On the Round Security of Symmetric-Key Cryptographic Primitives

Zulfikar Ramzan Leonid Reyzin[*]

November 30, 2000

Abstract

We put forward a new model for understanding the security of symmetric-key primitives, such as block ciphers. The model captures the fact that many such primitives often consist of iterating simpler constructs for a number of rounds, and may provide insight into the security of such designs.

We completely characterize the security of four-round Luby-Rackoff ciphers in our model, and show that the ciphers remain secure *even if the adversary is given black-box access to the middle two round functions.* A similar result can be obtained for message authentication codes based on universal hash functions.

1 Introduction

1.1 Block Ciphers

A *block cipher* is a family of permutations on a message space indexed by a secret key. Each permutation in the family deterministically maps *plaintext* blocks of some fixed length to *ciphertext* blocks of the same length; both the permutation and its inverse are efficiently computable given the key.

Motivated originally by the study of security of the block cipher DES [16], Luby and Rackoff provided a formal model for the security of block ciphers in their seminal paper [14]. They consider a block cipher to be secure ("super pseudorandom," or secure under both "chosen plaintext" and "chosen ciphertext" attacks) if, without knowing the key, a polynomial-time adversary with oracle access to both directions of the permutation is unable to distinguish it from a truly random permutation on the same message space. This definition is an extension of the definition of a pseudorandom function generator from [12], where the adversary has oracle access only to the forward direction of the function.[1]

1.2 The Natural Round Structure of Symmetric-Key Primitives

In addition to defining security of block ciphers, Luby and Rackoff also provided a construction of a secure block cipher based on a pseudorandom function generator. Their block cipher consists of four rounds of *Feistel [11] permutations*, each of which consists of an application of a pseudorandom

[*] Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139. E-mail: {zulfikar, reyzin}@theory.lcs.mit.edu. URL: http://theory.lcs.mit.edu/{~zulfikar, ~reyzin}. This work appears in *Advances in Cryptology — Crypto 2000,* Mihir Bellare, editor, Lecture Notes in Computer Science vol. 1880, Springer-Verlag, 2000. © IACR

[1] The paper [14] also considers block ciphers that are just *pseudorandom*, or secure against chosen plaintext attack only, where the adversary has access only to the forward direction of the permutation.

function and an exclusive-or operation. Each round's output is used for the next round's input, except for the last round, whose output is the output of the block cipher.

Much of the theoretical research that followed the work of [14] focused on efficiency improvements to this construction (e.g., see [15], [18] and references therein). All of these variations can also be naturally broken up into rounds.

This theme of an inherent round structure in block ciphers is also seen extensively in practice. For example, a number of ciphers, including DES [16] and many of the AES submissions [17] have an inherent round structure (though not necessarily involving Feistel permutations), where the output of one round is used as input to the next.

In addition to block ciphers, constructions of other cryptographic primitives often also proceed in rounds. For example, universal-hash-function-based message authentication codes (UHF MACs) [6], [22], [9] can be viewed as consisting of two rounds. Moreover, cryptographic hash functions (e.g., MD-5 [19]), and the various message authentication schemes that are built on top of them (e.g., HMAC [1]), have an induced round structure as well.

Consequently, it should come as little surprise that cryptanalysts have often considered looking at individual rounds in order to better understand the security properties of a given design; for example, a large number of papers have been written analyzing reduced-round variants of block ciphers and hash functions (see [5], [21], and the references therein).

It thus seems that a theoretical framework incorporating the notion of rounds would be desirable. This paper proposes such a framework. Although our model is a simple extension of the classical models of security for symmetric primitives ([14], [12], [2]), it allows one to obtain a number of interesting results not captured by the traditional models. In particular, we analyze the security of the original Luby-Rackoff construction, some of its variants, and UHF MACs within our framework.

1.3 Our Contributions

1.3.1 A New Model

The definition of a secure block cipher from [14], or of a secure MAC from [3], allows the adversary only black-box access to the primitive. We develop the notion *round security*, which considers what happens when the adversary has additional access to some of the internal rounds of the computation of the primitive. We focus on block ciphers, but our techniques can be extended to other primitives such as MACs.

For example, in the case of block ciphers, we study what happens when the adversary is allowed, in addition to its chosen-plaintext and chosen-ciphertext queries, to input a value directly to some round i of the block cipher and view the output after some round j, with restrictions on i and j. The adversary's job is still the same: to distinguish whether the chosen-ciphertext and chosen-plaintext queries are being answered by the block cipher or by a random permutation. The queries to internal rounds are always answered by the block cipher.

As discussed below, this model allows us gain a better understanding of what makes symmetric constructions secure, and enables us to make statements about security that are not captured by the traditional model.

1.3.2 Round Security of Luby-Rackoff Ciphers

We completely characterize the round security of the Luby-Rackoff construction and its more efficient variants from [15] and [18]. That is, we precisely specify the sets of rounds that the adversary can access for the cipher to remain secure, and show that access to other sets of rounds will make the cipher insecure.

The cipher proposed by Luby and Rackoff [14] operates on a $2n$-bit string (L, R) and can be described simply as follows:

$$\begin{aligned} S &= L \oplus h_1(R) \\ T &= R \oplus f_1(S) \\ V &= S \oplus f_2(T) \\ W &= T \oplus h_2(V), \end{aligned}$$

where h_1, h_2, f_1, f_2 are pseudorandom functions, $\oplus$ represents the exclusive-or, and the output is (V, W).

Naor and Reingold [15] demonstrated that pseudorandom functions h_1 and h_2 can be replaced by XOR-universal hash functions, thus suggesting that strong randomness is important only in the middle two rounds. We extend their observation by showing that, in fact, secrecy is important in the first and last rounds, while randomness (but no secrecy) is needed in the middle two rounds. Specifically, we show that:

- The cipher *remains secure* even if the adversary has oracle access to both f_1 and f_2.
- The cipher becomes *insecure* if the adversary is allowed access to any other round oracles.

Moreover, we demonstrate that instantiating h_1 and h_2 as hash functions instead of as pseudorandom functions does not significantly lower the round security of the block cipher, thus supporting the observation that strong randomness is not needed in the first and last rounds of the Luby-Rackoff construction.

1.3.3 Round Security of Universal Hash Function MACs

Using techniques in our paper, one can also characterize the round security of a class of Universal-Hash Function-based Message Authentication Codes (UHF MACs). In the first round, these UHF MACs apply a universal hash function h to a relatively large message, to get a shorter intermediary string. Then, in the second round, they use a pseudorandom function f on the shorter string to get a final tag. It turns out that:

- A UHF MAC remains *secure* if the adversary has oracle access to f.
- A UHF MAC is, in general, *insecure* if the adversary has oracle to h.

1.3.4 Implications for the Random Oracle Model

Our work has interesting implications for Luby-Rackoff ciphers and UHF MACs in the random oracle model. One can easily define security of block ciphers and MACs in this model given the work of [4]: one simply allows all parties (including the adversary) access to the same oracle, and the adversary has to succeed for a random choice of the oracle.

Our results imply that the Luby-Rackoff cipher remains secure in the random oracle model if one replaces the functions f_1 and f_2 with random oracles. That is, in the random oracle model, keying material will only be necessary for h_1 and h_2, which, as shown in [15] and [18], can be just (variants of) universal hash functions.

Similarly, the UHF MAC remains secure if the pseudorandom function, used in the second round, is replaced with a random oracle. Thus, again, in the random oracle model, keying material is needed only for the hash function.

