Building PRFs from PRPs*

Chris Hall[1], David Wagner[2], John Kelsey[1], and Bruce Schneier[1]

[1] Counterpane Systems
{hall,kelsey,schneier}@counterpane.com
101 E. Minnehaha Pkwy
Minneapolis, MN 55419
(612) 823-1098

[2] U.C. Berkeley
daw@cs.berkeley.edu
Soda Hall
Berkeley, CA 94720-1776

Abstract. We evaluate constructions for building pseudo-random functions (PRFs) from pseudo-random permutations (PRPs). We present two constructions: a slower construction which preserves the security of the PRP and a faster construction which has less security. One application of our construction is to build a wider block cipher given a block cipher as a building tool. We do not require any additional constructions—e.g. pseudo-random generators—to create the wider block cipher. The security of the resulting cipher will be as strong as the original block cipher.
Keywords. pseudo-random permutations, pseudo-random functions, concrete security, block ciphers, cipher feedback mode.

1 Introduction and Background

In this paper we examine building psuedo-random functions from pseudo-random permutations. There are several well known constructions for building pseudo-random permutations from pseudo-random functions, notably [LR88]. However, the only results we are aware of for going in the reverse directions are the recent results of Bellare et. al. in [BKR98][1].

One primary justification for building pseudo-random functions is that it allows one to use the results of Bellare et. al. [BDJR97] to produce an n-bit cipher that can be used to encrypt more than $2^{n/2}$ blocks. Due to birthday attacks, n-bit permutations will leak information about the plaintext after $2^{n/2}$ blocks. By closing the loop between pseudo-random functions and permutations, we can also accomplish a number of things: widening the block width of a cipher, creating a provably secure 1-bit cipher feedback mode, and building encryption functions secure for more than $2^{n/2}$ blocks. Given the plethora of existing practical block

---

* This is the full version of an extended abstract published in CRYPTO 98.

[1] We were unaware of these results when we originally wrote our paper, but they were instead pointed out to us by an anonymous referee.

ciphers, it would be nice to be able to create pseudo-random functions from them directly without having to resort to building new primitives from scratch.

Our work extends previous work on pseudo-random functions (PRFs) and permutations (PRPs). PRFs and PRPs were initially defined in [GGM86] as functions (resp. permutations) which a polynomially-bounded attacker cannot to distinguish from truly random functions (resp. permutations) with more than neglible probability. A more recent paper by Bellare et al. [BDJR97] evaluates four different notions of security and applies those notions to the definitions of PRFs and PRPs. In addition, M. Luby has written a book on pseudorandomness which provides an excellent summary of the theoretical constructions leading up to PRFs [Lub96].

Some authors have made a distinction between PRPs and super PRPs. With a super PRP, an adversary is allowed to query for inverse evaluations of the permutation [LR88]. For our applications, we require the "super" variety of PRP. Therefore, for the remainder of this paper we shall consider only super PRPs; we usually omit the "super" prefix for conciseness.

Extensive research has been conducted on building PRPs from PRFs. Many of the constructions are based on Luby and Rackoff's original work [LR88]. Let $F(l,r) = \Psi^m(f_1,\ldots,f_m)(l,r)$ denote an m-round Feistel network where $f_i \in \mathbb{F}_{n:n}$. Then $F(l,r) \in \mathbb{P}_{2n}$ where $\Psi^i(f_1,\ldots,f_i)$ is defined by

$$\Psi(f)(l,r) = (r, l \oplus f(r))$$
$$\Psi^k(f_1,\ldots,f_k) = \Psi(f_k) \circ \Psi(f_{k-1}) \circ \cdots \circ \Psi(f_1)(l,r).$$

Luby and Rackoff [LR88] showed that an adversary has advantage at most $m(m-1)/2^n$ if they make $m < Q(n)$ queries for some polynomial $Q(x)$. Recall that the advantage is computed as

$$\mathrm{Adv}\, A = |P[A^p = 1] - P[A^f = 1]|$$

where A is an adversary who returns 1 if they believe they are looking at a $2n$-bit permutation from the $\Psi^3(f_1,f_2,f_3)$ family and 0 otherwise. Then $P[A^p = 1]$ denotes the probability that an attacker returns 1 when given $p \in \{\Psi^3(f_1,f_2,f_3) : f_i \in F_{n:n}\}$ and $P[A^f = 1]$ denotes the probability that an attacker returns 1 when given $f \in P_{2n}$. The result was generalized for $m < 2^{n/2}$ [AV96,M92,P91b,P92] to

$$\mathrm{Adv}\, A = O(m^2/2^n).$$

Many different researchers have investigated variations of this construction [AV96,C97,Luc96,M92,P91b] [P92,P97,SP91,SP92,ZMI89a,ZMI89b] and even proposed different constructions [M92,P97]. The exact nature of these constructions is beyond the scope of this document; they investigate building PRPs from PRFs, and we are interested in going the other direction.

In addition to designing PRPs from PRFs, some researchers have studied designing PRFs from smaller PRFs. Aiello and Venkatesan built a $2n$-bit to $2n$-bit function from eight n-bit to n-bit functions using a Benes transform [AV96].

They achieved the notable bound that

$$\text{Adv}\, A = |P[A^B = 1] - P[A^f = 1]| = O(m/2^n)$$

after m queries, where A^B is the result of executing the adversary A with the oracle instantiated by a function B from the Benes family, and A^f is the result of running the adversary with a random $2n$-bit function.

Aside from building wider functions, some researchers have examined building variable-length input PRFs (VI-PRFs). In [BCK96], Bellare et al. formalize the notation of VI-PRFs and analyze the security of several constructions. Their functions are constructed using simpler primitives: fixed-length input PRFs (FI-PRFs) which were first defined in [BKR94] in order to model the Data Encryption Algorithm (DES). One important feature of these papers is that they focus on *concrete security analysis*, which attempts to provide precise estimates of security, rather than being satisfied with asymptotic results. Bellare et al. initiated this study in [BKR94,BGR95].

When it comes to building PRFs from PRPs, though several different people have noted that a PRP can be used as a PRF with advantage $O(m^2/2^n)$ for $m < 2^{n/2}$ (e.g. [AV96]), there has been a notable lack of research in this area. One recent exception is the excellent paper of Bellare et. al. [BKR98], which uses the notion of data-dependent keying to build a PRF from a PRP. Their results present strong evidence for the security of their PRP→PRF construction, and they take some initial steps towards a more complete analysis of its strength against computationally-bounded adversaries. One of the most appealing features of their construction is its practicality: the construction is very simple, and performance is degraded by only a factor of two (or less, when in stream cipher modes). It should be possible to use their re-keying construction in the applications found in Section 3, as a drop-in replacement for our PRP→PRF construction. This would provide corresponding improvements in performance; the tradeoff is that the available security results are weaker for the re-keying construction.

One interesting motivation for building PRFs from PRPs is that we could build larger PRPs from smaller PRPs by first constructing PRFs from the PRPs, using the results of [AV96] to strengthen the function, and finally using one of the many results available for building PRPs from PRFs. This is in fact the approach we take in our paper, and it provides the first useful technique (which we are aware of) for securely increasing the block width of a trusted block cipher.

The format of the rest of our paper is as follows. In Section 2 we introduce our two constructions for producing PRFs from PRPs: $\texttt{ORDER}(P)_i^{j:k}$ and $\texttt{TRUNCATE}(P)_n^{n-m}$. In Section 3 we apply our constructions to a few different problems and give descriptions of our solutions. Finally, in Section 4 we analyze the security of the constructions presented in Section 2.

1.1 Notation

In this section we introduce some of the notation we will use through the rest of the paper:

R_n $\mathbb{Z}_2^n$, the set of all n-bit blocks.

$\mathbb{P}_n$ The symmetric group on 2^n elements (specifically, the set of all permutations on R_n).

$\mathbb{F}_{n:m}$ The set of all functions with the signature $R_n \rightarrow R_m$.

$p(a, b)$ Denotes the evaluation of a permutation p with the bit concatentation of a and b.

(t, q, e)-secure Captures the notion of a cryptographic primitive which is secure, in the sense that any adversary who uses at most t units of offline work and issues at most q chosen-plaintext/ciphertext queries can only get advantage at most e for distinguishing the primitive from a random function (or permutation). This roughly means that an attacker must either do t work or implement q chosen-text queries to have any chance of breaking the primitive; however, even then the attacker is not guaranteed a practically-useful attack. Therefore, this is a very strong measure of security, in the sense that if a primitive is proven strong under this model, it is likely to be very strong indeed.

2 The Constructions

We introduce two constructions: $\texttt{ORDER}(P)_i^{j:k}$ and $\texttt{TRUNCATE}(P)_n^{n-m}$. The former uses a permutation $p \in \mathbb{P}_i$ to produce a function $f \in \mathbb{F}_{j:k}$. The latter uses a permutation $p \in \mathbb{P}_n$ to produce a function $f \in \mathbb{F}_{n:n-m}$. In this section we give a brief description of each of these constructions along with their security, but leave the analysis of their security until Section 4.

