Expanding Pseudorandom Functions; or: From Known-Plaintext Security to Chosen-Plaintext Security

Ivan Damgård and Jesper Buus Nielsen

BRICS[⋆] Department of Computer Science
University of Aarhus
Ny Munkegade
DK-8000 Arhus C, Denmark
{ivan,buus}@brics.dk

Abstract. Given any weak pseudorandom function, we present a general and efficient technique transforming such a function to a new weak pseudorandom function with an arbitrary length output. This implies, among other things, an encryption mode for block ciphers. The mode is as efficient as known (and widely used) encryption modes as CBC mode and counter (CTR) mode, but is provably secure against chosen-plaintext attack (CPA) already if the underlying symmetric cipher is secure against known-plaintext attack (KPA). We prove that CBC, CTR and Jutla's integrity aware modes do not have this property. In particular, we prove that when using a KPA secure block cipher, then: CBC mode is KPA secure, but need not be CPA secure, Jutla's modes need not be CPA secure, and CTR mode need not be even KPA secure. The analysis is done in a concrete security framework.

1 Introduction

A block cipher that is secure against known plaintext attacks (KPA) is a natural example of a *weak pseudorandom function*: as long as the key is unknown, and an adversary is given a set of *random* plaintext blocks and corresponding ciphertext blocks (of length, say, k bits each), he cannot distinguish the encryption function from a random function mapping k bits to k bits – or at least he can only do so with a small advantage. In general, a weak pseudorandom function is just a function F that maps a key K and input string x to a string y, we write $y \leftarrow F(K, x)$. It does not have to be invertible and y does not have to be the same length as x. Weak pseudorandomness means that even if an adversary is given $(x_1, F_K(x_1)), \ldots, (x_t, F_K(x_t))$, for uniformly random K and x_i, he cannot distinguish this from $(x_1, R(x_1)), \ldots, (x_t, R(x_t))$ where R is a random function. Pseudorandomness means that distinguishing remains hard, even if the adversary gets to pick the x_i's. An example of a weak pseudorandom function can be

⋆ Basic Research in Computer Science,
Centre of the Danish National Research Foundation.

derived in a natural way from the Decisional Diffie-Hellman Assumption (DDH). Here there are some public parameters, a large prime P and an element g of large prime order q such that the DDH assumption holds in the group generated by g. A random element $\alpha \in \mathbf{Z}_q$ functions as the secret key, the domain is the group generated by g, and the function maps h to $h^\alpha \bmod P$.

In this paper, we consider the case where we are given any weak pseudorandom function F with output length k. We present a general construction transforming F into a weak pseudorandom function G for which $G_K(x)$ is nk bits long and where n is any positive integer. The construction is efficient in the sense that to evaluate G, we need exactly n evaluations of F. We believe this is the best we can expect, since if nk random looking bits could be generated with fewer evaluations of F, it seems the construction would have to do some sort of pseudorandom generation "on its own". Whereas this construction requires key size logarithmic is n, we give a way to modify the construction such that variable output length can be handled using a constant amount of key material.

Note that G may also be used as a pseudorandom generator that expands K and a randomly chosen input into a random looking string of nk bits. If we apply our construction to the above function based on the DDH assumption, we obtain a pseudorandom generator that outputs k bits for each exponentiation done, where k is the security parameter. This is slightly faster than the generator that follows from Naor and Reingolds DDH-based pseudorandom function [NR97].

Our construction also implies an encryption mode for block ciphers that is secure against chosen plaintext attacks (CPA) already if the block cipher is KPA secure. We call this Pseudorandom Tree (PRT) mode. While an encryption scheme with such properties could be easily constructed using generic methods [BM84,GL89], the result would be very inefficient. In contrast, to encrypt/decrypt a message, PRT requires exactly the same number of block encryptions as the well known CBC and CTR modes and produces ciphertexts of the same lengths as does CBC/CTR. The only overhead compared to CBC/CTR is logarithmic in the length of the message, namely a key-schedule requiring $2\log_2(n))$ key expansions and block encryptions for a message containing n blocks. This can be precomputed, so that the actual throughput can be exactly the same as in the standard modes. Note that CBC and CTR also provide CPA security, but here we have to assume that the block cipher is also CPA secure. As in CTR mode, our mode allows easy random access decryption: To decrypt an arbitrary block in an n block message, we need at most $\log_2(n)$ block encryptions. Also as in CTR mode, we only use the block cipher in the encryption direction, even when we decrypt – in implementations, this can sometimes save code or chip area.

From a theoretical point of view, it is of course always better to prove security under the weakest possible assumption. But we believe the result is also useful from a more practical point of view: even though a block cipher was designed to be CPA secure, there may always be surprises and hence relying only on the KPA security of the block cipher gives extra protection. Moreover, we prove in the concrete security framework a bound on how CPA security of PRT mode relates to KPA security of the block cipher, i.e., if we assume the block cipher can be broken with at most a certain advantage under a KPA, we obtain a bound on the advantage with which PRT can be broken under a CPA. This bound behaves similarly to the bound that relates CPA security of CBC to CPA security of the block cipher. Now, considering that it may well be reasonable to assume that the block cipher performs better against KPA than against CPA, we see that we may get a better concrete bound on the security of PRT than we can get for CBC.

We mentioned that PRT introduces a slight overhead compared to using CBC or CTR. One might wonder whether it is possible to obtain the "amplification" from KPA to CPA with no overhead at all. As a partial answer, in Section 4, we analyse CBC mode, CTR mode and Jutla's integrity aware modes [Jut01] and show that none of these schemes can guarantee CPA security unless the underlying block cipher is CPA secure itself. Indeed, we prove that when using a KPA secure block cipher, CBC mode and Jutla's modes are KPA secure, but need not be CPA secure and counter mode need not even be KPA secure. The same simple techniques applied there can be applied to most other known modes of operations to demonstrate that they also do not guarantee CPA security unless the underlying block cipher is CPA secure itself (to our knowledge they can be applied to all *efficient* such modes). We therefore leave it as an interesting open question to investigate whether one can amplify from KPA to CPA and make do with a constant amount of overhead.

Finally, we discuss the problem of achieving chosen ciphertext (CCA) security (in addition to CPA). While we can argue that this is possible based only on a KPA secure block cipher, some overhead is introduced, and we do not know how to provide CCA security as efficiently as the CPA secure construction.

