Luby-Rackoff Ciphers: Why XOR Is Not So Exclusive

Sarvar Patel[1], Zulfikar Ramzan[2]*, and Ganpathy S. Sundaram[1]

[1] Bell Labs, Lucent Technologies
{sarvar, ganeshs}@bell-labs.com
[2] IP Dynamics, Inc.
zramzan@ipdynamics.com

Abstract. This work initiates a study of Luby-Rackoff ciphers when the bitwise exclusive-or (XOR) operation in the underlying Feistel network is replaced by a binary operation in an arbitrary finite group. We obtain various interesting results in this context:

– First, we analyze the security of three-round Feistel ladders over arbitrary groups. We examine various Luby-Rackoff ciphers known to be insecure when XOR is used. In some cases, we can break these ciphers over arbitrary *Abelian groups* and in other cases, however, the security remains an open problem.
– Next, we construct a four round Luby-Rackoff cipher, operating over *finite groups of characteristic greater than 2*, that is not only completely secure against *adaptive* chosen plaintext and ciphertext attacks, but has better time / space complexity and uses fewer random bits than all previously considered Luby-Rackoff ciphers of equivalent security in the literature. Surprisingly, when the group is of characteristic 2 (i.e., the underlying operation on strings is bitwise exclusive-or), the cipher can be completely broken in a constant number of queries.

Notably, for the former set of results dealing with three rounds (where we report no difference) we need new techniques. However for the latter set of results dealing with four rounds (where we prove a new theorem) we rely on a generalization of known techniques albeit requires a new type of hash function family, called a *monosymmetric hash function family*, which we introduce in this work. We also discuss the existence (and construction) of this function family over various groups, and argue the necessity of this family in our construction. Moreover, these functions can be very easily and efficiently implemented on most current microprocessors thereby rendering the four round construction very practical.

1 Introduction

Motivation. Let X be the n-bit string which represents the integer $2^n - 1$, and let Y be the n-bit string representing 1. Observe that, when we replace X

* Work done while this author was at Lucent Technologies and the Massachusetts Institute of Technology. This author would like to acknowledge DARPA grant DABT63-96-C-0018 and an NSF graduate fellowship.

© Springer-Verlag Berlin Heidelberg 2003

by $X + Y$ modulo 2^n we get the "zero" string whereas when we replace X by $X \oplus Y$ we get the string with $n - 1$ leading 1's and a 0 in the least significant position. In the former, all the bits of the orginial string X are affected, and in the latter only the least significant bit is affected. Surprisingly, this naive observation contributes to the complexity of various computational problems.

Consider, for example, the *subset sum problem.* This problem is known to be $\mathcal{NP}$-hard when the inputs are treated as elements in the group of integers modulo 2^n [6], [9]. Yet, when we treat the inputs as elements of $\mathrm{GF}(2^n)^+$ the subset sum problem can be efficiently solved via a system of linear equations.

Yet another example is the problem of factoring a bit string treated as an integer. This problem is believed to be hard and is central to various cryptosystems proposed in the past few decades (RSA and its variants). On the other hand, the same bit string treated as a polynomial in one variable over a binary field can be very easily factored; for example via Berlekamp's algorithm [2].

This idea of examining binary operations between bit strings has also been considered in the context of block ciphers. For example, Biham and Shamir [3] show that replacing some of the exclusive-or operations in DES with additions modulo 2^n, makes their differential attack less powerful. Carter, Dawson, and Nielsen [4] show a similar phenomenon when addition in DES is replaced by addition using a particular Latin Square. While these results show resistance to one particular type of attack, they are more ad-hoc since they do not demonstrate provable security against *all attacks.* Motivated by these examples, we embark on a formal study of the impact of the underlying binary operation on block ciphers. We focus our attention on three and four round *Luby-Rackoff ciphers.*

