distinguishing advantage of a distinguisher.

Definition IV.1 (distinguishing advantage)**.** *Let* $k, n, \sigma \in \mathbb{N}$ *and let* $\mathbf{layer} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^k \times \{0,1\}^n \times \{0,1\}^{\sigma n}$ *be a layer function. For* $\mu \in \mathbb{N}$*, let* $K_1, K_2, \ldots, K_\mu \xleftarrow{\$} \{0,1\}^k$*. The multi-key advantage of a distinguisher* $\mathcal{D}$ *in distinguishing* $\mathcal{G}[\mathbf{layer}]$ *from random is defined as:*

$$\mathbf{Adv}^{\mathrm{dist}[\mu]}_{\mathcal{G}[\mathbf{layer}]}(\mathcal{D}) = \Delta_{\mathcal{D}}\left(\mathcal{O}0_{K_1}, \ldots, \mathcal{O}0_{K_\mu}\,;\, \mathcal{O}1_{K_1}, \ldots, \mathcal{O}1_{K_\mu}\right). \tag{5}$$

The first call of the distinguisher must be an ***init*** *call. A* ***leak*** *call should always follow a* ***next*** *call, and should never be followed by a* ***next*** *call. The distinguisher may never initialize any of its oracles with a repeating nonce* N *(additional conditions on the nonce may apply).*

Note that in above definition, the distinguisher is not allowed to call **leak** after **init**: this would result in the silly case of a master key being leaked. Also, a **leak** call should either be the last query, or be followed by an **init** call: the state should be re-initialized. The distinguisher has full freedom beyond these restrictions: it typically records all information it obtains.

Pseudorandomness is defined as follows.

Definition IV.2 (pseudorandomness)**.** *For* $\mu, \ell, q, t \in \mathbb{N}$*, we denote by*

$$\mathbf{Adv}^{\mathrm{prf}[\mu]}_{\mathcal{G}[\mathbf{layer}]}(\ell, q, t) = \max_{\mathcal{D}} \mathbf{Adv}^{\mathrm{dist}[\mu]}_{\mathcal{G}[\mathbf{layer}]}(\mathcal{D})$$

the maximum advantage over any distinguisher that, to each of its μ *oracles, can make at most* q ***init*** *calls, at most* ℓ ***next*** *calls per* ***init*** *call,* 0 ***leak*** *calls, and that runs in time at most* t*.*

Forward security is defined as follows.

Definition IV.3 (forward security)**.** *For $\mu, \ell, q, t \in \mathbb{N}$, we denote by*

$$\mathbf{Adv}^{\mathrm{fwd}[\mu]}_{\mathcal{G}[\mathbf{layer}]}(\ell, q, t) = \max_{\mathcal{D}} \mathbf{Adv}^{\mathrm{dist}[\mu]}_{\mathcal{G}[\mathbf{layer}]}(\mathcal{D})$$

*the maximum advantage over any distinguisher that, to each of its μ oracles, can make at most q **init** calls, at most ℓ **next** calls per **init** call, at most 1 **leak** call (per user), and that runs in time at most t.*

Note that once a **leak** call is made, the distinguisher knows the state of the construction. From that moment on, the construction is not pseudorandom anymore, and one needs to reinitialize the construction to obtain a new state. However, due to forward security, the distinguisher cannot learn anything from the previous states with the leaked state.

To gain understanding in the definitions, we point out that there is a relation between pseudorandomness and forward security. The reduction is obvious, noting that a distinguisher against the forward security may opt *not* to query **leak**.

Proposition IV.1. *Let $k, n, \sigma \in \mathbb{N}$, and $\mathbf{layer} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^k \times \{0,1\}^n \times \{0,1\}^{\sigma n}$ be a layer function. Let $\mu, \ell, q, t \in \mathbb{N}$. Then,*

$$\mathbf{Adv}^{\mathrm{prf}[\mu]}_{\mathcal{G}[\mathbf{layer}]}(\ell, q, t) \le \mathbf{Adv}^{\mathrm{fwd}[\mu]}_{\mathcal{G}[\mathbf{layer}]}(\ell, q, t)\,. \tag{6}$$

V. Security of $\mathcal{G}$ with $\mathbf{layer}_1$ or $\mathbf{layer}_2$

We consider the security of our generalized key erasure stream cipher $\mathcal{G}$ of Section III, both with the $\mathbf{layer}_1$ function of Section III-A or the $\mathbf{layer}_2$ function of Section III-B, and both if it is instantiated with a block cipher $E : \{0,1\}^{2n} \times \{0,1\}^n \to \{0,1\}^n$ or a pseudorandom function $F : \{0,1\}^{2n} \times \{0,1\}^n \to \{0,1\}^n$.

For $a = 1, 2$, we define

$$\mathcal{G}_{a,E} := \mathcal{G}[\mathbf{layer}_a[E]]\,, \text{ and } \mathcal{G}_{a,F} := \mathcal{G}[\mathbf{layer}_a[F]]\,.$$

Note that if the state of the stream cipher is (K, N), a **next** call in $\mathbf{layer}_1$ activates the evaluation of the underlying primitive on inputs N, $N+1$, …, $N+\sigma+1$. Related to this, we define

$$\mathbf{call}_1(N) = \{N, N+1, \ldots, N+\sigma+1\}\,.$$

Likewise, in $\mathbf{layer}_2$, a **next** call activates one extra call to the underlying primitive (as opposed to $\mathbf{layer}_1$), and we define

$$\mathbf{call}_2(N) = \{N, N+1, \ldots, N+\sigma+2\}\,. \tag{7}$$

We are now ready to prove security of $\mathcal{G}_{a,F}$ for $a = 1, 2$.

Theorem V.1 (security of $\mathcal{G}_{a,F}$ $(a = 1, 2)$)**.** *Let $n, \sigma \in \mathbb{N}$ and consider $\mathcal{G}_{a,F}$ for any pseudorandom function $F : \{0,1\}^{2n} \times \{0,1\}^n \to \{0,1\}^n$. Let $\mu, \ell, q, t \in \mathbb{N}$. Then,*

$$\begin{aligned}\mathbf{Adv}^{\mathrm{prf}[\mu]}_{\mathcal{G}_{a,F}}(\ell, q, t),\ &\mathbf{Adv}^{\mathrm{fwd}[\mu]}_{\mathcal{G}_{a,F}}(\ell, q, t)\\ &\le \mu \cdot \mathbf{Adv}^{\mathrm{prf}}_{F}(\sigma_a q, t + \sigma_a \mu \ell q)\\ &\quad + \mu(\ell - 1)q \cdot \mathbf{Adv}^{\mathrm{prf}}_{F}(\sigma_a, t + 2\sigma_a \mu \ell q)\,,\end{aligned} \tag{8}$$

provided that for any two distinct initializations of the same oracle, the nonces N, N' satisfy $\mathbf{call}_a(N) \cap \mathbf{call}_a(N') = \emptyset$. Here, σ_a equals $\sigma + 2$ for $a = 1$ and $\sigma + 3$ for $a = 2$.

The proof is given in Section V-A and strongly relies on the lifting result of Lemma II.2. In Section V-B, we discuss the tightness of the bound.

Security of $\mathcal{G}_{a,E}$ for $a = 1, 2$ follows as a direct consequence of the PRP-PRF switch of Lemma II.1. Tightness is discussed in Section V-C.

Corollary V.1.1 (security of $\mathcal{G}_{a,E}$ $(a = 1, 2)$)**.** *Let $n, \sigma \in \mathbb{N}$ and consider $\mathcal{G}_{a,E}$ for any block cipher $E : \{0,1\}^{2n} \times \{0,1\}^n \to \{0,1\}^n$. Let $\mu, \ell, q, t \in \mathbb{N}$. Then,*

$$\begin{aligned}\mathbf{Adv}^{\mathrm{prf}[\mu]}_{\mathcal{G}_{a,E}}(\ell, q, t),\ &\mathbf{Adv}^{\mathrm{fwd}[\mu]}_{\mathcal{G}_{a,E}}(\ell, q, t)\\ &\le \mu \cdot \mathbf{Adv}^{\mathrm{prp}}_{E}(\sigma_a q, t + \sigma_a \mu \ell q)\\ &\quad + \mu(\ell - 1)q \cdot \mathbf{Adv}^{\mathrm{prp}}_{E}(\sigma_a, t + 2\sigma_a \mu \ell q)\\ &\quad + \mu \binom{\sigma_a q}{2}/2^n + \mu(\ell - 1)q \binom{\sigma_a}{2}/2^n\,,\end{aligned} \tag{9}$$

provided that for any two distinct initializations of the same oracle, the nonces N, N' satisfy $\mathbf{call}_a(N) \cap \mathbf{call}_a(N') = \emptyset$. Here, σ_a equals $\sigma + 2$ for $a = 1$ and $\sigma + 3$ for $a = 2$.

A. Proof of Theorem V.1

Let $F : \{0,1\}^{2n} \times \{0,1\}^n \to \{0,1\}^n$, and focus on $\mathcal{G}_{1,F}$ (extension to $\mathcal{G}_{2,F}$ is discussed at the end). Consider μ master keys $K_1, K_2, \ldots, K_\mu \xleftarrow{\$} \{0,1\}^{2n}$, and any distinguisher $\mathcal{D}$ that has access to $\mathcal{O}b_{K_1}, \ldots, \mathcal{O}b_{K_\mu}$ for $b \in \{0,1\}$. It can initialize each of its μ oracles at most q times, and the layer function is evaluated at most ℓ times for each initialization. In addition, it runs in time at most t (in which time it can make offline evaluations of F). As a first hybrid step, we transform to the single-key setting:

$$\mathbf{Adv}^{\mathrm{dist}[\mu]}_{\mathcal{G}_{1,F}}(\mathcal{D}) \le \mu \cdot \mathbf{Adv}^{\mathrm{dist}[1]}_{\mathcal{G}_{1,F}}(\mathcal{D}')\,, \tag{10}$$

where $\mathcal{D}'$ operates in time $t' = t + (\sigma+2)(\mu-1)\ell q$ as it needs to simulate $\mathcal{D}$'s oracles. Denote the master key for $\mathcal{D}'$'s game by $K \xleftarrow{\$} \{0,1\}^{2n}$: the distinguisher's goal is to distinguish $\mathcal{O}0_K$ from $\mathcal{O}1_K$.

Without loss of generality (as we will maximize over all possible distinguishers in the end), denote the q nonces for which $\mathcal{D}'$ queries its oracles by $N^{(1)}, \ldots, N^{(q)}$, and write

$$\boldsymbol{N} = (N^{(1)}, \ldots, N^{(q)})\,.$$

The distance between $\mathcal{O}b_K$ for $b \in \{0,1\}$ equals the distance of the ℓq σn-bit blocks from uniform random, and we will use the lifting result of Lemma II.2 to bound this term. For any $N \in \{0,1\}^n$, define

$$\begin{aligned} f_N &: \{0,1\}^{2n} \to \{0,1\}^{\sigma n}\,,\\ &\quad K \mapsto F_K(N+2) \,\|\, \cdots \,\|\, F_K(N+\sigma+1)\,,\\ g_N &: \{0,1\}^{2n} \to \{0,1\}^{2n}\,,\\ &\quad K \mapsto F_K(N) \,\|\, F_K(N+1)\,. \end{aligned}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2021.3109302, IEEE Transactions on Information Theory

For $j \in [0, \ell-1]$, write $fg_N^j = f_N \circ g_N \circ \cdots \circ g_N$. For brevity of notation, we will define the following extensions of f_N and g_N, where $\boldsymbol{K} = (K^{(1)}, \ldots, K^{(q)}) \in (\{0,1\}^{2n})^q$:

$$\begin{aligned} f_{\boldsymbol{N}}(K) &= \begin{pmatrix} f_{N^{(1)}}(K) & \ldots & f_{N^{(q)}}(K) \end{pmatrix}, \\ f_{\boldsymbol{N}}(\boldsymbol{K}) &= \begin{pmatrix} f_{N^{(1)}}(K^{(1)}) & \ldots & f_{N^{(q)}}(K^{(q)}) \end{pmatrix}, \\ g_{\boldsymbol{N}}(K) &= \begin{pmatrix} g_{N^{(1)}}(K) & \ldots & g_{N^{(q)}}(K) \end{pmatrix}, \\ g_{\boldsymbol{N}}(\boldsymbol{K}) &= \begin{pmatrix} g_{N^{(1)}}(K^{(1)}) & \ldots & g_{N^{(q)}}(K^{(q)}) \end{pmatrix}. \end{aligned}$$

Write $fg_{\boldsymbol{N}}^j$ as before, noting that for $j \geq 1$, $g_{\boldsymbol{N}}^j(K) \in (\{0,1\}^{2n})^q$.