2.1 $\texttt{ORDER}(\boldsymbol{P})_{n+1}^{n:1}$

Our first construction, $\texttt{ORDER}(P)_{n+1}^{n:1}$ uses the order of entries in a table representing a $n+1$-bit permutation to encode a function $f_p \in \mathbb{F}_{n:1}$. More specifically, if $p \in \mathbb{P}_{n+1}$ then we assign a function $f_p \in \mathbb{F}_{n:1}$ where

$$f_p(x) = \begin{cases} 0 & \text{if } p(0, x) < p(1, x) \\ 1 & \text{otherwise.} \end{cases}$$

Here the expression $p(0, x)$ stands for the result of applying p to the concatenation of the bit 0 and the n-bit value x.

Note, this construction is really a special instance of a much more general construction given in the next two sections. However, for our analysis we found it much simpler to analyze this simple case (in a later section) and extrapolate to the more general case.

2.2 Wider Outputs: $\texttt{ORDER}(\boldsymbol{P})_{n+km}^{n:2^{k-1}m}$

Of course, in practice a PRF with a 1-bit output is rarely useful. Fortunately, there are some simple techniques to build wide-output PRFs from PRPs with a 1-bit output.

One basic approach is to observe that $j = 2^k$ (independent) PRFs from $\mathbb{F}_{n:m}$ suffice to build a PRF on $\mathbb{F}_{n:jm}$. This yields the following construction. Given a PRF $F \in \mathbb{F}_{n+k:m}$, we build a wider PRF $G \in \mathbb{F}_{n:2^k m}$ by

$$G(x) = (F(0,x), F(1,x), \ldots, F(2^k - 1, x)).$$

This construction has the disadvantage that it reduces the input size slightly, which can be a problem for some applications.

Of course, this does not produce an optimal construction. We leave the analysis to Section 4, but in fact the next construction is optimal. By optimal we mean that it divides the set of permutations $\mathbb{P}_n$ into equally-sized equivalence classes such that each equivalence class has an odd number of permutations. This implies that we cannot divide the equivalence classes any further and still expect to have equally-sized equivalence classes.

2.3 Wider Outputs, Efficiently: $\mathtt{ORDER}(P)_{n+m}^{n:2^m-1}$

As one might expect, the construction of the previous section fails to extract all of the possible bits from a permutation. In some cases we can nearly double the number of bits that we obtain from a given permutation p by extracting more information about the sorted order of $p(x)$.

Suppose that we have a permutation $p \in \mathbb{P}_{n+m}$; then we build a function $f \in \mathbb{F}_{n:2^m-1}$ in the following fashion. First we determine 2^{m-1} bits of $f(x)$ by using the construction outlined in the previous section. That is, if $[f(x)]_i$ denotes bit i of $f(x)$ and $0 \le i < 2^{m-1}$, then

$$[f(x)]_i = \begin{cases} 0 & \text{if } p(0,i,x) < p(1,i,x) \\ 1 & \text{otherwise} \end{cases} .$$

Briefly, the remaining bits of information are obtained by comparing the minimum elements of two pairs of values $\min\{p(x_j), p(x'_j)\}$ for $j = 1, 2$.

To be more precise, we start by creating 2^n perfectly-balanced binary trees with $2^m - 1$ nodes (i.e. each one has height $m - 1$) and uniquely assign each tree to a different n-bit value x. Hence tree x will correspond to $f(x)$. For any given tree, each node has three values associated with it: a $n + m$-bit value $X(x)$, a 1-bit value $Y(x)$, and a m-bit value $Z(x)$, which serves to identify the node. For ease of exposition, we assign $Z(x)$ so that the root node has $Z(x) = 0$, the left child of a node has the value $2Z(x) + 1$, and the right child of a node has the value $2Z(x)+2$ (implying Z is independent of x so we can drop it). This assigns each node a unique m-bit value and allows us to associate bit i of $f(x)$ with the $Y(x)$ value of node i.[2]

Using these particular Z-values, the leaf nodes will have the values $2^{m-1} - 1$ to $2^m - 2$. Let $X_i(x)$ and $Y_i(x)$ denote the X and Y values respectively of the

[2] Some other ordering of the nodes will also work, such as one given by a postorder traversal of the tree.

leaf nodes with $Z = i$. Then for $i \geq\ 2^{m-1} - 1$

$$Y_i(x) = \begin{cases} 0 & \text{if } p(0, Z_i - 2^{m-1} - 1, x) < p(1, Z_i - 2^{m-1} - 1, x) \\ 1 & \text{otherwise.} \end{cases}$$
$$X_i(x) = \min\{p(0, Z_i - 2^{m-1} - 1, x), p(1, Z_i - 2^{m-1} - 1, x)\}$$

This is precisely the information obtained by the construction of the previous section. For the remaining $2^{m-1} - 1$ bits of information, we assign

$$Y_i(x) = \begin{cases} 0 & \text{if } X_{2i+1}(x) < X_{2i+2}(x) \\ 1 & \text{otherwise.} \end{cases}$$
$$X_i(x) = \min\{Y_{2i+1}(x), Y_{2i+2}(x)\}$$

It should be clear that this tree partially encodes the order of $p(0, x), \ldots, p(2^m - 1, x)$, but that more than one permutation may produce the same function. It requires 2^m invocations of the permutation p to produce the $2^m - 1$ bits of $f_p(x)$. For an example evaluation of $f(x)$ for $p \in \mathbb{P}_3$, see Figure 1.

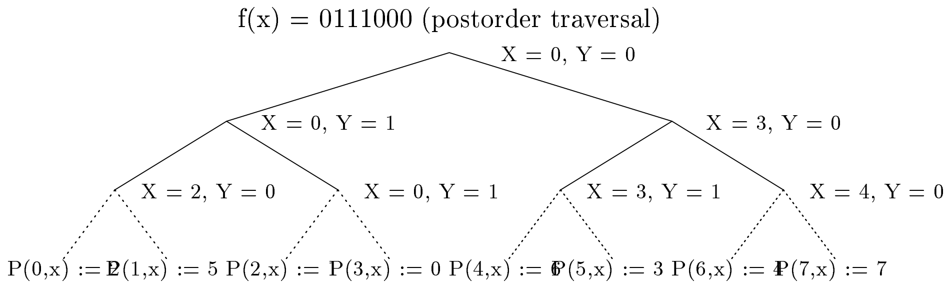

Fig. 1. Example Function from $p \in \mathbb{P}_3$.

We can use this latter technique to build a PRF in $\mathbb{F}_{m:n}$ using a permutation in $\mathbb{P}_a$ where $a = \lceil m + \log_2(n+1) + 1 \rceil$. It will require $\lceil \log_2(n+1) + 1 \rceil$ invocations of E per invocation of $\mathbb{F}_{m:n}$. Note, if $n \neq 2^l - 1$ for some l, we will actually obtain a wider output than n bits. In that case we can simply truncate the output to n bits and retain the security of the PRF.

While this construction provably transfers (essentially) all of the security of the underlying block cipher to the PRF, the disadvantage is that it has poor performance: we can get a 57:127 PRF that's provably as strong as DES (and hence a 57:64 PRF), but it requires 128 queries to DES per PRF computation.

2.4 $\texttt{TRUNCATE}(P)_n^{n-m}$

Our second construction has much better performance, but uses a very different idea: we merely truncate a few bits of the output of the underlying block cipher, so the PRF can be almost as fast as the block cipher. Formally, let $p \in \mathbb{P}_n$ be a random permutation. We assign a function $f_p \in \mathbb{F}_{n:n-m}$ by

$$f_p = g \circ p$$

where $g : R_n \to R_{n-m}$ denotes the function which truncates the high m bits from its input. For a PRP family $\{\pi_k\}$ the resulting PRF family would be $\{f_{\pi_k}\}$.

The disadvantage to this approach is that it doesn't preserve the security of the underlying block cipher nearly as well: our proofs only work when the attacker has at most $O(\min\{2^{(n+m)/2}, 2^{2(n-m)/3}\})$ chosen texts available to him, where n is the block width of the underlying cipher and m is the number of truncated bits. In practice, this means that we can prove security up to $O(2^{4n/7})$ chosen texts by truncating $m = n/7$ bits, but our analysis degrades too much to be provide better bounds for larger m.

3 Applications

There are several nice applications of our result. Probably one of the most interesting is that we "close the loop" between PRFs and PRPs. Luby-Rackoff [LR88] gave a nice PRF $\to$ PRP construction; we have now shown how to go the other direction[3].

We explore two additional possibilities in the next sections. There are others listed below, but due to a lack of time and space we have omitted further analysis of these ideas. Hence we pose them as open problems for further study.