Related Work. The first work to relate the security of modes of operations to the underlying block cipher is [BDJR97]. In [BDJR97] notions of CPA security and CCA security of block ciphers and symmetric encryption are developed in a concrete security framework, and the security of three well-known encryption modes, CBC mode and CTR mode (in its deterministic and probabilistic variants), are analysed. In [BDJR97] the modes of operations are analysed under the assumption that the underlying block cipher is CPA secure. In Section 2 we review their framework and make a straightforward extension to cover KPA security.

The first, and to our knowledge only, other paper to consider security enhancing modes of operations is [HN00]. What is investigated there is the possibility to generate a key stream given a one-way function, after which encryption is just Xoring the message and key stream. To be able to base their construction on block-ciphers, they consider the function $f : K \mapsto E_K(P)$ mapping a key to the encryption of a fixed plaintext under that key and assume that it is one-way. This assumption is of course well-motivated: If the encryption function should be secure in any reasonable sense, one should not be able to find the key given a plaintext/ciphertext pair. However, this assumption alone is not enough: The scheme, called Key Feedback Mode (KFB) is a variant of the technique of [BM84,Lev85,GL89], and works by iterating the one-way function, i.e., we compute $f(K), f(f(K)), \ldots$ and extract some number of bits from each value computed. For this to work, one must assume that f stays one-way even when iterated, i.e., from $f^i(K)$ it is hard to find $f^{i-1}(K)$. We believe that this assumption is closely related to our assumption on KPA security, but the assumptions do not seem to be directly comparable. In any case, KFB is significantly less efficient than PRT: in KFB, a key expansion is needed for every block encryption performed, and each block encryption can only contribute significantly less than k bits to the output stream (where k is block size) – from an asymptotic point of view, only a logarithmic number of bits can be produced. In contrast, PRT can use all k bits produced by each block encryption, and needs a logarithmic number of key expansions.

An Intuitive Introduction to the Main Idea. Assume that we have access to a length-preserving and weak pseudorandom function $F_\cdot(\cdot)$, where F_K could be, e.g., the encryption function of a KPA secure block cipher. From this we can build a length doubling weak pseudorandom function G_K by letting the key be a pair of keys $K = (K_1, K_2)$ for F, letting the input domain be that of F, and letting the output be $G_{K_1,K_2}(R) = F_{K_1}(R)\|F_{K_2}(R)$. We can repeat this operation as follows. Let $C_1\|C_2 = G_{K_1,K_2}(R)$ and assume we have another key (K_1', K_2') for G. We can then compute $C_3\|C_4 = G_{K_1',K_2'}(C_1)$ and $C_5\|C_6 = G_{K_1',K_2'}(C_2)$ and let the output (generated from input R) be $C_1\|C_2\|C_3\|C_4\|C_5\|C_6$. We then have a function six-doubling its input using six applications of F. Continuing this we will soon have a pseudorandom function with a very large output length. Using essentially the fact that a new key is used for each 'level' in this construction, it can be shown to be weak pseudorandom if F has this property. This construction can also handle variable-length output, if we can find a secure way to produce produce a key for each level in the construction (to an arbitrary depth) from a constant number of keys for F. How to do this using only weak pseudorandomness of F is shown in detail in Section 3. Finally, to use this construction to encrypt a message M, we can simply choose R at random, and let the ciphertext be $R, G_K(R) \oplus M$, where G is the function we construct from the encryption function of the block cipher. Clearly, a CPA attack on this scheme will allow the adversary to get the value of G_K on random inputs – and nothing more than that. Hence weak pseudorandomness of G is sufficient for CPA security.

2 Notions of Security

The following definitions are straightforward extensions of definitions from [BDJR97,Des00] to consider also KPA security. Of the four notions of security considered in [BDJR97] we have chosen real-or-random (ROR) indistinguishability. A symmetric encryption scheme $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ consists of three randomized algorithms. The key generation algorithm $\mathcal{K}$ returns a key K; we write $K \leftarrow \mathcal{K}$. The encryption algorithm $\mathcal{E}$ takes as input the key K and a plaintext M and returns a ciphertext C; we write $C \leftarrow \mathcal{E}_K(M)$. The decryption algorithm $\mathcal{D}$ takes as input the key K and a string C and returns a unique plaintext M or $\perp$; we write $x \leftarrow \mathcal{D}_K(C)$. We require that $\mathcal{D}_K(\mathcal{E}_K(M)) = M$ for all $M \in \{0,1\}^*$.

Definition 1 (ROR-KPA, ROR-CPA). *Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme. Let $b \in \{0,1\}$. Let A be an adversary that has access to an oracle. Let $\mathcal{R}_{K,b}$ be the oracle which on input $l \in \boldsymbol{N}$, if $b = 1$, outputs $(x, \mathcal{E}_K(x))$ for uniformly random $x \in \{0,1\}^l$, and, if $b = 0$, outputs $(x, \mathcal{E}_K(r))$ for uniformly random $x, r \in \{0,1\}^l$. Let $\mathcal{O}_{K,b}$ be the oracle which on input $x \in \{0,1\}^*$, if $b = 1$, outputs $\mathcal{E}_K(x)$, and, if $b = 0$, outputs $\mathcal{E}_K(r)$ for uniformly random r of the same length as x. Now consider the following experiments:*

```
proc $\mathbf{Exp}^{\text{ror-kpa-}b}_{\mathcal{SE},A} \equiv$
    $K \leftarrow \mathcal{K}$
    $d \leftarrow A^{\mathcal{R}_{K,b}}$
    return $d$
```

```
proc $\mathbf{Exp}^{\text{ror-cpa-}b}_{\mathcal{SE},A} \equiv$
    $K \leftarrow \mathcal{K}$
    $d \leftarrow A^{\mathcal{O}_{K,b}}$
    return $d$
```

We define the advantage *of the adversary via*

$$\mathbf{Adv}^{\text{ror-kpa}}_{\mathcal{SE},A} = \Pr[\mathbf{Exp}^{\text{ror-kpa-}1}_{\mathcal{SE},A} = 1] - \Pr[\mathbf{Exp}^{\text{ror-kpa-}0}_{\mathcal{SE},A} = 1]$$
$$\mathbf{Adv}^{\text{ror-cpa}}_{\mathcal{SE},A} = \Pr[\mathbf{Exp}^{\text{ror-cpa-}1}_{\mathcal{SE},A} = 1] - \Pr[\mathbf{Exp}^{\text{ror-cpa-}0}_{\mathcal{SE},A} = 1] \ .$$

We define the advantage function of the scheme *as follows. For any integers t, q, μ,*

$$\mathbf{Adv}^{\text{ror-kpa}}_{\mathcal{SE}}(t,q,\mu) = \max_A \left\{ \mathbf{Adv}^{\text{ror-kpa}}_{\mathcal{SE},A} \right\}$$
$$\mathbf{Adv}^{\text{ror-cpa}}_{\mathcal{SE}}(t,q,\mu) = \max_A \left\{ \mathbf{Adv}^{\text{ror-cpa}}_{\mathcal{SE},A} \right\}$$

where the maximum is over all A with "time complexity" t, making at most q queries to the oracle, these totaling at most μ bits.