Let $S^{(i,j)} \xleftarrow{\$} \{0,1\}^{\sigma n}$ for $(i,j) \in [1,q] \times [1,\ell]$, and write

$$\boldsymbol{S}^{(j)} = \begin{pmatrix} S^{(1,j)} & \ldots & S^{(q,j)} \end{pmatrix}.$$

Then,

$$\mathbf{Adv}^{\mathrm{dist}[1]}_{\mathcal{G}_{1,F}}(\mathcal{D}') = \Delta_{\mathcal{D}'}\left(\begin{pmatrix} f_{\boldsymbol{N}}(K) \\ fg_{\boldsymbol{N}}(K) \\ \vdots \\ fg_{\boldsymbol{N}}^{\ell-1}(K) \end{pmatrix} ; \begin{pmatrix} \boldsymbol{S}^{(1)} \\ \boldsymbol{S}^{(2)} \\ \vdots \\ \boldsymbol{S}^{(\ell)} \end{pmatrix} \right). \quad (11)$$

1) The First Layer: For $\mathcal{G}_{1,F}$, the first layers are different from the remaining layers. The reason for this is that all first layers are generated using one and the same master key K (all $(\ell-1)q$ subsequently layers are supposedly generated for independent keys). Also for this reason, we have opted for the compact definitions of $f_{\boldsymbol{N}}(K)$, $f_{\boldsymbol{N}}(\boldsymbol{K})$, $g_{\boldsymbol{N}}(K)$, $g_{\boldsymbol{N}}(\boldsymbol{K})$: they hide a q-column structure within (11) and each of these q columns are initiated using $K \xleftarrow{\$} \{0,1\}^{2n}$. The application of the lifting result of Lemma II.2 to "eliminate" the usage of K has to be performed on the entire first row of (11). The trick of the lifting result consists of "replacing" $g_{\boldsymbol{N}}(K)$ by q $2n$-bit random values. This will make all subsequent layers mutually independent.

Let $K^{(i,2)} \xleftarrow{\$} \{0,1\}^{2n}$ for $i \in [1,q]$, and write

$$\boldsymbol{K}^{(2)} = \begin{pmatrix} K^{(1,2)} & \ldots & K^{(q,2)} \end{pmatrix}.$$

Define

$$\begin{aligned} h_{\boldsymbol{N}} : (\{0,1\}^{2n})^q &\to (\{0,1\}^{\sigma n})^{(\ell-1)q}, \\ \boldsymbol{K} &\mapsto \begin{pmatrix} f_{\boldsymbol{N}}(\boldsymbol{K}) \\ \vdots \\ fg_{\boldsymbol{N}}^{\ell-2}(\boldsymbol{K}) \end{pmatrix}. \end{aligned}$$

By the lifting result of Lemma II.2:

$$\begin{aligned} (11) \leq & \Delta_{\mathcal{D}^{(1)}}\left(\begin{pmatrix} f_{\boldsymbol{N}}(K) \\ g_{\boldsymbol{N}}(K) \end{pmatrix} ; \begin{pmatrix} \boldsymbol{S}^{(1)} \\ \boldsymbol{K}^{(2)} \end{pmatrix} \right) \\ & + \Delta_{\mathcal{D}'}\left(\begin{pmatrix} f_{\boldsymbol{N}}(\boldsymbol{K}^{(2)}) \\ \vdots \\ fg_{\boldsymbol{N}}^{\ell-2}(\boldsymbol{K}^{(2)}) \end{pmatrix} ; \begin{pmatrix} \boldsymbol{S}^{(2)} \\ \vdots \\ \boldsymbol{S}^{(\ell)} \end{pmatrix} \right), \quad (12) \end{aligned}$$

where $\mathcal{D}^{(1)}$ runs in time t' plus the time to simulate h. Assuming that evaluating F takes one unit of time, evaluating h takes at most $(\sigma+2)(\ell-1)q$ evaluations of F. Therefore,

$$\begin{aligned} & \Delta_{\mathcal{D}^{(1)}}\left(\begin{pmatrix} f_{\boldsymbol{N}}(K) \\ g_{\boldsymbol{N}}(K) \end{pmatrix} ; \begin{pmatrix} \boldsymbol{S}^{(1)} \\ \boldsymbol{K}^{(2)} \end{pmatrix} \right) \\ & \quad \leq \mathbf{Adv}^{\mathrm{prf}}_F((\sigma+2)q, t' + (\sigma+2)(\ell-1)q) \\ & \quad \leq \mathbf{Adv}^{\mathrm{prf}}_F((\sigma+2)q, t + (\sigma+2)\mu\ell q). \end{aligned}$$

Having eliminated the first layer, the matrices in the second term of (12) consist of q independent columns. For arbitrary N, $K^{(2)} \xleftarrow{\$} \{0,1\}^{2n}$, and $S^{(j)} \xleftarrow{\$} \{0,1\}^{\sigma n}$ for $j \in [2,\ell]$, it satisfies

$$\begin{aligned} & \Delta_{\mathcal{D}'}\left(\begin{pmatrix} f_{\boldsymbol{N}}(\boldsymbol{K}^{(2)}) \\ \vdots \\ fg_{\boldsymbol{N}}^{\ell-2}(\boldsymbol{K}^{(2)}) \end{pmatrix} ; \begin{pmatrix} \boldsymbol{S}^{(2)} \\ \vdots \\ \boldsymbol{S}^{(\ell)} \end{pmatrix} \right) \\ & \quad \leq q \cdot \Delta_{\mathcal{D}''}\left(\begin{pmatrix} f_N(K^{(2)}) \\ \vdots \\ fg_N^{\ell-2}(K^{(2)}) \end{pmatrix} ; \begin{pmatrix} S^{(2)} \\ \vdots \\ S^{(\ell)} \end{pmatrix} \right), \end{aligned}$$

where $\mathcal{D}''$ operates in time $t'' = t' + (\sigma+2)(\ell-1)(q-1)$ as it needs to simulate $\mathcal{D}'$'s oracles. Thus (12) simplifies to

$$\begin{aligned} (12) \leq & \mathbf{Adv}^{\mathrm{prf}}_F((\sigma+2)q, t + (\sigma+2)\mu\ell q) \\ & + q \cdot \Delta_{\mathcal{D}''}\left(\begin{pmatrix} f_N(K^{(2)}) \\ \vdots \\ fg_N^{\ell-2}(K^{(2)}) \end{pmatrix} ; \begin{pmatrix} S^{(2)} \\ \vdots \\ S^{(\ell)} \end{pmatrix} \right). \quad (13) \end{aligned}$$

2) The Subsequent Layers: Having resolved the more complicated first layer, the remainder is a piece of cake. A recursive application of the lifting result of Lemma II.2 yields for the remaining distance of (13):

$$\begin{aligned} & \Delta_{\mathcal{D}''}\left(\begin{pmatrix} f_N(K^{(2)}) \\ \vdots \\ fg_N^{\ell-2}(K^{(2)}) \end{pmatrix} ; \begin{pmatrix} S^{(2)} \\ \vdots \\ S^{(\ell)} \end{pmatrix} \right) \\ & \quad \leq \sum_{j=2}^{\ell-1} \Delta_{\mathcal{D}^{(j)}}\left(\begin{pmatrix} f_N(K^{(j)}) \\ g_N(K^{(j)}) \end{pmatrix} ; \begin{pmatrix} S^{(j)} \\ K^{(j+1)} \end{pmatrix} \right) \\ & \quad + \Delta_{\mathcal{D}^{(\ell)}}\left(f_N(K^{(\ell)}) ; S^{(\ell)} \right), \quad (14) \end{aligned}$$

where $K^{(j)} \xleftarrow{\$} \{0,1\}^{2n}$ for $j \in [3,\ell]$. The time complexities of $\mathcal{D}^{(j)}$ are t'' plus the time to simulate h at that recursion: at most $(\sigma+2)(\ell-j)q$. As before,

$$\begin{aligned} & \Delta_{\mathcal{D}^{(j)}}\left(\begin{pmatrix} f_N(K^{(j)}) \\ g_N(K^{(j)}) \end{pmatrix} ; \begin{pmatrix} S^{(j)} \\ K^{(j+1)} \end{pmatrix} \right) \\ & \quad \leq \mathbf{Adv}^{\mathrm{prf}}_F(\sigma+2, t'' + (\sigma+2)(\ell-j)q), \end{aligned}$$

for $j \in [2, \ell-1]$, and

$$\Delta_{\mathcal{D}^{(\ell)}}\left(f_N(K^{(\ell)}) ; S^{(\ell)} \right) \leq \mathbf{Adv}^{\mathrm{prf}}_F(\sigma, t'').$$

Thus (14) simplifies to

$$\begin{aligned}(14) &\le \sum_{j=2}^{\ell} \mathbf{Adv}^{\mathrm{prf}}_{F}(\sigma+2, t''+(\sigma+2)(\ell-j)q) \\ &\le (\ell-1)\cdot \mathbf{Adv}^{\mathrm{prf}}_{F}(\sigma+2, t''+(\sigma+2)(\ell-2)q) \\ &\le (\ell-1)\cdot \mathbf{Adv}^{\mathrm{prf}}_{F}(\sigma+2, t+2(\sigma+2)\mu\ell q)\,. \qquad (15)\end{aligned}$$

3) Conclusion: From Eqns. (10), (11), (12), (13), (14), and (15), we immediately obtain:

$$\begin{aligned}\mathbf{Adv}^{\mathrm{dist}[\mu]}_{\mathcal{G}_{1,F}}(\mathcal{D}) &\le \mu\cdot \mathbf{Adv}^{\mathrm{prf}}_{F}((\sigma+2)q, t+(\sigma+2)\mu\ell q) \\ &\quad + \mu(\ell-1)q\cdot \mathbf{Adv}^{\mathrm{prf}}_{F}(\sigma+2, t+2(\sigma+2)\mu\ell q)\,.\end{aligned}$$

Maximizing over all $\mathcal{D}$, we obtain that both worlds are perfectly indistinguishable up to above bound. The bound applies to both pseudorandomness and forward security.

4) Extension to $\mathcal{G}_{2,F}$: In $\mathcal{G}_{2,F}$, not only the key evolves over layers, but also the nonce. This, concretely, means that the functions f_N and g_N will not be labeled by N anymore. Instead, they are functions that operate on a state from $\{0,1\}^{2n}\times\{0,1\}^n$ rather than $\{0,1\}^{2n}$, and any layer makes $\sigma+3$ primitive queries rather than $\sigma+2$. The remaining analysis is identical, noting that the core of the reduction is independence of keys across different evaluations of **next**.

B. Tightness of Theorem V.1

The bound of Theorem V.1 is pretty tight, barring any loss incurred by the use of the generic lifting result of Lemma II.2. If $F \xleftarrow{\$} \mathrm{Func}(k+n, n)$, then for $q \ge 1$, we have

$$\mathbf{Adv}^{\mathrm{prf}}_{F}(q,t) \approx t/2^k\,,$$

assuming that any evaluation of F takes 1 unit of time. In this case, for $\mathcal{G}_{1,F}$, the bound of (8) simplifies to

$$\frac{2(\sigma+2)\mu^2\ell^2q^2}{2^{2n}} + \frac{\mu\ell qt}{2^{2n}}\,.$$

The first term corresponds to collisions among the subkeys: all layers are assumed to be evaluated with independent keys, with the exception of the first layers after the initializations, which are all evaluated with the same master key. This means that, among all $\mu(\ell-1)q+\mu$ keys, the scheme exhibits non-random behavior if there are two colliding keys.

The second term corresponds to the time-memory trade-off attacks of Biham [28], [29], which demonstrates that if a distinguisher has access to μ oracles, can make ℓq calls per oracle, and can make t offline primitive queries (equating time with primitive evaluations), it succeeds in recovering *a* key with high probability if $\mu\ell qt \approx 2^{2n}$.

The bound admits a slight loss, most notable due to simplifications in the bounding of the time complexities of the distinguishers in the reduction. This also results in the fact that, contrary to intuition, the bound of $\mathcal{G}_{2,F}$ is worse than that of $\mathcal{G}_{1,F}$. In Section VII, we will consider security of the modes in the ideal primitive model, and derive more accurate and fine-grained security bounds.

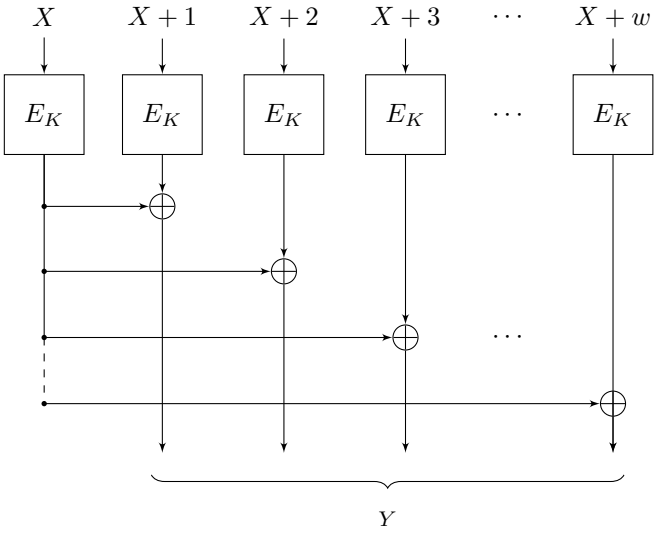


Figure 3: The $XORP_w[E]$ construction of (16).