1. Building MACs (or possibly hash functions) with provable security. The disadvantage is that they are likely to be very slow.
2. Building provably-strong PRNGs out of our constructions for provably-strong PRFs. Such a tool might be used for session key derivation, for example. The advantage is that in many cases (depending upon the application, of course) the PRNG isn't performance-critical, so slow techniques are still interesting if they have notable security advantages.
3. Building provably-strong stream ciphers from our constructions for provably-strong PRFs. By running a good PRF in counter mode, you can get security past the birthday bound. In contrast, 64-bit block ciphers typically run into security problems when used in a standard chaining mode to encrypt more than 2^{32} known texts, no matter how strong the cipher is.
   Bellare et. al. have explored this application further in [BKR98].

3.1 Building Wider Block Ciphers

Techniques for building wider block ciphers are especially relevant as the AES standards effort ramps up. The problem with most existing ciphers, such as

[3] We don't consider treating a PRP as a PRF as an example because it won't produce more general functions, and because its security level is limited.

Triple-DES, is that they offer only 64-bit blocks, and thus fall prey to certain birthday attacks that can work with only 2^{32} texts or so. (The matching ciphertext attack is one example.) As network communication speeds rise, this limit becomes increasingly concerning: for instance, on a 1 Gbit/sec encrypted link, we expect that information about two plaintext blocks will leak after only 6 minutes or so. The only solution is to move towards ciphers with wider block lengths, but if this involves a full cipher redesign, then we may forfeit the insights provided by more than two decades of analysis on DES and Triple-DES. This motivates our search for a construction which can provably retain the time-tested security level of Triple-DES while providing a wider block length.

This paper provides new results in this area. If we have a trusted cipher, then we can model it as a PRP family. Using one of our constructions we can construct a PRF family, use the Benes transform [AV96] to create a wider PRF family, and finally use Luby-Rackoff to create a PRP family again. The nice thing is that the resulting PRP family will be almost four times as wide as the original construction. Furthermore, we will be able to provide provable security reductions to show that the widened cipher is likely to be strong if the original cipher is secure.

We will focus on a particular example and consider Triple-DES. Hence let $n = 64$ and $P = \{E_k\}$ be the Triple-DES family where $E_k(X)$ denotes encryption with key k and plaintext X. Also suppose that P is (t, q, e) secure. Then using $\mathtt{ORDER}(P)^{58:63}_{64}$ we can construct a PRF family $F = \{f_{E_k}\} \subset \mathbb{F}_{58:58}$ (truncate the 5 extra bits). As later analysis will show, F is $(t, q/64, e)$ secure. In other words, F largely retains the security properties of Triple-DES.

Using the modified Benes transform, we can form a second PRF family $F^2 = \{g_f\}$ where $f \in F$ and $F^2 \subset \mathbb{F}_{110:110}$. The reason we do not obtain a family $F^2 \subset \mathbb{F}_{116:116}$ is that the modified Benes transform requires six independent functions. Hence we can use the first three bits of our functions in F to obtain eight independent function families for constructing $\mathbb{F}^2$. The results of [AV96] show that F^2 is a $(t, q/64, e+e')$-secure PRF, where e' is neglible for $q/64 < 2^{110}$.

Finally, using F^2 and Luby-Rackoff we can create a final PRP family $P^2 = \{p_{g_1,g_2,g_3}\}$ where $g_i \in F^2$ and $P^2 \subset P_{220}$. P^2 will have nearly identical security to F^2: $(t, q/64, e + e' + e'')$ where e'' is neglible for $q/64 < 2^{554}$.

The primary disadvantage of this construction is that the resulting widened cipher P^2 will be almost 450 times slower than the original cipher P. With a normal 4-round Luby-Rackoff construction, we will use 4 invocations of functions from F^2 for each invocation of a permutation in P^2. Each function in F^2 uses 6 invocations of functions from F. Finally, for each function in F we will use 64 invocations of a permutation in P. Hence we will require $4 \times 6 \times 64 = 1536$ invocations of Triple-DES per invocation of P^2; of course, each invocation of P^2 encrypts $220/64$ times as many bits as P, so the performance of the widened cipher P^2 will be $1536 \cdot 64/220 \approx 447$ times worse than Triple-DES. This is definitely very slow, but it is provably secure!

---

[4] Note: the security proof of the Luby-Rackoff construction given in [LR88] actually assume that the g_i are independent; but the proofs in [NR96] remove that restriction.

An alternative to using Luby-Rackoff is to use a construction by M. Naor and O. Reingold [NR96]. There you get a twofold speed-up in execution and we only require 768 invocations of Triple-DES. This translates to a total of roughly 223 times worse performance than Triple-DES. The advantage, of course, is that the resulting cipher is that we have removed the 2^{32}-texts limitation on the security of Triple-DES.

Our construction uses the Benes transform only for technical reasons. The reason we can't apply the Luby-Rackoff construction directly to our PRF family F is that a 4-round Luby-Rackoff cipher with m-bit blocks is only secure up to $2^{m/4}$ texts; with $m = 2 \cdot 58$, the security level would be too low, so we build a double-width PRF family F^2 to increase m. However, eliminating the Benes transform could produce significant performance speedups, so this motivates the search for PRP constructions with better security. For example, by using Patarin's recent results on the 6-round Luby-Rackoff construction [P98], we can build a widened cipher $P^3 \subset P_{116}$ that is secure with up to about $\min\{q, 2^{43.5}\}$ texts; by using single-DES instead of Triple-DES as our starting point, we can get a 116-bit cipher which is provably as secure as DES and has performance about 212 times worse than DES (or about 71 times worse than Triple-DES), though it has somewhat less security than our construction of P^2.

It would also be possible to use the $\mathtt{TRUNCATE}(P)_n^{n-m}$ construction to build a double-width block cipher, instead of $\mathtt{ORDER}(P)_6 4^{58:63}$ as above. This would provide significantly better performance (the widened cipher could be as fast as 1/3 the performance of the original cipher). However, at present the available proofs provide no guarantee of security past about 2^{36} texts, which is probably not a compelling advantage over the 2^{32} birthday bound.

As a third alternative, one could use the re-keying construction of Bellare et. al. [BKR98] to build F out of Triple-DES (say). Applying the Benes transform and the Naor-Reingold construction would then provide a 256-bit cipher which is only 3 times slower than Triple-DES. The disadvantage is that the available security results are difficult to compare with the figures given above.

The examples we gave here usually resulted in a block cipher with a peculiar width. It should be clear how to modify this example slightly to generate (say) a 192-bit block cipher, by truncating F or F^2 to the appropriate size.

3.2 Applications to 1-bit CFB mode

We note that our main construction provides a way to increase the robustness of 1-bit CFB mode by tweaking the mode slightly.

The standard 1-bit CFB mode builds a function $h : \mathbb{Z}_2^n \to \mathbb{Z}_2$ by letting $h(x)$ be the least significant bit of the encryption $E_k(x)$ of x under a block cipher E with key k. Then we build a stream cipher as $C_j = P_j \oplus h(C_{j-64}, \ldots, C_{j-1})$.

The problem is that we are not aware of any proof that CFB mode preserves the security of the underlying block cipher. Clearly all of the security of 1-bit CFB mode must come from the non-linear Boolean function h. Theorem 1 (in Section 4.1) guarantees the security of h against an adversary with access to $q \ll 2^{n/2}$ chosen-text queries, assuming the underlying block cipher is secure.

However, for typical block ciphers $2^{n/2} = 2^{32}$, which means that the "security warranty" provided by Theorem 1 is voided after 2^{32} chosen-text queries[5]. As most typical block ciphers are built to resist much more powerful adversaries, we would prefer to have better reductions.

We do not know of any better security proofs for 1-bit CFB in the literature, but we can improve the situation with a slight modification to the mode. Replace h by the function f_{E_k} defined in Section 2.1. (We will need to sacrifice one bit of feedback, so that $C_j = P_j \oplus f_{E_k}(C_{j-63}, \ldots, C_{j-1})$, but 63 bits is more than sufficient for practical purposes.) This requires two encryptions per invocation of f_{E_k}, so our mode will be twice as slow as the standard 1-bit CFB, but we do not expect this to be a serious problem, as implementors typically use 1-bit CFB mode for its resynchronization properties rather than for its performance characteristics.

Of course, the primary advantage of our 1-bit modified cipher feedback mode is that we can provide provable security reductions for f_{E_k}. If E is a (t, q, e)-PRP, then f_{E_k} will be a $(t, q/2, e)$-PRF. In short, our construction of f_{E_k} preserves the security level of the underlying block cipher extremely effectively. Therefore, this modification to 1-bit CFB mode looks attractive for practical use.

4 Analysis

In this section we provide analysis of our $\texttt{ORDER}(P)_i^{j:k}$ and $\texttt{TRUNCATE}(P)_n^{n-m}$ constructions. In addition, we evaluate the security of a PRP family when viewed as a PRF family.

4.1 Previous attempts

As we have mentioned earlier, every PRP can be viewed as a PRF with certain security parameters. We first analyze this trivial construction.

