Range Extension for Weak PRFs; The Good, the Bad, and the Ugly

Krzysztof Pietrzak[1] ⋆ and Johan Sjödin[2] ⋆⋆

[1] CWI Amsterdam
pietrzak@cwi.nl
[2] Department of Computer Science, ETH Zurich, CH-8092 Zurich, Switzerland
sjoedin@inf.ethz.ch

Abstract. We investigate a general class of (black-box) constructions for range extension of weak pseudorandom functions: a construction based on m independent functions $F_1, \ldots, F_m$ is given by a set of strings over $\{1, \ldots, m\}^*$, where for example $\{\langle 2\rangle, \langle 1, 2\rangle\}$ corresponds to the function $X \mapsto [F_2(X), F_2(F_1(X))]$. All efficient constructions for range expansion of weak pseudorandom functions that we are aware of are of this form.

We completely classify such constructions as *good*, *bad* or *ugly*, where the good constructions are those whose security can be proven via a black-box reduction, the bad constructions are those whose *in*security can be proven via a black-box reduction, and the ugly constructions are those which are neither good nor bad.

Our classification shows that the range expansion from [10] is optimal, in the sense that it achieves the best possible expansion ($2^m - 1$ when using m keys).

Along the way we show that for weak *quasirandom* functions (i.e. in the information theoretic setting), all constructions which are not bad – in particular all the ugly ones – are secure.

1 Introduction

Pseudorandomness, introduced by Blum and Micali, is a crucial concept in theoretical computer science in general, and cryptography in particular. Informally, an object is pseudorandom if no efficient adversary can distinguish it from a truly random one. The most popular pseudorandom objects are pseudorandom generators (PRG), functions (PRF), and permutations (PRP). A PRG is a function $prg : \{0,1\}^n \to \{0,1\}^m$ where $m > n$ and no efficient A can distinguish $prg(U_n)$ from U_m, where U_i denotes the uniform distribution over i bit strings. A PRF is a family of functions $F : \{0,1\}^\ell \times \{0,1\}^n \to \{0,1\}^m$, where no efficient

⋆ Supported by DIAMANT, the Dutch national mathematics cluster for discrete interactive and algorithmic algebra and number theory. This work was partially done while the author was a postdoc at the Ecole Normale Supérieure, Paris.

⋆⋆ This work was partially supported by the Zurich Information Security Center. It represents the views of the authors.

A can distinguish $F(U_\ell, .)$ from a uniformly random function. *Weak* PRFs, are defined similarly to PRFs, but where the adversary gets only to see the outputs on random inputs (and not on inputs of his choice). PRGs, PRFs, and PRPs are equivalent, i.e. black-box reducible, to one-way functions [4, 3, 6]. Unfortunately these reductions are quite inefficient, and therefore practical pseudorandom objects are either constructed from scratch (like the AES block-cipher, which is supposed to be a PRP) or from stronger assumptions than OWFs (in particular number theoretic assumptions like Decisional Diffie-Hellman).

Range Extension for PRGs and PRFs. From a PRG $prg : \{0,1\}^n \to \{0,1\}^{2n}$ one can efficiently construct a PRG with a larger range: on input $X \in \{0,1\}^n$ compute $Y_L \| Y_R \leftarrow prg(X)$ and output the $4n$-bit string $Z \leftarrow prg(Y_L)\|prg(Y_R)$. One can now recursively apply prg on input Z in order to get a pseudorandom $8n$-bit string and so on. The security of this construction follows by a simple hybrid argument.

From a PRF $prf : \{0,1\}^\ell \times \{0,1\}^n \to \{0,1\}^n$ we can get a PRF $prf' : \{0,1\}^{\ell t} \times \{0,1\}^n \to \{0,1\}^{nt}$ with larger range as

$$prf'(k_1, \ldots, k_t, x) = prf(k_1, x)\| \ldots \|prf(k_t, x)$$

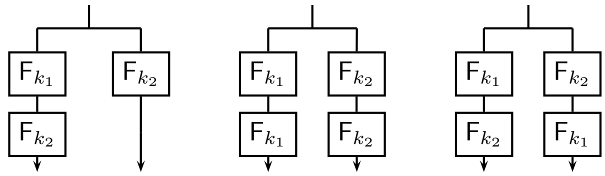
This construction also works for *weak* PRFs, but is not very practical as the number of keys is linear in the expansion factor. Let $bin(i)$ denote the binary representation of i padded with 0's to the length $\lceil \log t \rceil$. The following construction of a $\{0,1\}^\ell \times \{0,1\}^{n-\lceil \log t \rceil} \to \{0,1\}^{nt}$ function

$$prf''(k, x) = prf(k, x\|bin(0))\| \ldots \|prf(k, x\|bin(t-1))$$

just needs a single key, and prf'' is easily seen to be a PRF if prf is. Unfortunately this construction does not work for weak PRFs (just consider a weak PRF where the output does not depend on the last input bit).

Range Extension for weak PRFs. Efficient range extension for weak PRFs has been investigated in [2, 10, 11]. All constructions considered in these papers can be defined by an ordered set α of strings over $[m] \stackrel{\text{def}}{=} \{1, \ldots, m\}$. The input to the construction are m keys $k_1, \ldots, k_m$ for the fixed output length PRF F, and a single input x to F. Each string $s \in \alpha$ now defines how to compute a part of the output, for example $s = \langle 2, 1, 3 \rangle$ corresponds to the value $F_{k_3}(F_{k_1}(F_{k_2}(x)))$, thus the expansion factor is the size of α. We give a formal definition for such constructions, which we call expansions, in Section 3.

Classifying Expansions. Not all expansions are secure in the sense of being a weak PRF whenever the underlying component F is a weak PRF. Before we continue, the reader might take a look at the three expansions given in the figure below, and try to answer the following question: if F is a weak PRF, which of the three length doubling constructions will also be a weak PRF (here k_1, k_2 are two random independent keys).



$C_F^{\{\langle 1,2\rangle,\langle 2\rangle\}}$ $\quad C_F^{\{\langle 1,1\rangle,\langle 2,2\rangle\}}$ $\quad C_F^{\{\langle 1,2\rangle,\langle 2,1\rangle\}}$

In this paper we exactly classify which expansions are secure and which are not (cf. Theorem 1). Interestingly there are three (and not just two) natural classes which come up, we will call them good, bad, and ugly (the three constructions in the figure above are simple examples of a good, a bad, and an ugly expansion). We call expansions whose security can be proven by a black-box reduction[3] *good*. We call an expansion *bad*, if its *in*security can be proven by a black-box reduction. There are also expansion which are neither good nor bad, we call them *ugly*.

More on the Notion of Weak (Pseudo/Quasi)Random Functions. A function is a pseudorandom function if

(i) It cannot be distinguished from a uniformly random function by any efficient distinguisher.
(ii) It can be efficiently computed.

In this paper we also consider the setting where (ii) is not necessarily satisfied, as the function is realized by some oracle, we call such functions simply random functions (RF). If (i) only holds for distinguishers which may query the function on random inputs, we prepend the term "weak" (like weak PRF). Functions which cannot be distinguished from random by any (and not just any efficient) distinguisher making some bounded number of queries are called quasirandom functions (QRF).[4] In particular any function which is a RF relative to a PSPACE oracle is a QRF.[5]

We use the term random*ized* function to denote a function which is not deterministic. This could be an efficient family of functions, where a function is sampled by choosing a random key. It could also be an oracle implementing a function, where the oracle uses randomness. Clearly, any random function is

[3] In such a reduction one constructs an efficient adversary A, such that for every adversary B which breaks the security (as a weak PRF) of the expansion, the adversary A, given black-box access to B, breaks the security of the underlying randomized function (here A and B have only black-box access to the randomized function). Having black-box access to some component means that one only can query it on inputs of ones choice to get some output, but one does not get to see a description (say as a Turing machine) of the component.

[4] In the literature one often refers to such functions a almost k-wise independent functions, where k is a bound on the number of queries.

[5] This is the case as relative to a PSPACE oracle no computational hardness, and thus no pseudorandomness, exists. So if we have a RF relative to a PSPACE oracle, its randomness must be information theoretic, which means it is a QRF.

a randomized function as a deterministic function is easily distinguished from random, the converse is not true in general.