C. Tightness of Corollary V.1.1

The security bound of $\mathcal{G}_{1,E}$ differs from that of $\mathcal{G}_{1,F}$ in the addition of

$$\mu\binom{(\sigma+2)q}{2}/2^n + \mu(\ell-1)q\binom{\sigma+2}{2}/2^n\,,$$

due to the PRP-PRF switch. Recall that for q evaluations of $\mathcal{G}_{1,E}$, all first layers after the initializations are performed for identical master key, but all subsequent layers are mutually independent. This means that there are μ keys for which the underlying block cipher is evaluated $(\sigma+2)q$ times and $\mu(\ell-1)q$ keys for which the underlying block cipher is evaluated $\sigma+2$ times, precisely matching above term.

VI. Instantiating the Pseudorandom Function

Iwata [15] introduced the CENC construction, a beyond the birthday bound encryption scheme based on a block cipher $E:\{0,1\}^k\times\{0,1\}^n\to\{0,1\}^n$. The construction internally uses a pseudorandom function now known as the $XORP_w$ for $w\ge 1$ (see also Figure 3):

$$\begin{aligned}&XORP_w[E](K,X) \\ &= E_K(X)\oplus E_K(X+1)\parallel\cdots\parallel E_K(X)\oplus E_K(X+w)\,.\end{aligned} \qquad (16)$$

For $w=2$, it is equal to the sum of permutations [9]–[14]. For arbitrary $w\ge 2$, Iwata [15] proved security up to $2^{2n/3}$. Iwata et al. [16] proved optimal $2^n/w$ security using the Mirror Theory [13], [30], [31], and Bhattacharya and Nandi [17] derived a comparable bound using the Chi Squared Technique [14].

Lemma VI.1 (Bhattacharya and Nandi [17])**.** *Let $n, w\in\mathbb{N}$ and consider $XORP_w[E]$ of (16) for any block cipher $E:\{0,1\}^{2n}\times\{0,1\}^n\to\{0,1\}^n$. Let $q,t\in\mathbb{N}$. Then,*

$$\begin{aligned}&\mathbf{Adv}^{\mathrm{prf}}_{XORP_w[E]}(q,t) \\ &\qquad \le \mathbf{Adv}^{\mathrm{prp}}_{E}((w+1)q, t') + \frac{(1+\sqrt{2})(w+1)^2q}{2^n}\,, \qquad (17)\end{aligned}$$

with $t' = O(t)$.

A. Security of $\mathcal{G}$ Instantiated with XORP

The result of Lemma VI.1 shows that for any $w \in \mathbb{N}$, the function $XORP_w[E] : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^{wn}$ is as secure as a parallel evaluation of a random function $F \xleftarrow{\$} \mathrm{Func}(k+n, n)$ under w distinct inputs, up to around $2^n/w^2$. It would be tempting to take $\mathcal{G}$ of Section III with $\mathbf{layer}_1[XORP_w[E]]$ or $\mathbf{layer}_2[XORP_w[E]]$, but unfortunately, that hybrid construction does not work nicely: both would have a key of $2wn$ bits. Instead, we have to adapt the formalization of the layers slightly so as to suit instantiation with the wn-bit $XORP_w[E]$. We will exemplify this for $\mathbf{layer}_1$.

Recall that $\mathbf{layer}_1$ calls its underlying primitive on inputs $N, \ldots, N+\sigma+1$, and assume w.l.o.g. that $\sigma+2$ is a multiple of w. The adjusted $\mathbf{layer}'_1[\Pi]$ for $\Pi \in \mathrm{Func}(k+n, wn)$ gets as input (K, N) and simply makes $(\sigma+2)/w$ calls to Π on input $(K, N), (K, N+w+1), \ldots, (K, N+(\frac{\sigma+2}{w}-1)w+1)$. Related to this, we write $\mathcal{G}_{1,XORP_w[E]} := \mathcal{G}[\mathbf{layer}'_1[XORP_w[E]]]$, and define

$$\mathbf{call}'_1(N) = \{N, N+w+1, \ldots, N+\left(\frac{\sigma+2}{w}-1\right)w+1\}\,.$$

We can do the same for $\mathbf{layer}_2$, now assuming that $\sigma+3$ is a multiple of w.

In either case, a simple hybrid argument shows that, for $k = 2n$, the functions $\mathcal{G}_{a,F}$ and $\mathcal{G}_{a,XORP_w[E]}$ are equally secure up to the bound of Lemma VI.1. Here, we note, however, that the latter only makes σ_a/w calls to $XORP_w[E]$ and thus $\sigma_a(w+1)/w$ calls to E.

Corollary VI.1.1 (security of $\mathcal{G}_{a,XORP_w[E]}$ $(a=1,2)$)**.** *Let $n, \sigma, w \in \mathbb{N}$ and consider $\mathcal{G}_{a,XORP_w[E]}$ for any block cipher $E : \{0,1\}^{2n} \times \{0,1\}^n \to \{0,1\}^n$. Let $\mu, \ell, q, t \in \mathbb{N}$. Then,*

$$\begin{aligned}
&\mathbf{Adv}^{\mathrm{prf}[\mu]}_{\mathcal{G}_{a,XORP_w[E]}}(\ell,q,t),\ \mathbf{Adv}^{\mathrm{fwd}[\mu]}_{\mathcal{G}_{a,XORP_w[E]}}(\ell,q,t)\\
&\quad\le \mu\cdot\mathbf{Adv}^{\mathrm{prp}}_E(\sigma_a(w+1)/wq, t+\sigma_a/w\mu\ell q)\\
&\quad+\mu(\ell-1)q\cdot\mathbf{Adv}^{\mathrm{prp}}_E(\sigma_a(w+1)/w, t+2\sigma_a/w\mu\ell q)\\
&\quad+\mu\cdot\frac{(1+\sqrt{2})\sigma_a(w+1)^2/wq}{2^n}\\
&\quad+\mu(\ell-1)q\cdot\frac{(1+\sqrt{2})\sigma_a(w+1)^2/w}{2^n}\,,
\end{aligned}\tag{18}$$

provided that for any two distinct initializations of the same oracle, the nonces N, N' satisfy $\mathbf{call}'_a(N) \cap \mathbf{call}'_a(N') = \emptyset$. Here, σ_a equals $\sigma+2$ for $a=1$ and $\sigma+3$ for $a=2$.

B. Comparison with Block Cipher Based Schemes

We now have, for both $\mathbf{layer}_1$ and $\mathbf{layer}_2$, two block cipher based schemes: $\mathcal{G}_{a,E}$ and $\mathcal{G}_{a,XORP_w[E]}$ for $a = 1, 2$. The former makes σ_a block cipher calls per layer, whereas the latter makes $\sigma_a(w+1)/w$ calls. In addition, the latter has to maintain n bits of extra state to implement $XORP_w$. However, the $\mathcal{G}_{a,XORP_w[E]}$ *does* seem to achieve a higher level of security. In this section, we will make a more detailed comparison.

We will consider $\mathcal{G}_{a,E}$ and $\mathcal{G}_{a,XORP_w[E]}$ for $E \xleftarrow{\$} \mathrm{Perm}(2n, n)$. Note that for any such E, we have $\mathbf{Adv}^{\mathrm{prp}}_E(q,t) = t/2^{2n}$, discarding constants and assuming that any evaluation of E costs one unit of time. In this case, the bound of $\mathcal{G}_{a,E}$ of Corollary V.1.1 simplifies to (omitting constants)

$$\mathbf{Adv}^{\mathrm{prf/fwd}[\mu]}_{\mathcal{G}_{1,E}}(\ell,q,t) \le \frac{\sigma_a\mu^2\ell^2q^2}{2^{2n}} + \frac{\mu\ell qt}{2^{2n}} + \frac{\sigma_a^2\mu q(\ell+q)}{2^n}\,.$$

The bound of $\mathcal{G}_{a,XORP_w[E]}$ of Corollary VI.1.1 likewise simplifies to

$$\mathbf{Adv}^{\mathrm{prf/fwd}[\mu]}_{\mathcal{G}_{1,XORP_w[E]}}(\ell,q,t) \le \frac{\sigma_a\mu^2\ell^2q^2}{2^{2n}} + \frac{\mu\ell qt}{2^{2n}} + \frac{\sigma_a\mu\ell q}{2^n}\,.$$

Here, the value w is considered constant.

The security gain of using $\mathcal{G}_{1,XORP_w[E]}$ over $\mathcal{G}_{1,E}$ is immediate: in the original $\mathcal{G}_{1,E}$, the dominating term is of the form $\mu q^2/2^n$, i.e., security is only guaranteed up to around $2^{n/2}/\mu^{1/2}$ queries. In the case of $\mathcal{G}_{1,XORP_w[E]}$, the dominating term lasts until around $2^n/(\mu\ell)$ queries.

VII. Ideal Model Security

We study the security of $\mathcal{G}_{1,E}$, $\mathcal{G}_{1,F}$, $\mathcal{G}_{2,E}$, and $\mathcal{G}_{2,F}$ in the ideal model, meaning that the underlying block cipher or pseudorandom function is assumed to be perfectly random. This is a stronger security model, yet, it allows to perform a more fine-grained analysis. In particular, whereas the differences between the security bounds between $\mathcal{G}_{1,E}$ and $\mathcal{G}_{2,E}$ are negligible in the standard model (the same holds for $\mathcal{G}_{1,F}$ versus $\mathcal{G}_{2,F}$), in the ideal model the differences are more accurately derived.

A. Ideal Model Security Notions

For $k, n \in \mathbb{N}$, denote by $\mathrm{Perm}(k,n)$ the set of all block ciphers on $\{0,1\}^n$ keyed with a key from $\{0,1\}^k$. Note that $\mathrm{Perm}(k,n) \subseteq \mathrm{Func}(k+n, n)$.

The security definitions in the ideal model are very similar to those of Section IV, the only major difference is that the primitive Π used in $\mathbf{layer}$ is idealized and the distinguisher has query access to Π. It can make a total amount of $r \in \mathbb{N}$ queries. There is no need to limit the time complexity of $\mathcal{D}$ anymore; as of now, it is considered to be information-theoretic, and its complexity is measured by the amount of queries only.

The formal definitions of ideal model distinguishing advantage (i-dist), pseudorandomness (i-prf), and forward security (i-fwd) are given below.

Definition VII.1 (ideal model distinguishing advantage)**.** *Let $k, n, \sigma \in \mathbb{N}$, let $\Pi \xleftarrow{\$} \mathrm{Prims}$ be a primitive randomly selected from some finite set of primitives Prims, and $\mathbf{layer}[\Pi] : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^k \times \{0,1\}^n \times \{0,1\}^{\sigma n}$ be a layer function based on Π. For $\mu \in \mathbb{N}$, let $K_1, K_2, \ldots, K_\mu \xleftarrow{\$} \{0,1\}^k$. The multi-key advantage of a distinguisher $\mathcal{D}$ in distinguishing $\mathcal{G}[\mathbf{layer}]$ from random is defined as:*

$$\begin{aligned}
&\mathbf{Adv}^{\text{i-dist}[\mu]}_{\mathcal{G}[\mathbf{layer}[\Pi]]}(\mathcal{D}) =\\
&\Delta_{\mathcal{D}}\left(\mathcal{O}0[\Pi]_{K_1}, \ldots, \mathcal{O}0[\Pi]_{K_\mu}, \Pi\,;\, \mathcal{O}1[\Pi]_{K_1}, \ldots, \mathcal{O}1[\Pi]_{K_\mu}, \Pi\right).
\end{aligned}\tag{19}$$

The distinguisher has two-sided access to the underlying primitive Π *if* Π *is a block cipher, and forward access only if* Π *is a pseudorandom function. The first call of the distinguisher must be an* ***init*** *call. A* ***leak*** *call should always follow a* ***next*** *call, and should never be followed by a* ***next*** *call. The distinguisher may never initialize any of its oracles with a repeating nonce* N *(additional conditions on the nonce may apply).*

As before, the distinguisher is not allowed to call **leak** after **init**. Ideal model pseudorandomness and forward security are defined as follows.

Definition VII.2 (ideal model pseudorandomness)**.** *For* $\ell, q, r \in \mathbb{N}$*, we denote by*

$$\mathbf{Adv}^{\text{i-prf}[\mu]}_{\mathcal{G}[\mathbf{layer}[\Pi]]}(\ell, q, r) = \max_{\mathcal{D}} \mathbf{Adv}^{\text{i-dist}[\mu]}_{\mathcal{G}[\mathbf{layer}[\Pi]]}(\mathcal{D})$$

the maximum advantage over any distinguisher that, to each of its μ *oracles, can make at most* q ***init*** *calls, at most* ℓ ***next*** *calls per* ***init*** *call,* 0 ***leak*** *calls, and at most* r *calls to the underlying primitive.*

Definition VII.3 (ideal model forward security)**.** *For* $\ell, q, r \in \mathbb{N}$*, we denote by*

$$\mathbf{Adv}^{\text{i-fwd}[\mu]}_{\mathcal{G}[\mathbf{layer}[\Pi]]}(\ell, q, r) = \max_{\mathcal{D}} \mathbf{Adv}^{\text{i-dist}[\mu]}_{\mathcal{G}[\mathbf{layer}[\Pi]]}(\mathcal{D})$$

the maximum advantage over any distinguisher that, to each of its μ *oracles, can make at most* q ***init*** *calls, at most* ℓ ***next*** *calls per* ***init*** *call, at most* 1 ***leak*** *call (in total), and at most* r *calls to the underlying primitive.*

B. Ideal Model Security of $\mathcal{G}$ with $\mathbf{layer}_1$

We consider security of our generalized key erasure stream cipher $\mathcal{G}$ of Section III with the $\mathbf{layer}_1$ function of Section III-A, both if it is instantiated with an ideal cipher $E \xleftarrow{\$} \mathrm{Perm}(2n, n)$ or a random function $F \xleftarrow{\$} \mathrm{Func}(3n, n)$. The proofs are performed in an information-theoretic setting (i.e., using actual random primitives as components).