Theorem 1. *Let p be a random permutation on n bits. Then p is a (t, q, e)-PRF for $e = q^2/2^{n+1}$.*

Proof. Let (X, Y) be a transcript of any adversary that attempts to distinguish p from a random function: X is a q-vector listing the oracle queries, and Y is a q-vector listing the resulting outputs. (Without loss of generality we may assume X contains no repeated values: i.e. $X_i \neq X_j$ for all $i \neq j$.) Let f be a random function on n bits.

Then, conditioned on the event that Y contains no repeated values, the distribution of Y will be the same for both p and f. Let E_f (resp. E_p) be the event that Y contains no repeated values when the oracle is instantiated by f (resp.

[5] Theorem 7 can guarantee security up to $q = O(2^{4n/7})$, but this is still significantly smaller than the $O(2^n)$ query security we would ideally hope to see. As we shall see, this hope is not unreasonable.

p). Clearly $\mathrm{Prob}(E_p) = 1$. Since the output of A depends only on X, Y, we see that

$$\mathrm{Prob}(A^p = 1) = \mathrm{Prob}(A^p = 1 \mid E_p) = \mathrm{Prob}(A^f = 1 \mid E_f).$$

The birthday paradox lets us bound $\mathrm{Prob}(E_f)$ as follows:

$$\mathrm{Prob}(E_f) \geq 1 - q^2/2^{n+1}.$$

Finally, we may compute

$$\begin{aligned}
\mathrm{Adv}\, A &= |\mathrm{Prob}(A^p = 1) - \mathrm{Prob}(A^f = 1)| \\
&= |\mathrm{Prob}(A^f = 1 \mid E_f) - \mathrm{Prob}(A^f = 1 \mid E_f)\mathrm{Prob}(E_f) \\
&\qquad -\mathrm{Prob}(A^f = 1 \mid \neg E_f)\mathrm{Prob}(\neg E_f)| \\
&= (1 - \mathrm{Prob}(E_f))|\mathrm{Prob}(A^f = 1 \mid E_f) - \mathrm{Prob}(A^f = 1 \mid \neg E_f)| \\
&\leq q^2/2^{n+1}.
\end{aligned}$$

and this is the desired result. □

Furthermore, it is simple to show that this bound is tight. We can easily construct an adversary which distinguishes between p and a random function with advantage approximately $q(q-1)/2^{n+1}$ (for $q = O(2^{n/2})$): simply look for collisions and return 0 if you see a collision (where you are supposed to return 1 if you think it's not a random function).

It is worth noting that this analysis also establishes the security of another related construction. One might naively propose a construction based (loosely) on the Davies-Meyer hashing mode [MMO85]:

$$f_p(x) = p(x) \oplus x.$$

The final xor of x into the ciphertext destroys the bijective property of p, so at first glance f_p might look like a reasonable candidate for a better PRF. However, we note that this construction has no better security than before. It can be distinguished from a random function with advantage $q^2/2^{n+1}$: merely apply the adversary of the previous paragraph to the function g defined by $g(x) = f_p(x) \oplus x$.

The security reduction we showed in Theorem 1 is sufficient to show that PRFs exist if PRPs do, from a complexity-theoretic point of view, since the security bound it shows is exponential in n. Therefore, complexity theorists interested only in asymptotics need read no further. However, practical applications are a bit more demanding: they require concrete security guarantees.

We find this $O(2^{n/2})$ level of security inadequate for practical applications. Most block ciphers today offer 64-bit block widths, thus providing a convenient and efficient PRP with $n = 64$. For such ciphers, the above theorem provides no security assurances when adversaries are allowed to make $q \approx 2^{(n+1)/2} = 2^{32.5}$ chosen-text queries (or more). This is too weak for serious cryptologic use; we would prefer something that provides better resistance to chosen-text attacks. After all, the underlying block cipher typically provides better security than that—so it is natural to wonder whether we can do better. Is there a PRF construction that preserves the security of the underlying block cipher?

We show below that the answer is yes.

4.2 Analysis of $\mathtt{ORDER}(P)^{n:1}_{n+1}$

We gave a description of $\mathtt{ORDER}(P)^{n:1}_{n+1}$ in Section 2.1. Let π be a (keyed) family of permutations $\{\pi_k : k \in K\} \subset \mathbb{P}_{n+1}$ on R_{n+1}. Using this construction we obtain a family $f_\pi = \{f_{\pi_k} : k \in K\}$ of functions in $\mathbb{F}_{n:1}$. We can (by a slight abuse of notation) view π as a random variable, taking values in $\mathbb{P}_{n+1}$, by taking k to be a random variable uniformly distributed over K. (We drop the subscript, writing π instead of π_k as a slight abuse of notation, to avoid an unwieldy morass of distractingly nested subscripts.) Similarly, f_π can be viewed as a random variable, too.

We say that p is a random permutation (on R_{n+1}) to mean that it is a random variable which is uniformly distributed over all elements of $\mathbb{P}_{n+1}$. Similarly, we say that f is a random function (from $\mathbb{F}_{n:m}$) when we mean that it is a random variable which is uniformly distributed over $\mathbb{F}_{n:m}$.

We wish to show that f_π preserves the security level of the underlying PRP π. Most of the work to be done is handled by a purely information-theoretic analysis, which ignores all issues of computational complexity. We tackle this in Theorem 2.

Theorem 2. *If p is a random permutation on R_{n+1}, then f_p is a random function over $\mathbb{F}_{n:1}$.*

Proof. Take any $g \in \mathbb{F}_{n:1}$. It is clear that there exists a $p \in \mathbb{P}_{n+1}$ such that $g = f_p$: for example, take the p such that

$$p(2x) = 2x + g(x) \qquad p(2x+1) = 2x + 1 - g(x) \qquad \forall x \in R_n.$$

Next we show that $|\{p : g = f_p\}|$ is a constant that does not depend on g, i.e. that there are an equal number of representative permutations p for all g.

First, suppose that $g_1, g_2 \in \mathbb{F}_{n:1}$ are two functions that differ at exactly one point X (i.e. $g_1(x) = g_2(x)$ for all $x \neq X$ and $g_1(X) \neq g_2(X)$). Then we construct a bijective mapping $\phi : \mathbb{P}_{n+1} \to \mathbb{P}_{n+1}$, which has the property that $f_p = g_1$ exactly when $f_{\phi(p)} = g_2$. This will show that there are an equal number of representations for any two functions g_1, g_2 which differ at exactly one point. Then it will be easy to see that this implies the desired result, since for any two functions $g, h \in \mathbb{F}_{n:1}$ one can construct a sequence $g = g_0, g_1, g_2, \ldots, g_{k-1}, g_k = h$ such that all the consecutive pairs g_i, g_{i+1} differ at exactly one point.

The mapping ϕ is built as follows. Take any input p; we define $\phi(p) = p'$ by

$$p'(b,x) = \begin{cases} p(b,x) & \text{if } x \neq X \\ p(1-b,x) & \text{if } x = X. \end{cases}$$

Now it is clear that $f_{p'} = g_2$ if $f_p = g_1$, and vice versa. Furthermore, ϕ is an involution, so it is clear that it is a bijective mapping, as claimed. This completes the proof. □

Once we have this nice result, extending it to the setting of computationally-bounded adversaries is not so hard. It requires much unravelling of notation, but essentially no new ideas.

We first introduce the notion of pseudo-randomness, to handle the most important case where the adversary is computationally bounded. Informally, saying that π is a pseudo-random permutation (PRP) on R_{n+1} is supposed to convey the idea that it is computationally infeasible for an adversary to distinguish π from a random permutation on R_{n+1}. (Some authors use the phrase "pseudo-random permutation generator" to refer to this object; for conciseness, we will omit the "generator" term throughout this paper.)

We formalize this notion as follows. An adversary is an oracle machine $B^{p,p^{-1},\pi,\pi^{-1}}$ which outputs a 0 or 1 (according to whether it thinks p is truly random or is drawn from the family $\{\pi_k : k \in K\}$). It takes four oracles as inputs: a test permutation p (which outputs $p(x)$ on input x) along with its inverse p^{-1}, and an oracle for π (which outputs $\pi_k(x)$ on input k, x) as well as an oracle for π^{-1}. Its advantage $\mathrm{Adv}\, B$ is

$$\mathrm{Adv}\, B = |\mathrm{Prob}(B^{\pi_k,\pi_k^{-1},\pi,\pi^{-1}} = 1) - \mathrm{Prob}(B^{r,r^{-1},\pi,\pi^{-1}} = 1)|,$$

where r is a random permutation and k is uniformly distributed over K. More formally, we say that π is a (t, q, e)-PRP if the advantage of any adversary which is allowed at most q queries (total) to the first two oracles and t offline work is at most e. This models a block cipher which is secure with up to q adaptive chosen-plaintext/ciphertext queries and t trial encryptions. See [BKR94,BGR95] for more information about (t, q, e) security.