Block ciphers have been analyzed in the random-oracle model before. For example, Even and Mansour [10] construct a cipher using a public random *permutation* oracle P (essentially, the construction is $y = P(k_1 \oplus x) \oplus k_2$, where k_1 and k_2 constitute the key, x is the plaintext, and y is the resulting ciphertext). They show their construction is hard to invert and to existentially forge. We can recast their construction in our model, as a three-round cipher, where the adversary has access to the second round. Using the techniques in our paper, we can, in fact, obtain a stronger result; namely, that their cipher is super pseudorandom.

Of course, whether a scheme in the random oracle model can be instantiated securely in the real world (that is, with polynomial-time computable functions in place of random oracles) is uncertain, particularly in light of the results of Canetti, Goldreich and Halevi [7]. However, our results open up an interesting direction: is it possible to replace pseudorandom functions with unkeyed functions in any of the constructions we discuss?

2 Prior Definitions and Constructions

Below we describe the relevant definitions and prior constructions. Our presentation is in the "concrete" (or "exact") security model as opposed to the asymptotic model (though our results can be made to hold for either). Our treatment follows that of Bellare, Kilian, and Rogaway [3], and Bellare, Canetti, Krawczyk [2].

2.1 Definitions

2.1.1 Notation

For a bit string x, we let $|x|$ denote its length. If x has even length, then x^L and x^R denote the left and right halves of the bits respectively; we sometimes write $x = (x^L, x^R)$. If x and y are two bit strings of the same length, $x \oplus y$ denotes their bitwise exclusive OR. If S is a probability space, then $x \overset{R}{\leftarrow} S$ denotes the process of picking an element from S according to the underlying probability distribution. Unless otherwise specified, the underlying distribution is assumed to be uniform. We let I_n denote the set of bit strings of length n: $\{0,1\}^n$.

By a finite function (or permutation) family $\mathcal{F}$, we denote a set of functions with common domain and common range. Let $\mathsf{Rand}^{k \to l}$ be the set of all functions going from I_k to I_l, and let Perm^m be the set of all permutations on I_m. We call a finite function (or permutation) family *keyed* if every function in it can be specified (not necessarily uniquely) by a key a. We denote the function given by a as f_a. We assume that given a, it is possible to efficiently evaluate f_a at any point (as well as f_a^{-1} in case of a keyed permutation family). For a given keyed function family, a key can be any string from I_s, where s is known as "key length." (Sometimes it is convenient to have keys from a set other than I_s; we do not consider such function families simply for clarity of exposition—our results do not change in such a case.) For functions f and g, $g \circ f$ denotes the function $x \mapsto g(f(x))$.

2.1.2 Model of Computation

The adversary $\mathcal{A}$ is modeled as a program for a Random Access Machine (RAM) that has black-box access to some number k of oracles, each of which computes some specified function. If $(f_1, \ldots, f_k)$

is a k-tuple of functions, then $\mathcal{A}^{f_1,\ldots,f_k}$ denotes a k-oracle adversary who is given black-box oracle access to each of the functions $f_1,\ldots,f_k$. We define $\mathcal{A}$'s "running time" to be the number of time steps it takes plus the length of its description (to prevent one from embedding arbitrarily large lookup tables in $\mathcal{A}$'s description).

2.1.3 Pseudorandom Functions and Block Ciphers

The pseudorandomness of a keyed function family $\mathcal{F}$ with domain I_k and range I_l captures its computational indistinguishability from $\mathsf{Rand}^{k\to l}$. This definition is a slightly modified version of the one given by Goldreich, Goldwasser and Micali [12].

Definition 1 *A pseudorandom function family $\mathcal{F}$ is a keyed function family with domain I_k, range I_l, and key length s. Let $\mathcal{A}$ be a 1-oracle adversary. Then we define $\mathcal{A}$'s advantage as*

$$\mathsf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(\mathcal{A}) = \left|\Pr[a \overset{R}{\leftarrow} I_s : \mathcal{A}^{f_a} = 1] - \Pr[f \overset{R}{\leftarrow} \mathsf{Rand}^{k\to l} : \mathcal{A}^{f} = 1]\right|.$$

For any integers $q, t \geq 0$, we define an insecurity function $\mathsf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(q,t)$:

$$\mathsf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(q,t) = \max_{\mathcal{A}}\{\mathsf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(\mathcal{A})\}.$$

The above maximum is taken over choices of adversary $\mathcal{A}$ such that:

- *$\mathcal{A}$ makes at most q oracle queries, and*
- *the running time of $\mathcal{A}$, plus the time necessary to select $a \overset{R}{\leftarrow} I_s$ and answer $\mathcal{A}$'s queries, is at most t.*

We are now ready to define a secure block cipher, or what Luby and Rackoff [14] call a *super pseudorandom* permutation. The notion captures the pseudorandomness of a permutation family on I_l in terms of its indistinguishability from Perm^l, where the adversary is given access to both directions of the permutation. In other words, it measures security of a block cipher against chosen plaintext and ciphertext attacks.

Definition 2 *A block cipher $\mathcal{F}$ is a keyed permutation family with domain and range I_l and key length s. Let $\mathcal{A}$ be a 2-oracle adversary. Then we define $\mathcal{A}$'s advantage as*

$$\mathsf{Adv}^{\mathsf{sprp}}_{\mathcal{F}}(\mathcal{A}) = \left|\Pr[a \overset{R}{\leftarrow} I_s : \mathcal{A}^{f_a, f_a^{-1}} = 1] - \Pr[f \overset{R}{\leftarrow} \mathsf{Perm}^l : \mathcal{A}^{f, f^{-1}} = 1]\right|.$$

For any integers $q, t \geq 0$, we define an insecurity function $\mathsf{Adv}^{\mathsf{sprp}}_{F}(q,t)$ similarly to Definition 1.

2.1.4 Hash Functions

Our definitions of hash functions follow those given in [8], [18], [22], [13], [20].

Definition 3 *Let H be a keyed function family with domain I_k, range I_l, and key length s. Let $\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4 \geq 2^{-l}$. H is an ϵ_1-uniform family of hash functions if for all $x \in I_k, z \in I_l$, $\Pr[a \overset{R}{\leftarrow} I_s : h_a(x) = z] \leq \epsilon_1$. H is ϵ_2-XOR-universal if for all $x \neq y \in I_k, z \in I_l$, $\Pr[a \overset{R}{\leftarrow} I_s : h_a(x) \oplus h_a(y) = z] \leq \epsilon_2$. It is ϵ_3-bisymmetric if for all $x, y \in I_k$ (here we allow $x = y$), $z \in I_l$, $\Pr[a_1 \overset{R}{\leftarrow} I_s, a_2 \overset{R}{\leftarrow} I_s : h_{a_1}(x) \oplus h_{a_2}(y) = z] \leq \epsilon_3$. It is ϵ_4-universal if for all $x \neq y \in I_k$, $\Pr[a \overset{R}{\leftarrow} I_s : h_a(x) = h_a(y)] \leq \epsilon_4$.*

We note that in some of the past literature, hash functions are assumed to be uniform by default. We prefer to separate uniformity from other properties.

An example of a family that has all four properties for $\epsilon_1 = \epsilon_2 = \epsilon_3 = \epsilon_4 = 2^{-l}$ is a family keyed by a random $l \times k$ matrix A over $GF(2)$ and a random l-bit vector v, with $h_{A,v}(x) = Ax + v$ [8].

Remark 1 *We will use the phrase "h is a uniform (XOR-universal, bisymmetric, universal) hash function" to mean "h is drawn from a uniform (XOR-universal, bisymmetric, universal) family of hash functions."*

2.2 Constructions of Luby-Rackoff Ciphers

We now define Feistel structures, which are the main tool for constructing pseudorandom permutations on $2n$ bits from functions on n bits.