1.1 Related Work

OPTIMAL EXPANSIONS. Efficient range expansion for weak PRFs have been investigated by Damgård and Nielsen [2]. They prove that there are good expansions which achieve an exponential expansion factor of roughly $2^{m/2} - 1$ (using m keys). This has been improved to roughly $3^{m/2} - 1$ in [11] and to $2^m - 1$ in [10]. From our classification it follows (Corollary 1) that $2^m - 1$ is indeed the best possible.[6]
EXPANSIONS IN MINICRYPT. In [13], we show that in Minicrypt, i.e. under the assumption that public-key cryptography does not exist[7], *some* ugly constructions[8] are secure. We do not know if relative to this assumption *all* ugly constructions are secure (in this paper we show that relative to a PSPACE oracle all ugly constructions are secure).

1.2 Applications

Weak PRFs are a strictly weaker primitive than PRFs, and thus requiring that some construction (like AES) is only a weak PRF is less of an assumption than assuming it to be a "regular" PRF.[9] Still, for many applications, weak PRFs are enough. An example is symmetric encryption [12, 2, 10]. The scheme defined by encrypting a message M as $(r, F(k, r) \oplus M)$, where r is sampled uniformly at random, is IND-CPA secure if F is a weak PRF [12]. There is some overhead as the ciphertext is $|r|$ bits longer than the plaintext, but using range extension for weak PRFs, a message of any length can be encrypted [2], and thus the overhead is independent of the message length.

In particular, when using the optimal expansion from [10] in the above scheme one needs $m = \lceil \log_2(|M|/n + 1) \rceil$ shared keys (n being the block-length) for the (fixed output length) weak PRF (those keys can also be computed by expanding a single key, see [10]). This expansion has a "depth" of m, by which we mean that to compute some elements of the output, one will have to invoke the weak PRF up to m times sequentially. Let us, however, stress that to compute all

[6] In [10], it is shown – under the Inverse Decisional Diffie Hellman (IDDH) assumption – that their expansion α of size $2^m - 1$ is optimal for expansions containing strings of length logarithmic in the expansion factor (this corresponds to log-time random access to the output blocks). However, this still leaves open the possibility that a different expansion of larger size exists. In fact, [11] claim to have found a construction with a better expansion, but their proof is flawed (see [10]).

[7] This means relative to an oracle where one-way functions do exist, but key-agreement does not, such an oracle was constructed by Impagliazzo and Rudich [5].

[8] In particular (using notation introduced in the next section) $\alpha = \{\langle 1, 2\rangle, \langle 2, 1\rangle\}$.

[9] Block ciphers like AES are usually not only assumed to be PRFs, but even super pseudorandom-permutations, i.e. indistinguishable from a uniformly random permutation when adaptively queried from both directions.

$2^m - 1$ outputs, one only needs a total number of $2^m - 1$ invocations. This is no contradiction, as if we compute an element with depth c, all the $c-1$ values computed on the way will also be part of the output.

Although a depth of m is only logarithmic in the expansion factor, this might already be too much (say, due to hardware restrictions). We show (Corollary 2) that if we require a smaller depth $c < m$, then the best expansion factor we can get is $\sum_{i=0}^{c} \binom{m}{c} - 1$. Note that for $m = c$, this indeed gives the $2^m - 1$ bound.

2 Basic Definitions

By ${}_LX$ and ${}_RX$ we denote the left and right half of a bit string X of even length, respectively. We denote with $[m]$ the set $\{1, \ldots, m\}$.

An *expansion* α is a set of strings over an alphabet $[m]$ for some $m \in \mathbb{N}$. Consider an expansion $\alpha = \{s_1, \ldots, s_t\}$, each $s_i \in [m]^*$. With $s_i[j]$ we denote the j'th letter of s_i. We denote with $\#\alpha \stackrel{\text{def}}{=} m$ the alphabet size, with $|\alpha| \stackrel{\text{def}}{=} t$ the size, with $\|\alpha\| = \sum_{i=1}^{t} |s_i|$ the total length, and for $1 \leq i \leq m$ with α_i the number of occurrences of the letter i in α. Note that $\sum_{i=1}^{\#\alpha} \alpha_i = \|\alpha\|$.

For an expansion α, $\#\alpha = m$, $|\alpha| = t$, and functions $F_1, \ldots, F_m$, each $\mathcal{X} \to \mathcal{X}$, we define the function

$$C^{\alpha}_{F_1,\ldots,F_m} = \mathcal{X} \to \mathcal{X}^t$$

as follows. On input $X \in \mathcal{X}$, the i'th component ($i \in [t]$) of the output is computed using s_i as

$$F_{s_i[|s_i|]}(F_{s_i[|s_i|-1]}(\ldots F_{s_i[2]}(F_{s_i[1]}(X))\ldots)).$$

We will refer to the above computation as the evaluation of the i'th *chain*. For a randomized function F, we denote with C^{α}_F the function $C^{\alpha}_{F_1,\ldots,F_m}$ where $m = \#\alpha$ and each F_i as an independent instantiation of F.

3 The Good, the Bad & the Ugly

We classify the expansions into three classes depending on the security they guarantee for C^{α}_F.

THE GOOD: α is good if the security of C^{α}_F as a weak random function can be efficiently black-box reduced to the security of F as a weak random function.[10] So whenever F is a weak random function, also C^{α}_F is, and moreover this holds relative to any oracle.

THE BAD: α is bad if there is an efficient construction F' which uses some function F as a black-box, such that the security of F' as a weak random function

[10] The reduction being efficient means that from any adversary A which breaks the security of C^{α}_F, we construct an adversary B where $B^{A,F}$ breaks the security of F, and the size of B (as an oracle circuit) is polynomial in the size of α and the range of F.

can be efficiently black-box reduced to the security of F as a weak random function, but $C^{\alpha}_{F'}$ is not a weak random function.
THE UGLY: α is ugly if it is neither good nor bad.

We now give a simple classification of all expansions into three classes $\mathfrak{G}, \mathfrak{B}$ and $\mathfrak{U}$, which by Theorem 1 below are exactly the good, the bad, and the ugly expansions.

Definition 1 *An expansion $\alpha = \{s_1, \ldots, s_t\}$ is*

- *of type $\mathfrak{B}$ if it does contain a string with two consecutive identical letters or two identical strings, i.e.*

$$\exists i, k \text{ where } s_i[k] = s_i[k+1] \quad \text{or} \quad \exists i, j, 1 \leq i < j \leq m : s_i = s_j.$$

- *of type $\mathfrak{G}$ if it is not of type $\mathfrak{B}$ and whenever a letter c appears before a letter d in some $s \in \alpha$, then d does not appear before c in any string $s' \in \alpha$, i.e.*[11]

$$\forall s, s' \in \alpha, i, j, i', j' : s[i] = s'[i'] \wedge s[j] = s'[j'] \wedge i < j \Rightarrow i' < j'.$$

- *of type $\mathfrak{U}$ if it is not of type $\mathfrak{G}$ or $\mathfrak{B}$.*

Theorem 1 (main) .

(i) An expansion is **good** *if and only if it is of type $\mathfrak{G}$.*
(ii) An expansion is **bad** *if and only if it is of type $\mathfrak{B}$.*
(iii) An expansion is **ugly** *if and only if it is of type $\mathfrak{U}$.*

That $\mathfrak{G}$ expansions are good and $\mathfrak{B}$ expansions are bad follows by rather simple black-box reductions (Lemmata 1 and 2), the "only if" part is much harder. In order to show that the $\mathfrak{U}$ expansions are ugly, one has to come up with an oracle implementing a random function, such that relative to this oracle the expansion is not secure (thus it is not good), and another oracle relative to which it is secure (thus it is not bad). For the latter oracle we use a PSPACE oracle, as we show (Theorem 2) that for QRFs (recall that any RF is a QRF relative to a PSPACE oracle) any expansion which is not of type $\mathfrak{B}$, is secure. The following table summarizes the proof of the theorem.