By the "time complexity" we mean the worst case total running time of the experiment with $b = 1$, plus the size of the code of the adversary, in some fixed RAM model of computation. We stress that the total execution time of the experiment includes the time of *all* operations in the experiment, including the time for key generation and the encryptions done by the oracle. For a discussion of this time complexity, see [Des00].

A function family with key-space $\mathcal{K}$, input-length l, and output-length L is a map $F : \mathcal{K} \times \{0,1\}^l \to \{0,1\}^L$. For each key $K \in \mathcal{K}$ we define a map $F_K : \{0,1\}^l \to \{0,1\}^L$ by $F_K(x) = F(K,x)$. We write $f \overset{R}{\leftarrow} F$ for the operation $K \overset{R}{\leftarrow} \mathcal{K}$; $f \leftarrow F_K$. We call F a family of permutations if for all $K \in \mathcal{K}$, F_K is a permutation. We use $\text{Rand}^{l \to L}$ to denote the family of all functions $\{0,1\}^l \to \{0,1\}^L$.

If a random function from the function family looks like a random function from $\text{Rand}^{l \to L}$, we call the family a pseudorandom function family. Below we define this notion formally for KPAs. The definitions for CPAs and CCAs can be found in [BDJR97], but will not be used in the present paper.

A variable-length output function family with key-space $\mathcal{K}$ and input-length l is a map $F : \mathcal{K} \times \boldsymbol{N} \times \{0,1\}^l \to \{0,1\}^*$. For each key $K \in \mathcal{K}$ we define a map $F_K : \boldsymbol{N} \times \{0,1\}^l \to \{0,1\}^*$ by $F_K(L,x) = F(K,L,x)$. We require that $|F(K,L,x)| = L$ for all inputs. We use VO-Rand^l to denote the probabilistic function generated as follows: On input (L,r) check if a string o_r is defined, if not define it to be the empty string. Then check whether o_r has length at least L, if not append to o_r a uniformly random string from $\{0,1\}^{L-|o_r|}$. Then output the L first bits of o_r. We define what it means for at variable-length output function family to be KPA secure.

Definition 2 (VO-PRF-KPA). *Let F be a variable-length output function family with input-length l. Let D be a distinguisher that has access to an oracle. Let $\mathcal{R}_f$ be the oracle which on input $L \in \boldsymbol{N}$ generates a uniformly random $r \in \{0,1\}^l$ and outputs $(r, f(L,r))$. Now consider the following experiments:*

```
proc $\mathbf{Exp}^{\text{vo-prf-kpa-}0}_{F,D}$ ≡
  $d \leftarrow D^{\mathcal{R}_{\text{VO-Rand}^l}}$
  return $d$

proc $\mathbf{Exp}^{\text{vo-prf-kpa-}1}_{F,D}$ ≡
  $f \overset{R}{\leftarrow} F$
  $d \leftarrow D^{\mathcal{R}_f}$
  return $d$
```

We define the advantage *of the distinguisher via*

$$\mathbf{Adv}^{\text{vo-prf-kpa}}_{F,D} = \Pr[\mathbf{Exp}^{\text{vo-prf-kpa-}1}_{F,D} = 1] - \Pr[\mathbf{Exp}^{\text{vo-prf-kpa-}0}_{F,D} = 1] \ .$$

We define the advantage function of the function family *as follows. For any t, q, l,*

$$\mathbf{Adv}^{\text{vo-prf-kpa}}_{F}(t,q,\mu) = \max_{D} \left\{ \mathbf{Adv}^{\text{vo-prf-kpa}}_{F,D} \right\} \ .$$

where the maximum is over all D with time complexity t, making at most q queries to the oracle, these totaling at most μ bits..

The notion of KPA security of a fixed output function family with input-length l can easily be derived from the above definition, giving rise to the notions $\mathbf{Adv}^{\text{prf-kpa}}_{F,D}$ and $\mathbf{Adv}^{\text{prf-kpa}}_{F}(t,q)$ – we skip the explicit mentioning of μ as it is given by q.

3 PRT Mode

PRT mode is a construction of a VO-PRF-KPA secure variable-length output function family from a PRF-KPA secure function family. The encryption will then be done using the variable-length output function family F as

$$\text{VO-PRF-ENC}[F]_K(M) = (r, F_K(|M|, r) \oplus M) \ ,$$

where r is uniformly random in $\{0,1\}^l$. The following theorem relates the ROR-CPA security of VO-PRF-ENC$[F]$ to the VO-PRF-KPA security of F.

Theorem 1. *Suppose F is a variable-length output function family. If F is VO-PRF-KPA secure, then* VO-PRF-ENC[F] *is ROR-CPA secure*[1]*. Specifically, for any t, q, μ,*

$$\mathbf{Adv}^{\text{ror-cpa}}_{\text{VO-PRF-ENC}[F]}(t,q,\mu) \leq \mathbf{Adv}^{\text{vo-prf-kpa}}_{F}(t,q,\mu) + \frac{q(q-1)}{2^{l+1}} \ .$$

Proof: We prove the specific bound. Consider an ROR-CPA distinguisher $\overline{D}$ expecting access to an oracle $\mathcal{O}_{K,b}$ for the VO-PRF-ENC[F] scheme. We construct a distinguisher D having access to a VO-PRF-KPA oracle $\mathcal{R}_f$ for the variable-length output function family F as follows. The distinguisher D runs the code of $\overline{D}$. Each time D request an encryption of message M, request a pair (r, R), where r is uniformly random in $\{0,1\}^l$ and $R = f(|M|, r)$. Then return $c = (r, M \oplus R)$. When D returns with some value d, return d.

If $b = 1$, then f is a random function from F and the values c are distributed as values from $\mathcal{O}_{K,1}$. If on the other hand $b = 0$, then f is VO-Rand^l, and in that case the values c are distributed as values from $\mathcal{O}_{K,0}$, as long as there are no collisions among the r-values returned by $\mathcal{R}_f$. Since the r values are uniformly random l-bit values and q of them are drawn, the probability of collision are well-known to be upper bounded by $q(q-1)/2^{l+1}$, which proves the theorem.