Definition 4 (Basic Feistel Permutation) *Let f be a mapping from I_n to I_n. Let $x = (x^L, x^R)$ with $x^L, x^R \in I_n$. We denote by $\overline{f}$ the permutation on I_{2n} defined as $\overline{f}(x) = (x^R, x^L \oplus f(x^R))$. Note that it is a permutation because $\overline{f}^{-1}(y) = (y^R \oplus f(y^L), y^L)$.*

Definition 5 (Feistel Network) *If $f_1, \ldots, f_s$ are mappings with domain and range I_n, then we denote by $\Psi(f_1, \ldots, f_s)$ the permutation on I_{2n} defined as $\Psi(f_1, \ldots, f_s) = \overline{f_s} \circ \ldots \circ \overline{f_1}$*

Luby and Rackoff [14] were the first to construct pseudorandom permutations. They did so using four independently-keyed pseudorandom functions. The main theorem in their paper is:

Theorem 1 (Luby-Rackoff) *Let h_1, f_1, f_2, h_2 be independently-keyed functions from a keyed function family $\mathcal{F}$ with domain and range I_n and key space I_s. Let $\mathcal{P}$ be the family of permutations on I_{2n} with key space I_{4s} defined by $\mathcal{P} = \Psi(h_1, f_1, f_2, h_2)$ (the key for an element of $\mathcal{P}$ is simply the concatenation of keys for h_1, f_1, f_2, h_2). Then*

$$\mathsf{Adv}^{\mathsf{sprp}}_{\mathcal{P}}(q,t) \leq \mathsf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(q,t) + \binom{q}{2}\left(2^{-n+1} + 2^{-2n+1}\right).$$

Naor and Reingold [15] optimized the above construction by enabling the use of XOR-universal hash functions in the first and last rounds.

Theorem 2 (Naor-Reingold) *Let f_1 and f_2 be independently-keyed functions from a keyed function family $\mathcal{F}$ with domain and range I_n and key space I_{s_1}. Let h_1, h_2 be ϵ-XOR-universal hash functions, keyed independently of each other and of f_1, f_2, from a keyed function family H with domain and range I_n and key space I_{s_2}. Let $\mathcal{P}$ be the family of permutations on I_{2n} with key space $I_{2s_1+2s_2}$ defined by $p = \Psi(h_1, f_1, f_2, h_2)$. Then*

$$\mathsf{Adv}^{\mathsf{sprp}}_{\mathcal{P}}(q,t) \leq \mathsf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(q,t) + \binom{q}{2}\left(2\epsilon + 2^{-2n+1}\right).$$

Patel, Ramzan, and Sundaram [18], following a suggestion in [15], optimized the construction further by allowing the same pseudorandom function to be used in the middle rounds, thus reducing the key size. This required an additional condition on the hash function.

Theorem 3 (Patel-Ramzan-Sundaram) *Let f be a function from a keyed function family $\mathcal{F}$ with domain and range I_n and key space I_{s_1}. Let h_1, h_2 be ϵ_1-bisymmetric ϵ_2-XOR-universal hash functions, keyed independently of each other and of f, from a keyed function family H with domain and range I_n and key space I_{s_2}. Let $\mathcal{P}$ be the family of permutations on I_{2n} with key space $I_{s_1+2s_2}$ defined by $\mathcal{P} = \Psi(h_1, f, f, h_2)$. Then*

$$\mathsf{Adv}^{\mathsf{sprp}}_{\mathcal{P}}(q,t) \leq \mathsf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(2q,t) + q^2\epsilon_1 + \binom{q}{2}\left(2\epsilon_2 + 2^{-2n+1}\right)$$

3 New Model: Round Security

Having presented the classical definitions and constructions of block ciphers, we are now ready to define the new model of round security. The definitions can be easily extended to other symmetric primitives, such as MACs.

Let $\mathcal{P}, \mathcal{F}^1, \mathcal{F}^2, \ldots, \mathcal{F}^r$ be keyed permutation families, each with domain and range I_l and key length s, such that for any key $a \in I_s$, $p_a = f^r_a \circ \ldots \circ f^1_a$. Then $\mathcal{F}^1, \ldots, \mathcal{F}^r$ is called an *r-round decomposition* for $\mathcal{P}$. For $i \leq j$, denote by $(i \rightarrow j)_a$ the permutation $f^j_a \circ \ldots \circ f^i_a$, and by $(i \leftarrow j)_a$ the permutation $\left(f^j_a \circ \ldots \circ f^i_a\right)^{-1}$. Denote by $i \rightarrow j$ and $i \leftarrow j$ the corresponding keyed function families.

Note that having oracle access to a member of $i \rightarrow j$ means being able to give inputs to round i of the forward direction of a block cipher and view outputs after round j. Likewise, having oracle access to $i \leftarrow j$ corresponds to being able to give inputs to round j of the *reverse* direction of the block cipher and view outputs after round i. Thus, the oracle for $1 \rightarrow r = \mathcal{P}$ corresponds to the oracle for chosen plaintext attack, and the oracle for $1 \leftarrow r$ corresponds to the oracle for chosen ciphertext attack.

We are now ready to define security in this round-based model. This definition closely mimics Definition 2. The difference is that the adversary is allowed oracle access to some subset K of the set $\{i \rightarrow j, i \leftarrow j : 1 \leq i \leq j \leq r\}$, and the insecurity function additionally depends on K.

Definition 6 *Let $\mathcal{P}$ be a block cipher with domain and range I_l, key length s and some r-round decomposition $\mathcal{F}^1, \ldots, \mathcal{F}^r$. Fix some subset $K = \{\pi^1, \ldots, \pi^k\}$ of the set $\{i \rightarrow j, i \leftarrow j : 1 \leq i \leq j \leq r\}$, and let $\mathcal{A}$ be a $k+2$-oracle adversary. Then we define $\mathcal{A}$'s advantage as*

$$\begin{aligned}&\mathsf{Adv}^{\mathsf{sprp}}_{\mathcal{P},\mathcal{F}^1,\ldots,\mathcal{F}^r,K}(\mathcal{A}) = \\ &\qquad \left|\Pr[a \overset{R}{\leftarrow} I_s : \mathcal{A}^{p_a, p_a^{-1}, \pi^1_a, \ldots, \pi^k_a} = 1] - \Pr[p \overset{R}{\leftarrow} \mathsf{Perm}^l, a \overset{R}{\leftarrow} I_s : \mathcal{A}^{p, p^{-1}, \pi^1_a, \ldots, \pi^k_a} = 1]\right|\end{aligned}$$

For any integers $q, t \geq 0$ and set K, we define an insecurity function

$$\mathsf{Adv}^{\mathsf{sprp}}_{\mathcal{P},\mathcal{F}^1,\ldots,\mathcal{F}^r}(q,t,K)$$

similarly to Definition 2.

4 Round Security of Luby-Rackoff Ciphers

Having developed a round security framework for block ciphers, we examine the specific case of a four-round cipher described in Section 2.2. Our goal is to characterize the insecurity function defined above depending on the set K of oracles.

We are able to do so completely, in the following sense. We place every set K in one of two categories: either the insecurity function is unacceptably high, or it is almost as low as in the standard model. That is, we completely characterize the acceptable sets of oracles for the construction to remain secure in our model.