<table>
<tr><th>$\mathfrak{G}$</th><th>$\mathfrak{U}$</th><th>$\mathfrak{B}$</th></tr>
<tr><td>good by Lemma 2 (and [10])</td><td colspan="2">not good by Lemma 3</td></tr>
<tr><td colspan="2">not bad by Theorem 2</td><td>bad by Lemma 1</td></tr>
</table>

So Theorem 1.(i) follows from Lemma 2 and 3, Theorem 1.(ii) follows from Theorem 2 and Lemma 1, and Theorem 1.(iii) follows from (i) and (ii).
COROLLARIES. For every m, [10] construct a good expansion of size $2^m - 1$ using m keys: let α contain all $2^m - 1$ distinct s (of length at least 1) over $[m]$ where $s[i-1] < s[i]$ for all $2 \leq i \leq |s|$. From our classification it follows that this is best possible, and moreover, this expansion is the unique good expansion of size $2^m - 1$ (up to relabellings of the keys).

[11] Note that we do not require $c \neq d$, so this condition implies that no letter appears more than once in any string.

Corollary 1 *For any m and α with alphabet size $\#\alpha = m$, if α is good then*

$$|\alpha| \leq 2^m - 1,$$

and this is tight for $\alpha = \{s \in [m]^* \ ; \ s[1] < s[2] < \ldots < s[|s|]\}$.

For some $c < m$, consider the expansion we get by removing all $s \in \alpha$ of length more than c from the optimal expansion just described. This expansion is still good, and it is not hard to show that it is the best good expansion of depth c using m keys.

Corollary 2 *For any $m, c \leq m$, and α with alphabet size $\#\alpha = m$, if α is good then*

$$|\alpha| \leq \sum_{i=0}^{c} \binom{m}{i} - 1,$$

and this is tight for $\alpha = \{s \in [m]^* \ ; \ s[1] < s[2] < \ldots < s[|s|], |s| \leq c\}$.

Note that Corollary 1 is just a special case of Corollary 2 for the case $c = m$.

4 The Bad Expansions are Exactly $\mathfrak{B}$

To prove that expansions outside of $\mathfrak{B}$ are not bad, we use the random systems framework of Maurer [7]. Here we only give a rather informal and restricted exposition of the framework, in particular we only consider known-plaintext attacks (KPA), as this is the only attack relevant for this paper.

Notation. We use capital calligraphic letters like $\mathcal{X}$ to denote sets, capital letters like X to denote random variables, and small letters like x denote concrete values. To save on notation we write X^i for $X_1, X_2, \ldots, X_i$.

Random Systems. Informally, a *random system* is a system which takes inputs $X_1, X_2, \ldots$ and generates, for each new input X_i, an output Y_i which depends probabilistically on the inputs and outputs seen so far. We define random systems in terms of the distribution of the outputs Y_i conditioned on $X^i Y^{i-1}$, more formally: An $(\mathcal{X}, \mathcal{Y})$*-random system* $\mathbf{F}$ is a sequence of conditional probability distributions $\mathsf{P}^{\mathbf{F}}_{Y_i|X^iY^{i-1}}$ for $i \geq 1$. Here we denote by $\mathsf{P}^{\mathbf{F}}_{Y_i|X^iY^{i-1}}(y_i, x^i, y^{i-1})$ the probability that $\mathbf{F}$ will output $y_i \in \mathcal{Y}$ on input $x_i \in \mathcal{X}$ conditioned on the fact that $\mathbf{F}$ did output $y_j \in \mathcal{Y}$ on input $x_j \in \mathcal{X}$ for $j = 1, \ldots, i-1$.

Uniformly random functions (URFs) are random systems which will be important in this paper, throughout $\mathbf{R}_{n,m}$ will denote a URF $\{0,1\}^n \to \{0,1\}^m$.

Conditions for Random Systems. With $\mathbf{F}^{\mathcal{A}}$ we denote the random system $\mathbf{F}$, but which additionally defines an internal binary random variable after each query (called a condition). Let $A_i \in \{0,1\}$ denote the condition after the i'th query. We set $A_0 = 0$ and require the condition to be monotone which means that $A_i = 1 \Rightarrow A_{i+1} = 1$ (i.e. when the condition failed, it will never hold again). Let $\overline{a}_i$ denote the event $A_i = 1$, then with $\nu^{\mathsf{KPA}}(\mathbf{F}^{\mathcal{A}}, \overline{a}_k)$ we denote the probability of the event $\overline{a}_k$ occurring when $\mathbf{F}^{\mathcal{A}}$ is queried on random inputs, i.e.

$$\begin{aligned}\nu^{\mathsf{KPA}}(\mathbf{F}^{\mathcal{A}}, \overline{a}_k) &\stackrel{\text{def}}{=} \sum_{x^k \in \mathcal{X}^k} \mathsf{Pr}[X^k = x^k] \cdot \mathsf{Pr}[\overline{a}_k \text{ holds in } \mathbf{F}^{\mathcal{A}}(x^k)] \\ &= \frac{1}{|\mathcal{X}|^k} \sum_{x^k \in \mathcal{X}^k} \mathsf{Pr}[\overline{a}_k \text{ holds in } \mathbf{F}^{\mathcal{A}}(x^k)].\end{aligned}$$

Indistinguishability. For $(\mathcal{X}, \mathcal{Y})$-random systems $\mathbf{F}$ and $\mathbf{G}$, we denote with $\Delta_k^{\mathsf{KPA}}(\mathbf{F}, \mathbf{G})$ the distinguishing advantage of any unbounded distinguisher in a k query known-plaintext attack. This advantage is simply the statistical distance, i.e. with X^k being uniformly random over $\mathcal{X}^k$

$$\begin{aligned}\Delta_k^{\mathsf{KPA}}(\mathbf{F}, \mathbf{G}) &\stackrel{\text{def}}{=} \frac{1}{2} \sum_{x^k \in \mathcal{X}^k, y^k \in \mathcal{Y}^k} \mathsf{Pr}[X^k = x^k] \cdot \left|\mathsf{Pr}[\mathbf{F}(x^k) = y^k] - \mathsf{Pr}[\mathbf{G}(x^k) = y^k]\right| \\ &= \frac{1}{2 \cdot |\mathcal{X}|^k} \sum_{x^k \in \mathcal{X}^k, y^k \in \mathcal{Y}^k} \left|\mathsf{Pr}[\mathbf{F}(x^k) = y^k] - \mathsf{Pr}[\mathbf{G}(x^k) = y^k]\right|.\end{aligned}$$

$\mathbf{F}^{\mathcal{A}} \stackrel{\circ}{=} \mathbf{G}^{\mathcal{B}}$ denotes that $\mathbf{F}^{\mathcal{A}}$ is equivalent to $\mathbf{G}^{\mathcal{B}}$ while the respective condition holds:

$$\mathbf{F}^{\mathcal{A}} \stackrel{\circ}{=} \mathbf{G}^{\mathcal{B}} \iff \forall x^i, y^i : \mathsf{Pr}^{\mathbf{F}^{\mathcal{A}}}_{a_i \wedge Y^i | X^i}(y^i, x^i) = \mathsf{Pr}^{\mathbf{G}^{\mathcal{B}}}_{b_i \wedge Y^i | X^i}(y^i, x^i).$$

We say that $\mathbf{F}^{\mathcal{A}}$ is dominated by $\mathbf{G}$, which is denoted by $\mathbf{F}^{\mathcal{A}} \preceq \mathbf{G}$, if on any input x^i and for any possible output y^i the probability that $\mathbf{F}^{\mathcal{A}}(x^i)$ output y^i *and* the condition $\mathcal{A}$ holds, is at most the probability that $\mathbf{G}(x^i) = y^i$.

$$\mathbf{F}^{\mathcal{A}} \preceq \mathbf{G} \iff \forall x^i, y^i : \mathsf{Pr}^{\mathbf{F}^{\mathcal{A}}}_{a_i \wedge Y^i | X^i}(y^i, x^i) \leq \mathsf{Pr}^{\mathbf{G}}_{Y^i | X^i}(y^i, x^i)$$

or equivalently $\forall x^i, y^i : \mathsf{Pr}^{\mathbf{F}^{\mathcal{A}}}_{a_i \wedge Y_i | X^i Y^{i-1} a_{i-1}}(y_i, x^i, y^{i-1}) \leq \mathsf{Pr}^{\mathbf{G}}_{Y_i | X^i Y^{i-1}}(y_i, x^i, y^{i-1})$

Note that $\mathbf{F}^{\mathcal{A}} \stackrel{\circ}{=} \mathbf{G}^{\mathcal{B}}$ implies $\mathbf{F}^{\mathcal{A}} \preceq \mathbf{G}$ and $\mathbf{G}^{\mathcal{B}} \preceq \mathbf{F}$. The following are the two main propositions of the framework (restricted to the case of KPA attacks).