We can define a (t, q, e)-PRF (pseudo-random function) in a similar fashion. In this definition, an adversary is an oracle machine $A^{g,\gamma,\pi,\pi^{-1}}$ with access to four oracles: a function g which outputs $g(x)$ on input x, an oracle γ which outputs $\gamma_k(x)$ on input k, x, and two oracles π, π^{-1} for the PRP class (as above). We define its advantage by

$$\mathrm{Adv}\, A = |\mathrm{Prob}(A^{\gamma_k,\gamma,\pi,\pi^{-1}} = 1) - \mathrm{Prob}(A^{s,\gamma,\pi,\pi^{-1}} = 1)|,$$

where s is a random function and k is uniformly distributed over K. In the cases that we are most interested in, we have $\gamma_k = f_{\pi_k}$. We say that γ is a (t, q, e)-PRF if all adversaries A which make at most q oracles queries (total) to g, γ and perform at most t computations obey $\mathrm{Adv}\, A \le e$.

Note that it is important to include the oracles for π, π^{-1} in the definition of a (t, q, e)-PRF. In what follows, we will be interested in PRFs built from a PRP π. Here π models a block cipher; we assume the algorithm is publicly known (by Kerchkoff's principle), so anyone trying to attack f_π can freely compute $\pi_k(x)$ on any chosen inputs k, x. This required us to extend the standard definition of a PRF to model this situation.

With those preliminaries out of the way, we may proceed to the rest of the analysis. We get the following pleasing consequence of Theorem 2 whose proof we leave to the appendices.

Theorem 3. *If π is a (t, q, e)-PRP on R_{n+1}, then f_π is a $(t, q/2, e)$-PRF over $\mathbb{F}_{n:1}$.*

Proof. See Appendix A. $\square$

4.3 Analysis of $\texttt{ORDER}(P)_{n+m}^{n:2^m-1}$

In Section 2.3 we introduced the general $\texttt{ORDER}(P)_{n+m}^{n:2^m-1}$ construction. There are two corresponding theorems whose full proofs we omit due to a lack of space.

Theorem 4. *If π is a random permutation on R_{n+m}, then f_π is a random function over $\mathbb{F}_{n:2^m-1}$.*

Proof. (sketch) The basic idea is to again consider two functions $f_1, f_2 \in \mathbb{F}_{n:2^m-1}$ which differ in exactly one output (say $f_1(0) \neq f_2(0)$). We can build a map $\phi : \mathbb{P}_{n+m} \to \mathbb{P}_{n+m}$ such that $f_p = f_1$ exactly when $f_{\phi(p)} = f_2$. Again this will show that all $f \in \mathbb{F}_{n:2^m-1}$ have an equal number of representative permutations p (existance is trivial).

To build the map ϕ, we need merely look at the binary tree we constructed for $x = 0$. (Actually we must consider a slightly expanded version of the tree in which the leaf nodes of our original tree are expanded into two children containing the values $p(0, Z, x)$ and $p(1, Z, x)$.) Starting with $i = 0$, we compare bit i of $f_1(0)$ and $f_2(0)$ and swap the left and right subtrees of node i if $f_1(0)$ and $f_2(0)$ differ in bit i. Note, this may destroy the original equality of bits $j > i$ for $f_1(0)$ and $f_2(0)$ so in evaluating bit i we assume that $f_1(0)$ has the value denoted by the most recent tree. The end result is a series of subtree swaps for evaluating $f_1(0)$ which are clearly reversible.

The subtree swaps specified will remap values of a permutation to values of another permutation and we take ϕ to be that map. It is clearly onto and hence bijective. This completes the proof of the theorem. $\square$

Theorem 5. *If π is a (t, q, e)-PRP on R_{n+m}, then f_π is a $(t, q/2^m, e)$-PRF over $\mathbb{F}_{n:2^m-1}$.*

Proof. The proof is nearly identical to that of Theorem 3. $\square$

In Section 2.3 we made a claim that this construction was optimal. By that we meant that one could not create a map from $\mathbb{P}_{n+m}$ to $\mathbb{F}_{n:l}$ for $l > 2^m - 1$. We state this in the following lemma.

Lemma 1. *There exists no map $\phi : \mathbb{P}_{n+m} \to \mathbb{F}_{n:l}$ for $l > 2^m - 1$ such that $N(f) = |\{p \in \mathbb{P}_{n+m} : \phi(p) = f\}|$ is constant for all $f \in \mathbb{F}_{n:l}$.*

Proof. Let ϕ be a map such that $N(f)$ is constant for all $f \in \mathbb{F}_{n:l}$, then we will show that $l \leq 2^m - 1$. There are 2^{l2^n} functions in $\mathbb{F}_{n:l}$ hence we are dividing $\mathbb{P}_{n+m}$ into 2^{l2^n} equivalence classes. For any j, $|\mathbb{P}_j| = (2^j)!$ and it is not hard to show that 2^{2^j-1} exactly divides $(2^j)!$. Hence the power of 2 dividing the size of each equivalence class is $2^{2^{n+m}-1}/2^{l2^n} = 2^{2^n(2^m-l)-1}$. In order for $N(f)$ to be constant, we must have that $2^n(2^m - l) - 1 \geq 0$, which implies $2^m - l \geq 1$, whence $l \leq 2^m - 1$. This completes the proof of the lemma. $\square$

Our analysis above used a strongly information-theoretic framework: first, we showed that the construction produces a random function when fed a random permutation (Theorem 2), and then all the desired pseudo-randomness results just fall out trivially from that. This framework is desirable because it makes the analysis relatively simple; however, we showed in Lemma 1 that it imposes serious limitations on the performance of the resulting constructions. The above bound essentially shows that, to achieve better performance, we'll need to do abandon the information-theoretic framework and take another approach. This we do below.

4.4 Analysis of $\mathtt{TRUNCATE}(P)_n^{n-m}$

The construction of Section 4.2 is probably most attractive because it is so amenable to theoretical analysis, and because it preserves the security of the underlying block cipher so efficiently no matter how many chosen-text queries are issued. However, it also has a severe disadvantage for practical use: it is quite slow.

In Section 2.4 we defined a PRF family $\mathtt{TRUNCATE}(P)_n^{n-m}$ based on truncating bits of a permutation. The result trades off security for performance. Recall the construction: for any permutation π_k on R_n, we define a function $f \in \mathbb{F}_{n:n-m}$ by

$$f_{\pi_k} = g \circ \pi_k$$

where $g : R_n \to R_{n-m}$ denotes the function which truncates the high m bits from its input.

We could instead have taken g to be any fixed function $g : R_n \to R_{n-m}$ such that each $y \in R_{n-m}$ has 2^m easily-computable pre-images $g^{-1}(y)$, and the results would still apply. However, bit-truncation is attractive because it is both fast and amenable to a simple mathematical description[6]. Therefore, for clarity of exposition we concentrate hereafter solely on bit-truncation.

First we show that if π is a random permutation, then f_π is a pseudo-random function. The following theorem proves that, roughly speaking, $\operatorname{Adv} A$ is negligible while $q \ll \min\{2^{(n+m)/2}, 2^{2(n-m)/3}\}$.

Theorem 6. *If π is a random permutation on R_n, then f_π is a (t, q, e)-PRF over $\mathbb{F}_{n:n-m}$, where $e = 5(q^2/2^{n+m})^{1/3} + q^3/2^{2(n-m)+1}$.*

Proof. See Appendix B. □

This shows that truncating some bits from the output of π_k gives slightly better security. For $m \leq n/7$, the theorem says that truncating m bits adds nearly $m/2$ bits of security against adaptive chosen-text attacks to the PRF π_k.

However, for $m > n/7$, the second term in e dominates (in that it is largest and hence limits q), and our analysis does not provide better security reductions when increasing m past $n/7$. We believe that these limits are not inherent, but

[6] The study of the properties of bit-truncation may also have some independent interest, as several authors have already suggested applying a bit-truncation output transform to existing MAC constructions in hopes of improving their security (see, e.g., [PO95]).

rather are a reflection of the inadequacy of our analysis. As an illustration, the best attack we can find needs $q = O(2^{(n+m)/2)}$ texts to distinguish f_π from random with significant advantage (see Theorem 8), so this leaves a substantial gap between the upper and lower bounds. We suspect that a better analysis could provide a better security reduction. (However, we could be wrong.)

The main idea of the proof is to show that the probability of getting any particular set of outputs Y to a given set of oracle-queries X is roughly the same whether the oracle is instantiated "under the hood" by a PRF or by a truncated-PRP. We use this to show that any oracle algorithm A must behave almost exactly the same regardless of which type of oracle it is given. That follows just because A's execution can depend only on the list of inputs and outputs to the oracle. This can then be used to show that $\mathrm{Adv}\, A$ is small. Of course, our bounds only hold when q is small enough.