Moreover, we do so for all three ciphers presented in Section 2.2 (although we need to add an ϵ-uniformity condition on the hash functions in the second and third constructions in order for them to remain secure; this is a mild condition, often already achieved by a hash function family). As it turns out, the round security of the three constructions is the same. Specifically, all three ciphers remain secure if the adversary is given access to the second and third rounds. These results suggest, in some sense, that the so-called "whitening" steps, performed in the first and last rounds, require secrecy but only weak randomness, whereas the middle rounds require strong randomness but no secrecy.

We present our results in two parts. First, in Section 4.1, we examine what combinations of oracles make the cipher insecure. Then, in Section 4.2, we show that any other combination leaves it secure.

4.1 Negative Results

In this section we demonstrate which oracles make the cipher insecure. Our negative results are strong, in the sense that they hold regardless of what internal functions h_1, h_2, f_1, f_2 are used. That is, the cipher can be distinguished from a random permutation even if each of these functions is chosen truly at random. Thus, our results hold for all three ciphers presented in Section 2.2.

Theorem 4 *Regardless of how the functions h_1, f_1, f_2, h_2 are chosen from the set of all functions with domain and range I_n, let $P = \Psi(h_1, f_1, f_2, h_2)$. Let t be the time required to compute 17 n-bit XOR operations, a comparison of two n-bit strings, and 9 oracle queries.*[2] *Then*

$$\mathsf{Adv}^{\mathsf{sprp}}_{\mathcal{P},\overline{h_1},\overline{f_1},\overline{f_2},\overline{h_2}}(9, t, K) \geq 1 - 2^{-n},$$

as long as K is not a subset of $\{2 \to 2, 2 \leftarrow 2, 3 \to 3, 3 \leftarrow 3, 2 \to 3, 2 \leftarrow 3\}$. That is, P is insecure as long as the adversary has access to an oracle that includes the first or fourth rounds.

We will prove the theorem by eliminating oracles that allow the adversary to distinguish the cipher from a random permutation. This involves using the attack against a three-round cipher from [14]. The complete proof is given in Appendix A.

4.2 Positive Results

In this section, we prove what is essentially the converse of the results of the previous section. Namely, we show that if K is the set given in Theorem 4, then the cipher is secure. Of course, if K is a subset of it, then the cipher is also secure.

Theorem 5 *Suppose $K \subseteq \{2 \to 2, 2 \leftarrow 2, 3 \to 3, 3 \leftarrow 3, 2 \to 3, 2 \leftarrow 3\}$.*
Let h_1, f_1, f_2, h_2 and P be as in Theorem 1. Then

$$\mathsf{Adv}^{\mathsf{sprp}}_{\mathcal{P},\overline{h_1},\overline{f_1},\overline{f_2},\overline{h_2}}(q, t, K) \leq \mathsf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(q, t) + \binom{q}{2}\left(2^{-n+1} + 2^{-2n+1}\right) + q^2\left(2^{-n-1}\right).$$

[2] The values 17 and 9 can be reduced by more careful counting; it is unclear, however, if there is any reason to expend effort finding the minimal numbers that work.

If h_1, f_1, f_2, h_2 and P are as in Theorem 2, with the additional condition that h_1 and h_2 be ϵ_3-uniform, then

$$\mathsf{Adv}^{\mathsf{sprp}}_{\mathcal{P},\overline{h_1},\overline{f_1},\overline{f_2},\overline{h_2}}(q,t,K) \leq \mathsf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(q,t) + \binom{q}{2}\left(2\epsilon + 2^{-2n+1}\right) + q^2\epsilon_3/2.$$

Finally, if h_1, f, h_2 and P are as in Theorem 3, with the additional condition that h_1 and h_2 be ϵ_3-uniform, then

$$\mathsf{Adv}^{\mathsf{sprp}}_{\mathcal{P},\overline{h_1},\overline{f},\overline{f},\overline{h_2}}(q,t) \leq \mathsf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(2q,t) + q^2(\epsilon_1 + \epsilon_3) + \binom{q}{2}\left(2\epsilon_2 + 2^{-2n+1}\right).$$

We focus our proof on the last part of the theorem. The proofs of other cases are very similar. Our proof technique is a generalization of the techniques of Naor and Reingold [15] designed to deal with the extra queries. Moreover, we analyze concrete, rather than asymptotic, security.

First, in the following simple claim, we reduce the statement to the case when f is a truly random function.

Claim 1 *Suppose*

$$\mathsf{Adv}^{\mathsf{sprp}}_{\mathcal{P},\overline{h_1},\overline{f},\overline{f},\overline{h_2}}(q,t) \leq \delta$$

when f is picked from $\mathsf{Rand}^{n \to n}$, rather than from a pseudorandom family. Then

$$\mathsf{Adv}^{\mathsf{sprp}}_{\mathcal{P},\overline{h_1},\overline{f},\overline{f},\overline{h_2}}(q,t) \leq \delta + \mathsf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(2q,t)$$

when is f picked from $\mathcal{F}$.

Proof. Indeed, suppose $\mathcal{A}$ is an adversary for the block cipher $\mathcal{P}$, with advantage γ. Build an adversary $\mathcal{A}'$ for pseudorandom function family $\mathcal{F}$ as follows: $\mathcal{A}'$ selects at random h_1 and h_2 from a suitable family, and runs $\mathcal{A}$ on the cipher $\Psi(h_1, f, f, h_2)$. In order to answer the queries of $\mathcal{A}$, $\mathcal{A}'$ simply queries f where appropriate and computes the answer according to the Feistel structure. $\mathcal{A}'$ then outputs the same result as $\mathcal{A}$.

Note that $\mathcal{A}$ has advantage at least γ if f is from $\mathcal{F}$, and at most δ for a truly random f. By a standard application of the triangle inequality, $\mathsf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(\mathcal{A}') \geq \gamma - \delta$. ■

We note that access to the oracles of K is equivalent to access to the oracle for f (although one query to $2 \to 3$ or $3 \to 2$ can be simulated by two queries to f). Thus, it suffices to prove the following theorem.

Theorem 6 *Let f be a random function, and let h_1, h_2 be ϵ_1-bisymmetric ϵ_2-XOR-universal ϵ_3-uniform hash functions with domain and range I_n, $\Psi = \Psi(h_1, f, f, h_2)$, and R be a random permutation on I_{2n}. Then, for any 3-oracle adversary $\mathcal{A}$ (we do not restrict the running time of $\mathcal{A}$) that makes at most q_c queries to its first two oracles and at most q_o queries to its third oracle,*

$$\left|\Pr[\mathcal{A}^{\Psi(h_1,f,f,h_2),\Psi^{-1}(h_1,f,f,h_2),f} = 1] - \Pr[A^{R,R^{-1},f} = 1]\right|$$
$$\leq q_c^2\epsilon_1 + 2q_oq_c\epsilon_3 + \binom{q_c}{2}\left(2\epsilon_2 + 2^{-2n+1}\right).$$

The remainder of this section gives the proof of this theorem. To summarize, the first part of the proof focuses on the transcript (a.k.a. the "view") of the adversary, and shows that each possible transcript is about as likely to occur when $\mathcal{A}$ is given Ψ as when $\mathcal{A}$ is given R. The second part uses a probability argument to show that this implies that $\mathcal{A}$ will have a small advantage in distinguishing Ψ from R.