Proposition 1 *If* $\mathbf{F}^{\mathcal{A}} \preceq \mathbf{G}$ *then* $\Delta_q^{\mathsf{KPA}}(\mathbf{F}, \mathbf{G}) \leq \nu^{\mathsf{KPA}}(\mathbf{F}^{\mathcal{A}}, \overline{a}_q)$.

Proposition 2 *For any random systems* $\mathbf{F}$ *and* $\mathbf{G}$*, there exist conditions* $\mathcal{A}$ *and* $\mathcal{B}$ *such that*

$$\mathbf{F}^{\mathcal{A}} \stackrel{\circ}{=} \mathbf{G}^{\mathcal{B}} \quad \textit{and} \qquad \Delta_q^{\mathsf{KPA}}(\mathbf{F}, \mathbf{G}) = \nu^{\mathsf{KPA}}(\mathbf{F}^{\mathcal{A}}, \overline{a}_q) = \nu^{\mathsf{KPA}}(\mathbf{G}^{\mathcal{B}}, \overline{b}_q).$$

Proposition 1 is quite easy to prove and appeared in the original paper [7]. Proposition 2 is from (the yet unpublished) [9], a weaker version of this proposition appeared in [8].

4.1 Expansions not in $\mathfrak{B}$ are not Bad

By the following theorem, $C^{\alpha}_{\mathbf{F}_1,\ldots,\mathbf{F}_m}$ is a weak quasirandom function whenever the $\mathbf{F}_i$'s are weak QRFs and α is not in $\mathfrak{B}$. The distance of the output of $C^{\alpha}_{\mathbf{F}_1,\ldots,\mathbf{F}_m}$ on q random queries can be upper bounded by the sum of the distances of the $\mathbf{F}_i$'s on $q\alpha_i$ random queries (recall that α_i is the number of invocations of $\mathbf{F}_i$ on an invocation of $C^{\alpha}_{\mathbf{F}_1,\ldots,\mathbf{F}_m}$), plus some term which is small unless $q \cdot \|\alpha\|$ is in the order of $2^{n/2}$.

Theorem 2 *For any expansion $\alpha = \{s_1, \ldots, s_t\}$ which is not of type $\mathfrak{B}$, any randomized functions $\mathbf{F}_i : \{0,1\}^n \to \{0,1\}^n, 1 \le i \le \#\alpha := m$, and every $q \in \mathbb{N}$:*

$$\Delta^{\mathsf{KPA}}_q(C^{\alpha}_{\mathbf{F}_1,\ldots,\mathbf{F}_m}, \mathbf{R}_{n,n\cdot t}) \le \sum_{i=1}^{m} \Delta^{\mathsf{KPA}}_{q\cdot\alpha_i}(\mathbf{F}_i, \mathbf{R}_{n,n}) + \frac{q^2\|\alpha\|^2}{2^n}.$$

Proof. To save on notation let

$$\mathbf{I} \stackrel{\text{def}}{=} C^{\alpha}_{\mathbf{R}_1,\ldots,\mathbf{R}_m} \quad \text{and} \quad \mathbf{C} \stackrel{\text{def}}{=} C^{\alpha}_{\mathbf{F}_1,\ldots,\mathbf{F}_m},$$

where the $\mathbf{R}_i$'s are independent instantiations of $\mathbf{R}_{n,n}$. By the triangle inequality

$$\Delta^{\mathsf{KPA}}_q(\mathbf{C}, \mathbf{R}_{n,n\cdot t}) \le \Delta^{\mathsf{KPA}}_q(\mathbf{C}, \mathbf{I}) + \Delta^{\mathsf{KPA}}_q(\mathbf{I}, \mathbf{R}_{n,n\cdot t}). \tag{1}$$

The theorem follows from the two claims below, which bound the two terms on the right hand side of (1) respectively.

Claim 1

$$\Delta^{\mathsf{KPA}}_q(\mathbf{I}, \mathbf{R}_{n,n\cdot t}) \le \frac{q^2\|\alpha\|^2}{2^{n+1}}$$

Proof (of Claim 1). We define a condition $\mathcal{D}$ on $\mathbf{I}$ as follows: the condition is satisfied as long as for all $i, 1 \le i \le m$, there was no nontrivial collision on the inputs to the component $\mathbf{R}_i$. Here the trivial collisions are the "unavoidable" collisions which occur when two chains have the same prefix. For example in $C^{\{\langle 1,2,3,4\rangle,\langle 1,2,4,3\rangle\}}_{\mathbf{R}_1,\ldots,\mathbf{R}_4}$ the inputs to $\mathbf{R}_1$ in the two different chains are always identical, the same holds for the inputs to $\mathbf{R}_2$ (but not for $\mathbf{R}_3$ or $\mathbf{R}_4$). We now show (using that α is not of type $\mathfrak{B}$) that this condition satisfies $\mathbf{I}^{\mathcal{D}} \preceq \mathbf{R}_{n,n\cdot t}$, i.e.

$$\forall x^i, y^i : \mathsf{Pr}^{\mathbf{I}}_{Y_i \wedge d_i | X^i Y^{i-1} \wedge d_{i-1}}(y_i, x^i, y^{i-1}) \le \mathsf{Pr}^{\mathbf{R}_{n,n\cdot t}}_{Y_i | X^i Y^{i-1}}(y_i, x^i, y^{i-1}) = 2^{-n\cdot t}. \tag{2}$$

Assume we invoke $\mathbf{I}$ on the i'th query $x_i \in \{0,1\}^n$, and that d_{i-1}, i.e. the condition was satisfied after the $(i-1)$'th query. We evaluate the $t = |\alpha|$ chains of $\mathbf{I} = C^{\alpha}_{\mathbf{R}_1,\ldots,\mathbf{R}_m}$ one by one and assume that the s_i's are ordered by increasing length.[12] For any j, when computing the j'th chain we stop just before we invoke the last component $\mathbf{R}_{s_j[|s_j|]}$. Now, if the input to this component is fresh (i.e. $\mathbf{R}_{s_j[|s_j|]}$ was never invoked on that input before), then every output has

[12] This will only be important if one chain is the prefix on another.

probability exactly 2^{-n}. The probability that we get fresh inputs (to the last components) on all t chains and the outputs will be consistent with y_i in all chains is thus at most $2^{-n\cdot t}$. On the other hand, if at some point we have an input which is not fresh, then there has been a collision. Now, as no two chains are equivalent (as α is not in $\mathfrak{B}$) and we process them by increasing length, it follows that this collision was a nontrivial one, and thus $\overline{d}_i$. This concludes the proof of (2). The first step of

$$\Delta_q^{\mathsf{KPA}}(\mathbf{I}, \mathbf{R}_{n,n\cdot t}) \leq \nu^{\mathsf{KPA}}(\mathbf{I}^{\mathcal{D}}, \overline{d}_q) \leq \frac{\sum_{i=1}^{m}(q\cdot\alpha_i)^2}{2^{n+1}} \leq \frac{q^2\|\alpha\|^2}{2^{n+1}} \tag{3}$$

follows by Proposition 1 using $\mathbf{I}^{\mathcal{D}} \preceq \mathbf{R}_{n,n\cdot t}$. The second step follows by the birthday bound: the fact that $\overline{d}_q$ means that at some point for some $i \in [m]$ the uniformly random output of $\mathbf{R}_i$ did collide with some "old" input to $\mathbf{R}_i$. As $\mathbf{R}_i$ is invoked $q \cdot \alpha_i$ times, the probability that there will be a collision is at most $(q\cdot\alpha_i)^2/2^{n+1}$. To get the probability that there will be a collision for any $\mathbf{R}_i, i \in [m]$, we take the union bound. △

Claim 2

$$\Delta_q^{\mathsf{KPA}}(\mathbf{C}, \mathbf{I}) \leq \frac{q^2\|\alpha\|^2}{2^{n+1}} + \sum_{i=1}^{m} \Delta_{q\cdot\alpha_i}^{\mathsf{KPA}}(\mathbf{F}_i, \mathbf{R}_{n,n})$$