■

Security Preserving Operations on KPA Secure PRFs. Before presenting the actual construction, we present some operations on PRFs which preserves KPA security. Assume that we are given PRFs

$$F : \{0,1\}^{m_1} \to \{0,1\}^{n_1}, \qquad G : \{0,1\}^{m_2} \to \{0,1\}^{n_2}$$

with key-length k_1 resp. k_2. For operations only involving F, we use the notation $m = m_1, n = n_1, k = k_1$. Our first operation makes the output domain larger. It gives the function family

$$F^{\to\beta} : \{0,1\}^m \to \{0,1\}^{\beta n}, \qquad F^{\to\beta}_{K_1,\ldots,K_\beta}(R) = F_{K_1}(R) \| \cdots \| F_{K_\beta}(R) \ ,$$

where we generate a key for $F^{\to\beta}$ by generating β independent keys $K_1, \ldots, K_\beta$ for F. Our second operation is similar, but has shorter key-length. Assume that $k \leq m$, so that an output-block can be used as key, and consider the following function family

[1] Actually, we have not assigned a meaning to the claim that VO-PRF-ENC[F] is ROR-CPA secure if F is VO-PRF-KPA secure, as we have no definition of security: In this paper we consider a concrete security framework without a security parameter. If, however, we introduced a security parameter k, then in the asymptotic security framework, all of t, q, μ, l, and L would be polynomial in k and typically $l = \Theta(k)$. We would then define security by requiring that the advantage of all probabilistic polynomial time (in k) adversaries is negligible (in k). The claim would then follow from the specific bound on $\mathbf{Adv}^{\text{ror-cpa}}_{\text{VO-PRF-ENC}[F]}(t,q,\mu)$ given by the theorem. In the following we will use the term “secure” with this meaning.

$$F^{\to\overline{\beta}} : \{0,1\}^m \to \{0,1\}^{\beta n}, \qquad F^{\to\overline{\beta}}_{K,R_1}(R_2) = F_{K_1}(R_2) \| \cdots \| F_{K_\beta}(R_2) \ ,$$

where $K_1 = K$ and inductively $K_{i+1} = F_{K_i}(R_1)$. Now consider the operation making both the input and the output domain larger

$$F^{\alpha\to\alpha} : \{0,1\}^{\alpha m} \to \{0,1\}^{\alpha n}, \qquad F^{\alpha\to\alpha}_K(R) = F_K(R_1) \| \cdots \| F_K(R_\alpha) \ ,$$

where $R = (R_1, \ldots, R_\alpha)$. For completeness we name the following operation

$$F^{\alpha\to\alpha\beta} : \{0,1\}^{\alpha m} \to \{0,1\}^{\alpha\beta n}, \qquad F^{\alpha\to\beta} = (F^{\to\beta})^{\alpha\to\alpha} \ .$$

Finally assume that $n_1 \geq m_2$ and consider the following composition operation

$$G \circ F : \{0,1\}^{m_1} \to \{0,1\}^{n_1+n_2}, \qquad (G \circ F)_{K_1,K_2}(R) = F_{K_1}(R) \| G_{K_2}(F_{K_1}(R)) \ .$$

We give a short example of the use of these operations. Assume that we are given any KPA secure PRF $F : \{0,1\}^m \to \{0,1\}^m$. From this family, we can using the $\to 2$ operation and construct a KPA secure PRF $G : \{0,1\}^m \to \{0,1\}^{2m}$. From G we can then construct the KPA secure PRF $G^{2\to 2} : \{0,1\}^{2m} \to \{0,1\}^{4m}$, and can using the composition operation construct the KPA secure PRF $G^{2\to 2} \circ G : \{0,1\}^m \to \{0,1\}^{6m}$. This can be iterated. In Fig. 1 the PRF $G^{8\to 8} \circ G^{4\to 4} \circ G^{2\to 2} \circ G : \{0,1\}^m \to \{0,1\}^{30m}$ is depicted. This construction works even if F is not length preserving. We can always define G by computing $F^{\to\beta}$ for appropriate choice of β and use the first $2m$ bits of the output.

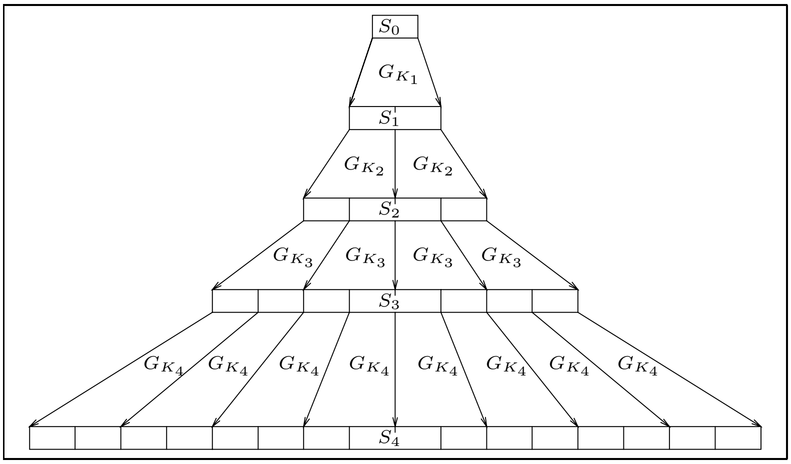

Fig. 1. The structure of the PRF $G^{8\to 8} \circ G^{4\to 4} \circ G^{2\to 2} \circ G$. Note that the output of the function is all levels except the root level, which is the input. The key $K = K_1 \| K_2 \| K_3 \| K_4$ consists of four keys for the PRF G.

Lemma 1. *If F is PRF-KPA secure, then $F^{\to\beta}$ and $F^{\to\overline{\beta}}$ are PRF-KPA secure. Specifically, for any t, q,*

$$\begin{aligned}\mathbf{Adv}^{\text{prf-kpa}}_{F^{\to\beta}}(t,q) &\leq \beta\mathbf{Adv}^{\text{prf-kpa}}_{F}(t,q)\\ \mathbf{Adv}^{\text{prf-kpa}}_{F^{\to\overline{\beta}}}(t,q) &\leq \beta\mathbf{Adv}^{\text{prf-kpa}}_{F}(t,q+1)+\frac{q}{2^m}\ .\end{aligned}$$

Proof: We do the proof for $F^{\to\overline{\beta}}$. The proof for $F^{\to\beta}$ is equivalent.