4.2.1 Proof of Theorem 6

To start with, let P denote the permutation oracle (either $\Psi(h_1, f, f, h_2)$ or R) that $\mathcal{A}$ accesses. Let $\mathcal{O}^f$ denote the oracle that computes the function f (note that when $\mathcal{A}$ gets Ψ as its permutation oracle, f is actually used as the round function in the computation of the oracle $P = \Psi$; when $\mathcal{A}$ gets R as its permutation oracle, f is completely independent of $P = R$). The machine $\mathcal{A}$ has two possibilities for queries to the oracle P: $(+, x)$ which asks to obtain the value of $P(x)$, or $(-, y)$ which asks to obtain the value of $P^{-1}(y)$ – where both x and y are in I_{2n}. We call these cipher queries. We define the query-answer pair for the i^{th} cipher query as $\langle x_i, y_i\rangle \in I_{2n} \times I_{2n}$ if $\mathcal{A}$'s query was $(+, x)$ and y is the answer it received from P or its query was $(-, y)$ and x is the answer it received. We assume that $\mathcal{A}$ makes exactly q_c queries and we call the sequence $\{\langle x_1, y_1\rangle, \ldots, \langle x_{q_c}, y_{q_c}\rangle\}_P$ the cipher-transcript of $\mathcal{A}$.

In addition, $\mathcal{A}$ can make queries to $\mathcal{O}^f$. We call these oracle queries. We denote these queries as: $(\mathcal{O}^f, x')$ which asks to obtain $f(x')$. We define the query-answer pair for the i^{th} oracle query as $\langle x'_i, y'_i\rangle \in I_n \times I_n$ if $\mathcal{A}$'s query was $(\mathcal{O}^f, x')$ and the answer it received was y'. We assume that $\mathcal{A}$ makes q_o queries to this oracle. We call the sequence $\{\langle x'_1, y'_1\rangle, \ldots, \langle x'_{q_o}, y'_{q_o}\rangle\}_{\mathcal{O}^f}$ the oracle-transcript of $\mathcal{A}$.

Note that since $\mathcal{A}$ is computationally unbounded, we can make the standard assumption that $\mathcal{A}$ is a deterministic machine. Under this assumption, the exact next query made by $\mathcal{A}$ can be determined by the previous queries and the answers received. We formalize this as follows:

Definition 7 *Let $C_{\mathcal{A}}[\{\langle x_1, y_1\rangle, \ldots, \langle x_i, y_i\rangle\}_P, \{\langle x'_1, y'_1\rangle, \ldots, \langle x'_j, y'_j\rangle\}_{\mathcal{O}^f}]$, where either $i < q_c$ or $j < q_o$, denote the $i + j + 1^{st}$ query $\mathcal{A}$ makes as a function of the first $i + j$ query-answer pairs in $\mathcal{A}$'s cipher and oracle transcripts. Let $C_{\mathcal{A}}[\{\langle x_1, y_1\rangle, \ldots, \langle x_{q_c}, y_{q_c}\rangle\}_P, \{\langle x'_1, y'_1\rangle, \ldots, \langle x'_{q_o}, y'_{q_o}\rangle\}_{\mathcal{O}^f}]$ denote the output $\mathcal{A}$ gives as a function of its cipher and oracle transcripts.*

Definition 8 *Let σ be the pair of sequences*

$$(\{\langle x_1, y_1\rangle, \ldots, \langle x_{q_c}, y_{q_c}\rangle\}_P, \{\langle x'_1, y'_1\rangle, \ldots, \langle x'_{q_o}, y'_{q_o}\rangle\}_{\mathcal{O}^f}),$$

where for $1 \le i \le q_c$ we have that $\langle x_1, y_1\rangle \in I_{2n} \times I_{2n}$, and for $1 \le j \le q_o$, we have that $\langle x', y'\rangle \in I_n$. Then, σ is a consistent $\mathcal{A}$-transcript if for every $1 \le i \le q_c$:

$$C_{\mathcal{A}}[\{\langle x_1, y_1\rangle, \ldots, \langle x_i, y_i\rangle\}_P, \{\langle x'_1, y'_1\rangle, \ldots, \langle x'_j, y'_j\rangle\}_{\mathcal{O}^f}] \in \{(+, x_{i+1}), (-, y_{i+1}), (\mathcal{O}^f, x'_{j+1})\}.$$

We now consider another process for answering $\mathcal{A}$'s cipher queries that will be useful to us.

Definition 9 *The random process $\tilde{R}$ answers the i^{th} cipher query of $\mathcal{A}$ as follows:*

1. *If $\mathcal{A}$'s query is $(+, x_i)$ and for some $1 \le j < i$ the j^{th} query-answer pair is $\langle x_i, y_i\rangle$, then $\tilde{R}$ answers with y_i.*

2. *If $\mathcal{A}$'s query is $(-, y_i)$ and for some $1 \le j < i$ the j^{th} query-answer pair is $\langle x_i, y_i\rangle$, then $\tilde{R}$ answers with x_i.*

3. *If neither of the above happens, then $\tilde{R}$ answers with a uniformly chosen element in I_{2n}.*

Note that $\tilde{R}$'s answers may not be consistent with any function, let alone any permutation. We formalize this concept.

Definition 10 *Let $\sigma = \{\langle x_1, y_1\rangle, \ldots, \langle x_{q_c}, y_{q_c}\rangle\}_P$ be any possible $\mathcal{A}$-cipher transcript. We say that σ is inconsistent if for some $1 \le j < i \le q_c$ the corresponding query-answer pairs satisfy $x_i = x_j$ but $y_i \neq y_j$, or $x_i \neq x_j$ but $y_i = y_j$.*

Note 1 *If $\sigma = (\{\langle x_1, y_1\rangle, \ldots, \langle x_{q_c}, y_{q_c}\rangle\}_P, \{\langle x'_1, y'_1\rangle, \ldots, \langle x'_{q_o}, y'_{q_o}\rangle\}_{\mathcal{O}^f})$ is a possible $\mathcal{A}$-transcript, we assume from now on that if σ is consistent and if $i \neq j$ then $x_i \neq x_j$, $y_i \neq y_j$, and $x'_i \neq x'_j$. This formalizes the concept that $\mathcal{A}$ never repeats a query if it can determine the answer from a previous query-answer pair.*

Fortunately, we can show that the process $\tilde{R}$ often "behaves" exactly like a permutation. It turns out that if $\mathcal{A}$ is given oracle access to either $\tilde{R}$ or R to answer its cipher queries, it will have a negligible advantage in distinguishing between the two. We prove this more formally in proposition 1. Before doing so, we first consider the distributions on the various transcripts seen by $\mathcal{A}$ as a function of the different distributions on answers it can get.

Definition 11 *The random variables $T_\Psi, T_R, T_{\tilde{R}}$ denote the cipher-transcript / oracle transcript pair seen by $\mathcal{A}$ when its cipher queries are answered by Ψ, R, $\tilde{R}$ respectively, and its oracle queries are all answered by $\mathcal{O}^f$.*

Remark 2 *Observe that according to our definitions and assumptions, $\mathcal{A}^{\Psi,\Psi^{-1},f}$ and $C_\mathcal{A}(T_\Psi)$ denote the same random variable. The same is true for $\mathcal{A}^{R,R^{-1},f}$ and $C_\mathcal{A}(T_R)$.*

Proposition 1 $|\Pr_{\tilde{R}}[C_\mathcal{A}(T_{\tilde{R}}) = 1] - \Pr_R[C_\mathcal{A}(T_R) = 1]| \le \binom{q_c}{2} \cdot 2^{-2n}$

Proof.