Luby-Rackoff Ciphers and Variants. Luby and Rackoff [10] formalize the notion of a provably-secure block cipher as a *Pseudorandom Permutation* (PRP) family (indexed by the key). They achieve a construction satisfying this definition which relies on the existence of one-way functions. This seminal piece of work has received a lot of attention, and ciphers based on this principle are now called *Luby-Rackoff ciphers.* A Luby-Rackoff cipher involves the use of a *Feistel permutation* which sends a $2n$-bit string (L, R) to $(R, L \oplus f(R))$ where f is a length-preserving function on n bit strings and $\oplus$ represents the XOR operation on bit strings. The Feistel permutation appears in a number of well-known block cipher constructions such as DES. Luby and Rackoff show that composing three Feistel permutations with independently keyed pseudorandom functions, which we denote as $\Psi(f_1, f_2, f_3)$, yields a cipher secure against chosen plaintext attacks. They also show that composing four Feistel permutations with independent keys, denoted as $\Psi(f_1, f_2, f_3, f_4)$, yields a cipher secure against adaptive chosen plaintext and ciphertext attacks. This path-breaking work, has stimulated much research which can be broadly classified as follows:

– Reducing the number of independent functions in the three or four round construction, and examine which constructions continue to guarantee pseudorandomness or super pseudorandomness, (example, [15], [17]).

- Replacing the outer rounds of the four round construction by various flavors of universal hash functions, and examine which constructions continue to guarantee super pseudorandomness, (example, [11]).
- Increasing the number of rounds to enhance the security guarantees, (example, [12], [13]).

Our work fits in the first two broad areas of research to obtain variants of Luby-Rackoff constructions where the number of different functions used in the 3 or 4 rounds is minimized. The kind of questions asked by this research are, for example: is the three-round variant $\Psi(f, f, f)$ pseudorandom? Alternatively, is the four-round variant $\Psi(f_1, f_2, f_2, f_1)$ strongly pseudorandom?

The answer to these questions are important not just because of the apparent savings of key material achieved by minimizing different pseudorandom functions, but also because of the insight they provide into the design of secure block ciphers. In this context, negative results are also very useful. *For example, knowing that $\Psi(f, f, f)$ is not a pseudorandom permutation for any pseudorandom function f means that block cipher designers should avoid such symmetrical structures independent of what is being used for f.* Also the description of the specific attack which works on $\Psi(f, f, f)$ is useful to the designer because he can check if that attack or similar attacks may also work with many more feistel rounds used in practical block ciphers. Thus the question of security of variants of Luby-Rackoff ciphers is fundamental. Similarly variants of Luby-Rackoff ciphers where the first and last rounds are replaced by universal hash functions, should be considered; for example, is $\Psi(h, f, f, h)$ strongly pseudorandom?

OUR CONTRIBUTIONS. All of the previous constructions rely on the bitwise exclusive-or (XOR) operation. We depart from this paradigm and we hope that utilizing different operations, with different algebraic properties, may yield new more powerful constructions. A more direct motivation is the result in [14], where the authors optimize a Naor-Reingold variant using "non-XOR" operations. Naor and Reingold proved that $\Psi(h_1, f, f, h_2)$ using XOR is super pseudorandom where the universal hash functions h_1 and h_2 work on $2n$ bits. Following this Patel, Ramzan, and Sundaram, [14] optimized this construction by using specialized hash functions and working over addition mod 2^n to make h_1 and h_2 work over n bits. Although, they proved an interesting optimization, their construction did not further minimize the different kinds of functions when compared to the constructions in the Naor-Reingold construction [11]. Both require h in round 1 and 4 to be different. Thus it remained an open question whether there are Luby-Rackoff variants which are secure when a non-XOR operation is used and are insecure when XOR is used.

Our efforts focus on both three-round and four-round Feistel networks that permute $2n$-bits of data as in the original Luby-Rackoff construction and some of its variants discussed above. Our point of departure is that we treat the n-bit strings as elements in an arbitrary algebraic structure which is not necessarily

$\mathrm{GF}(2^n)^+$.[1] This idea seems to open up new lines of research and we obtain results which are both surprising and useful. Specifically:

– We examine various three-round Luby-Rackoff ciphers known to be insecure over $\mathrm{GF}(2^n)^+$. In some cases, we can break these ciphers over arbitrary *Abelian groups* – though we have to employ different attacks. In other cases, however, the security remains an open problem.
– Next we construct a Luby-Rackoff style cipher, whose Feistel ladder operates over various *finite groups* of characteristic greater than 2, that is not only super-pseudorandom, but has better time/space complexity and uses fewer random bits than all previously considered Luby-Rackoff ciphers of equivalent security in the literature. Interestingly, when we use the bit-wise exclusive-or operation instead, we can distinguish the cipher from random with near certainty using only two queries.
– We show that the requirements on our construction are precise when operations are performed in a *finite field.* In particular, eliminating one of the statistical requirements on the hash function used in the first and last round results in the cipher becoming distinguishable from random with near certainty using only a small constant number of queries.

The four-round construction is fairly interesting since one can then construct a Luby-Rackoff cipher that can be broken with two plaintext/ciphertext queries when the bit-wise exclusive-or is the group operation; yet, if one simply changes four of those bit-wise exclusive-or operations to, for example, additions mod 2^n, the cipher becomes completely secure against both adaptive chosen plaintext and ciphertext attacks. Note that both operations can be implemented very efficiently on most current microprocessors, and both are frequently used in popular block ciphers. A more careful analysis shows that we only need to change the exclusive-or operation in the first and last round to addition mod 2^n to achieve security. Thus, in some sense, there are two very simple operations at the heart of the cipher's security. Our construction utilizes the notion of a *monosymmetric* universal hash function, which we intoduce in this work. In particular the monosymmetric property does not really hold in groups of characteristic 2, which explains why our constructions fail to be secure in this case. In addition we discuss the existence and necessity of this hash function family in our construction.

Organization. The next section provides relevant definitions and reviews prior art on Luby-Rackoff ciphers; we focus on results that are relevant to this paper. In sections three and four we analyze three and four-round Luby-Rackoff ciphers over arbitrary groups. Section five discusses monosymmetric hash functions.

2 Definitions and Prior Work

Notation. We let I_n denote the set of bit strings of length n. For a bit string x, if x has even length, then x^L and x^R denote the left and right halves of

[1] Recall that $\mathrm{GF}(2^n)^+$ refers to the additive group attached to the field $\mathrm{GF}(2^n)$. The addition of two elements in this group amounts to computing their bit-wise exclusive-or.

the bits respectively; we sometimes write $x = (x^L, x^R)$ or $x^L \cdot x^R$. If S is a set whose elements can be sampled according to some pre-specified underlying probability distribution, then $x \overset{R}{\leftarrow} S$ denotes the process of picking an element x from S according to this distribution. Unless otherwise specified, the underlying distribution is assumed to be uniform. We let $|x|$ denote its length.

If n is a positive integer, we let $\mathrm{GF}(2^n)$ denote the Galois Field of order 2^n, and we let $\mathrm{GF}(2^n)^+$ denote the additive group attached to the field $\mathrm{GF}(2^n)$. Recall that elements of $\mathrm{GF}(2^n)$ can be represented as bit strings of length n, and that the addition operation on two elements merely amounts to taking their bit-wise exclusive-or.

By a finite function (permutation) family $\mathcal{F}$, we mean a set of functions (permutations) on a fixed domain and range. If k and l are positive integers, then $\mathsf{Rand}^{k \to l}$ denotes the set of all functions going from I_k to I_l. Similarly, Perm^l denotes the set of all permutations on the set I_l. We call a finite function (permutation) family keyed if every function in it can be specified (not necessarily uniquely) by a key a. We denote the function corresponding to a as f_a. For a given keyed function family, a key can be any string from I_s, where s is known as the "key length." While it is possible to consider key spaces other than I_s, we avoid doing so for clarity of exposition. For functions f and g, where the range of f is contained in the domain of g, we let $g \circ f$ denote their functional composition; i.e. $x \mapsto g(f(x))$. If S is a set contained in a universe U, then S^C denotes the set-theoretic complement of S – that is the set of elements of U that are not in S.

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. The adversary $\mathcal{A}$ will have a one-bit output. 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). This convention was used by Bellare, Kilian, and Rogaway [1].