Let $S = S_1\|\cdots\|S_\beta$ be a random function from $\{0,1\}^m$ to $\{0,1\}^{\beta n}$. Let K be a random key for F, let R_1 be uniformly random in $\{0,1\}^m$ and for $i = 1,\ldots,\beta+1$ let

$$H^i_{S,K,R_1}(R_2) = S_1(R_2)\|\cdots\|S_{i-1}(R_2)\|F_{K_i}(R_2)\|\cdots\|F_{K_\beta}(R_2)\ ,$$

where $K_i = K$ and inductively $K_{j+1} = F_{K_j}(R_1)$. Let D be any distinguisher running in time t using q queries. We want to prove that D cannot distinguish $F^{\to\overline{\beta}}$ and S with advantage better than $\beta\mathbf{Adv}^{\text{prf-kpa}}_{F}(t,q+1)+\frac{q}{2^m}$. Since the event that R_1 occurs as one of the uniformly random R_2 values is at most $\frac{q}{2^m}$ it is enough to prove that D cannot distinguish $F^{\to\overline{\beta}}$ and S with advantage better than $\beta\mathbf{Adv}^{\text{prf-kpa}}_{F}(t,q+1)$ when this event does not occur. So, since $H^1_{S,K,R_1} = F^{\to\overline{\beta}}_{K,R_1}(R_2)$ and $H^{\beta+1}_{S,K,R_1} = S$ it is in turn enough to prove that for $i = 1,\ldots,\beta$, D cannot distinguish H^i_{S,K,R_1} from H^{i+1}_{S,K,R_1} with better advantage than $\mathbf{Adv}^{\text{prf-kpa}}_{F}(t,q+1)$.

To prove this assume we have access to an oracle $\mathcal{R}_f$ where f is a random function from F or a random function $\{0,1\}^m \to \{0,1\}^n$. We simulate an oracle to D as follows: First request a generation from $\mathcal{R}_f$ and obtain (R_1,T_1). Let $K_{i+1} = T_1$ and inductively $K_{j+1} = F_{K_j}(R_1)$. Further more maintain a random function $S = S_1\|\cdots\|S_{i-1}$ using a dictionary. When D request a generation, request a generation from $\mathcal{R}_f$ to obtain (R_2,T_2), and return to D the value $(R_2, H_{S,f,R_1}(R_2))$, where

$$H_{S,f,R_1}(R_2) = S_1(R_2)\|\cdots\|S_{i-1}(R_2)\|T_2\|F_{K_{i+1}}(R_2)\|\cdots\|F_{K_\beta}(R_2)\ .$$

If $f = F_K$, then using the renaming $K_i = K$ we have that $K_{i+1} = F_{K_i}(R_1)$ and $T_2(R_2) = F_{K_i}(R_2)$ and thus $H_{S,f,R_1}(R_2) = H^i_{S,K,R_1}(R_2)$. If f is a random function $R : \{0,1\}^m \to \{0,1\}^n$, then $K_{i+1} = R(R_1)$ and

$$\begin{aligned}H_{S,f,R_1}(R_2) &= S_1(R_2)\|\cdots\|S_{i-1}(R_2)\|R(R_2)\|F_{K_{i+1}}(R_2)\|\cdots\|F_{K_\beta}(R_2)\\ &= H^{i+1}_{S,K_{i+1},R_1}(R_2)\ .\end{aligned}$$

Since K_{i+1} is uniformly random and independent of all the other values as long as no R_2 equals R_1, the theorem follows. ■

Lemma 2. *If $F : \{0,1\}^m \to \{0,1\}^n$ is PRF-KPA secure, then $F^{\alpha\to\alpha}$ is PRF-KPA secure. Specifically, for any t, q,*

$$\mathbf{Adv}^{\text{prf-kpa}}_{F^{\alpha\to\alpha}}(t,q) \leq \mathbf{Adv}^{\text{prf-kpa}}_{F}(t,\alpha q)+\frac{q\alpha(q\alpha-1)}{2^{m+1}}\ .$$

Proof: Assume that D can distinguish $D_1 = (R_1, \ldots, R_\alpha, F_K(R_1), \ldots, F_K(R_\alpha))$ from $D_0 = (R_1, \ldots, R_\alpha, S_1, \ldots, S_\alpha)$ with probability δ when the S_i are uniformly random values. We use D to distinguish values of the form (R_i, T_i) where $T_i = F_K(R_i)$ if $b = 1$ and $T_i = S_i$ if $b = 1$. When D asks for a value, get α values $(R_1, T_1), \ldots, (R_\alpha, T_\alpha)$ and hand $V = (R_1, \cdots, R_\alpha, T_1, \cdots, T_\alpha)$ to D. If $b = 1$, then $V = D_1$ and if $b = 0$, then $V = D_0$ unless there are identical R_i values – if there are identical R_i values, the corresponding S_i values will also be identical, which would typically not occur if $S = S_1 \| \cdots \| S_\alpha$ was a uniformly random αn-bit-string. Since the R_i values are m-bit values and there is generated a total of $q\alpha$ of them, the probability of collisions is bounded by $\frac{q\alpha(q\alpha-1)}{2^{m+1}}$, which proves the theorem. ■

Lemma 3. *If F and G are PRF-KPA secure, then $G \circ F$ is PRF-KPA secure. Specifically, for any t, q,*

$$\mathbf{Adv}_{G \circ F}^{\text{prf-kpa}}(t, q) \leq \mathbf{Adv}_F^{\text{prf-kpa}}(t, q) + \mathbf{Adv}_G^{\text{prf-kpa}}(t, q) + \frac{q(q-1)}{2^{m_1+1}} .$$

Proof: We must show that one cannot distinguish D_0 and D_1, where $D_1 = (R, F_{K_1}(R), G_{K_2}(F_{K_1}(R)))$, where R is random, and $D_0 = (R, R_1(R), R_2(R))$, where R is random and R_1 and R_2 are random functions. We consider two hybrids: $H_1 = (R, R_1(R), G_{K_2}(R_1(R)))$, where R is random and R_1 is a random function and $H_2 = (R, R_1(R), R_2(R_1(R)))$, where R is random and R_1 and R_2 are random functions. It is easy to see that $H_2 = D_0$ unless identical R_1 values occur. To prove the lemma it is therefore enough to prove that D_1 cannot be distinguished from H_1 with better advantage than $\mathbf{Adv}_F^{\text{prf-kpa}}(t, q)$, and that H_1 cannot be distinguished from H_2 with better advantage than $\mathbf{Adv}_G^{\text{prf-kpa}}(t, q)$. Both claims follows using trivial reductions. ■

The PRT Family. As the basic primitive in our PRT construction, we will need a KPA secure PRF $G : \{0,1\}^m \to \{0,1\}^n$ with key length k, where $n \geq \max(2m, k)$. Using Lemmas 1 and 2 such a function can be constructed from any KPA secure PRF F with k-bit keys.