For any possible and consistent $\mathcal{A}$-transcript σ we have that:

$$\Pr_R[T_R = \sigma] = \frac{(2^{2n} - q_c)!}{2^{2n}!} \cdot 2^{-q_o n} = \Pr_{\tilde{R}}[T_{\tilde{R}} = \sigma \mid T_{\tilde{R}} \text{ is consistent}].$$

Thus T_R and $T_{\tilde{R}}$ have the same distribution conditioned on $T_{\tilde{R}}$ being consistent. We now bound the probability that $T_{\tilde{R}}$ is inconsistent. Recall that $T_{\tilde{R}}$ is inconsistent if there exists an i and j with $1 \le j < i \le q_c$ for which $x_i = x_j$ but $y_i \neq y_j$, or $x_i \neq x_j$ but $y_i = y_j$. For a particular i and j this event happens with probability 2^{-2n}. So,

$$\Pr_{\tilde{R}}[T_{\tilde{R}} \text{ is inconsistent}] \le \binom{q_c}{2} \cdot 2^{-2n}.$$

We complete the proof via a standard argument:

$$\begin{aligned}
&\left|\Pr_{\tilde{R}}[C_M(T_{\tilde{R}}) = 1] - \Pr_R[C_M(T_R) = 1]\right| \\
\le &\left|\Pr_{\tilde{R}}[T_{\tilde{R}} = \sigma \mid T_{\tilde{R}} \text{ is consistent}] - \Pr_R[C_M(T_R) = 1]\right| \cdot \Pr_{\tilde{R}}[T_{\tilde{R}} \text{ is consistent}] \\
+ &\left|\Pr_{\tilde{R}}[T_{\tilde{R}} = \sigma \mid T_{\tilde{R}} \text{ is inconsistent}] - \Pr_R[C_M(T_R) = 1]\right| \cdot \Pr_{\tilde{R}}[T_{\tilde{R}} \text{ is inconsistent}] \\
&\le \Pr_{\tilde{R}}[T_{\tilde{R}} \text{ is inconsistent}] \le \binom{q_c}{2} \cdot 2^{-2n}.
\end{aligned}$$

This completes the proof of the proposition. ■

We now proceed to obtain a bound on the advantage that $\mathcal{A}$ will have in distinguishing between T_Ψ and $T_{\tilde{R}}$. It turns out that T_Ψ and $T_{\tilde{R}}$ are identically distributed unless the same value is input to f on two different occasions (we show this in Lemma 1). This depends *only* on the choice of h_1 and h_2. We call this event "BAD" (in the next definition) and obtain a bound on the probability that it actually occurs (in Proposition 2).

Definition 12 *For every specific pair of functions h_1, h_2 define $BAD(h_1, h_2)$ to be the set of all possible and consistent transcripts*

$$\sigma = (\{\langle x_1, y_1\rangle, \ldots, \langle x_{q_c}, y_{q_c}\rangle\}_P, \{\langle x'_1, y'_1\rangle, \ldots, \langle x'_{q_o}, y'_{q_o}\rangle\}_{\mathcal{O}^f})$$

satisfying at least one of the following events:

- **B1:** *there exists $1 \leq i < j \leq q_c$ such that $h_1(x_i^R) \oplus x_i^L = h_1(x_j^R) \oplus x_j^L$, or*
- **B2:** *there exists $1 \leq i < j \leq q_c$ such that $y_i^R \oplus h_2(y_i^L) = y_j^R \oplus h_2(y_j^L)$, or*
- **B3:** *there exists $1 \leq i, j \leq q_c$ such that $h_1(x_i^R) \oplus x_i^L = y_j^R \oplus h_2(y_j^L)$, or*
- **B4:** *there exists $1 \leq i \leq q_c$, $1 \leq j \leq q_o$ such that $h_1(x_i^R) \oplus x_i^L = x'_j$, or*
- **B5:** *there exists $1 \leq i \leq q_c$, $1 \leq j \leq q_o$ such that $y_i^R \oplus h_2(y_i^L) = x'_j$.*

Proposition 2 *Let h_1, h_2 be ϵ_1-bisymmetric ϵ_2-XOR-universal ϵ_3-uniform hash functions. Then, for any possible and consistent $\mathcal{A} - transcript$ σ, we have that*

$$\Pr_{h_1, h_2}[\sigma \in BAD(h_1, h_2)] \leq q_c^2 \epsilon_1 + 2 q_o q_c \epsilon_3 + \binom{q_c}{2} \cdot 2\epsilon_2$$

Proof. Recall that a transcript $\sigma \in BAD(h_1, h_2)$ if one of the events B_i occur. It is straightforward to determine the individual probabilities of each of these events separately by using the properties of h, and apply the union bound to add up the probabilities for each event. ■

Lemma 1 *Let $\sigma = (\{\langle x_1, y_1\rangle, \ldots, \langle x_{q_c}, y_{q_c}\rangle\}_P, \{\langle x'_1, y'_1\rangle, \ldots, \langle x'_{q_o}, y'_{q_o}\rangle\}_{\mathcal{O}^f})$ be any possible and consistent $M - transcript$, then*

$$\Pr_\Psi[T_\Psi = \sigma | \sigma \notin BAD(h_1, h_2)] = \Pr_{\tilde{R}}[T_{\tilde{R}} = \sigma].$$

Proof. It is not hard to see that $\Pr_{\tilde{R}}[T_{\tilde{R}} = \sigma] = 2^{-(2q_c + q_o)n}$ (see [15] for more details).

Now, fix h_1, h_2 to be such that $\sigma \notin BAD(h_1, h_2)$. We will now compute $\Pr_f[T_\Psi = \sigma]$ (note that the probability is now only over the choice of f). Since σ is a possible $\mathcal{A}$-transcript, it follows that $T_{\Psi(h_1, f, f, h_2)} = \sigma$ iff $y_i = \Psi(h_1, f, f, h_2)(x_i)$ for all $1 \leq i \leq q_c$ and $y'_j = f(x'_j)$ for all $1 \leq j \leq q_o$. If we define

$$\begin{aligned} S_i &= x_i^L \oplus h_1(x_i^R) \\ T_i &= y_i^R \oplus h_2(y_i^L), \end{aligned}$$

then

$$(y_i^L, y_i^R) = \Psi(x_i^L, x_i^R) \Leftrightarrow f(S_i) = T_i \oplus x_i^R \text{ and } f(T_i) = y_i^L \oplus S_i.$$

Now observe that for all $1 \leq i < j \leq q_c$, $S_i \neq S_j$ and $T_i \neq T_j$ (otherwise $\sigma \in BAD(h_1, h_2)$). Similarly, for all $1 < i, j < q_c$, $S_i \neq T_j$. In addition, it follows again from the fact that $\sigma \notin BAD(h_1, h_2)$ that for all $1 \leq i \leq q_c$ and $1 \leq j \leq q_o$, $x_i' \neq S_j$ and $x_i' \neq T_j$. So, if $\sigma \notin BAD(h_1, h_2)$ all the inputs to f are distinct. Since f is a random function, $\Pr_f[T_\Psi = \sigma] = 2^{-(2q_c+q_o)n}$ (The cipher transcript contributes 2^{-2nq_c} and the oracle transcript contributes $2^{-q_o n}$ to the probability).

Thus, for every choice of h_1, h_2 such that $\sigma \notin BAD(h_1, h_2)$, the probability that $T_\Psi = \sigma$ is exactly the same: $2^{-(2q_c+q_o)n}$. Therefore:

$$\Pr_\Psi[T_\Psi = \sigma | \sigma \notin BAD(h_1, h_2)] = 2^{-(2q_c+q_o)n}.$$

which completes the proof of the lemma. ■

The rest of the proof consists of using the above lemma and Propositions 1 and 2 in a probability argument.