Theorem VII.1 (ideal model security of $\boldsymbol{\mathcal{G}_{1,E}}$)**.** *Let* $n, \sigma \in \mathbb{N}$ *and consider* $\mathcal{G}_{1,E}$ *with* $E \xleftarrow{\$} \mathrm{BC}(2n, n)$*. Let* $\mu, \ell, q, r \in \mathbb{N}$*. Then,*

$$\mathbf{Adv}^{\text{i-prf}[\mu]}_{\mathcal{G}_{1,E}}(\ell, q, r),\ \mathbf{Adv}^{\text{i-fwd}[\mu]}_{\mathcal{G}_{1,E}}(\ell, q, r) \le \frac{\mu^2\ell^2q^2}{2^{2n}} + \frac{\mu\ell qr}{2^{2n-1}} + \frac{(\sigma+2)^2\mu(\ell+q)q}{2^{n+1}}, \quad (20)$$

provided that for any two distinct initializations of the same oracle, the nonces N, N' *satisfy* $\mathbf{call}_1(N) \cap \mathbf{call}_1(N') = \emptyset$*.*

The proof is given in Supplementary Material A. We will interpret the bound in Section VII-D.

The last term in (20) is the only one with 2^n in the denominator, which makes it the dominating term in Theorem VII.1. This term is the probability of an n-bit block collision, and therefore, in the security analysis we have defined the remaining bad events in a looser and simpler way. If, instead, $\mathcal{G}[\mathbf{layer}_1[F]]$ is initiated with a random function $F \xleftarrow{\$} \mathrm{Func}(3n, n)$, block collisions are no problem anymore and it is beneficial to use slightly more involved bad events. This yields the following theorem, as one can see from equation (22), this bound is not affected by the block collision, every term has 2^{2n} in the denominator. The proof of Theorem VII.2 is given in Supplementary Material B. Also these bounds are interpreted in Section VII-D.

Theorem VII.2 (ideal model security of $\boldsymbol{\mathcal{G}_{1,F}}$)**.** *Let* $n, \sigma \in \mathbb{N}$ *and consider* $\mathcal{G}_{1,F}$ *with* $F \xleftarrow{\$} \mathrm{Func}(3n, n)$*. Let* $\mu, \ell, q, r \in \mathbb{N}$*. Then,*

$$\mathbf{Adv}^{\text{i-prf}[\mu]}_{\mathcal{G}_{1,F}}(\ell, q, r) \le \frac{(2\sigma+3)\mu^2\ell^2q}{2^{2n}} + \frac{\mu\ell r}{2^{2n}}, \quad (21)$$

$$\mathbf{Adv}^{\text{i-fwd}[\mu]}_{\mathcal{G}_{1,F}}(\ell, q, r) \le \frac{(2\sigma+3)\mu^2\ell^2q}{2^{2n}} + \frac{\mu\ell r}{2^{2n}} + \frac{\mu(\ell-1)q+\mu}{2^{2n}}, \quad (22)$$

provided that for any two distinct initializations of the same oracle, the nonces N, N' *satisfy* $\mathbf{call}_1(N) \cap \mathbf{call}_1(N') = \emptyset$*.*

C. Ideal Model Security of $\mathcal{G}$ with $\mathbf{layer}_2$

Next, we consider $\mathcal{G}$ of Section III with the $\mathbf{layer}_2$ function of Section III-B, again both in the case it is instantiated with an ideal cipher or a random function. We can derive comparable results for $\mathcal{G}_{2,E}$ and $\mathcal{G}_{2,F}$. The proofs are given in Supplementary Material C and Supplementary Material D, respectively. The bounds are interpreted in Section VII-D.

Theorem VII.3 (ideal model security of $\boldsymbol{\mathcal{G}_{2,E}}$)**.** *Let* $n, \sigma \in \mathbb{N}$ *and consider* $\mathcal{G}_{2,E}$ *with* $E \xleftarrow{\$} \mathrm{BC}(2n, n)$*. Let* $\mu, \ell, q, r \in \mathbb{N}$*. Then,*

$$\mathbf{Adv}^{\text{i-prf}[\mu]}_{\mathcal{G}_{2,E}}(\ell, q, r),\ \mathbf{Adv}^{\text{i-fwd}[\mu]}_{\mathcal{G}_{2,E}}(\ell, q, r) \le \frac{\mu^2\ell^2q^2}{2^{2n}} + \frac{\mu\ell qr}{2^{2n-1}} + \frac{(\sigma+3)^2\mu(\ell+q)q}{2^{n+1}} + \frac{(2\sigma+5)\mu^2\ell^2q^2}{2^{3n-1}}, \quad (23)$$

provided that for any two distinct initializations of the same oracle, the nonces N, N' *satisfy* $\mathbf{call}_2(N) \cap \mathbf{call}_2(N') = \emptyset$*.*

Theorem VII.4 (ideal model security of $\boldsymbol{\mathcal{G}_{2,F}}$)**.** *Let* $n, \sigma \in \mathbb{N}$ *and consider* $\mathcal{G}_{2,F}$ *with* $F \xleftarrow{\$} \mathrm{Func}(3n, n)$*. Let* $\mu, \ell, q, r \in \mathbb{N}$*. Then,*

$$\mathbf{Adv}^{\text{i-prf}[\mu]}_{\mathcal{G}_{2,F}}(\ell, q, r) \le \frac{\mu^2}{2^{2n+1}} + \frac{(2\sigma+5)\mu^2\ell^2q^2}{2^{3n}} + \frac{\mu r}{2^{2n}} + \frac{(\sigma+3)\mu(\ell-1)qr}{2^{3n}}, \quad (24)$$

$$\mathbf{Adv}^{\text{i-fwd}[\mu]}_{\mathcal{G}_{2,F}}(\ell, q, r) \le \frac{\mu^2}{2^{2n+1}} + \frac{(2\sigma+5)\mu^2\ell^2q^2}{2^{3n}} + \frac{\mu r}{2^{2n}} + \frac{(\sigma+3)\mu(\ell-1)qr}{2^{3n}} + \frac{\mu(\ell-1)q+\mu}{2^{2n}}, \quad (25)$$

provided that for any two distinct initializations of the same oracle, the nonces N, N' *satisfy* $\mathbf{call}_2(N) \cap \mathbf{call}_2(N') = \emptyset$*.*

D. Interpretation

In this section, we interpret the bounds for the block cipher based schemes and the bounds for the random function based schemes, both bounds are achieved using the ideal cipher model.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2021.3109302, IEEE Transactions on Information Theory

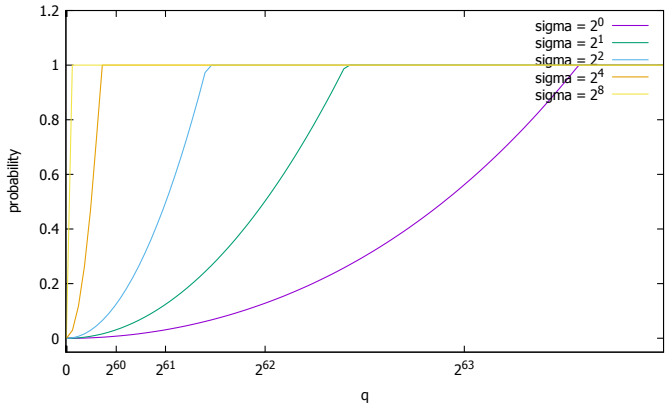


Figure 4: The bound of (26) plotted as a function of q, for $n = 128$, $\sigma \in \{2^0, 2^1, 2^2, 2^4, 2^8\}$ (right to left), $\mu = 1$, $\ell = q$, and $r = \ell q$.

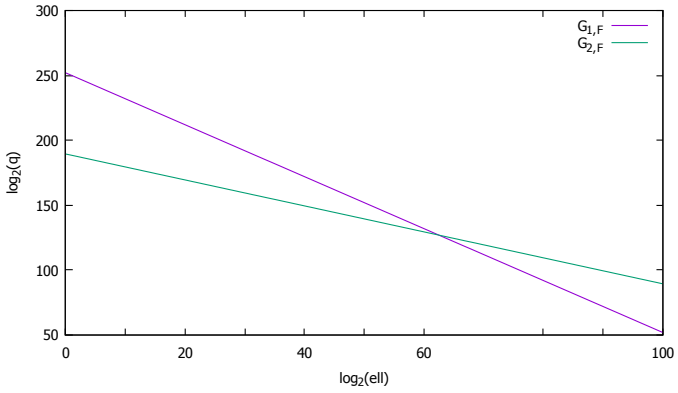


Figure 5: The bounds of (27) and (29), for $n = 128$, $\sigma = 2^4$, $\mu = 1$, and $r = \ell q$. The two bounds are equated to 1, and interpreted as $\log_2(q)$ as function of $\log_2(\ell)$.

1) Block Cipher Based Schemes: Discarding the difference between $\sigma + 2$ versus $\sigma + 3$, for both pseudorandomness and forward security the bounds of both $\mathcal{G}_{1,E}$ (Theorem VII.1) and $\mathcal{G}_{2,E}$ (Theorem VII.3) are of the form

$$\frac{\mu\ell qr}{2^{2n-1}} + \frac{\sigma^2\mu(\ell+q)q}{2^n}. \tag{26}$$

This bound is tight, as explained in Section A.

In Figure 4, we plot this bound for $n = 128$ and various parameters of σ, μ, ℓ, r. Noting that (26) is linear in μ, we restrict our focus to $\mu = 1$. Noting that the second term will dominate, ℓ will not significantly influence the bound, and we simply set $\ell = q$. We assume that $r = \ell q$. The plots are given for $\sigma \in \{2^0, 2^1, 2^2, 2^4, 2^8\}$ and are, subsequently, a function in q.

2) Random Function Based Schemes: The bounds of $\mathcal{G}_{1,F}$ (Theorem VII.2) and $\mathcal{G}_{2,F}$ (Theorem VII.4) expose small differences. Discarding constants and assuming that $\mu \leq r$, the bounds are of the following form:

$$\mathbf{Adv}^{\mathrm{prf/fwd}[\mu]}_{\mathcal{G}_{1,F}}(\ell, q, r) \lesssim \frac{\sigma\mu^2\ell^2 q}{2^{2n}} + \frac{\mu\ell r}{2^{2n}}, \tag{27}$$

$$\mathbf{Adv}^{\mathrm{prf}[\mu]}_{\mathcal{G}_{2,F}}(\ell, q, r) \lesssim \frac{\sigma\mu^2\ell^2 q^2}{2^{3n}} + \frac{\mu r}{2^{2n}} + \frac{\sigma\mu\ell qr}{2^{3n}}, \tag{28}$$

$$\mathbf{Adv}^{\mathrm{fwd}[\mu]}_{\mathcal{G}_{2,F}}(\ell, q, r) \lesssim \frac{\sigma\mu^2\ell^2 q^2}{2^{3n}} + \frac{\mu r}{2^{2n}} + \frac{\sigma\mu\ell qr}{2^{3n}} + \frac{\mu\ell q}{2^{2n}}. \tag{29}$$

In Figure 5, we plot the forward security bounds (27) and (29). In this case, plotting the probability as function in q is not so informative. Therefore, we depict the bounds in a different way. We consider $n = 128$, $\sigma = 2^4$, $\mu = 1$, and $r = \ell q$. We consider various possibilities of ℓ and compute q for which the right hand side of (27) resp. (29) equals 1. For example, if we consider $\ell = 2^{20}$, the dominant term of (27) equals 1 for $q = 2^{212}$. The plots, as such, consider q as a function of ℓ, or more formally $\log_2(q)$ as a function of $\log_2(\ell)$.

Unlike for the block cipher based case, the plots of Figure 5 give a more surprising picture: for small values of ℓ, $\mathcal{G}_{1,F}$ performs better and for larger values of ℓ, $\mathcal{G}_{2,F}$ performs better. This is mainly caused by the first term in the bounds, and becomes apparent by comparing the terms for ℓ constant and for $\ell \approx q$. In the former case (ℓ constant) $\mathcal{G}_{1,F}$ achieves around $2^{2n}/\mathrm{const}$ security and $\mathcal{G}_{2,F}$ only $2^{3n/2}/\mathrm{constant}$. In the latter case ($\ell \approx q$), $\mathcal{G}_{1,F}$ achieves $2^{2n/3}$ security and $\mathcal{G}_{2,F}$ reaches $2^{3n/4}$.