Proof (of Claim 2). For every $i, 1 \leq i \leq m$, let $\mathcal{A}^i$ and $\mathcal{B}^i$ be conditions such that (the existence of such conditions follows by Proposition 2)

$$\mathbf{F}_i^{\mathcal{A}^i} \mathrel{\mathring{=}} \mathbf{R}_{n,n}^{\mathcal{B}^i} \quad \text{and} \quad \Delta_q^{\mathsf{KPA}}(\mathbf{F}_i, \mathbf{R}_{n,n}) = \nu^{\mathsf{KPA}}(\mathbf{R}_{n,n}^{\mathcal{B}^i}, \overline{b}_q^i) = \nu^{\mathsf{KPA}}(\mathbf{F}_i^{\mathcal{A}^i}, \overline{a}_q^i). \tag{4}$$

To save on notation let $\mathcal{B} \stackrel{\text{def}}{=} \mathcal{B}^1 \wedge \cdots \wedge \mathcal{B}^m$, $\mathcal{A} \stackrel{\text{def}}{=} \mathcal{A}^1 \wedge \cdots \wedge \mathcal{A}^m$ and $q_i = q \cdot \alpha_i$. . As for all $\mathbf{F}_i^{\mathcal{A}^i} \equiv \mathbf{R}_{n,n}^{\mathcal{B}^i}$ for $1 \leq i \leq m$, it follows that

$$\mathbf{C}^{\mathcal{A}} \mathrel{\mathring{=}} \mathbf{I}^{\mathcal{B}}. \tag{5}$$

Let $b \Rightarrow_q d$ denote the event defined on $\mathbf{I}^{\mathcal{B}\wedge\mathcal{D}}$ which holds if at any timepoint up to after the q'th query, either $\mathcal{D}$ holds or $\mathcal{B}$ does not hold (or equivalently, either $\mathcal{D}$ does not fail, or it only fails after $\mathcal{B}$ fails). The first step below follows by Proposition 1 using (5). The last step follows by the union bound and observing that $\overline{b}_q \vee \overline{d}_q$ holds iff $\overline{d}_q \vee [\overline{b}_q \wedge [b \Rightarrow_q d]]$.

$$\begin{aligned}
\Delta_q^{\mathsf{KPA}}(\mathbf{C}, \mathbf{I}) &\leq \nu^{\mathsf{KPA}}(\mathbf{I}^{\mathcal{B}}, \overline{b}_q) \\
&\leq \nu^{\mathsf{KPA}}(\mathbf{I}^{\mathcal{B}\wedge\mathcal{D}}, \overline{b}_q \vee \overline{d}_q) \\
&\leq \nu^{\mathsf{KPA}}(\mathbf{I}^{\mathcal{D}}, \overline{d}_q) + \nu^{\mathsf{KPA}}(\mathbf{I}^{\mathcal{B}\wedge\mathcal{D}}, \overline{b}_q \wedge [b \Rightarrow_q d]).
\end{aligned} \tag{6}$$

We can bound the first term of (6) using (3) as $\nu^{\mathsf{KPA}}(\mathbf{I}^{\mathcal{D}}, \overline{d}_q) \leq q^2\|\alpha\|^2/2^{n+1}$. We now bound the second term of (6), using $\overline{b}_q \iff \overline{b}_{q_1}^1 \vee \ldots \vee \overline{b}_{q_m}^m$ in the first inequality, and the union bound in the second step:

$$\begin{aligned}
\nu^{\mathsf{KPA}}(\mathbf{I}^{\mathcal{B}\wedge\mathcal{D}}, \overline{b}_q \wedge [b \Rightarrow_q d]) &= \nu^{\mathsf{KPA}}(\mathbf{I}^{\mathcal{B}\wedge\mathcal{D}}, [\overline{b}_{q_1}^1 \vee \ldots \vee \overline{b}_{q_m}^m] \wedge [b \Rightarrow_q d]) \\
&\leq \sum_{i=1}^{m} \nu^{\mathsf{KPA}}(\mathbf{I}^{\mathcal{B}\wedge\mathcal{D}}, \overline{b}_{q_i}^i \wedge [b \Rightarrow_q d]).
\end{aligned}$$

The term $\nu^{\mathsf{KPA}}(\mathbf{I}^{\mathcal{B}\wedge\mathcal{D}}, \overline{b}^i_{q_i} \wedge [b \Rightarrow_q d])$ is the probability that when querying $\mathbf{I}$ on q random inputs, the condition $\mathcal{B}^i$ defined on $\mathbf{R}^i$ will fail, and it will do so before $\mathcal{D}$ fails. Now, as long as $\mathcal{D}$ holds, $\mathbf{R}^i$ is invoked on uniformly random inputs: the inputs are either part of the global input (which is random in a KPA attack), or it is the output of some URF $\mathbf{R}^j$. It is important to note that in this case always $j \neq i$,[13] so $\mathbf{R}_i$ is never invoked on its own output, which guarantees that (while $\mathcal{D}$ holds) the inputs to $\mathbf{R}_i$ are not only random, but also independent of $\mathbf{R}_i$. So the probability that $\overline{b}^i_{q_i} \wedge [b \Rightarrow_q d]$ in $\mathbf{I}$ can be upper bounded by the probability that $\overline{b}_{q_i}$ in $\mathbf{R}_i^{\mathcal{B}^i}$ in a normal KPA attack, i.e.

$$\nu^{\mathsf{KPA}}(\mathbf{I}^{\mathcal{B}\wedge\mathcal{D}}, \overline{b}^i_{q_i} \wedge [b \Rightarrow_q d]) \leq \nu^{\mathsf{KPA}}(\mathbf{R}_i^{\mathcal{B}^i}, \overline{b}^i_{q_i}) = \Delta^{\mathsf{KPA}}_{q_i}(\mathbf{F}_i, \mathbf{R}_{n,n}),$$

where the second step follows by (4). $\triangle$

$\square$

4.2 Type $\mathfrak{B}$ Expansions are Bad

Lemma 1 *Expansions of type* $\mathfrak{B}$ *are bad.*

Proof. To prove the lemma we show a black-box construction of a random function G^P based on a permutation P such that:

(i) The security of G^P as a weak random function can be black-box reduced to the security of P as a random permutation.

(ii) For every bad expansion α, $C^{\alpha}_{G^P}$ is not a weak random function.

Note that we assume that G has access to an oracle which implements a random permutation,[14] and not just a weak RF as required by the lemma. We can do this as random permutations and weak random functions are equivalent, in the sense that both can be constructed from (and imply the existence of) functions which are hard to invert on random inputs[15] via a black-box reduction [3, 4, 6].

To simplify the argument, in the proof we assume that the random permutation $P : \{0,1\}^{2n} \to \{0,1\}^{2n}$ is a uniformly random permutation (URP). As by definition a random permutation is indistinguishable from a URP, this does not change the statement. $G^P(X) : \{0,1\}^{2n} \to \{0,1\}^{2n}$ is defined as follows, first let $Y = {}_LY \| {}_RY \leftarrow P^{-1}(X)$, now

$$G^P(X) = \begin{cases} 0^{2n} & \text{if } {}_LY = 0^n \text{ or } X = 0^{2n} \\ P(0^n \| {}_RX) & \text{otherwise.} \end{cases}$$

We first prove statement (i), namely that G^P is a weak random function (in fact, as we assume that P is a URP, we can even show that G^P is a weak quasirandom function).

---

[13] This is because α is not of type $\mathfrak{B}$ and thus no $s \in \alpha$ has two identical consecutive letters.

[14] A random permutation is a random bijective function (with same range and domain).

[15] Such functions are called one-way functions in the special (and most interesting) case where the function can be efficiently computed in forward direction.