If we use Rijndael with 128 bit keys and 128-bit blocks (call this function family Rin), we can let $G = \text{Rin}^{\to 2}$. Then G has 256-bit keys, 128-bit input, and 256-bit output. Keys will simply consist of two Rijndael keys, and the function will be $\text{Rin}^{\to 2}_{K_1,K_2}(R) = \text{Rin}_{K_1}(R) \| \text{Rin}_{K_2}(R)$. If DES is used, then we can let $G = \text{DES}^{\to 2}$. Then G has 112-bit keys, 64-bit input, and 128-bit output. Since for all families which one would use in practice, the loss of security in going from F to G is minimal, we will in the following express the security of our construction in that of G.

Starting with a random input S_0 for G and keys $K = (K_1, \ldots, K_d)$, we can compute a pseudorandom output of length exponential in d by computing $(G^{2^{d-1} \to 2^{d-1}} \circ \cdots \circ G^{2 \to 2} \circ G)_K(S_0)$. However, we are after a VO-PRF which the construction will not provide for any fixed number of key. This is the reason for

```
proc PRT_K(R, l) ≡
  $T_0 = K[1..k]$
  $R_1 = K[(k+1)..(k+m)]$
  $R_2 = K[(k+m+1)..(k+2m)]$
  $S_0 = R$
  $O = \epsilon$
  $i = 1$
  while $|O| < l$ do
        $K_i = (G_{T_{i-1}}(R_1))[1..k]$
        $T_i = (G_{T_{i-1}}(R_2))[1..k]$
        $S_{i+1} = \epsilon$
        $j = 1$
        while $j \leq |S_{i-1}| - m + 1$ do
              $S_i = S_i \| G_{K_i}(S_{i-1}[j..(j+m-1)])$
              $j = j + m$ od
        $O = O \| S_i$
        $i = i + 1$ od
  return $O[1..l]$
```

Fig. 2. The VO-PRF $\mathrm{PRT}_K(R, l)$ obtained from a PRF $G : \{0,1\}^m \to \{0,1\}^n$ with key-length k where $n \geq \max(2m, k)$. The domains of the inputs are $K \in \{0,1\}^{k+2m}$ and $R \in \{0,1\}^m$. The first two lines of the outer loop constitutes the key scheduling and can be preprocessed to prepare the functions $G_{K_1}, \ldots, G_{K_l}$ to some appropriate depth.

the second requirement that the output of G is at least as long as the key. This allows to schedule an arbitrary number of keys from one key K using Lemma 1: Pick $R \in \{0,1\}^{2m}$ at random and to schedule d keys, compute $G^{\to \overline{d}}(R)$. This gives a random string of length $dn \geq dk$ and allows to define d random keys. The entire construction is given in more detail in Fig. 2 using pseudo-code.

Theorem 2. *If $G : \{0,1\}^m \to \{0,1\}^n$ is PRF-KPA secure, then* $\mathrm{PRT}[G]$ *is VO-PRF-KPA secure. Specifically, for any t, q, μ,*

$$\mathbf{Adv}^{\text{prf-kpa}}_{\mathrm{PRT}[G]}(t, q, \mu) \leq d(\mathbf{Adv}^{\text{prf-kpa}}_{G}(t, ql) + \mathbf{Adv}^{\text{prf-kpa}}_{G}(t, q+1)) + \frac{ql(ql-1)+q}{2^m}$$

where L is the maximal length of a query in bits, $l = \lceil L/m \rceil$, and $d = \log_2(l+1)$ is the maximal depth of any PRT used. Using $L \leq \mu$ this easily translates into a bound depending only on t, q and μ.

Proof: A pseudorandom tree of depth d is used to generate between $2^d - 1$ and $2^{d+1} - 2$ blocks. Thus the maximal depth of the pseudorandom trees used in each evaluation is upper bounded by d.

Note that a pseudorandom tree, $\mathrm{PRT}[F]_K(R, l)$, of depth d can be computed by first computing $\overline{K} = G^{\to \overline{d}}_{K[1..(k+m)]}(K[(k+m+1)..(k+2m)])$ and then computing $O = (\circ_{i=1}^{d} G^{2^{i-1} \to 2^{i-1}})_{\overline{K}}(R)$ and outputting $O[1..l]$.

If $\overline{K}$ was uniformly random, then by the above observations and Lemmas 2 and 3

$$\begin{aligned}\mathbf{Adv}^{\text{prf-kpa}}_{\text{PRT}[F]}(t,q,\mu) &\le \sum_{i=0}^{d-1} \mathbf{Adv}^{\text{prf-kpa}}_{G^{2^i \to 2^i}}(t,q) + \sum_{i=0}^{d-2} \frac{(q(q-1)}{2^{m+1}} \\ &\le \sum_{i=0}^{d-1} \left(\mathbf{Adv}^{\text{prf-kpa}}_{G}(t,2^i q) + \frac{q2^i(q2^i-1)}{2^{m+1}} + \frac{q(q-1)}{2^{m+1}} \right) \\ &\le \sum_{i=0}^{d-1} \mathbf{Adv}^{\text{prf-kpa}}_{G}(t,2^i q) + \frac{q2^{d-1}(q2^{d-1}-1)}{2^m} \ .\end{aligned}$$

The theorem then follows using that $2^{d-1} \le ql$ and using Lemma 1 in a hybrids argument. ■

The theorem tells us that even if G was a perfect PRF, i.e. $\mathbf{Adv}^{\text{prf-kpa}}_{G}(t,q) = 0$, the PRF that we build out of it will not necessarily be perfect. Intuitively it is easy to see that this imperfectness is not by failure of our analysis. It is the birthday attack, to which almost all encryption modes must surrender. The intuition is that if at some level in a pseudorandom tree as that in Fig. 1 a collision occurs, then because the next levels are build using functions, sub-trees under collisions will be identical. On the other hand, identical sub-tree will occur very seldom if each bit in the tree is chosen uniformly at random and independent of the other bits. A careful analysis of the probability of find such collisions will allow to prove that the bound in the theorem is fairly sharp.

To see why it is essential to the construction that different keys are used at each level, we refer the reader to Theorem 4 and the discussion following it.