Let Γ be the set of all possible and consistent transcripts σ such that $C_\mathcal{A}(\sigma) = 1$. Then

$$\begin{aligned} &\left|\Pr_\Psi[\mathcal{A}^{\Psi,\Psi^{-1},f} = 1] - \Pr_R[\mathcal{A}^{R,R^{-1},f} = 1]\right| \\ &= \left|\Pr_\Psi[C_\mathcal{A}(T_\Psi) = 1] - \Pr_R[C_\mathcal{A}(T_R) = 1]\right| \\ &\leq \left|\Pr_\Psi[C_\mathcal{A}(T_\Psi) = 1] - \Pr_{\tilde{R}}[C_\mathcal{A}(T_{\tilde{R}}) = 1]\right| + \binom{q_c}{2} \cdot 2^{-2n} \end{aligned}$$

The last inequality follows from the previous by proposition 1. Now, let $\mathcal{T}$ denote the set of all possible transcripts (whether or not they are consistent), and let Δ denote the set of all possible inconsistent transcripts σ such that $C_\mathcal{A}(\sigma) = 1$. Notice that $\Gamma \cup \Delta$ contains all the possible transcripts such that $C_\mathcal{A}(\sigma) = 1$, and $\mathcal{T} - (\Gamma \cup \Delta)$ contains all the possible transcripts such that $C_\mathcal{A}(\sigma) = 0$. Then:

$$\begin{aligned} &\left|\Pr_\Psi[C_\mathcal{A}(T_\Psi) = 1] - \Pr_{\tilde{R}}[C_\mathcal{A}(T_{\tilde{R}}) = 1]\right| \\ &= \left|\sum_{\sigma \in \mathcal{T}} \Pr_\Psi[C_\mathcal{A}(\sigma) = 1] \cdot \Pr_\Psi[T_\Psi = \sigma] - \sum_{\sigma \in \mathcal{T}} \Pr_{\tilde{R}}[C_\mathcal{A}(\sigma) = 1] \cdot \Pr_{\tilde{R}}[T_{\tilde{R}} = \sigma]\right| \\ &\leq \left|\sum_{\sigma \in \Gamma} (\Pr_\Psi[T_\Psi = \sigma] - \Pr_{\tilde{R}}[T_{\tilde{R}} = \sigma])\right| + \left|\sum_{\sigma \in \Delta} (\Pr_\Psi[T_\Psi = \sigma] - \Pr_{\tilde{R}}[T_{\tilde{R}} = \sigma])\right| \\ &\leq \left|\sum_{\sigma \in \Gamma} (\Pr_\Psi[T_\Psi = \sigma] - \Pr_{\tilde{R}}[T_{\tilde{R}} = \sigma])\right| + \Pr_{\tilde{R}}[T_{\tilde{R}} \text{ is inconsistent}]. \end{aligned}$$

Recall (from the proof of Proposition 1) that $\Pr_{\tilde{R}}[T_{\tilde{R}} \text{ is inconsistent}] \leq \binom{q_c}{2} \cdot 2^{-2n}$. We now want to bound the first term of the above expression.

$$\begin{aligned} &\left|\sum_{\sigma \in \Gamma} (\Pr_\Psi[T_\Psi = \sigma] - \Pr_{\tilde{R}}[T_{\tilde{R}} = \sigma])\right| \\ &\leq \left|\sum_{\sigma \in \Gamma} (\Pr_\Psi[T_\Psi = \sigma | \sigma \in BAD(h_1, h_2)] - Pr_{\tilde{R}}[T_{\tilde{R}} = \sigma]) \cdot \Pr_\Psi[\sigma \in BAD(h_1, h_2)]\right| \end{aligned}$$

$$+\left|\sum_{\sigma\in\Gamma}(\Pr_{\Psi}[T_{\Psi}=\sigma|\sigma\notin BAD(h_1,h_2)]-\Pr_{\tilde{R}}[T_{\tilde{R}}=\sigma])\cdot\Pr_{\Psi}[\sigma\notin BAD(h_1,h_2)]\right|$$

Now, we can apply Lemma 1 to get that the last term of the above expression is equal to 0. All that remains is to find a bound for the first term:

$$\left|\sum_{\sigma\in\Gamma}(\Pr_{\Psi}[T_{\Psi}=\sigma|\sigma\in BAD(h_1,h_2)]-\Pr_{\tilde{R}}[T_{\tilde{R}}=\sigma])\cdot\Pr_{\Psi}[\sigma\in BAD(h_1,h_2)]\right|$$
$$\le\max_{\sigma}\Pr_{\Psi}[\sigma\in BAD(h_1,h_2)]\times$$
$$\max\left\{\sum_{\sigma\in\Gamma}(\Pr_{\Psi}[T_{\Psi}=\sigma|\sigma\in BAD(h_1,h_2)],\sum_{\sigma\in\Gamma}\Pr_{\tilde{R}}[T_{\tilde{R}}=\sigma])\right\}.$$

Note that the last two sums of probabilities are both between 0 and 1, so the above expression is bounded by $\max_{\sigma}\Pr_{\Psi}[\sigma\in BAD(h_1,h_2)]$, which is, by Proposition 2, bounded by $q_c^2\epsilon_1+2q_oq_c\epsilon_3+\binom{q_c}{2}\cdot 2\epsilon_2$.

Finally, combining the above computations, we get:

$$\left|\Pr_{\Psi}[\mathcal{A}^{\Psi,\Psi^{-1},f}=1]-\Pr_{R}[\mathcal{A}^{R,R^{-1},f}=1]\right|\le q_c^2\epsilon_1+2q_oq_c\epsilon_3+\binom{q_c}{2}(2\epsilon_2+2^{-2n+1}),$$

which completes the proof of Theorem 6. ■

5 Acknowledgments

The authors would like to thank Ron Rivest and Salil Vadhan for helpful discussions, and the anonymous referees for many detailed suggestions. Both authors were supported by National Science Foundation Graduate Research Fellowships. The first author was supported by a grant from Merrill-Lynch, and the second author was supported by a grant from the NTT corporation.

References

[1] M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. In *Advances in Cryptology—CRYPTO '96*. Springer-Verlag.

[2] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Pseudorandom functions revisited: The cascade construction and its concrete security. In *37th Annual Symposium on Foundations of Computer Science*, pages 514–523. IEEE, 1996.

[3] Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of cipher block chaining. In Yvo G. Desmedt, editor, *Advances in Cryptology—CRYPTO '94*, volume 839 of *Lecture Notes in Computer Science*, pages 341–358. Springer-Verlag, 21–25 August 1994.

[4] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *First ACM Conference on Computer and Communications Security*, pages 62–73, Fairfax, 1993.

[5] E. Biham and A. Shamir. *Differential Cryptanalysis of the Data Encryption Standard.* Springer Verlag, 1993. ISBN: 0-387-97930-1, 3-540-97930.

[6] J. Black, S. Halevi, H. Krawczyk, T. Krovetz, and P. Rogaway. UMAC: fast and secure message authentication. In M. Wiener, editor, *Proc. CRYPTO 99*, volume 1666 of *Springer-Verlag*, pages 216–233, August 1999. Full version can be found at: `http://www.cs.ucdavis.edu/~rogaway/umac`.

[7] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. In *Proc. 30th ACM Symp. on Theory of Computing*, 1998.

[8] J. L. Carter and M. N. Wegman. Universal classes of hash functions. *JCSS*, 18(2):143–154, April 1979.

[9] M. Etzel, S. Patel, and Z. Ramzan. Square hash: Fast message authentication via optimized universal hash functions. In *Proc. CRYPTO 99*, Lecture Notes in Computer Science. Springer-Verlag, 1999.

[10] Shimon Even and Yishay Mansour. A construction of a cipher from a single pseudorandom permutation. *Journal of Cryptology*, 10(3):151–162, Summer 1997.