VIII. Conclusion

In general, forward security comes at only a marginal cost over pseudorandomness, the only exception being $\mathcal{G}_{2,F}$. This is, perhaps, not surprising in light of the definitions of pseudorandomness and forward security and their relation given in Proposition IV.1. The observation suggests that the bound of Proposition IV.1 is reasonably tight.

For the block cipher based stream ciphers, the security bounds of $\mathcal{G}_{1,E}$ and $\mathcal{G}_{2,E}$ are comparable: this is because the dominating term is the one corresponding to block collisions, and in both schemes the amount of generated blocks is of the same order. The term disappears when we move to pseudorandom based instantiations $\mathcal{G}_{1,F}$ and $\mathcal{G}_{2,F}$. We have made use of this in our instantiation of F as $XORP_w$ in Section VI, obtaining a block cipher based scheme akin to CENC encryption [15] that achieves security well beyond the birthday bound.

For random function based schemes, generically $\mathcal{G}_{2,F}$ performs better than $\mathcal{G}_{1,F}$. Yet, for specific parameters of ℓ, the situation is reversed. This is because $\mathcal{G}_{2,F}$ generates random nonces every round; these nonces may collide and give the distinguisher additional power in breaking the pseudorandomness and forward security.

Appendix A
Proof of Theorem VII.1 ($\mathcal{G}_{1,E}$)

Let $E \xleftarrow{\$} \mathrm{BC}(2n, n)$, and consider μ master keys $K_1, K_2, \ldots, K_\mu \xleftarrow{\$} \{0,1\}^{2n}$. Consider any distinguisher $\mathcal{D}$ that has access to $\mathcal{O}b[E]_{K_1}, \cdots, \mathcal{O}b[E]_{K_\mu}$ for $b \in \{0,1\}$. It can initialize each of its μ oracles at most q times, and the layer function is evaluated at most ℓ times for each initialization. In addition, it can make r queries to E (forward or inverse).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2021.3109302, IEEE Transactions on Information Theory

For $(h,i,j) \in [1,\mu] \times [1,q] \times [1,\ell]$, denote by $(K_h^{(i,j)}, N_h^{(i,j)})$ the state used in the generation of the j-th stream in the i-th initialization of the h-th oracle. For any $K \in \{0,1\}^k$, let σ_K denote the number of construction-induced primitive evaluations for key K, where $\sum_{K\in\{0,1\}^k} \sigma_K \leq (\sigma+2)\mu\ell q$. Denote the input-outputs tuples for a key K by $(X_K^{(m)}, Y_K^{(m)})$ for $m \in [1,\sigma_K]$, where any order is adopted. Note that for any $K \in \{0,1\}^k$,

$$\{X_K^{(m)} \mid m \in [1,\sigma_K]\} = \bigcup_{(h,i,j),K=K_h^{(i,j)}} \mathbf{call}_1(N_h^{(i,j)}) .$$

Denote the r primitive queries made by $\mathcal{D}$ by (L_l, X_l, Y_l) for $l \in [1,r]$.

In the remainder, we treat pseudorandomness and forward security of (20) separately.

A. Pseudorandomness

For the case of pseudorandomness, the distinguisher $\mathcal{D}$ is not allowed to call $\mathcal{O}b$.**leak**. In the real world, security breaks in case two identical evaluations of E occur (either among construction queries or between a construction and a primitive query). This means, for example, that $K_h^{(i,j)} = K_{h'}^{(i',j')}$ for two distinct $(h,i,j),(h',i',j')$. In addition, problems occur if two distinct evaluations of E give the same n-bit output, a block collision. The latter term will dominate. It turns out that a lossier description of the former bad event allows for a more refined analysis of the second bad event. Formally, we define a bad event $\mathsf{BAD}^{\text{i-prf}} = \mathsf{BAD}^{\text{i-prf}}_{\text{cons}} \vee \mathsf{BAD}^{\text{i-prf}}_{\text{prim}} \vee \mathsf{BAD}^{\text{i-prf}}_{\text{block}}$ as follows:

$$\begin{aligned}
\mathsf{BAD}^{\text{i-prf}}_{\text{cons}} &: \exists (h,i,j),(h',i',j') \in [1,\mu]\times[1,q]\times[1,\ell] : \\
&\quad (h,i,j) \neq (h',i',j') \wedge \neg(h=h' \wedge j=j'=1) \wedge \\
&\quad K_h^{(i,j)} = K_{h'}^{(i',j')} , \\
\mathsf{BAD}^{\text{i-prf}}_{\text{prim}} &: \exists l \in [1,r] , (h,i,j) \in [1,\mu]\times[1,q]\times[1,\ell] : \\
&\quad L_l = K_h^{(i,j)} , \\
\mathsf{BAD}^{\text{i-prf}}_{\text{block}} &: \exists K \in \{0,1\}^k , m,m' \in [1,\sigma_K] : \\
&\quad m \neq m' \wedge X_K^{(m)} \neq X_K^{(m')} \wedge Y_K^{(m)} = Y_K^{(m')} .
\end{aligned}$$

As long as $\mathsf{BAD}^{\text{i-prf}}_{\text{prim}}$ never happens, primitive queries do not influence the distribution of the outcomes of the construction oracles. Given that, among the construction queries, key collisions never occur, block inputs are always distinct (as $\mathbf{call}_1(N) \cap \mathbf{call}_1(N') = \emptyset$), and the individual blocks never collide, the real and ideal oracles behave identically. Therefore, both oracles are identical until $\mathsf{BAD}^{\text{i-prf}}$, and by the fundamental lemma of game playing [27], [32],

$$\mathbf{Adv}^{\text{i-dist}[\mu]}_{\mathcal{G}_{1,E}}(\mathcal{D}) \leq \Pr\left[\mathsf{BAD}^{\text{i-prf}}\right] . \tag{30}$$

Clearly,

$$\begin{aligned}
\Pr\left[\mathsf{BAD}^{\text{i-prf}}\right] &\leq \Pr\left[\mathsf{BAD}^{\text{i-prf}}_{\text{cons}}\right] + \Pr\left[\mathsf{BAD}^{\text{i-prf}}_{\text{prim}}\right] \\
&\quad + \Pr\left[\mathsf{BAD}^{\text{i-prf}}_{\text{block}} \;\middle|\; \neg\mathsf{BAD}^{\text{i-prf}}_{\text{cons}} \wedge \neg\mathsf{BAD}^{\text{i-prf}}_{\text{prim}}\right] ,
\end{aligned} \tag{31}$$

and we want to bound the probability that $\mathsf{BAD}^{\text{i-prf}}$ happens in the ideal world. For the first probability of (31), noting that $K_h^{(i,1)} = K_h^{(i',1)} = K_h$ for any h,i,i', there are a total amount of $\mu(\ell-1)q+\mu$ keys. Note that the marginal distribution of the master keys is $1/2^{2n}$, and the marginal distribution of the session keys is $1/2^n \cdot 1/(2^n-1)$, since the session keys are derived as two concatenated calls to an n-bit block cipher. Any two keys collide with probability at most $2/2^{2n}$. Summing over all options, we obtain

$$\Pr\left[\mathsf{BAD}^{\text{i-prf}}_{\text{cons}}\right] \leq 2\binom{\mu(\ell-1)q+\mu}{2}/2^{2n} \leq \frac{\mu^2\ell^2q^2}{2^{2n}} . \tag{32}$$

For the second probability of (31), consider any possible L_l (at most r choices), and noting that $K_h^{(i,1)} = K_h^{(i',1)} = K_h$ for any h,i,i', there are a total amount of at most $\mu(\ell-1)q+\mu$ keys that may set the bad event. Any of the master keys collides with L_l with probability at most $1/2^{2n}$, and any of the session keys collides with L_l with probability at most $2/2^{2n}$. Since the session keys are derived as two concatenated calls to an n-bit block cipher, and the marginal distribution of the session keys are $1/2^n \cdot 1/(2^n-1)$. Summing over all possible choices of L_l, we obtain

$$\Pr\left[\mathsf{BAD}^{\text{i-prf}}_{\text{prim}}\right] \leq \frac{\mu r}{2^{2n}} + \frac{2\mu(\ell-1)qr}{2^{2n}} \leq \frac{\mu\ell qr}{2^{2n-1}} . \tag{33}$$

For the third probability of (31), basic probability gives

$$\Pr\left[\mathsf{BAD}^{\text{i-prf}}_{\text{block}} \;\middle|\; \neg\mathsf{BAD}^{\text{i-prf}}_{\text{cons}} \wedge \neg\mathsf{BAD}^{\text{i-prf}}_{\text{prim}}\right] \leq \sum_{K\in\{0,1\}^k} \binom{\sigma_K}{2}/2^n ,$$

where we will use the negation $\neg\mathsf{BAD}^{\text{i-prf}}_{\text{cons}} \wedge \neg\mathsf{BAD}^{\text{i-prf}}_{\text{prim}}$ to determine the σ_K's. By $\neg\mathsf{BAD}^{\text{i-prf}}_{\text{prim}}$, none of the primitive queries influences the analysis. By $\neg\mathsf{BAD}^{\text{i-prf}}_{\text{cons}}$, there are no "unexpected" collisions among the keys $K_h^{(i,j)}$: the only "expected" collisions are that $K_h^{(i,1)} = K_h^{(i',1)} = K_h$ for any h,i,i'. Therefore, the μ master keys $K_1,\ldots,K_\mu$ occur $\sigma_{K_h} = (\sigma+2)q$ times each. Furthermore, any other session key $K_h^{(i,j)}$ for $(h,i,j) \in [1,\mu]\times[1,q]\times[2,\ell]$ is used $\sigma_{K_h^{(i,j)}} = \sigma+2$ times. We conclude that

$$\begin{aligned}
&\Pr\left[\mathsf{BAD}^{\text{i-prf}}_{\text{block}} \;\middle|\; \neg\mathsf{BAD}^{\text{i-prf}}_{\text{cons}} \wedge \neg\mathsf{BAD}^{\text{i-prf}}_{\text{prim}}\right] \\
&\qquad \leq \mu\binom{(\sigma+2)q}{2}/2^n + \mu(\ell-1)q\binom{\sigma+2}{2}/2^n \\
&\qquad \leq \frac{(\sigma+2)^2\mu(\ell+q)q}{2^{n+1}} .
\end{aligned} \tag{34}$$

From (30), (31), and (32-34), we obtain

$$\mathbf{Adv}^{\text{i-dist}[\mu]}_{\mathcal{G}_{1,E}}(\mathcal{D}) \leq \frac{\mu^2\ell^2q^2}{2^{2n}} + \frac{\mu\ell qr}{2^{2n-1}} + \frac{(\sigma+2)^2\mu(\ell+q)q}{2^{n+1}} .$$

Maximizing over all $\mathcal{D}$ that, to each of their μ oracles, can make at most q **init** calls, at most ℓ **next** calls per **init** call, 0 **leak** calls, and at most r calls to the underlying block cipher E. We obtain that both worlds are perfectly indistinguishable up to above bound.

B. Forward Security

For the case of forward security, the distinguisher $\mathcal{D}$ is allowed to make at most 1 call to $\mathcal{O}b$.**leak**. We would like that if at some point distinguisher $\mathcal{D}$ calls $\mathcal{O}b$.**leak** and recovers a state $\bar{K}$, all other data still has a certain amount of randomness. This applies first to the data $\mathcal{D}$ has seen so far, as well as all future data blocks, noting that by Definition VII.3 the distinguishing game continues once a state is leaked (but $\mathcal{D}$ must re-initialize the state). Denote the index of the leaked state with overlines: $(\bar{h},\bar{i},\bar{j})$, such that $\bar{K} = K_{\bar{h}}^{(\bar{i},\bar{j})}$. The restrictions on the distinguisher's **leak** query impose that $\bar{j} > 1$ and the distinguisher never queries the stream for index $(\bar{h},\bar{i},j)$ with $j \geq \bar{j}$.

Informally, we have that any stream learned by $\mathcal{D}$ has a certain amount of randomness if E is never evaluated twice for the same input (as in the pseudorandomness proof), and in addition, the leaked key never appears as master or session key elsewhere. The latter is strictly required to ensure that any state still maintains some randomness. Formally, we define a bad event $\mathsf{BAD}^{\text{i-fwd}} = \mathsf{BAD}^{\text{i-prf}} \vee \mathsf{BAD}^{\text{i-fwd}}_{\text{hit}}$, with $\mathsf{BAD}^{\text{i-prf}}$ as above, with the restriction that a primitive query colliding with a leaked state is invalid, and with $\mathsf{BAD}^{\text{i-fwd}}_{\text{hit}}$ as follows:

$$\begin{aligned}\mathsf{BAD}^{\text{i-fwd}}_{\text{hit}} \;:\; &\exists (h,i,j) \in [1,\mu]\times[1,q]\times[1,\ell] \;: \\ &(h,i,j) \neq (\bar{h},\bar{i},\bar{j}) \wedge K_h^{(i,j)} = K_{\bar{h}}^{(\bar{i},\bar{j})}\,,\end{aligned}$$

recalling that a **leak** call should always follow a **next** call, hence $\bar{j} > 1$.