4 Analysis and Comparison of CBC, CTR, Jutla's Modes, and PRT

We will now compare the security of our new encryption mode to that of the well-known encryption modes CBC and CTR, and also the integrity aware modes of Jutla[Jut01] – he proposed two modes, namely IACBC and IAPM, both of which provide both integrity and confidentiality – we will only consider confidentiality in this section, however. We are going to prove the results given by the table below, which gives the "maximal" security that holds in general for various combinations of encryption and attack modes. For instance the entry CBC \ PRF-KPA being equal to ROR-KPA means that CBC-encryption using a KPA-secure PRF family is ROR-KPA secure, and there exists a KPA-secure PRF family G such that CBC[G] is not CPA-secure.

MODE \ ATK$_{\text{impl}}$	**PRF-KPA**	**PRF-CPA**
CBC	ROR-KPA	ROR-CPA
CTR	insecure	ROR-CPA
Jutla	ROR-KPA	ROR-CPA
PRT	ROR-CPA	ROR-CPA

The bottom row and the right-most column follows from known results from [BDJR97,Jut01] and Section 3. We now prove the remaining claims in the following theorems. The CBC and CTR encryption modes are given in Fig. 3.

```
proc CBC[P]_K(M) ≡
  $m \leftarrow \lceil |M|/l \rceil$
  $n \leftarrow ml - |M|$
  $r \overset{R}{\leftarrow} \{0,1\}^n$
  $M \leftarrow M \| r$
  $c_0 \overset{R}{\leftarrow} \{0,1\}^l$
  for $i = 0$ to $m-1$ do
      $c_{i+1} \leftarrow P_K(M[il..(il+l-1)] \oplus c_i)$
  od
  return $(n, c_0 \| c_1 \| \dots \| c_m)$
```

```
proc CTR[F]_K(M) ≡
  $m \leftarrow \lceil |M|/L \rceil$
  $n \leftarrow |M| - (m-1)L$
  $r \overset{R}{\leftarrow} \{0,1\}^l$
  for $i = 1$ to $m$ do
      $r_i \leftarrow F_K(r + i \bmod 2^l)$
  od
  return $(r, M \oplus (r_1 \| \dots \| r_{m-1} \| r_m[1..n]))$
```

Fig. 3. CBC[P] mode and CTR[F] mode.

Jutla's IACBC mode is essentially CBC encryption, but where the sequence of blocks coming from the CBC encryption is Xor'ed by a sequence of pseudorandom blocks generated using an independent key. The IAPM mode first Xor's the sequence of plaintext blocks by a pseudorandom sequence of blocks, then encrypts in ECB mode, and finally Xor's the result by the same pseudorandom sequence. Both IACBC and IAPM also generate a checksum that receives special treatment, but this is not relevant for our discussion.

Theorem 3. *Suppose P is a permutation family with length l. If P is PRF-KPA secure, then* CBC[P], IACBC[P], *and* IAPM[P] *are ROR-KPA secure. Specifically, for any t, q,*

$$\mathbf{Adv}^{\text{ror-kpa}}_{\text{CBC}[P]}(t,q,\mu), \mathbf{Adv}^{\text{ror-kpa}}_{\text{IACBC}[P]}(t,q,\mu), \mathbf{Adv}^{\text{ror-kpa}}_{\text{IAPM}[P]}(t,q,\mu) \le \mathbf{Adv}^{\text{prf-kpa}}_{P}(t,\nu) + \frac{\nu(\nu-1)}{2^{l+1}} ,$$

where $\nu = \lfloor \mu/l \rfloor + q$.

Proof: Consider an ROR-KPA distinguisher $\overline{D}$ expecting access to an oracle $\mathcal{R}_{K,b}$ for the CBC[P] scheme. We construct a distinguisher D having access to a PRF-KPA oracle $\mathcal{R}_f$ for the permutation family P as follows. The distinguisher D runs the code of $\overline{D}$. Each time D requests an encryption of length m', request $m = \lceil m'/l \rceil$ pairs $(x_i, f(x_i))$ from $\mathcal{R}_f$. Then generate a random l-bit string c_0 and for $i = 1, \dots, m$ let $c_i = f(x_i)$ and let $p_i = x_i \oplus c_{i-1}$. Then output $(M, C) = (p_1 \| \dots \| p_m, (ml - m', c_0 \| c_1 \| \dots \| c_m))$.

In all cases M is uniformly random and C is distributed exactly as a CBC encryption of p using f. So, if $f = P_K$ is a random permutation from P, then (M, C) is distributed exactly as values from $\mathcal{R}_{K,1}$, and if f is a random function, then C is uniformly random and independent of M, unless M has collisions among the blocks, which proves the theorem for CBC.

Since IACBC is clearly no weaker than CBC under any notion of security the result also covers IACBC, and the result for IAPM follows from Lemma 2. ■

Theorem 4. *For any permutation family P with length l, there exists a permutation family $\overline{P}$ such that $\overline{P}$ is PRF-KPA secure if P is PRF-KPA secure, but neither* CBC$[\overline{P}]$ *nor IACBC$[\overline{P}]$ are ROR-CPA secure.*

Proof: Given some permutation family P, consider the permutation family $\overline{P}$ given by $\overline{P}_K(x_1, x_2) = (P_K^{-1}(x_2), P_K(x_1))$. A random evaluation of $\overline{P}$ just consists of two random evaluations of P, and so $\overline{P}$ is PRF-KPA secure if P is PRF-KPA secure. To see that $\overline{P}$ is not PRF-CPA secure in CBC mode, ask for an encryption of the all-zero-string of length $4l$ and use that permutations from $\overline{P}$ are their own inverses.

This can be generalized to IACBC mode: In one version of this mode, the pseudorandom sequence S that is Xor'ed to the result of CBC encryption is of form $s_i = e(i)W$, where $e()$ is a public injective map from the integers mod $2^l - 1$ to $GF(2^l)^*$, l is the block length of the cipher, and W is a pseudorandomly generated block. The multiplication is in $GF(2^l)$. Now, it is easy to see that if we request the IACBC encryption of a message with 4 zero-blocks, the first and third block output from the CBC part will be equal. Hence, the Xor of the corresponding blocks in the IACBC encryption will equal $(e(1) + e(3))W$. Since $e()$ is public and $e(1) + e(3) \neq 0$ we can compute W and hence all of S, Xor with the ciphertext and obtain the output from the CBC part. Now we are in a situation equivalent to what we had for CBC. Similar arguments apply to the other suggested variants of IACBC. ■

Note that the function $\overline{P}$ constructed in the above proof demonstrates that it is essential to the PRT construction that different keys are used at each level. The function family $\overline{P}$ is KPA-PRF secure, but if it was used in a PRT construction with the same key at each level, the tree would be scattered with repeating blocks.