Sometimes we abuse notation by listing an entire function family as the oracle, rather than just a single function. In this case, the oracle is considered to be a function (or some set of functions) chosen from the family, according to some induced probability distribution. That is, we can think of the oracle as a random variable, which denotes a function, and outputs the value of the function on any input queries it receives. For example, $\mathcal{A}^{\mathsf{Rand}^{n \to n}}$ would be used to denote an adversary whose oracle computes a function chosen at random from the set of all functions with domain and range I_n. Similarly, it may be the case that an adversary has access to multiple oracles, all of which are drawn from the same family. For example, if we deal with oracles chosen from the family Perm^n, we could conceive of giving oracle access to the permutations $f, f^{-1} \in \mathsf{Perm}^n$. These kinds of scenarios apply when we talk about attacks on block ciphers. We also remark that oracles need not simply represent deterministic functions.

Instead there could be some degree of randomness in their answers. In this case the oracle's output is determined by the input together with some internal coin tosses. Both deterministic and randomized oracles are used in this paper.

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 [7].

Definition 1. *Let $\mathcal{F}$ be a keyed function family with domain $\mathcal{D}$ and range $\mathcal{R}$. Let $\mathcal{A}$ be a 1-oracle adversary. Then we define $\mathcal{A}$'s advantage in distinguishing between $\mathcal{F}$ and $\mathsf{Rand}^{\mathcal{D} \to \mathcal{R}}$ as $\mathsf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(\mathcal{A}) = \Pr[a \overset{R}{\leftarrow} \mathsf{Keys}(\mathcal{F}) : \mathcal{A}^{f_a} = 1] - \Pr[f \overset{R}{\leftarrow} \mathsf{Rand}^{\mathcal{D} \to \mathcal{R}} : \mathcal{A}^{f} = 1]$. For any integers $q, t \geq 0$, we define an insecurity function $\mathsf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(q, t)$: $\mathsf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(q, t) = \max_{\mathcal{A}}\{\mathsf{Adv}^{\mathsf{prf}}_{\mathcal{F}}(\mathcal{A})\}$, where the maximum is taken over choices of adversary $\mathcal{A}$ such that $\mathcal{A}$ makes at most q queries to its oracle, and the running time of $\mathcal{A}$, plus the time necessary to select the key a, and answer $\mathcal{A}$'s queries, is at most t.*

We are now ready to formally define security for a block cipher. The first notion we consider is that of a pseudorandom permutation. This notion, which is due to Luby and Rackoff [10], 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 the forward diretion of the permutation. In other words, it measures security of a block cipher against adaptive chosen plaintext attacks.

Definition 2. *Let $\mathcal{F}$ be a keyed permutation family with domain and range $\mathcal{D}$. Let $\mathcal{A}$ be a 1-oracle adversary. Then we say that $\mathcal{A}$ is an ϵ pseudorandom permutation distinguisher for $\mathcal{F}$ if $\mathsf{Adv}^{\mathsf{prp}}_{\mathcal{F}}(\mathcal{A}) = \Pr[a \overset{R}{\leftarrow} \mathsf{Keys}(\mathcal{F}) : \mathcal{A}^{f_a} = 1] - \Pr[f \overset{R}{\leftarrow} \mathsf{Perm}^{\mathcal{D}} : \mathcal{A}^{f} = 1] \leq \epsilon$. For any integers $q, t \geq 0$, we define an insecurity function $\mathsf{Adv}^{\mathsf{prp}}_{F}(q, t)$ just like the one in definition 1. We say that $\mathcal{F}$ is a (t, q, ϵ)-secure pseudorandom permutation family if $\mathsf{Adv}^{\mathsf{prp}}_{\mathcal{F}}(q, t) \leq \epsilon$.*

Luby and Rackoff [10] also define the notion of a *super pseudorandom* permutation which 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 thereby measuring the security of a block cipher against adaptive interleaved chosen plaintext and ciphertext attacks.