The first step in the analysis is to compute the probabilities that a random function F and a truncated-PRP f_π will map X to Y. For F this is easy, but for f_π it is substantially harder. In the general case this gets quite messy, so we restrict ourselves to the special case where there are no three-way collisions in Y; this makes the calculation tractable. (This restriction adds an artificial contribution $q^3/2^{2(n-m)+1}$ to our bound on the advantage, so we have sacrificed tightness for tractability.) After that, all that is left is relatively straightforward computations (albeit in large quantities).

Theorem 7. *If π is a (t, q, e)-PRP on R_n, then f_π is a (t, q, e')-PRF over $\mathbb{F}_{n:n-m}$, where $e' = e + 5(q^2/2^{n+m})^{1/3} + q^3/2^{2(n-m)+1}$.*

Proof. Omitted. Follows directly from Theorem 6 along lines analogous to the proof of Theorem 3. □

Theorem 8. *Let π be a permutation family on R_n. Then for all $q = O(2^{(n+m)/2})$, there is an adversary which can distinguish f_π from a random function with q known texts, $O(q)$ work, and advantage $\Omega(q^2/2^{n+m})$.*

Proof. (sketch) Let r count the number of collisions in the outputs $f_\pi(1), \ldots, f_\pi(q)$. Then our adversary outputs 1 (guessing that the oracle is f_π) if $r < q(q-1)/2^{n-m+1}$, and 0 otherwise. Using the techniques found in the proof of Theorem 6, we find that this adversary operates with advantage $\Omega(q^2/2^{n+m})$. □

5 Conclusion

We have presented two constructions for generating pseudo-random functions given pseudo-random permutations: $\texttt{ORDER}(P)_i^{j:k}$ and $\texttt{TRUNCATE}(P)_n^{n-m}$. The former had the notable property that it preserved the security of the underlying pseudo-random permutation whereas the latter had the property that it was much more efficient. Unfortunately, the gain in speed results in a trade-off in security and the latter construction fails to preserve the strength of the underlying pseudo-random permutation.

Using our constructions we were able to solve a few different problems, including stretching the width of a block cipher while preserving the security. We also examined a secure 1-bit cipher feedback mode using a pseudo-random permutation.

6 Acknowledgements

Thanks to Dan Boneh, Ian Goldberg, and the anonymous referees for many helpful comments.

References

[AV96] W. Aiello, R. Venkatesan, "Foiling birthday attacks in length doubling transformations," *Advances in Cryptology—EUROCRYPT '96 Proceedings*, Springer-Verlag, pp. 307–320.

[BCK96] M. Bellare, R. Canetti, H. Krawczyk, "Pseudorandom Functions Revisited: The Cascade Construction and its Concrete Security," *Proceedings of the 37th Symposium on Foundations of Computer Science*, IEEE, 1996.

[BDJR97] M. Bellare, A. Desai, E. Jokipii, P. Rogaway, "A Concrete Security Treatment of Symmetric Encryption: Analysis of the DES Modes of Operation," Full version, *Extended abstract in Proceedings of 38th Annual Symposium on Foundations of Computer Science (FOCS 97)*, IEEE, 1997.

[BGR95] M. Bellare, R. Guérin, P. Rogaway, "XOR MACs: New methods for message authentication using finite pseudorandom functions," *Advances in Cryptology—CRYPTO '95 Proceedings*, Springer-Verlag, 1995, pp 15–28.

[BKR94] M. Bellare, J. Kilian, P. Rogaway, "The security of cipher block chaining," *Advances in Cryptology—CRYPTO '94 Proceedings*, Springer-Verlag, 1994.

[BKR98] M. Billare, T. Krovetz, P. Rogaway, "Luby-Rackoff Backwards: Increasing Security by Making Block Ciphers Non-Invertible (Extended Abstract)," *Advances in Cryptology—EUROCRYPT '98 Proceedings*, Springer-Verlag, 1998.

[BM84] M. Blum, S. Micali, "How to Generate Cryptographically Strong Sequences of Pseudo-random Bits," *SIAM J. Comput.*, 13 (Nov. 1984), pp. 850–864.

[C97] D. Coppersmith, "Luby-Rackoff: Four rounds is not enough," IBM Research Report, RC 20674 (12/24/96), Mathematics.

[GGM86] O. Goldreich, S. Goldwasser, S. Micali, "How to Construct Random Functions," *Journal of the ACM*, Vol. 33, No. 4, October 1986, pp. 792–807.

[LR88] M. Luby, C. Rackoff, "How to Construct Pseudorandom Permutations from Pseudorandom Functions," SIAM J. Comput., Vol. 17, No. 2, April 1988, pp. 373–386.

[Lub96] M. Luby, *Pseudorandomness and Cryptographic Applications*, Princeton University Press, 1996.

[Luc96] S. Lucks, "Faster Ruby-Lackoff Ciphers," *Proceedings of Third Fast Software Encryption Workshop*, Springer-Verlag, pp. 189–203.

[M92] U.M. Maurer, "A Simplified and Generalized Treatment of Luby-Rackoff Pseudorandom Permutation Generators," *Advances in Cryptology—EUROCRYPT '92 Proceedings*, Springer-Verlag, 1992, pp. 239–255.

[MMO85] S.M. Matyas, C.H. Meyeter, J. Oseas, "Generating strong one-way functions with cryptographic algorithm," *IBM Technical Disclosure Bulletin*, 27 (1985), 5658–5659.

[NR96] M. Naor, O. Reingold, "On the construction of pseudo-random permutations: Luby-Rackoff revisited.," preliminary version, `http://www.wisdom.weizmann.ac.il/Papers/trs/CS96-10/abstract.html`

[P90] J. Pieprzyk, "How to Construct Pseudorandom Permutations from Single Pseudorandom Functions," *Advances in Cryptology—EUROCRYPT '90*, Springer-Verlag, pp. 140–150.

[P91a] J. Patarin, "Etude des gńŕateurs de permutations basés sure le Schéma du D.E.S.," Ph. D. Thesis, INRIA, Domaine de Voluceau, Le Chesnay, France, 1991.

[P91b] J. Patarin, "New Results on Pseudorandom Permutation Generators Based on the DES Scheme," *Advances in Cryptology—CRYPTO '91 Proceedings*, Springer-Verlag, pp. 301–312.

[P92] J. Patarin, "How to Consruct Pseudorandom and Super Pseudorandom Permutations from One Single Pseudorandom Function," *Advances in Cryptology—EUROCRYPT '92 Proceedings*, Springer-Verlag, pp. 256–266.

[P97] J. Patarin, "Improved Security Bounds for Pseudorandom Permutations," *Proceedings of the Fourth ACM Conference on Computer and Communications Security*, April 1–4, 1997, pp. 142–150.

[P98] J. Patarin, "About Feistel Schemes with Six (or More) Rounds," *Proceedings of the Fifth Fast Software Encryption Workshop*, LNCS 1372, Springer, 1998, pp. 103–121.

[PO95] B. Preneel, P. van Oorschot, "MDx MAC and building fast MACs from hash functions," *Advances in Cryptology—CRYPTO '95 Proceedings*, LNCS 1070, Springer-Verlag, 1996.

[SP91] B. Sadeghiyan, J. Pieprzyk, "On Necessary and Sufficient Conditions for the Construction of Super Pseudorandom Permutations," *Advances in Cryptology—ASIACRYPT '91*, Springer-Verlag, pp. 194–209.

[SP92] B. Sadeghiyan, J. Pieprzyk, "A Construction for Super Pseudorandom Permutations from A Single Pseudorandom Function," *Advances in Cryptology—EUROCRYPT '92*, Springer-Verlag, pp. 267–284.

[Y82] A.C. Yao, "Theory and Applications of Trapdoor Functions," *Proceedings of the 23rd IEEE Symposium on Foundations of Computer Science*, IEEE, New York, 1982, pp. 80–91.

[ZMI89a] , Y. Zheng, T. Matsumoto, H. Imai, "On the Construction of Block Ciphers Provably Secure and Not Relying on Any Unproved Hypothesis," *Advances in Cryptology—CRYPTO '89 Proceedings*, Springer-Verlag, pp. 461-480.

[ZMI89b] Y. Zheng, T. Matsumoto, H. Imai, "Impossibility and Optimality Results on Constructing Pseudorandom Permutations," *Advances in Cryptology—EUROCRYPT '89*, Springer-Verlag, pp. 412–421.

A Proof of Theorem 3

Proof. Our proof proceeds as follows. Suppose we have an adversary A which (t', q', e')-breaks f_π.[7] We construct an adversary B which $(t', 2q', e')$-breaks π. The result will follow.

[7] This means that with at most q' queries and t' offline encryptions, there is an adversary who has advantage greater than e'.

The construction for B requires very little creativity. $B^{p,p^{-1},\pi,\pi^{-1}}$ performs the same computations as $A^{g,\gamma,\pi,\pi^{-1}}$; anytime A makes an oracle query, we simulate the oracle and return the result to the computation in progress.