Theorem 5. *For any permutation family P with length l, there exists a permutation family $\overline{P}$ such that $\overline{P}$ is PRF-KPA secure if P is PRF-KPA secure, but* CTR$[\overline{P}]$ *is not ROR-KPA secure.*

Proof: Given some permutation family P, consider the permutation family $\overline{P}$ given by $\overline{P}_K(x_1, x_2) = (P_K(x_1), P_K(x_2))$. By Lemma 2, $\overline{P}$ is PRF-KPA secure if P is PRF-KPA secure. To see that $\overline{P}$ is not PRF-KPA secure in CTR mode, note that if a permutation from $\overline{P}$ is evaluated on two consecutive elements $x, x+1$, where $x = x_1 x_2$, then the result will typically be of the form $(P_K(x_1), P_K(x_2)), (P_K(x_1), P_K(x_2 + 1))$, which will not look random. ■

Theorem 6. *There exists a function family P such that P is PRF-KPA secure, but IAPM$[P]$ is not ROR-CPA secure.*

Proof: In IAPM mode, a ciphertext block is of form $s_i \oplus P_K(p_i \oplus s_i)$ where s_i is a pseudorandom block and p_i is the i'th plaintext block. Suppose that $s_i = e(i)W$

for a random block W as described in the proof of Theorem 4. Then s_i, s_j for $i \neq j$ are related by $s_j = e(j)e(i)^{-1}s_i$. Since $e()$ ranges over all non-zero values in $GF(2^l)$, we can choose i, j such that $\alpha := e(j)e(i)^{-1}$ is a generator of $GF(2^l)^*$. Now, a function P_K in our family is constructed as follows: we will think of it as a mapping from $GF(2^l)$ to itself. We choose two random values as the images of 0 and 1. For every element α^m, where $1 \leq m < 2^l - 1$ is odd we choose a random value as image, whereas we set $P_K(\alpha^{m+1}) = \alpha P_K(\alpha^m)$. Now, P is PRF-KPA secure, because a set of random inputs has to be exponentially large in l in order to contain both of α^m, α^{m+1} for odd m with significant probability. But in a CPA on IAPM, an attacker can choose all $p_i = 0$, which means that P_K receives the sequence of s_i's as inputs. If W is random, then with probability 1/2, $s_i = \alpha^m$ for an odd m. Hence $s_j = \alpha^{m+1}$, and so we have for ciphertext blocks C_i and C_j that $C_j = s_j + P_K(s_j) = \alpha s_i + \alpha P_K(s_i) = \alpha(s_i + P_K(s_i)) = \alpha C_i$. This correlation allows to distinguish from a random encryption. Other ways to generate the s_i's can be handled in a similar way. ■

5 CCA Security

Having constructed CPA secure encryption, we can construct CCA secure encryption using a number of known techniques. One can e.g. do with a KPA secure VO-PRF $G : \{0,1\}^k \to \{0,1\}^*$ acting as key-stream generator and a CPA secure variable-length input PRF (VI-PRF) $MAC : \{0,1\}^* \to \{0,1\}^k$ acting as a MAC. Given a message M one generates a uniformly random input R for G and computes $C = R\|(G_{K_1}(|M|, R) \oplus M)$ and lets the encryption be $E_{K_1,K_2}(M) = (C, MAC_{K_2}(C))$. This scheme can be proven CCA secure using standard techniques, see e.g. [Des00].

We can construct a CPA secure VI-PRF from a KPA secure PRF using known techniques. From any KPA secure PRF F one can build a pseudorandom *generator* by mapping key K and input R for the PRF to $F_K(R)$. Using the technique in Section 3 the PRF can be modified to give this pseudorandom generator expansion factor two. Using the technique in [GGM86] this then allows to build a CPA secure VI-PRF using in the order of l applications of F per evaluation, where l is the length of the message.

To do a CCA secure encryption using PRT mode, one will then need $l/k + \log_2(l/k)$ applications of F for the key-stream, where k is the block-size, and l applications of F for the MACing. This is too large a overhead for the solution to be practical and leaves the open problem of finding an efficient CCA secure encryption scheme relying only on the KPA security of the underlying block cipher.

If one is willing to make the extra assumption that a collision resistant hash-function $H : \{0,1\}^* \to \{0,1\}^h$ is given the above scheme can be made practical. By first hashing the message and then MACing the hash, the result is still a CPA secure VI-PRF. However, now the price for encrypting is (neglecting the price for hashing) $l/k + \log_2(l/k) + h$ applications of F, where h in current practice could be 160.

6 Conclusion

We have shown how to efficiently enlarge the output-length of a weak pseudorandom function and how to use this for constructing CPA secure encryption from any weak pseudorandom function without essential loss of efficiency compared to known modes as CBC and CTR. We showed that also CCA secure encryption can be based on a KPA secure PRF, and opened the problem of finding an *efficient* CCA secure encryption scheme based on a KPA secure PRF.

References

BDJR97. 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* [IEE97].

BM84. Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM Journal on Computing*, 13(4):850–864, November 1984.

Des00. Anand Desai. New paradigms for constructing symmetric encryption schemes secure against chosen-ciphertext attack. In Mihir Bellare, editor, *Advances in Cryptology - Crypto 2000*, pages 394–412, Berlin, 2000. Springer-Verlag. Lecture Notes in Computer Science Volume 1880.

GGM86. Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986.

GL89. Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the Twenty First Annual ACM Symposium on Theory of Computing*, pages 25–32, Seattle, Washington, 15–17 May 1989.

HN00. Johan Håstad and Mats Näslund. Key feedback mode: a keystream generator with provable security. 2000.

IEE97. IEEE. *38th Annual Symposium on Foundations of Computer Science*, Miami Beach, FL, 19–22 October 1997.

Jut01. Charanjit S. Jutla. Encryption modes with almost free message integrity. In *Advances in Cryptology - EuroCrypt 2001*, pages 529–544, Berlin, 2001. Springer-Verlag. Lecture Notes in Computer Science Volume 2045.

Lev85. Leonid A. Levin. One-way functions and pseudorandom generators. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, pages 363–365, Providence, Rhode Island, 6–8 May 1985.

NR97. Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions (extended abstract). In *38th Annual Symposium on Foundations of Computer Science* [IEE97], pages 458–467.