As explained above, as long as $\mathsf{BAD}^{\text{i-fwd}}$ never happens, leakage does not help the distinguisher, and we are virtually back at pseudorandomness. We therefore find, by the fundamental lemma of game playing [27], [32],

$$\begin{aligned}\mathbf{Adv}^{\text{i-dist}[\mu]}_{\mathcal{G}_{1,E}}(\mathcal{D}) &\leq \Pr\left[\mathsf{BAD}^{\text{i-fwd}}\right] \\ &\leq \Pr\left[\mathsf{BAD}^{\text{i-prf}}\right] + \Pr\left[\mathsf{BAD}^{\text{i-fwd}}_{\text{hit}} \;\middle|\; \neg\mathsf{BAD}^{\text{i-prf}}\right]. \end{aligned} \tag{35}$$

For the first probability of (35), the bound on (31) with (32-34) carries over (the limitation that a primitive query colliding with a leaked state is invalid is irrelevant for the bounding). The second probability of (35) equals 0 as $\neg\mathsf{BAD}^{\text{i-prf}}_{\text{cons}} \implies \neg\mathsf{BAD}^{\text{i-fwd}}_{\text{hit}}$.[1]

We conclude that

$$\mathbf{Adv}^{\text{i-dist}[\mu]}_{\mathcal{G}_{1,E}}(\mathcal{D}) \leq \frac{\mu^2\ell^2q^2}{2^{2n}} + \frac{\mu\ell qr}{2^{2n-1}} + \frac{(\sigma+2)^2\mu(\ell+q)q}{2^{n+1}}\,.$$

Maximizing over all $\mathcal{D}$ that, to each of their μ oracles, can make at most q **init** calls, at most ℓ **next** calls per **init** call, 1 **leak** calls, and at most r calls to the underlying block cipher E. We obtain that both worlds are perfectly indistinguishable up to above bound.

[1]Recall that, as the dominant term in our block cipher based analysis is the block collision ($\mathsf{BAD}^{\text{i-prf}}_{\text{block}}$) anyway, the definitions of $\mathsf{BAD}^{\text{i-prf}}_{\text{cons}}$ and $\mathsf{BAD}^{\text{i-prf}}_{\text{prim}}$ were a bit looser to allow for an easier proof. The formalization of $\mathsf{BAD}^{\text{i-fwd}}_{\text{hit}}$ is strictly seen unnecessary, but included for intuition.

APPENDIX B
PROOF OF THEOREM VII.2 ($\mathcal{G}_{1,F}$)

Let $F \xleftarrow{\$} \mathrm{Func}(3n,n)$, and consider μ master keys $K_1,K_2,\ldots,K_\mu \xleftarrow{\$} \{0,1\}^{2n}$. Consider any distinguisher $\mathcal{D}$ that has access to $\mathcal{O}b[F]_{K_1},\cdots,\mathcal{O}b[F]_{K_\mu}$ for $b \in \{0,1\}$. It can initialize each of its μ oracles at most q times, and the layer function is evaluated at most ℓ times for each initialization. In addition, it can make r queries to F. We use the same convention for indices as in Section A.

In the remainder, we treat pseudorandomness (21) and forward security (22) separately.

A. Pseudorandomness

As before, the distinguisher $\mathcal{D}$ is not allowed to call $\mathcal{O}b$.**leak**, and security breaks in case two identical evaluations of F occur. This means, for example, $(K_h^{(i,j)},N_h^{(i,j)}) = (K_{h'}^{(i',j')},N_{h'}^{(i',j')})$ for two distinct $(h,i,j),(h',i',j')$. We can note, however, that $N_h^{(i,j)} = N_h^{(i,1)}$ for any (h,i,j). There is furthermore no need to consider block collisions (as we use F instead of a block cipher E). Formally, we define a bad event $\mathsf{BAD}^{\text{i-prf}} = \mathsf{BAD}^{\text{i-prf}}_{\text{cons}} \vee \mathsf{BAD}^{\text{i-prf}}_{\text{prim}}$ as follows:

$$\begin{aligned}\mathsf{BAD}^{\text{i-prf}}_{\text{cons}} \;:\; &\exists (h,i,j),(h',i',j') \in [1,\mu]\times[1,q]\times[1,\ell] \;: \\ &(h,i,j) \neq (h',i',j') \wedge K_h^{(i,j)} = K_{h'}^{(i',j')} \wedge \\ &\mathbf{call}_1(N_h^{(i,1)}) \cap \mathbf{call}_1(N_{h'}^{(i',1)}) \neq \emptyset\,, \\ \mathsf{BAD}^{\text{i-prf}}_{\text{prim}} \;:\; &\exists l \in [1,r]\,,\, (h,i,j) \in [1,\mu]\times[1,q]\times[1,\ell] \;: \\ &L_l = K_h^{(i,j)} \wedge X_l \in \mathbf{call}_1(N_h^{(i,1)})\,.\end{aligned}$$

As long as $\mathsf{BAD}^{\text{i-prf}}_{\text{prim}}$ never happens, primitive queries do not influence the distribution of the outcomes of the construction oracles. Given that, among the construction queries, primitive queries are never evaluated twice under the same input, the real and ideal oracles behave identically. Therefore, both oracles are identical until $\mathsf{BAD}^{\text{i-prf}}$, and by the fundamental lemma of game playing [27], [32],

$$\mathbf{Adv}^{\text{i-dist}[\mu]}_{\mathcal{G}_{1,F}}(\mathcal{D}) \leq \Pr\left[\mathsf{BAD}^{\text{i-prf}}\right]. \tag{36}$$

Clearly,

$$\Pr\left[\mathsf{BAD}^{\text{i-prf}}\right] \leq \Pr\left[\mathsf{BAD}^{\text{i-prf}}_{\text{cons}}\right] + \Pr\left[\mathsf{BAD}^{\text{i-prf}}_{\text{prim}}\right], \tag{37}$$

and we want to bound the probability that $\mathsf{BAD}^{\text{i-prf}}$ happens in the ideal world. For the first probability of (37), we make a distinction depending on the choice of h,h'.

- $h = h'$. In this case, different initializations of the oracle are done under difference nonces (such that $\mathbf{call}_1(N_h^{(i,1)}) \cap \mathbf{call}_1(N_h^{(i',1)}) = \emptyset$), and hence $\mathsf{BAD}^{\text{i-prf}}_{\text{cons}}$ happens with probability 0 if $i \neq i'$. For the case of $i = i'$, any two layers have the same nonce, and their keys collide with probability at most $\binom{\ell}{2}/2^{2n}$. Summing over all possible choices of $h = h'$ and $i = i'$, $\mathsf{BAD}^{\text{i-prf}}_{\text{cons}}$ is set with probability at most $\mu\binom{\ell}{2}q/2^{2n}$;
- $h \neq h'$. Consider any query $i \in \{1,\ldots,q\}$, there are at most $2\sigma+3$ possible $i' \in \{1,\ldots,q\}$ such that $\mathbf{call}_1(N_h^{(i,1)}) \cap \mathbf{call}_1(N_{h'}^{(i',1)}) \neq \emptyset$. For any such choice,

the keys of two layers collide with probability $\ell^2/2^{2n}$. Summing over all possible choices of $h \neq h'$, i, and i', $\mathsf{BAD}^{\text{i-prf}}_{\text{cons}}$ is set with probability at most $\binom{\mu}{2}(2\sigma+3)\ell^2 q/2^{2n}$.

Summing over all options, we obtain

$$\Pr\left[\mathsf{BAD}^{\text{i-prf}}_{\text{cons}}\right] \leq \mu\binom{\ell}{2}\frac{q}{2^{2n}} + \binom{\mu}{2}\frac{(2\sigma+3)\ell^2 q}{2^{2n}} \leq \frac{(2\sigma+3)\mu^2\ell^2 q}{2^{2n}}\,. \tag{38}$$

For the second probability of (37), consider any possible (L_l, X_l) (r choices). We have at most μ possible choices h, as the distinguisher must choose its nonces so that $\mathbf{call}_1(N_h^{(i,1)}) \cap \mathbf{call}_1(N_h^{(i',1)}) = \emptyset$ there is at most 1 possible i such that $X_l \in \mathbf{call}_1(N_h^{(i,1)})$, and there are ℓ possible choices of j. For any selection of parameters, we have $L_l = K_h^{(i,j)}$ with probability $1/2^{2n}$. Hence,

$$\Pr\left[\mathsf{BAD}^{\text{i-prf}}_{\text{prim}}\right] \leq \frac{\mu\ell r}{2^{2n}}\,. \tag{39}$$

From (36), (37), and (38-39), we obtain

$$\mathbf{Adv}^{\text{i-dist}[\mu]}_{\mathcal{G}_{1,F}}(\mathcal{D}) \leq \frac{(2\sigma+3)\mu^2\ell^2 q}{2^{2n}} + \frac{\mu\ell r}{2^{2n}}\,.$$

Maximizing over all $\mathcal{D}$ that, to each of their μ oracles, can make at most q **init** calls, at most ℓ **next** calls per **init** call, 0 **leak** calls, and at most r calls to the underlying block cipher F. We obtain that both worlds are perfectly indistinguishable up to above bound.

B. Forward Security

The additional bad event needed to guarantee indistinguishability for any distinguisher $\mathcal{D}$ that is allowed to make at most 1 call to $\mathcal{O}b$.**leak** remains the same as the case of block cipher based schemes. Formally, we define a bad event $\mathsf{BAD}^{\text{i-fwd}} = \mathsf{BAD}^{\text{i-prf}} \vee \mathsf{BAD}^{\text{i-fwd}}_{\text{hit}}$, with $\mathsf{BAD}^{\text{i-prf}}$ as above, with the restriction that a primitive query colliding with a leaked state is invalid, and with $\mathsf{BAD}^{\text{i-fwd}}_{\text{hit}}$ exactly as before:

$$\begin{aligned}\mathsf{BAD}^{\text{i-fwd}}_{\text{hit}} \;:\; &\exists (h,i,j) \in [1,\mu]\times[1,q]\times[1,\ell]\;:\\ &(h,i,j) \neq (\bar{h},\bar{i},\bar{j}) \wedge K_h^{(i,j)} = K_{\bar{h}}^{(\bar{i},\bar{j})}\,,\end{aligned}$$

recalling that a **leak** call should always follow a **next** call, hence $\bar{j} > 1$.

As before, as long as $\mathsf{BAD}^{\text{i-fwd}}$ never happens, leakage does not help the distinguisher, and we are virtually back at pseudorandomness. We therefore find, by the fundamental lemma of game playing [27], [32],

$$\begin{aligned}\mathbf{Adv}^{\text{i-dist}[\mu]}_{\mathcal{G}_{1,F}}(\mathcal{D}) &\leq \Pr\left[\mathsf{BAD}^{\text{i-fwd}}\right]\\ &\leq \Pr\left[\mathsf{BAD}^{\text{i-prf}}\right] + \Pr\left[\mathsf{BAD}^{\text{i-fwd}}_{\text{hit}} \;\middle|\; \neg\mathsf{BAD}^{\text{i-prf}}\right]\,.\end{aligned} \tag{40}$$

For the first probability of (40), the bound on (37) with (38-39) carries over (the limitation that a primitive query colliding with a leaked state is invalid is irrelevant for the bounding). For the second probability of (40), noting that $K_h^{(i,1)} = K_h^{(i',1)} = K_h$ for any h, i, i', there are a total amount of at most $\mu(\ell-1)q+\mu$ keys that may set the bad event. Any of the keys collides with $K_{\bar{h}}^{(\bar{i},\bar{j})}$ with probability at most $1/2^{2n}$. Summing over all options, we obtain

$$\Pr\left[\mathsf{BAD}^{\text{i-fwd}}_{\text{hit}} \;\middle|\; \neg\mathsf{BAD}^{\text{i-prf}}\right] \leq \frac{\mu(\ell-1)q+\mu}{2^{2n}}\,. \tag{41}$$

From (40), (37), and (41), we obtain

$$\mathbf{Adv}^{\text{i-dist}[\mu]}_{\mathcal{G}_{1,F}}(\mathcal{D}) \leq \frac{(2\sigma+3)\mu^2\ell^2 q}{2^{2n}} + \frac{\mu\ell r}{2^{2n}} + \frac{\mu(\ell-1)q+\mu}{2^{2n}}\,.$$

Maximizing over all $\mathcal{D}$ that, to each of their μ oracles, can make at most q **init** calls, at most ℓ **next** calls per **init** call, 1 **leak** calls, and at most r calls to the underlying block cipher F. We obtain that both worlds are perfectly indistinguishable up to above bound.

Appendix C
Proof of Theorem VII.3 ($\mathcal{G}_{2,E}$)

The proof is very similar to that in Section A, most importantly due to the relaxed description of $\mathsf{BAD}^{\text{i-prf}}_{\text{cons}}$ and $\mathsf{BAD}^{\text{i-prf}}_{\text{prim}}$. There are only two differences.