[11] H. Feistel. Cryptography and computer privacy. *Scientific American*, 228(5):15–23, May 1973.

[12] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, October 1984.

[13] H. Krawczyk. LFSR-based hashing and authentication. In *Proc. CRYPTO 94*, Lecture Notes in Computer Science. Springer-Verlag, 1994.

[14] M. Luby and C. Rackoff. How to construct pseudorandom permutations and pseudorandom functions. *SIAM J. Computing*, 17(2):373–386, April 1988.

[15] M. Naor and O. Reingold. On the construction of pseudo-random permutations: Luby-Rackoff revisited. *J. of Cryptology*, 12:29–66, 1999. Preliminary version in: *Proc. STOC 97*.

[16] National Bureau of Standards. FIPS publication 46: Data encryption standard, 1977. Federal Information Processing Standards Publication 46.

[17] National Institute of Standards and Technology. Advanced encryption standard home page. `http://www.nist.gov/aes`.

[18] S. Patel, Z. Ramzan, and G. Sundaram. Towards making Luby-Rackoff ciphers optimal and practical. In *Proc. Fast Software Encryption 99*, Lecture Notes in Computer Science. Springer-Verlag, 1999.

[19] R. Rivest. The MD5 message digest algorithm. IETF RFC-1321, 1992.

[20] P. Rogaway. Bucket hashing and its application to fast message authentication. In *Proc. CRYPTO 95*, Lecture Notes in Computer Science. Springer-Verlag, 1995.

[21] B. Schneier. A self-study course in block cipher cryptanalysis. Available from: `http://www.counterpane.com/self-study.html`, 1998.

[22] Mark N. Wegman and J. Lawrence Carter. New hash functions and their use in authentication and set equality. *JCSS*, 22(3):265–279, June 1981.

A Proof of Theorem 4

First, we note the following fact.

Lemma 2 *If we give the adversary $\mathcal{A}$ a way to compute the values of h_1 on arbitrary inputs, then there exists $\mathcal{A}$ that asks three queries to h_1, two queries to the chosen-plaintext oracle p, and one query to the chosen-ciphertext oracle p^{-1}, performs 8 XOR operations, and has an advantage of $1 - 2^{-n}$.*

Proof. This is so because access to h_1 allows the adversary to "peel off" the first round of the cipher, and then use the attack of [14] against a three-round cipher.

Consider an adversary who performs the following steps:

1. pick three arbitrary n-bit strings L_1, R_1, R_2;
2. query the plaintext oracle on (L_1, R_1) to get (V_1, W_1)
3. query the plaintext oracle on $(L_1 \oplus h_1(R_1) \oplus h_1(R_2), R_2)$ to get (V_2, W_2)
4. query the ciphertext oracle on $(V_2, W_2 \oplus R_1 \oplus R_2)$
5. output 1 if $h_1(R_3) \oplus L_3 = V_1 \oplus V_2 \oplus L_1 \oplus h_1(R_1)$

Recall the the goal of the adversary is to output 1 when given the plaintext and ciphertext oracles for a random permutation with noticeably different probability than when given oracles for the block cipher.

Clearly, if the plaintext and ciphertext oracles are truly random, then the adversary will output 1 with probability 2^{-n}, because V_1 and L_3 are then random and independent of the rest of the terms. However, if the plaintext and ciphertext oracles are for the block cipher, then the adversary would output 1 with probability 1. Here is why.

Let S_i, T_i $(1 \le i \le 3)$ be the intermediate values computed in rounds 1 and 2 of the block cipher for the three queries. Let $L_2 = L_1 \oplus h_1(R_1) \oplus h_1(R_2)$, $V_3 = V_2$ and $W_3 = W_2 \oplus R_1 \oplus R_2$. Note that $S_1 = L_1 \oplus h_1(R_1) = L_2 \oplus h_1(R_2) = S_2$. Then $T_3 = W_3 \oplus h_2(V_3) = W_2 \oplus R_1 \oplus R_2 \oplus h_2(V_2) = T_2 \oplus R_1 \oplus R_2 = f_2(S_2) \oplus R_2 \oplus R_1 \oplus R_2 = f_2(S_1) \oplus R_1 = T_1$. Finally, $h_1(R_3) \oplus L_3 = S_3 = V_3 \oplus f_3(T_3) = V_2 \oplus f_3(T_1) = V_2 \oplus V_1 \oplus S_1 = V_2 \oplus V_1 \oplus L_1 \oplus h_1(R_1)$. ■

Note that this fact can be similarly shown for h_2. The lemma above allows us to easily prove the following result.

Lemma 3 *If K contains at least one of the following oracles: $1 \to 4$, $1 \leftarrow 4$, $2 \to 4$, $2 \leftarrow 4$, $1 \to 3$, $1 \leftarrow 3$, $1 \to 1$, $1 \to 2$, $1 \leftarrow 1$, $1 \leftarrow 2$, $4 \leftarrow 4$, $3 \leftarrow 4$, $4 \to 4$ or $3 \leftarrow 4$, then there exists $\mathcal{A}$ making no more than 9 queries to the oracles and performing no more than 17 XOR operations whose advantage is $1 - 2^{-n}$.*

Proof. If K contains $1 \to 4$ or $1 \to 3$, then $\mathcal{A}$ can input an arbitrary pair (L, R) to either of these and receive (V, W) or (T, V). $\mathcal{A}$ then inputs (L, R) to the chosen plaintext oracle p to receive (V', W'), and checks if $V = V'$.

Similarly for $1 \leftarrow 4$ or $2 \leftarrow 4$.

If K contains $2 \to 4$, then $\mathcal{A}$ can input an arbitrary pair (R, S) to it to receive (V, W). $\mathcal{A}$ then inputs (V, W) to the chosen ciphertext oracle p^{-1} to receive (L, R') and checks if $R = R'$. Similarly for $1 \leftarrow 3$.

If K contains $1 \to 1$ or $1 \to 2$, then $\mathcal{A}$ can input (L, R) and receive, in particular, $S = L \oplus h_1(R)$. $\mathcal{A}$ can then compute $h_1(R) = S \oplus L$, and use the procedure of Lemma 2.

Access to $1 \leftarrow 1$ allows $\mathcal{A}$ to input (R, S) and receive $(L = S \oplus h_1(R), R)$. $\mathcal{A}$ can then compute $h_1(R) = L \oplus S$.

Access to $1 \leftarrow 2$ allows $\mathcal{A}$ to compute $h_1(R)$ as follows:

1. query the $1 \leftarrow 2$ oracle on an arbitrary pair (S_1, T_1) to get (L_1, R_1);
2. let $T_2 = T_1 \oplus R_1 \oplus R$ and $S_2 = S_1$;
3. query the $1 \leftarrow 2$ oracle on (S_2, T_2) to get (L_2, R_2); then $R_2 = T_2 \oplus f_1(S_2) = (T_1 \oplus R_1 \oplus R) \oplus (R_1 \oplus T_1) = R$;
4. compute $h_1(R) = L_2 \oplus S_2$.

Thus, any of the oracles $1 \to 1, 1 \to 2, 1 \leftarrow 1, 1 \leftarrow 2$ gives $\mathcal{A}$ access to h_1 and thus makes the cipher insecure.

Similarly for $4 \leftarrow 4, 3 \leftarrow 4$, $4 \to 4$ and $3 \to 4$. ■

Finally, to prove Theorem 4, note that there are 20 possible oracles. Of those, 14 are ruled out by the above lemma, leaving only 6 possible oracles to choose from.