The simulation of oracle queries goes like this. If A queries the g oracle with x, then B issues two queries to p with inputs $(0,x)$ and $(1,x)$ and compares the results; if $p(0,x) < p(1,x)$, B uses the result 0, and otherwise uses 1. If A queries the γ oracle with k,x, then B issues the two queries $k,(0,x)$ and $k,(1,x)$ to its oracle for π, and constructs the result similarly. Finally, A's oracle queries for π and π^{-1} can be satisfied trivially.

Let $e' = \mathrm{Adv}\, A$, and let t', q' count the time and number of oracle queries that A requires. Clearly $t', 2q'$ counts the time and number of oracle queries that B requires. It merely remains to bound $\mathrm{Adv}\, B$, which we achieve with the following series of observations.

Lemma 2. *Let r be a random permutation, and s be a random function. (Recall that according to our terminology both will be uniformly distributed on their respective spaces.)*

(i) For any permutation p, $A^{f_p,f_\pi,\pi,\pi^{-1}} = B^{p,p^{-1},\pi,\pi^{-1}}$.

(ii) The random variable f_r has the same distribution as s; in other words, f_r is a random function.

(iii) With the random variables r,s as before, we have $Prob(A^{s,f_\pi,\pi,\pi^{-1}} = 1) = Prob(A^{f_r,f_\pi,\pi,\pi^{-1}} = 1)$.

(iv) $Prob(A^{s,f_\pi,\pi,\pi^{-1}} = 1) = Prob(B^{r,r^{-1},\pi,\pi^{-1}} = 1)$.

(v) $Prob(A^{f_{\pi_k},f_\pi,\pi,\pi^{-1}} = 1) = Prob(B^{\pi_k,\pi_k^{-1},\pi,\pi^{-1}} = 1)$.

Proof. (i) follows by construction of B.

(ii) is exactly Theorem 2.

(iii) follows immediately from (ii).

(iv) follows from (i) and (iii).

(v) is merely a special case of (i), substituting $p = \pi_k$. $\square$

Lemma 3. *$\mathrm{Adv}\, B = \mathrm{Adv}\, A$.*

Proof. Apply part (v) of Lemma 2 to the first term in $\mathrm{Adv}\, B$, and apply part (iv) of Lemma 2 to the second term in $\mathrm{Adv}\, B$. We get exactly the expression for $\mathrm{Adv}\, A$. $\square$

Now this suffices to prove the theorem. Given that π is a (t,q,e)-PRP, we show that f_π is a $(t,q/2,e)$-PRF by examining the contrapositive. Suppose there exists an adversary A who $(t,q/2,e')$-breaks f_π with advantage $e' > e$. We have shown how to construct an adversary B which breaks π. By Lemma 3, $\mathrm{Adv}\, B = \mathrm{Adv}\, A = e'$; also, B requires at most t time and q oracle queries. In other words, B (t,q,e')-breaks π, for $e' > e$. But this contradicts our assumption that π is a (t,q,e)-PRP. Therefore, such A cannot exist, and the theorem follows. $\square$

B Proof of Theorem 6

Proof. Let $A^{f_{\pi_k},\pi,\pi^{-1}}$ be any adversary that breaks f_{π_k}. Let $X = (X_1, \ldots, X_q)$ be a q-vector over R_n of q different "inputs," and let $Y = (Y_1, \ldots, X_q)$ be a q-vector over R_{n-m} of "outputs." The random variables (X, Y) will be a "transcript" of a run of the adversary A: X_j records the j-th query to the f_π oracle, and Y_j records the corresponding output from the oracle.

Without loss of generality, we may take $X = (1, \ldots, q)$, since any adversary A which makes repeated queries $X_i = X_j$ $(i < j)$ can be easily converted to an adversary A' with the same advantage such that A' makes no repeated queries. (Our construction of A' merely simulates the action of A, for each oracle query X_j made by A: if X_j is a new query, A' behaves identically to A; but if $X_j = X_i$ for some $i < j$, then A' sets $Y_j = Y_i$ and continues to simulate the action of A without querying the f_π oracle.) Furthermore, since π is presumed to be a truly random permutation, the values of the oracle queries don't matter, so long as they don't repeat.

Let $F \in \mathbb{F}_{n:n-m}$ be a truly random function. Define $p_F(X, Y) = \text{Prob}(F(X) = Y)$ to be the probability that F maps each $X_j \mapsto Y_j$; define $p_f(X, Y) = \text{Prob}(f_\pi(X) = Y)$ in a similar fashion. We often leave off the (X, Y) and simply write p_f or p_F when the choice of (X, Y) is clear from context. Also, we sometimes write $p_f(S)$ to mean the probability (with respect to f) of a set S, i.e. $p_f(S) = \sum_{(X,Y)\in S} p_f(X, Y)$.

We wish to show that f_π is roughly indistinguishable from F if q is not too large. The main idea is to show that $p_f(X, Y) \approx p_F(X, Y)$. Our argument proceeds as follows. We bound p_f/p_F, showing that it is close to 1. This bound doesn't hold uniformly for all choices of (X, Y), but it holds for nearly all of them—or more precisely, the set S where the bound holds has probability very close to 1. Formally, we prove that $|p_f/p_F - 1| \le \delta$ for all $(X, Y) \in S$; we also show that both $p_f(\neg S)$ and $p_F(\neg S)$ are small. This can be informally viewed as a sort of "probabilistic bound."

We prove, in another crucial lemma, that

$$\text{Adv}\, A \le \max\{p_f(\neg S), p_F(\neg S)\} + \delta.$$

This is a generic result that relies only on the bound on p_f/p_F; no domain-specific knowledge is required. Therefore, it suffices to bound p_f/p_F tightly enough that $p_f(\neg S)$, $p_F(\neg S)$, and δ are small.

We move to the details of the proof. We take S to be the set of all q-vectors Y over R_{n-m} which have r repeated values, no triply-repeated values, and for which

$$|r - q(q-1)/2^{n-m+1}| \le cq/2^{(n-m+1)/2},$$

where $c \ge 1$ is a small constant left free for the moment. Lemma 5 helps us show that $|p_f/p_F - 1| \le \delta$ for $(X, Y) \in S$. Lemma 6 bounds $p_F(\neg S)$, and Lemma 7 bounds $p_f(\neg S)$. Finally, Lemma 4 proves that $\text{Adv}\, A \le \max\{p_f(\neg S), p_F(\neg S)\} + \delta$. Combining these four lemmas, we get the big result

$$\text{Adv}\, A \le 1/c^2 + 4cq/2^{(n+m+1)/2} + q^3/2^{2(n-m)+1} + q^2/2^{n+m}$$

for all $c \geq 1$. Finally, we optimize over c to obtain the best bound; taking $c = (2^{(n+m-1)/2}/q)^{1/3}$ yields

$$\mathrm{Adv}\, A \leq 4(q^2/2^{n+m})^{1/3} + q^3/2^{2(n-m)+1} + q^2/2^{n+m} \leq 5(q^2/2^{n+m})^{1/3} + q^3/2^{2(n-m)+1},$$

as claimed[8].

We now proceed to the text of the lemmas.

Lemma 4. *With S defined as above, we have*

$$Adv\, A \leq \max\{p_f(\neg S), p_F(\neg S)\} + \delta$$

when $\delta \geq \max_{(X,Y)\in S} |p_f(X,Y)/p_F(X,Y) - 1|$.

Proof. Let $T = \{(X,Y) : A(X,Y) = 1\}$ be the set of transcripts (X,Y) such that A outputs 1. Of course, T does not depend on whether the oracle is instantiated by f_{π_k} or F, but the probability of observing any pair (X,Y) does depend on the choice of oracle.

The definition of $\mathrm{Adv}\, A$ says that

$$\mathrm{Adv}\, A = \left|\mathrm{Prob}(A^{f_{\pi_k},\pi,\pi^{-1}} = 1) - \mathrm{Prob}(A^{F,\pi,\pi^{-1}} = 1)\right|.$$

The first term is just $\sum_{(X,Y)\in T} p_f(X,Y)$, and the second term similarly becomes $\sum_{(X,Y)\in T} p_F$, so we see that $\mathrm{Adv}\, A = |\sum_{(X,Y)\in T} p_f - p_F|$. We may break this sum up into a sum over $T\cap\neg S$ and a sum over $T\cap S$: $\mathrm{Adv}\, A = |\sum_{(X,Y)\in T\cap\neg S} p_f - p_F + \sum_{(X,Y)\in T\cap S} p_f - p_F| \;\leq\; |\sum_{T\cap\neg S} p_f - p_F| + |\sum_{T\cap S} p_f - p_F|$. We will use the bounds on $p_f(S)$ and $p_F(S)$ to bound the sum over $T \cap \neg S$, and use the bound on $|p_f/p_F - 1|$ to bound the sum over $T \cap S$. For the first, we find

$$\Big|\sum_{(X,Y)\in T\cap\neg S} p_f - p_F\Big| = \big|p_f(T\cap\neg S) - p_F(T\cap\neg S)\big| \leq \max\{p_f(T\cap\neg S), p_F(T\cap\neg S)\} \leq \max\{p_f(\neg S), p_F(\neg S)\}.$$

For the second term, we obtain

$$\Big|\sum_{(X,Y)\in T\cap S} p_f - p_F\Big| \;=\; \Big|\sum_{T\cap S} p_F\cdot(p_f/p_F - 1)\Big| \;\leq\; \delta\Big|\sum_{T\cap S} p_F\Big| \;\leq\; \delta$$

since $|p_f/p_F - 1| \leq \delta$ for all $(X,Y) \in T\cap S \subset S$. The lemma follows immediately. $\square$

Lemma 5. *Let Y have r repeated values and no triply-repeated values, with $X = (1,2,\ldots,q)$. Then $p_F = 2^{-q(n-m)}$, and*

$$p_f = (1-2^{-m})^r \prod_{i=0}^{q-1} \frac{2^m}{2^n - i}.$$

[8] We conjecture that it may be possible to remove the exponent 1/3 from the first term above by improving our current crude Chebyshev bound in Lemma 6 (possibly using the theory of martingales); however, we have not confirmed this.