Claim 3

$$\varDelta_q^{\mathsf{KPA}}(G^P, \mathbf{R}_{2n,2n}) \le \frac{3q^2}{2^n}$$

Proof (of Claim 3). By the triangle inequality

$$\varDelta_q^{\mathsf{KPA}}(G^P, \mathbf{R}_{2n,2n}) \le \varDelta_q^{\mathsf{KPA}}(G^P, P) + \varDelta_q^{\mathsf{KPA}}(P, \mathbf{R}_{2n,2n}) \tag{7}$$

G^P is equivalent to P unless we happen to query G^P on input 0^{2n} or an input X where the first n bits of $P^{-1}(X)$ are 0^n. For a random X, this happens with probability $\le 2^{-2n} + 2^{-n}$. By the union bound

$$\varDelta_q^{\mathsf{KPA}}(G^P, P) \le \frac{2q}{2^n}. \tag{8}$$

By the so called PRF/PRP switching lemma (see e.g. [1]) we have

$$\varDelta_q^{\mathsf{KPA}}(P, \mathbf{R}_{2n,2n}) \le \frac{q^2}{2^{2n+1}}. \tag{9}$$

The claim follows from (7), (8), and (9). $\triangle$

Now we prove statement (ii), i.e. that for every bad expansion α, $C^\alpha_{G^P}$ is not a weak random function. Recall that $\alpha = \{s_1, \ldots, s_t\}$ is bad if either $s_i = s_j$ for some $i \ne j$ or there is a s_i with two consecutive identical letters, i.e. for some $j : s_i[j] = s_i[j+1]$. When $s_i = s_j$ then also the i'th and j'th tuple in the output of $C^\alpha_{G^P}(X)$ are identical for any X, and thus easy to distinguish from random.

We now consider the other case. Let α be any expansion where for some element $s \in \alpha$ we have for some j that $s[j] = s[j+1]$. As we prove a negative statement, we can without loss of generality assume that s is the only element in α. We claim that $C^\alpha_{G^P}$ is not random as for any $m = \#\alpha$ instantiations $G_1^P, \ldots, G_m^P$ of G^P and any X we have $C^\alpha_{G_1^P,\ldots,G_m^P}(X) = 0^{2n}$. To see this let $X_0 = X$ and for $i = 1, \ldots, |s| : X_i = G^P_{s[i]}(X_{i-1})$, then $C^\alpha_{G_1^P,\ldots,G_m^P}(X) = X_{|s|}$. Now by the definition of G^P, for any Z and $i \in [m]$, $G_i^P(G_i^P(Z)) = 0^{2n}$, in particular $X_{j+1} = G^P_{s[j+1]}(G^P_{s[j]}(X_{j-1})) = 0^{2n}$, and as $G_i^P(0^{2n}) = 0^{2n}$ for any i we get $X_\ell = 0^{2n}$ for all $\ell \ge j$. For concreteness let us illustrate this computation on the example $\alpha = \{\langle 1, 2, 2, 3\rangle\}$. Here P_2 is the P used by G_2^P, and $X_3 = 0^{2n}$ holds as ${}_LP_2^{-1}(X_2) = 0^n$.

$$X = X_0 \stackrel{G_1^P}{\to} X_1 \stackrel{G_2^P}{\to} X_2 = P_2(0^n \|_R X_1) \stackrel{G_2^P}{\to} X_3 = 0^{2n} \stackrel{G_3^P}{\to} X_4 = 0^{2n}.$$

$\square$

5 The Good Expansions are Exactly $\mathfrak{G}$

5.1 Type $\mathfrak{G}$ Expansions are Good

The following lemma is from [10], for completeness we give a proof in the appendix.

Lemma 2 *Expansions of type $\mathfrak{G}$ are good.*

5.2 Expansions not in $\mathfrak{G}$ are Not Good

Lemma 3 *Expansions not in $\mathfrak{G}$ are not good.*

By Lemma 1 expansions of type $\mathfrak{B}$ are not good. It remains to show that there exists an oracle $\mathcal{O}$ relative to which a weak random function $F^{\mathcal{O}}$ exists, but where for any expansion $\alpha = \{s_1, \dots, s_t\}$ of type $\mathfrak{U}$ the function $C^{\alpha}_{F^{\mathcal{O}}}$ is not weakly random. The oracle $\mathcal{O}$ we construct will consist of two parts, which can be accessed by setting the first part of the input to either "eval" or "break".

Let n be our security parameter (think of $\mathcal{O}$ as a family of oracles, one for each $n \in \mathbb{N}$). Let $m = \max_i |s_i|$ and $\ell = n^3 m$. Let $F^{\mathcal{O}} : \{0,1\}^n \times \{0,1\}^\ell \to \{0,1\}^\ell$ ($\mathcal{O}$ still to be defined)

$$F^{\mathcal{O}}(k, x) = \mathcal{O}(eval, k, x).$$

We will often write the key as a subscript $F^{\mathcal{O}}_k(.) = F^{\mathcal{O}}(k, .)$. The all zero string 0^n is excluded from the valid keys as later 0^n will have the special meaning of "no key".

We now define the "eval" part of the oracle. Initially, $2^{mn} - 1$ disjoint subsets of $\{0,1\}^\ell$, each of size 2^n, are sampled. Each such set corresponds to an ordered sequence of at most m (and at least one) keys, the set corresponding to the keys $k_1, \dots, k_{m'}, m' \leq m$ is denoted $S_{0^{(m-m')n}\|k_1\|k_2\|\dots\|k_{m'}}$. With S_0 we denote the elements from $\{0,1\}^\ell$ which are in no set, i.e. $S_0 = \{0,1\}^\ell \setminus \bigcup_{x \in \{0,1\}^{mn} \setminus 0^{mn}} S_x$ (we have $|S_0| = 2^{mn^3} - 2^{mn^2} + 2^n$, i.e. all but a 2^{-n} fraction of elements from $\{0,1\}^\ell$ are in S_0).

Now for any key k, $\mathcal{O}(eval, k, .)$ maps the elements from S_0 at random to $S_{0^{(m-1)n}\|k}$. As for the inputs not in S_0, for any key k and keys $k_1, \dots, k_{m'}$, $\mathcal{O}(eval, k, .)$ is defined as a random bijective function from $S_{0^{(m-m')n}\|k_1\|k_2\|\dots\|k_{m'}}$ to $S_{0^{(m-m'-1)n}\|k_1\|k_2\|\dots\|k_{m'}\|k}$ (where for $m = m'$, we shift the leftmost key out, i.e. we map $S_{k_1\|k_2\|\dots\|k_m}$ to $S_{k_2\|\dots\|k_m\|k}$). Note that this means that for any $t \leq m$ and $x \in S_0$ a value computed as $y = F^{\mathcal{O}}_{k_t}(F^{\mathcal{O}}_{k_{t-1}} \dots F^{\mathcal{O}}_{k_1}(x))$ is in $S_{0^{(m-t)n}\|k_1\|\dots\|k_t}$. For a computationally bounded distinguisher, this y will look random, but the computationally unbounded "break" part of the oracle (defined below) can learn the keys $k_1, \dots, k_t$ used.

We now define the "break" part of the oracle. $\mathcal{O}(break, .)$ is a $(\{0,1\}^\ell)^2 \to \{0,1\}$ function and defined as follows. For any $Y_1 \in S_{0^{m-m'n}\|a_1\|\dots\|a_{m'}}$ and $Y_2 \in S_{0^{m-m''n}\|b_1\|\dots\|b_{m''}}$, we define $\mathcal{O}(break, \{Y_1, Y_2\}) = 1$ if there are i, i', j, j' where

$$a_i = b_{i'} \qquad a_j = b_{j'} \qquad i < j \qquad i' > j',.$$

and $\mathcal{O}(break, \{Y_1, Y_2\}) = 0$ otherwise. In particular, $\mathcal{O}(break, \{Y_1, Y_2\})$ outputs 0 if either $Y_1 \in S_0$ or $Y_2 \in S_0$.