The first difference is that $\mathbf{call}_1()$ is replaced with $\mathbf{call}_2()$, a change dealt with by replacing $\sigma+2$ by $\sigma+3$.

The second difference is that, among different layers, $N_h^{(i,j)}$ changes. This means that the condition "block inputs are always distinct (as $\mathbf{call}_2(N) \cap \mathbf{call}_2(N') = \emptyset$)" does not apply anymore, and we will have to include an additional block collision event that captures input collisions. Formally, we define a bad event $\mathsf{BAD}^{\text{i-prf}} = \mathsf{BAD}^{\text{i-prf}}_{\text{cons}} \vee \mathsf{BAD}^{\text{i-prf}}_{\text{prim}} \vee \mathsf{BAD}^{\text{i-prf}}_{\text{block}} \vee \mathsf{BAD}^{\text{i-prf}}_{\text{nonce}}$, where the first three bad events are as in Section A, and the fourth is as follows:

$$\begin{aligned}\mathsf{BAD}^{\text{i-prf}}_{\text{nonce}} \;:\; &\exists (h,i,j),(h',i',j') \in [1,\mu]\times[1,q]\times[1,\ell]\;:\\ &(h,i,j) \neq (h',i',j') \wedge K_h^{(i,j)} = K_{h'}^{(i',j')} \wedge\\ &\mathbf{call}_2(N_h^{(i,j)}) \cap \mathbf{call}_2(N_{h'}^{(i',j')}) \neq \emptyset\,.\end{aligned}$$

The same reasoning as before applies, yielding

$$\begin{aligned}\mathbf{Adv}^{\text{i-dist}[\mu]}_{\mathcal{G}_{2,E}}(\mathcal{D}) &\leq \Pr\left[\mathsf{BAD}^{\text{i-prf}}\right]\\ &\leq \Pr\left[\mathsf{BAD}^{\text{i-prf}}_{\text{cons}}\right] + \Pr\left[\mathsf{BAD}^{\text{i-prf}}_{\text{prim}}\right]\\ &\quad + \Pr\left[\mathsf{BAD}^{\text{i-prf}}_{\text{block}} \;\middle|\; \neg\mathsf{BAD}^{\text{i-prf}}_{\text{cons}} \wedge \neg\mathsf{BAD}^{\text{i-prf}}_{\text{prim}}\right]\\ &\quad + \Pr\left[\mathsf{BAD}^{\text{i-prf}}_{\text{nonce}} \;\middle|\; \neg\mathsf{BAD}^{\text{i-prf}}_{\text{cons}} \wedge \neg\mathsf{BAD}^{\text{i-prf}}_{\text{prim}}\right]\,,\end{aligned} \tag{42}$$

and we want to bound the probability that $\mathsf{BAD}^{\text{i-prf}}$ happens in the ideal world. For the first three probabilities of (42), the bounds of (32-34) in Section A carries over, with $\sigma+2$ replaced by $\sigma+3$. Recall that the marginal distribution of the master keys is $1/2^{2n}$, of the session keys is $1/2^n \cdot 1/(2^n-1)$, and of the session nonces is $1/(2^n-2)$. For the fourth probability of (42), we make a distinction depending on the choice of j, j'.

- $j=1$, $j'=1$. If $h=h'$, the event is set with probability 0 as the distinguisher must choose its nonces so that $\mathbf{call}_2(N_h^{(i,1)}) \cap \mathbf{call}_2(N_h^{(i',1)}) = \emptyset$. On the other hand, if $h \neq h'$, we have $K_h^{(i,1)} = K_h$ and $K_{h'}^{(i',1)} = K_{h'}$ and the event is set with probability $1/2^{2n}$. However, we do

not need to include this event as it does not happen by virtue of $\neg\mathsf{BAD}^{\text{i-prf}}_{\text{cons}}$.
- $j = 1$, $j' > 1$. Note that $K_h^{(i,1)} = K_h$ denotes the h-th master key. The states are always randomly generated and for every possible (h', i', j') the conditions are set with probability $4(2\sigma+5)/2^{3n}$. We have μ possible master keys, and $\mu(\ell-1)q$ choices (h', i', j') such that $j' \in [2, \ell]$. Hence, $\mathsf{BAD}^{\text{i-prf}}_{\text{nonce}}$ is set with probability at most $4(2\sigma+5)\mu^2(\ell-1)q/2^{3n}$.
- $j > 1$, $j' > 1$. The states are always randomly generated and for every possible $(h, i, j), (h', i', j')$ the conditions are set with probability $4(2\sigma+5)/2^{3n}$. There are at most $\binom{\mu(\ell-1)q}{2}$ possible choices for $(h, i, j), (h', i', j')$ such that $j, j' \in [2, \ell]$. Hence $\mathsf{BAD}^{\text{i-prf}}_{\text{nonce}}$ is set with probability at most $4\binom{\mu(\ell-1)q}{2}(2\sigma+5)/2^{3n}$.

Summing over all options, we obtain

$$\begin{aligned}\Pr\left[\mathsf{BAD}^{\text{i-prf}}_{\text{nonce}}\right] &\le \frac{4(2\sigma+5)\mu^2(\ell-1)q}{2^{3n}} + 4\binom{\mu(\ell-1)q}{2}\frac{2\sigma+5}{2^{3n}} \\ &\le \frac{(2\sigma+5)\mu^2\ell^2q^2}{2^{3n-1}}\,. \end{aligned} \tag{43}$$

From (42), (32-34), and (43), we obtain

$$\begin{aligned}\mathbf{Adv}^{\text{i-dist}[\mu]}_{\mathcal{G}_{2,E}}(\mathcal{D}) \le \frac{\mu^2\ell^2q^2}{2^{2n}} + \frac{\mu\ell qr}{2^{2n-1}} + \frac{(\sigma+3)^2\mu(\ell+q)q}{2^{n+1}} \\ + \frac{(2\sigma+5)\mu^2\ell^2q^2}{2^{3n-1}}\,.\end{aligned}$$

Maximizing over all $\mathcal{D}$, we obtain that both worlds are perfectly indistinguishable up to above bound. The bound applies to both pseudorandomness and forward security.

Appendix D
Proof of Theorem VII.4 ($\mathcal{G}_{2,F}$)

Let $F \xleftarrow{\$} \mathrm{Func}(3n, n)$, and consider μ master keys $K_1, K_2, \ldots, K_\mu \xleftarrow{\$} \{0,1\}^{2n}$. Consider any distinguisher $\mathcal{D}$ that has access to $\mathcal{O}b[F]_{K_1}, \cdots, \mathcal{O}b[F]_{K_\mu}$ for $b \in \{0,1\}$. It can initialize each of its μ oracles at most q times, and the layer function is evaluated at most ℓ times for each initialization. In addition, it can make r queries to F. We use the same convention for indices as in Section A.

In the remainder, we treat pseudorandomness (24) and forward security (25) separately.

A. Pseudorandomness

As in the analysis of $\mathbf{layer}_1[F]$ in Section B, security breaks in case two identical evaluations of F occur. This means, for example, $(K_h^{(i,j)}, N_h^{(i,j)}) = (K_{h'}^{(i',j')}, N_{h'}^{(i',j')})$ for two distinct $(h, i, j), (h', i', j')$. The current scheme, however, uses $\mathbf{layer}_2$, which changes its nonce every layer. This results in a slightly different bad event that captures construction queries. Formally, we define a bad event $\mathsf{BAD}^{\text{i-prf}} = \mathsf{BAD}^{\text{i-prf}}_{\text{cons}} \vee \mathsf{BAD}^{\text{i-prf}}_{\text{prim}}$ as follows:

$$\begin{aligned}\mathsf{BAD}^{\text{i-prf}}_{\text{cons}} &: \exists (h,i,j),(h',i',j') \in [1,\mu]\times[1,q]\times[1,\ell] : \\ &\quad (h,i,j) \ne (h',i',j') \wedge K_h^{(i,j)} = K_{h'}^{(i',j')} \wedge \\ &\quad \mathbf{call}_2(N_h^{(i,j)}) \cap \mathbf{call}_2(N_{h'}^{(i',j')}) \ne \emptyset\,, \\ \mathsf{BAD}^{\text{i-prf}}_{\text{prim}} &: \exists l \in [1,r]\,,\ (h,i,j) \in [1,\mu]\times[1,q]\times[1,\ell] : \\ &\quad L_l = K_h^{(i,j)} \wedge X_l \in \mathbf{call}_2(N_h^{(i,j)})\,.\end{aligned}$$

In line with the reasoning of Section B,

$$\begin{aligned}\mathbf{Adv}^{\text{i-dist}[\mu]}_{\mathcal{G}_{2,F}}(\mathcal{D}) &\le \Pr\left[\mathsf{BAD}^{\text{i-prf}}\right] \\ &\le \Pr\left[\mathsf{BAD}^{\text{i-prf}}_{\text{cons}}\right] + \Pr\left[\mathsf{BAD}^{\text{i-prf}}_{\text{prim}}\right]\,, \end{aligned} \tag{44}$$

and we want to bound the probability that $\mathsf{BAD}^{\text{i-prf}}$ happens in the ideal world. For the first probability of (44), we can almost inherit the computation on $\mathsf{BAD}^{\text{i-prf}}_{\text{nonce}}$ of (43): the definitions of bad events is identical, but in (43) we assumed $\neg\mathsf{BAD}^{\text{i-prf}}_{\text{cons}}$ as assumption. We make a distinction depending on the choice of j, j'.

- $j = 1$, $j' = 1$. If $h = h'$, the event is set with probability 0 as the distinguisher must choose its nonces so that $\mathbf{call}_2(N_h^{(i,1)}) \cap \mathbf{call}_2(N_h^{(i',1)}) = \emptyset$. On the other hand, if $h \ne h'$, we have $K_h^{(i,1)} = K_h$ and $K_{h'}^{(i',1)} = K_{h'}$ and the event is set with probability $1/2^{2n}$. Hence, $\mathsf{BAD}^{\text{i-prf}}_{\text{nonce}}$ is set with probability at most $\binom{\mu}{2}/2^{2n} \le \mu^2/2^{2n+1}$.
- $j = 1$, $j' > 1$. Note that $K_h^{(i,1)} = K_h$ denotes the h-th master key. The states are always randomly generated and for every possible (h', i', j') the conditions are set with probability $(2\sigma+5)/2^{3n}$. We have μ possible master keys, and $\mu(\ell-1)q$ choices (h', i', j') such that $j' \in [2, \ell]$. Hence, $\mathsf{BAD}^{\text{i-prf}}_{\text{nonce}}$ is set with probability at most $(2\sigma+5)\mu^2(\ell-1)q/2^{3n}$.
- $j > 1$, $j' > 1$. The states are always randomly generated and for every possible $(h, i, j), (h', i', j')$ the conditions are set with probability $(2\sigma+5)/2^{3n}$. There are at most $\binom{\mu(\ell-1)q}{2}$ possible choices for $(h, i, j), (h', i', j')$ such that $j, j' \in [2, \ell]$. Hence $\mathsf{BAD}^{\text{i-prf}}_{\text{nonce}}$ is set with probability at most $\binom{\mu(\ell-1)q}{2}(2\sigma+5)/2^{3n}$.

Summing over all options, we obtain

$$\begin{aligned}\Pr\left[\mathsf{BAD}^{\text{i-prf}}_{\text{cons}}\right] &\le \frac{\mu^2}{2^{2n+1}} + \frac{(2\sigma+5)\mu^2(\ell-1)q}{2^{3n}} + \binom{\mu(\ell-1)q}{2}\frac{2\sigma+5}{2^{3n}} \\ &\le \frac{\mu^2}{2^{2n+1}} + \frac{(2\sigma+5)\mu^2\ell^2q^2}{2^{3n}}\,. \end{aligned} \tag{45}$$

For the second probability of (44), consider any possible (L_l, X_l) (r choices). We likewise make a distinction depending on the choice of j.

- $j = 1$. Note that $K_h^{(i,1)} = K_h$ denotes the h-th master key. The bad event is set if $L_l = K_h$, which happens with probability $1/2^{2n}$. Summing over all possible choices, $\mathsf{BAD}^{\text{i-prf}}_{\text{prim}}$ is set (for this primitive query) with probability at most $\mu/2^{2n}$.

- $j > 1$. The states are always randomly generated and for every possible (h, i, j) the conditions are set with probability $(\sigma+3)/2^{3n}$. There are at most $\mu(\ell-1)q$ possible choices for (h, i, j) such that $j \in [2, \ell]$, and hence $\mathsf{BAD}^{\text{i-prf}}_{\text{prim}}$ is set (for this primitive query) with probability at most $(\sigma+3)\mu(\ell-1)q/2^{3n}$.