We have $p_f/p_F \approx \exp\{q(q-1)/2^{n+1} - r/2^m\}$ for large q, r. Finally, if $(X,Y) \in S$ and $q \leq 2^{(n+m)/2}/c$ and $q \leq 2^{2n/3}$, then $|p_f/p_F - 1| \leq \delta$ for $\delta = 2cq/2^{(n+m+1)/2} + 2q^3/3 \cdot 2^{-2n} + q^2/2^{n+m+1}$.

Proof. The formula for p_F follows immediately from the definition of F.

Verifying the result for p_f requires a bit of calculation. When $r = 0$, it follows easily: $\mathrm{Prob}(f_\pi(x_1) = y_1) = 2^m/2^n$, and $\mathrm{Prob}(f_\pi(x_2) = y_2 \mid f_\pi(x_1) = y_1) = 2^m/(2^n - 1)$ (for $y_1 \neq y_2$), and so on; then

$$\mathrm{Prob}(f_\pi(X) = Y) = \prod_{i=0}^{q-1} \mathrm{Prob}(f_\pi(x_{i+1}) = y_{i+1} \mid f_\pi(X_{1..i}) = Y_{1..i}) = \prod_{i=0}^{q-1} \frac{2^m}{2^n - i}.$$

Next we consider the case $r > 0$. Each repeated value $y_j = y_i$ ($j > i$) diminishes the probability by a factor $(2^m - 1)/2^m$, because $\mathrm{Prob}(f_\pi(x_j) = y_j \mid f_\pi(X_{1..j-1}) = Y_{1..j-1}) = (2^m - 1)/(2^n - j + 1)$ instead of $2^m/(2^n - j + 1)$ as computed earlier. (Note that π is a permutation and $x_i \neq x_j$, so $\pi(x_i) \neq \pi(x_j)$, which implies that there are only $2^m - 1$ possible values for $\pi(x_j)$ satisfying $g(\pi(x_j)) = y_j$.) This establishes the formula for p_f.

Dividing, we see that

$$p_f/p_F = (1 - 2^{-m})^r / \prod_{i=0}^{q-1} (1 - i \cdot 2^{-n}).$$

The second term is approximately $\exp -q(q-1)/2^{n+1}$, for large q; the first term is approximately $\exp -r/2^m$ for large r. When $(X,Y) \in S$, some cancellation occurs, and we get

$$p_f/p_F \approx \exp(q(q-1)/2^{n-m+1} - r)/2^m = \exp x$$

where $|x| \leq cq/2^{(n-m+1)/2} \cdot 2^{-m} = cq/2^{(n+m+1)/2}$. Thus, heuristically we expect $|p_f/p_F - 1| \approx |x| \leq cq/2^{(n+m+1)/2}$; we shall formalize this estimate below.

We can get an explicit bound by some standard tricks. For instance, we have the bound $e^{-ab-a^2b} \leq (1-a)^b \leq e^{-ab}$ ($0 \leq a \leq 1/2$). Also, taking the logarithm of $P = 1/\prod_{i=0}^{q-1}(1 - i/2^n)$, we get $\log P = \sum_{i=0}^{q-1} - \log(1 - i/2^n)$, and by using the Taylor series expansion $\log(1+x) = x - x^2/2 + x^3/3 - \ldots$, we see $i/2^n \leq -\log(1 - i/2^n) \leq i/2^n + i^2/2^{2n}$. Taking sums, we get that $0 \leq \log P - q(q-1)/2^{n+1} \leq q(q-1)(2q-1)/(6 \cdot 2^{2n})$. Therefore, we get

$$q(q-1)/2^{n+1} - r/2^m - r/2^{2m} \leq \log(p_f/p_F) \leq q(q-1)/2^{n+1} + q(q-1)(2q-1)/(6 \cdot 2^{2n}) - r/2^m.$$

Exponentiating this expression will give a concrete bound on p_f/p_F.

Finally, we will derive an explicit bound on $|p_f/p_F - 1|$ when $(X,Y) \in S$. Let $x = q(q-1)/2^{n+1} - r/2^m = (q(q-1)/2^{n-m+1} - r)/2^m$; then because $(X,Y) \in S$, we get $|x| \leq cq/2^{(n+m+1)/2}$. Also $r/2^{2m} \leq q^2/2^{n+m+1} + cq/2^{(n+3m+1)/2}$ since $(X,Y) \in S$. We obtain

$$p_f/p_F \geq e^{x - r/2^{2m}} \geq 1 + x - r/2^{2m} \geq 1 - 1.5cq/2^{(n+m+1)/2} - q^2/2^{n+m+1}.$$

Also, letting $y = q(q-1)(2q-1)/(6 \cdot 2^{2n})$, we may conclude $y \leq q^3/3 \cdot 2^{-2n}$. When $q \leq 2^{(n+m)/2}/c$ and $q \leq 2^{2n/3}$, we may deduce $|x| + y \leq 1.05$ and hence

$$p_f/p_F \leq e^{x+y} \leq 1 + 2x + 2y \leq 1 + 2cq/2^{(n+m+1)/2} + 2q^3/3 \cdot 2^{-2n}.$$

Therefore, $|p_f/p_F - 1| \leq \delta$ when $\delta = 2cq/2^{(n+m+1)/2} + 2q^3/3 \cdot 2^{-2n} + q^2/2^{n+m+1}$. □

Lemma 6. *We have*

$$p_F(\neg S) \leq 1/c^2 + q^3/6 \cdot 2^{-2(n-m)}.$$

Proof. We can easily obtain the bound $p_F(S) \leq a + b$ where $a = \text{Prob}_F(|r - q(q-1)/2^{n-m+1}| > cq/2^{(n-m+1)/2})$ and b stands for the probability of obtaining a triply-repeated value in Y (i.e. i, j, k with $Y_i = Y_j = Y_k$). It is easy to see that

$$b \leq \binom{q}{3}/2^{2(n-m)} \leq q^3/6 \cdot 2^{-2(n-m)}.$$

Next we bound a, the probability of obtaining too many or too few collisions. We define the random variable $Z_{i,j}$ $(1 \leq i < j \leq q)$ to be 1 if $Y_i = Y_j$ or 0 otherwise, and let $Z = \sum_{i<j} Z_{i,j}$. Then each $Z_{i,j}$ is a Boolean variable with $\mathbb{E}\, Z_{i,j} = 1/2^{n-m}$ and $\text{Var}\, Z_{i,j} = (1 - 1/2^{n-m})/2^{n-m}$. By linearity of expectations, we obtain $\mathbb{E}\, Z = q(q-1)/2^{n-m+1}$. Also, since the $Z_{i,j}$ are pairwise independent, we find that $\text{Var}\, Z = q(q-1) \cdot (1 - 1/2^{n-m})/2^{n-m+1}$. Now a standard Chebyshev bound shows that $\text{Prob}(|Z - \mathbb{E}\, Z| \geq c \cdot \sqrt{\text{Var}\, Z}) \leq 1/c^2$, whence $a \leq 1/c^2$ and the result follows. □

Lemma 7. *We have*

$$p_f(\neg S) \leq 1/c^2 + q^3/6 \cdot 2^{-2(n-m)} + \delta.$$

Proof. Lemma 5 assures us that $p_f(S) \geq (1-\delta)p_F(S)$. Also, the previous lemma shows that $p_F(S) \geq 1 - a$ when $a = 1/c^2 + q^3/6 \cdot 2^{-2(n-m)}$. Then a simple computation suffices to obtain the desired result:

$$p_f(\neg S) = 1 - p_f(S) \leq 1 - (1-\delta)p_F(S) \leq 1 - (1-\delta)(1-a) \leq \delta + a.$$

□

This completes the proof of Theorem 6. □