Definition 3. *Let $\mathcal{F}$ be a keyed permutation family with domain and range $\mathcal{D}$. Let $\mathcal{A}$ be a 2-oracle adversary. Then we define $\mathcal{A}$'s advantage in distinguishing between $\mathcal{F}$ and $\mathsf{Perm}^{\mathcal{D}}$ as $\mathsf{Adv}^{\mathsf{sprp}}_{\mathcal{F}}(\mathcal{A}) = \Pr[a \overset{R}{\leftarrow} \mathsf{Keys}(\mathcal{F}) : \mathcal{A}^{f_a, f_a^{-1}} = 1] - \Pr[f \overset{R}{\leftarrow} \mathsf{Perm}^{\mathcal{D}} : \mathcal{A}^{f, f^{-1}} = 1]$. For any integers $q, t \geq 0$, we define an insecurity function $\mathcal{A}^{\mathsf{sprp}}_{F}(q, t)$ similar to the one in definition 1. We say that $\mathcal{F}$ is a (t, q, ϵ)-secure super pseudorandom permutation family if $\mathsf{Adv}^{\mathsf{sprp}}_{\mathcal{F}}(q, t) \leq \epsilon$.*

Universal Hash Functions. We define various families of universal hash functions, which appear in many of the constructions in this paper. Stinson [16] prepared an excellent note accurately outlining the history and evolution of these function families.

Definition 4. *Let H be a keyed function family with domain $\mathcal{D}$, range $\mathcal{R}$, and key space $\mathcal{K}$. We assume that the elements of the key space $\mathcal{K}$ are picked according to some underlying distribution. We also assume that the elements of $\mathcal{R}$ form a group with additive notation ('+' and '-'). Let $\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4 \geq 1/|\mathcal{R}|$. Then, H is an ϵ_1-uniform family of hash functions if for all $x \in \mathcal{D}, z \in \mathcal{R}$, $\Pr[a \overset{R}{\leftarrow} \mathcal{K} : h_a(x) = z] \leq \epsilon_1$. H is ϵ_2-almost-Δ-universal if for all $x \neq y \in \mathcal{D}, z \in \mathcal{R}$, $\Pr[a \overset{R}{\leftarrow} \mathcal{K} : h_a(x) - h_a(y) = z] \leq \epsilon_2$. H is ϵ_3-monosymmetric if for all $x, y \in \mathcal{D}$ (here we allow $x = y$) and $z \in \mathcal{R}$, $\Pr[a \overset{R}{\leftarrow} \mathcal{K} : h_a(x) + h_a(y) = z] \leq \epsilon_2$. H is ϵ_4-universal if for all $x \neq y \in \mathcal{D}$, $\Pr[a \overset{R}{\leftarrow} \mathcal{K} : h_a(x) = h_a(y)] \leq \epsilon_4$.*

We remark that the notion of monosymmetric hash function families is a novel contribution of this paper. An example of a family that has all four properties for $\epsilon_1 = \cdots = \epsilon_4 = 1/|\mathcal{R}|$ is a family keyed by a random pair $a = (a_1, a_2)$ with $a_1 \in \mathcal{Z}_p^*$, $a_2 \in \mathcal{Z}_p$, and $h_a(x) = a_1 x + a_2 \bmod p$ where p is a prime, $\mathcal{Z}_p^*$ is the multiplicative group of nonzero integers modulo p, and $\mathcal{Z}_p$ is the additive group of integers modulo p. We also remark that we use the phrase "h is a Δ-universal (monosymmetric) hash function" to mean "h is drawn from a Δ-universal (monosymmetric) family of hash functions."

Constructions of Luby-Rackoff Ciphers. We now formally define Feistel ladders which are the main tool for constructing pseudorandom permutations on $2n$-bit strings from length-preserving functions on n-bits strings. Feistel ladders have been used in a large number of popular block cipher such as DES.