Claim 4 *For any α of type $\mathfrak{U}$, $C^{\alpha}_{F^{\mathcal{O}}}$ is not a weak random function (relative to the oracle $\mathcal{O}$).*

Proof (of Claim 4). Let $X \in \{0,1\}^\ell$ be a random input, and $Y = C^{\alpha}_{F^{\mathcal{O}}}(X)$. Let Y_i denote the i'th ℓ-bit block of $Y \stackrel{\text{def}}{=} Y_1\|\dots\|Y_t$. As $\alpha = \{s_1, \dots, s_t\}$ is of type

$\mathfrak{U}$, there are i, j and letters c, d such that $s_i = *c * d*$, and $s_j = *d * c*$, where each $*$ is a wildcard, i.e. stands for "any" string. As $Y_i \in S_{*s_i} = S_{*c*d*}$ and $Y_j \in S_{*s_j} = S_{*d*c*}$, it follows that $\mathcal{O}(break, \{Y_i, Y_j\}) = 1$. On the other hand, for a random $Y' = Y'_1 \| \ldots \| Y'_t$ the probability that $\mathcal{O}(break, \{Y'_i, Y'_j\}) = 1$ is very small: we get a rough (but already exponentially small) upper bound on this probability by using that the oracle will output 0 whenever Y'_i is in S_0, i.e.

$$\mathsf{Pr}[\mathcal{O}(break, \{Y'_i, Y'_j\}) = 1] \leq \mathsf{Pr}[Y'_i \notin S_0] < 1/2^n.$$

Thus we can distinguish the output Y of $C^{\alpha}_{F^{\mathcal{O}}}$ from random Y' with advantage almost 1. $\triangle$

Claim 5 *$F^{\mathcal{O}}$ is a weak random function relative to $\mathcal{O}$.*

Proof (sketch of Claim 5). Clearly the function $F^{\mathcal{O}}$ is a random function relative to the oracle $\mathcal{O}(eval, ., .)$ alone (i.e. where there is no $\mathcal{O}(break, .)$).

Now we will show that adding the oracle $\mathcal{O}(break, .)$ will not break the security of $F^{\mathcal{O}}$ as a weak random function (but note that it trivially does break the security of $F^{\mathcal{O}}$ as a (non weak) random function[16]), as if an adversary $A^{\mathcal{O}}$ can distinguish $F^{\mathcal{O}}(k, .)$ from random on random inputs and access to the oracle $\mathcal{O}(break, .)$, then there is an adversary $B^{\mathcal{O},A}$ which uses A as "black-box" and which can distinguish $F^{\mathcal{O}}$ without querying the oracle $\mathcal{O}(break, .)$ at all (this is a contradiction as $F^{\mathcal{O}}$ is a random function relative to $\mathcal{O}(eval, .)$ alone). The adversary $B^{\mathcal{O},A}$ on input $Q = \{(X_1, Y_1), \ldots, (X_q, Y_q)\}$ (where the X_i's are random and the Y_i's are either random or $Y_i = F^{\mathcal{O}}(k, X_i)$ for a random k) runs A on input Q. Here A has no access to the oracle $\mathcal{O}$, but B controls A's oracle gates. B initializes an empty set T, this T will be used to remember the queries made by A. Whenever A requests the output of $\mathcal{O}(eval, .)$ on some input k, x, $B^{\mathcal{O},A}$ correctly answers with $y = \mathcal{O}(eval, k, x)$ and adds (k, x, y) to T. When A requests the output of $\mathcal{O}(break, .)$ on an input $\{Y, Y'\}$, B guesses the answer itself, and we will show that $B^{\mathcal{O},A}$ can indeed guess $\mathcal{O}(break, \{Y, Y'\})$ correctly with high probability. We now describe how $B^{\mathcal{O},A}$ guesses $\mathcal{O}(break, \{Y, Y'\})$.

$B^{\mathcal{O},A}$ first looks up the sequence $(k_1, x_1, y_1), \ldots, (k_t, x_t, y_t) \in T$ where $Y = y_t$ and for $i = 2, \ldots, t : x_i = y_{i-1}$ (where t is maximal, i.e. $(k, x, x_1) \notin T$ for any k, x). Similarly it looks up the sequence $(k'_1, x'_1, y'_1), \ldots, (k'_{t'}, x'_{t'}, y'_{t'})$ where $y_{t'} = Y'$. Note that this means that Y and Y' were computed as

$$Y = F^{\mathcal{O}}_{k_t}(F^{\mathcal{O}}_{k_{t-1}}(\ldots F^{\mathcal{O}}_1(x_1)\ldots)) \qquad Y' = F^{\mathcal{O}}_{k_{t'}}(F^{\mathcal{O}}_{k_{t'-1}}(\ldots F^{\mathcal{O}}_1(x'_1)\ldots)). \quad (10)$$

Now, if there are i, j, i', j' where $i < j \leq m, j' < i' \leq m$ and $k_i = k'_{i'}, k_j = k'_{j'}$, then $B^{\mathcal{O},A}$ guesses that $\mathcal{O}(break, \{Y, Y'\})$ is 1 and guesses that it is 0 otherwise.

[16] Having chosen plaintext access to a function $T(.)$, we pick some key k and evaluate $C^{\langle 1,2\rangle,\langle 2,1\rangle}_{T(.),F^{\mathcal{O}}(k,.)}$ on some input X to get an output $Y = Y_1 \| Y_2$. As $\{\langle 1, 2\rangle, \langle 2, 1\rangle\}$ is ugly, if $T(.)$ is of the form $F^{\mathcal{O}}(k', .)$, then $\mathcal{O}(break, \{Y_1, Y_2\})$ will be 1, and if $T(.)$ is a URF, then $\mathcal{O}(break, \{Y_1, Y_2\})$ will almost certainly be 0. Thus we can distinguish $F^{\mathcal{O}}(k, .)$ with random k from a URF.

When the guess is 1, it is always correct by the definition of $\mathcal{O}(break, .)$. So we must show that when the guess is 0 then $\mathcal{O}(break, \{Y, Y'\}) = 1$ is very unlikely.

First we assume that the random $X_1, \ldots, X_q \in Q$ are all in S_0, this will hold but with probability $q/2^n$. Next, we assume that for the case where the Y_i are computed as $F^{\mathcal{O}}(k, X_i)$, A never makes a query $\mathcal{O}(eval, k, X)$ for any X. As k is random this will be true with probability at least $q/2^n$. Now if $B^{\mathcal{O},A}$ wrongly guesses that $\mathcal{O}(break, \{Y, Y'\})$ is 0, then the initial input x_1 or x'_1 from (10) was not in S_0. As x_1 and x'_1 were not received as an output from $\mathcal{O}$ (otherwise we could extend one of the sequences of (10)), A has guessed a value outside of S_0. As S_0 is a random subset which covers all but a $1/2^n$ fraction of possible inputs, the probability that A could have guessed an x_1 outside of S_0 is at most $1/2^n$ (same for x'_1). $\triangle$

The lemma follows from the two claims above.

Acknowledgments

We would like to the thank the Eurocrypt committee for their suggestions.

References

1. Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In *Advances in Cryptology — EUROCRYPT '06*, volume 4004 of *LNCS*, pages 409–426. Springer, 2006.
2. Ivan Damgård and Jesper B. Nielsen. Expanding pseudorandom functions; or: From known-plaintext security to chosen-plaintext security. In *Advances in Cryptology — CRYPTO '02*, volume 2442 of *LNCS*, pages 449–464. Springer, 2002.
3. Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. ACM*, 33(4):792–807, 1986.
4. Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.
5. Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *Proc, 21th ACM Symposium on the Theory of Computing (STOC)*, pages 44–61, 1989.
6. Michael Luby and Charles Rackoff. Pseudo-random permutation generators and cryptographic composition. In *Proc, 18th ACM Symposium on the Theory of Computing (STOC)*, pages 356–363, 1986.
7. Ueli Maurer. Indistinguishability of random systems. In *Advances in Cryptology — EUROCRYPT '02*, volume 2332 of *LNCS*, pages 110–132. Springer, 2002.
8. Ueli Maurer and Krzysztof Pietrzak. Composition of random systems: When two weak make one strong. In *Theory of Cryptograpy — TCC '04*, volume 2951 of *LNCS*, pages 410–427. Springer, 2004.
9. Ueli Maurer, Krzysztof Pietrzak, and Renato Renner. Indistinguishability amplification, 2006. Cryptology ePrint Archive: Report 2006/456, 2006.
10. Ueli Maurer and Johan Sjödin. A fast and key-efficient reduction of chosen-ciphertext to known-plaintext security. In *Advances in Cryptology — EUROCRYPT '07*, LNCS. Springer, 2007. This proceedings.

11. Kazuhiko Minematsu and Yukiyasu Tsunoo. Expanding weak PRF with small key size. In *ICISC '05*, volume 3935 of *LNCS*, pages 284–298. Springer, 2005.
12. Moni Naor and Omer Reingold. From unpredictability to indistinguishability: A simple construction of pseudo-random functions from MACs. In *Advances in Cryptology — CRYPTO '98*, LNCS, pages 267–282. Springer, 1998.
13. Krzysztof Pietrzak and Johan Sjödin. Weak pseudorandom functions in minicrypt, November 2006. Manuscript.