Summing over all possible choices of (L_l, X_l), we obtain

$$\Pr\left[\mathsf{BAD}^{\text{i-prf}}_{\text{prim}}\right] \leq \frac{\mu r}{2^{2n}} + \frac{(\sigma+3)\mu(\ell-1)qr}{2^{3n}}. \qquad (46)$$

From (44) and (45-46), we obtain

$$\mathbf{Adv}^{\text{dist}[\mu]}_{\mathcal{G}_{2,F}}(\mathcal{D}) \leq \frac{\mu^2}{2^{2n+1}} + \frac{(2\sigma+5)\mu^2\ell^2q^2}{2^{3n}} + \frac{\mu r}{2^{2n}} + \frac{(\sigma+3)\mu(\ell-1)qr}{2^{3n}}.$$

Maximizing over all $\mathcal{D}$ that, to each of their μ oracles, can make at most q **init** calls, at most ℓ **next** calls per **init** call, 0 **leak** calls, and at most r calls to the underlying block cipher F. We obtain that both worlds are perfectly indistinguishable up to above bound.

B. Forward Security

The analysis is identical to the analysis of forward security of $\mathcal{G}_{1,F}$, noting that the bad event introduced there is independent of the nonce anyway. We obtain

$$\mathbf{Adv}^{\text{i-dist}[\mu]}_{\mathcal{G}_{2,F}}(\mathcal{D}) \leq \frac{\mu^2}{2^{2n+1}} + \frac{(2\sigma+5)\mu^2\ell^2q^2}{2^{3n}} + \frac{\mu r}{2^{2n}} + \frac{(\sigma+3)\mu(\ell-1)qr}{2^{3n}} + \frac{\mu(\ell-1)q+\mu}{2^{2n}}.$$

Maximizing over all $\mathcal{D}$ that, to each of their μ oracles, can make at most q **init** calls, at most ℓ **next** calls per **init** call, 1 **leak** calls, and at most r calls to the underlying block cipher F. We obtain that both worlds are perfectly indistinguishable up to above bound.

References

[1] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway, "A Concrete Security Treatment of Symmetric Encryption," in *38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997*. IEEE Computer Society, 1997, pp. 394–403. [Online]. Available: https://doi.org/10.1109/SFCS.1997.646128

[2] M. Bellare and B. S. Yee, "Forward-security in private-key cryptography," in *Topics in Cryptology - CT-RSA 2003, The Cryptographers' Track at the RSA Conference 2003, San Francisco, CA, USA, April 13-17, 2003, Proceedings*, ser. Lecture Notes in Computer Science, M. Joye, Ed., vol. 2612. Springer, 2003, pp. 1–18. [Online]. Available: https://doi.org/10.1007/3-540-36563-X_1

[3] E. Barker and J. Kelsey, "NIST SP 800-90 Revision 1: Recommendation for Random Number Generation Using Deterministic Random Bit Generators," 2015.

[4] D. J. Bernstein, "Fast-key-erasure random-number generators," July 2017, https://blog.cr.yp.to/20170723-random.html.

[5] T. Shrimpton and R. S. Terashima, "A provable-security analysis of Intel's secure key RNG," in *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, ser. Lecture Notes in Computer Science, E. Oswald and M. Fischlin, Eds., vol. 9056. Springer, 2015, pp. 77–100. [Online]. Available: https://doi.org/10.1007/978-3-662-46800-5_4

[6] J. Woodage and D. Shumow, "An analysis of NIST SP 800-90A," in *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part II*, ser. Lecture Notes in Computer Science, Y. Ishai and V. Rijmen, Eds., vol. 11477. Springer, 2019, pp. 151–180. [Online]. Available: https://doi.org/10.1007/978-3-030-17656-3_6

[7] J. Daemen and V. Rijmen, *The Design of Rijndael: AES - The Advanced Encryption Standard*, ser. Information Security and Cryptography. Springer, 2002. [Online]. Available: https://doi.org/10.1007/978-3-662-04722-4

[8] B. Mennink and S. Neves, "Optimal PRFs from blockcipher designs," *IACR Trans. Symmetric Cryptol.*, vol. 2017, no. 3, pp. 228–252, 2017. [Online]. Available: https://doi.org/10.13154/tosc.v2017.i3.228-252

[9] M. Bellare, T. Krovetz, and P. Rogaway, "Luby-Rackoff backwards: Increasing security by making block ciphers non-invertible," in *Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding*, ser. Lecture Notes in Computer Science, K. Nyberg, Ed., vol. 1403. Springer, 1998, pp. 266–280. [Online]. Available: https://doi.org/10.1007/BFb0054132

[10] M. Bellare and R. Impagliazzo, "A tool for obtaining tighter security analyses of pseudorandom function based constructions, with applications to PRP to PRF conversion," Cryptology ePrint Archive, Report 1999/024, 1999.

[11] S. Lucks, "The sum of PRPs is a secure PRF," in *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*, ser. Lecture Notes in Computer Science, B. Preneel, Ed., vol. 1807. Springer, 2000, pp. 470–484. [Online]. Available: https://doi.org/10.1007/3-540-45539-6_34

[12] J. Patarin, "A proof of security in $o(2^n)$ for the xor of two random permutations," in *Information Theoretic Security, Third International Conference, ICITS 2008, Calgary, Canada, August 10-13, 2008, Proceedings*, ser. Lecture Notes in Computer Science, R. Safavi-Naini, Ed., vol. 5155. Springer, 2008, pp. 232–248. [Online]. Available: https://doi.org/10.1007/978-3-540-85093-9_22

[13] ——, "Introduction to Mirror Theory: Analysis of Systems of Linear Equalities and Linear Non Equalities for Cryptography," Cryptology ePrint Archive, Report 2010/287, 2010.

[14] W. Dai, V. T. Hoang, and S. Tessaro, "Information-theoretic indistinguishability via the chi-squared method," in *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, ser. Lecture Notes in Computer Science, J. Katz and H. Shacham, Eds., vol. 10403. Springer, 2017, pp. 497–523. [Online]. Available: https://doi.org/10.1007/978-3-319-63697-9_17

[15] T. Iwata, "New blockcipher modes of operation with beyond the birthday bound security," in *Fast Software Encryption, 13th International Workshop, FSE 2006, Graz, Austria, March 15-17, 2006, Revised Selected Papers*, ser. Lecture Notes in Computer Science, M. J. B. Robshaw, Ed., vol. 4047. Springer, 2006, pp. 310–327. [Online]. Available: https://doi.org/10.1007/11799313_20

[16] T. Iwata, B. Mennink, and D. Vizár, "CENC is Optimally Secure," Cryptology ePrint Archive, Report 2016/1087, 2016.

[17] S. Bhattacharya and M. Nandi, "Revisiting variable output length XOR pseudorandom function," *IACR Trans. Symmetric Cryptol.*, vol. 2018, no. 1, pp. 314–335, 2018. [Online]. Available: https://doi.org/10.13154/tosc.v2018.i1.314-335

[18] D. J. Bernstein, "Mess of Proofs," Dagstuhl Seminar "Symmetric Cryptography", January 2018.

[19] T. Shrimpton and R. S. Terashima, "Salvaging weak security bounds for blockcipher-based constructions," in *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, ser. Lecture Notes in Computer Science, J. H. Cheon and T. Takagi, Eds., vol. 10031, 2016, pp. 429–454. [Online]. Available: https://doi.org/10.1007/978-3-662-53887-6_16

[20] F. Denis, "AES-STREAM," December 2017, https://github.com/jedisct1/aes-stream.

[21] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche, "Sponge-based pseudo-random number generators," in *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*, ser. Lecture Notes in Computer Science, S. Mangard and F. Standaert,

Eds., vol. 6225. Springer, 2010, pp. 33–47. [Online]. Available: https://doi.org/10.1007/978-3-642-15031-9_3

[22] P. Gazi and S. Tessaro, "Provably robust sponge-based PRNGs and KDFs," in *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, ser. Lecture Notes in Computer Science, M. Fischlin and J. Coron, Eds., vol. 9665. Springer, 2016, pp. 87–116. [Online]. Available: https://doi.org/10.1007/978-3-662-49890-3_4

[23] D. Hutchinson, "A robust and sponge-like PRNG with improved efficiency," in *Selected Areas in Cryptography - SAC 2016 - 23rd International Conference, St. John's, NL, Canada, August 10-12, 2016, Revised Selected Papers*, ser. Lecture Notes in Computer Science, R. Avanzi and H. M. Heys, Eds., vol. 10532. Springer, 2016, pp. 381–398. [Online]. Available: https://doi.org/10.1007/978-3-319-69453-5_21

[24] M. Hamann, M. Krause, and W. Meier, "LIZARD - A lightweight stream cipher for power-constrained devices," *IACR Trans. Symmetric Cryptol.*, vol. 2017, no. 1, pp. 45–79, 2017. [Online]. Available: https://doi.org/10.13154/tosc.v2017.i1.45-79

[25] S. Banik, T. Isobe, T. Cui, and J. Guo, "Some cryptanalytic results on lizard," *IACR Trans. Symmetric Cryptol.*, vol. 2017, no. 4, pp. 82–98, 2017. [Online]. Available: https://doi.org/10.13154/tosc.v2017.i4.82-98

[26] M. Hamann, M. Krause, and A. Moch, "Tight security bounds for generic stream cipher constructions," in *Selected Areas in Cryptography - SAC 2019 - 26th International Conference, Waterloo, ON, Canada, August 12-16, 2019, Revised Selected Papers*, ser. Lecture Notes in Computer Science, K. G. Paterson and D. Stebila, Eds., vol. 11959. Springer, 2019, pp. 335–364. [Online]. Available: https://doi.org/10.1007/978-3-030-38471-5_14

[27] M. Bellare and P. Rogaway, "The security of triple encryption and a framework for code-based game-playing proofs," in *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, ser. Lecture Notes in Computer Science, S. Vaudenay, Ed., vol. 4004. Springer, 2006, pp. 409–426. [Online]. Available: https://doi.org/10.1007/11761679_25

[28] E. Biham, "How to Forge DES-Encrypted Messages in 2^{28} Steps," September 1996, http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-get.cgi/1996/CS/CS0884.pdf.

[29] ——, "How to decrypt or even substitute DES-encrypted messages in 2^{28} steps," *Inf. Process. Lett.*, vol. 84, no. 3, pp. 117–124, 2002. [Online]. Available: https://doi.org/10.1016/S0020-0190(02)00269-7

[30] B. Mennink and S. Neves, "Encrypted Davies-Meyer and its dual: Towards optimal security using mirror theory," in *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, ser. Lecture Notes in Computer Science, J. Katz and H. Shacham, Eds., vol. 10403. Springer, 2017, pp. 556–583. [Online]. Available: https://doi.org/10.1007/978-3-319-63697-9_19

[31] J. Patarin, "On linear systems of equations with distinct variables and small block size," in *Information Security and Cryptology - ICISC 2005, 8th International Conference, Seoul, Korea, December 1-2, 2005, Revised Selected Papers*, ser. Lecture Notes in Computer Science, D. Won and S. Kim, Eds., vol. 3935. Springer, 2005, pp. 299–321. [Online]. Available: https://doi.org/10.1007/11734727_25

[32] V. Shoup, "Sequences of Games: A Tool for Taming Complexity in Security Proofs," Cryptology ePrint Archive, Report 2004/332, 2004.

[33] J. Katz and H. Shacham, Eds., *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, ser. Lecture Notes in Computer Science, vol. 10403. Springer, 2017. [Online]. Available: https://doi.org/10.1007/978-3-319-63697-9

Yu Long Chen is currently pursuing the Ph.D. degree with the imec-COSIC Research Group, KU Leuven, Leuven, Belgium, under the supervision of Prof. B. Preneel and Prof. B. Mennink. His research focus is on pseudorandom permutations and functions. He is supported by a Ph.D. Fellowship from the Research Foundation - Flanders (FWO). He completed the M.Sc. thesis on Efficient Length Doubling From Tweakable Block Ciphers, under the supervision of Prof. B. Preneel and Prof. V. Rijmen.

Atul Luykx is a Senior Software Engineer at Google, where he helps Google with its cryptography needs and works on the Tink cryptographic library.

Previously he was head of cryptography at Swirlds, a staff research scientist at Visa Research, and held a postdoctoral appointment at the UC Davis computer science department with Phil Rogaway and the COSIC research group of the KU Leuven headed by Bart Preneel.

Bart Mennink received the Ph.D. degree titled Provable Security of Cryptographic Hash Functions, in 2013, under the supervision of Prof. B. Preneel and Prof. V. Rijmen. He was an NWO Veni Postdoctoral Researcher with Radboud University, Nijmegen, The Netherlands, and the FWO Postdoctoral Researcher with imec-COSIC, KU Leuven, Leuven, Belgium. He completed the M.Sc. thesis on Encrypted certificate schemes and their security and privacy analysis during a nine-month internship with Philips, Eindhoven, The Netherlands. He is an Assistant Professor with the Digital Security Group, Radboud University Nijmegen, Nijmegen, The Netherlands. His current research focus is on authentication and encryption.

Bart Preneel (Member, IEEE) is currently a Full Professor with KU Leuven, Leuven, Belgium, where he heads the imec-COSIC Research Group. He has authored over 400 scientific publications and is the inventor of four patents. His main research interests are cryptography, information security, and privacy. Prof. Preneel was the Program Chair of 15 international conferences. He has been an Invited Speaker at over 90 conferences in 40 countries.