A Proof of Lemma 2

Proof (of Lemma 2). To show that expansions of type $\mathfrak{G}$ are good, we must show that for any expansion α of type $\mathfrak{G}$, the security of C_F^α as a weak random function can be black-box reduced to the security of F as a weak random function.

Let $\mathbf{Adv}_q^A(F,G)$ denote the advantage of the distinguisher A to distinguish the randomized function F from G in a q query known-plaintext attack. More formally, consider the random variable $Q^F = (X_1,\ldots,X_q,Y_1,\ldots,Y_q)$ where the X_i's are uniformly random and $Y_i = F'(X_i)$ for an instantiation F' of F, then

$$\mathbf{Adv}_q^A(F,G) = \mathsf{Pr}[A(Q^F) \rightarrow 1] - \mathsf{Pr}[A(Q^G) \rightarrow 1].$$

We prove the following statement:

For any expansion α of type $\mathfrak{G}$, any randomized function F with range and domain $\{0,1\}^n$, there exists an adversary B such that for any adversary A

$$\mathbf{Adv}_{q\cdot\alpha_{max}}^{B^{A,F}}(F,\mathbf{R}_{n,n}) \geq \frac{\mathbf{Adv}_q^A(C_F^\alpha,\mathbf{R}_{n,n\cdot t})}{\#\alpha} - \frac{q^2\cdot\alpha_{max}^2}{2^n}. \qquad (11)$$

Where $\alpha_{max} = \max(\alpha_1,\ldots,\alpha_{\#\alpha})$. Moreover B only uses A and F as a black-box and it is efficient (basically, all that B has to do is to simulate C_F^α on q inputs and it invokes A only once).

So if A breaks the security of C_F^α as a weak RF, then B breaks the security of the underlying F as a weak RF. For the special case of pseudorandom functions, this statement implies that if F is a weak PRF, then so is C_F^α. We now prove (11).

Consider an expansion $\alpha = \{s_1,\ldots,s_t\}$ of type $\mathfrak{G}$. We can assume without loss of generality that for all $s \in \alpha$ and $0 < i < j \leq t$ it holds that $s[i] < s[j]$ (as we can always permute the letters of an α of type $\mathfrak{G}$ so that this holds).

For the proof it will be convenient to introduce a new random system. With $\mathbf{B}_{a,b}$ we denote a random beacon $\{0,1\}^a \rightarrow \{0,1\}^b$, this system is simply a random source which outputs a new uniformly random value in $\{0,1\}^b$ on each input. As $\mathbf{B}_{a,b}$ and $\mathbf{R}_{a,b}$ have exactly the same output distribution unless queried twice on the same input, it is easy to show that for any A (e.g. using the framework from section 4 for the second step)

$$\mathbf{Adv}_q^A(\mathbf{B}_{a,b},\mathbf{R}_{a,b}) \leq \Delta_q^{\mathsf{KPA}}(\mathbf{B}_{a,b},\mathbf{R}_{a,b}) \leq \frac{q^2}{2^{a+1}}. \qquad (12)$$

Let $m := \#\alpha$ and consider the hybrid systems $C_i \stackrel{\text{def}}{=} C^{\alpha}_{\mathbf{B}_1,\ldots,\mathbf{B}_i,F_{i+1},\ldots,F_m}$, where each $\mathbf{B}_i$ denotes an instantiation of $\mathbf{B}_{n,n}$. As $C_0 \equiv C^{\alpha}_{F_1,\ldots,F_m}$, $C_m \equiv C^{\alpha}_{\mathbf{B}_1,\ldots,\mathbf{B}_m} \equiv \mathbf{B}_{n,t\cdot n}$ we have

$$\mathbf{Adv}_q^A(C^{\alpha}_{F_1,\ldots,F_m}, \mathbf{B}_{n,t\cdot n}) = \sum_{i=1}^{m} \mathbf{Adv}_q^A(C_{i-1}, C_i). \tag{13}$$

For $i \in [m]$ let $B_i^{A,F}$ be an adversary which on input $(X_1, \ldots, X_{q\alpha_i}, Y_1, \ldots, Y_{q\alpha_i})$ simulates the computation of $C^{\alpha}_{\mathbf{B}_1,\ldots,\mathbf{B}_{i-1},T,F_{i+1},\ldots,F_m}$ (T to be defined) on q random inputs $X'_1, \ldots, X'_q$ to get outputs $Y'_1, \ldots, Y'_q$, and then outputs the output of $A(X'_1, \ldots, X'_q, Y'_1, \ldots, Y'_q)$. In this simulation the component T is only queried on uniformly random inputs[17]. Instead of choosing those inputs at random, we require that $B_i^{A,F}$ uses the values $X_1, X_2, \ldots$ if it has to define the random values which are used as inputs to T. Now, if T is a beacon $\mathbf{B}_{n,n}$, then $C^{\alpha}_{\mathbf{B}_1,\ldots,\mathbf{B}_{i-1},T,F_{i+1},\ldots,F_m}$ is C_i, and if T is an instance of F then it is C_{i-1}, so

$$\mathbf{Adv}_{q\alpha_i}^{B_i^{A,F}}(F, \mathbf{B}_{n,n}) = \mathbf{Adv}_q^A(C_{i-1}, C_i). \tag{14}$$

Now consider an adversary $B^{A,F}$ which first chooses a random $i \in [m]$ and then runs $B_i^{A,F}$. Using (14) in the second and (13) in the third step, we get:

$$\begin{aligned}
\mathbf{Adv}_{q\cdot\alpha_{max}}^{B^{A,F}}(F, \mathbf{B}_{n,n}) &= \frac{1}{m}\sum_{i=1}^{m} \mathbf{Adv}_{q\alpha_i}^{B_i^{A,F}}(F, \mathbf{B}_{n,n}) \\
&= \frac{1}{m}\sum_{i=1}^{m} \mathbf{Adv}_q^A(C_{i-1}, C_i) \\
&= \frac{\mathbf{Adv}_q^A(C^{\alpha}_{F_1,\ldots,F_m}, \mathbf{B}_{n,t\cdot n})}{m}.
\end{aligned} \tag{15}$$

To conclude the proof of (11) we must "replace" the beacons $\mathbf{B}$ in (15) by URFs $\mathbf{R}$. Below we use the triangle inequality in the first and third, and (15) in the second step. In the last step we use (12) twice.

$$\begin{aligned}
&\mathbf{Adv}_{q\cdot\alpha_{max}}^{B^{A,F}}(F, \mathbf{R}_{n,n}) \\
\geq\ &\mathbf{Adv}_{q\cdot\alpha_{max}}^{B^{A,F}}(F, \mathbf{B}_{n,n}) - \mathbf{Adv}_{q\cdot\alpha_{max}}^{B^{A,F}}(\mathbf{B}_{n,n}, \mathbf{R}_{n,n}) \\
=\ &\frac{\mathbf{Adv}_q^A(C^{\alpha}_{F_1,\ldots,F_m}, \mathbf{B}_{n,t\cdot n})}{m} - \mathbf{Adv}_{q\cdot\alpha_{max}}^{B^{A,F}}(\mathbf{B}_{n,n}, \mathbf{R}_{n,n}) \\
\geq\ &\frac{\mathbf{Adv}_q^A(C^{\alpha}_{F_1,\ldots,F_m}, \mathbf{R}_{n,t\cdot n})}{m} - \frac{\mathbf{Adv}_q^A(\mathbf{B}_{n,t\cdot n}, \mathbf{R}_{n,t\cdot n})}{m} - \mathbf{Adv}_{q\cdot\alpha_{max}}^{B^{A,F}}(\mathbf{B}_{n,n}, \mathbf{R}_{n,n}) \\
\geq\ &\frac{\mathbf{Adv}_q^A(C^{\alpha}_{F_1,\ldots,F_m}, \mathbf{R}_{n,t\cdot n})}{m} - \underbrace{\frac{q^2}{2^{n+1}\cdot m} - \frac{q^2\cdot\alpha_{max}^2}{2^{n+1}}}_{q^2\cdot\alpha_{max}^2/2^n} \qquad \square
\end{aligned}$$

[17] As $s[i] < s[j]$ if $i < j$, so T is invoked on either the global input or on the output of some $\mathbf{B}_j, j < i$.