Definition 5 (Feistel Ladders). *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 basic Feistel permutation on I_{2n} defined as $\overline{f}(x) = (x^R, x^L \oplus f(x^R))$. Note that it is a permutation because $\overline{f}^{-1}(y) = (y^R \oplus f(y^L), y^L)$, which can be computed if the function f is known. 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 Feistel ladder, which is a permutation on I_{2n} defined by $\Psi(f_1, \ldots, f_s) = \overline{f_s} \circ \cdots \circ \overline{f_1}$.*

Note that this definition (although stated using exclusive-or) easily extends to any binary group operation. We also note that, we will often need to talk about the intermediate stages of the computation in Feistel ladders as plaintext is transformed to ciphertext (and vice-versa). We denote the right halves of the values attained, as each successive basic Feistel permutations is applied, by the letters S, T, V, and W respectively. In addition, we refer to the left half and right halves of the plaintext input to the cipher as L and R respectively. Similarly, we refer to the left and right halves of the ciphertext output as V and W respectively. We can, for example, describe $\Psi(f_1, f_2, f_3, f_4)$ by:

$$\begin{aligned} S &= L \oplus f_1(R); \\ T &= R \oplus f_2(S); \\ V &= S \oplus f_3(T); \\ W &= T \oplus f_4(V). \end{aligned}$$

Observe that a Feistel ladder is invertible since it is simply a composition of basic Feistel permutations. Sometimes we refer to Feistel ladders as *Feistel networks.*

Luby and Rackoff [10] show that three independently-keyed pseudorandom functions in a Feistel ladder yields a pseudorandom permutation, whereas four independently-keyed pseudorandom functions yields a super pseudorandom permutation. Here by *key* to a pseudorandom permutation we mean the concatenation of the individual keys of the underlying pseudorandom functions. The main theorem in their paper is:

Theorem 1 (Luby-Rackoff). *Let $f_1, \ldots, f_4$ be independently-keyed functions from $\mathsf{Rand}^{n \to n}$. Let $\mathcal{B}_1$ be the family of permutations on I_{2n} consisting of permutations of the form $P = \Psi(f_1, f_2, f_3)$. Then $\mathsf{Adv}^{\mathsf{prp}}_{\mathcal{B}}(q,t) \le \binom{q}{2}\left(2^{-n+1} + 2^{-2n}\right)$. Let $\mathcal{B}_2$ be the family of permutations on I_{2n} with key consisting of permutations of the form $P = \Psi(f_1, f_2, f_3, f_4)$. Then $\mathsf{Adv}^{\mathsf{sprp}}_{\mathcal{B}}(q,t) \le \binom{q}{2}\left(2^{-n+1} + 2^{-2n}\right)$.*

We remark that in the original Luby-Rackoff paper, the main theorem statement is written using complexity-theoretic security; we have recast the statement to the concrete security setting. Also, our treatment is in the information-theoretic case, where our functions are chosen from $\mathsf{Rand}^{n \to n}$. It is straightforwad to translate these results to the computational case, where the functions might be chosen from a finite function family $\mathcal{F}$, using a hybrid argument [11]. These same remarks apply for the remaining theorems given in this paper.

Naor and Reingold [11] optimize the above construction by replacing the first and last rounds in the Feistel ladder with strongly universal hash functions. Here the underlying hash function family must consist of permutations to ensure that the cipher is invertible. Their construction is more efficient since universal hash functions only involve specific statistical properties, so can typically be implemented much faster than pseudorandom functions. Also, universal hash function constructions do not require making any cryptographic assumption or conjecture. By reducing the number of pseudorandom function invocations from four to two, Naor and Reingold achieve a significant savings.

Theorem 2 (Naor-Reingold). *Let f_1 and f_2 be independently-keyed functions from $\mathsf{Rand}^{n \to n}$. Let h_1, h_2 be strongly-universal hash functions, keyed independently of each other and of f_1, f_2, from a keyed permutation family H with domain and range I_{2n}. Let $\mathcal{B}$ be the family of permutations on I_{2n} consisting of permutations of the form $P = h_2^{-1} \circ \Psi(f_1, f_2) \circ h_1$. Then $\mathsf{Adv}^{\mathsf{sprp}}_{\mathcal{B}}(q,t) \le \binom{q}{2}\left(2^{-n+1} + 2^{-2n}\right)$.*

Naor and Reingold gave two improvements to their construction. Patel, Ramzan, and Sundaram [14] observed that trying to securely achieve both simultaneously requires different conditions on the universal hash functions:

Theorem 3 (Patel-Ramzan-Sundaram). *Let f be a function from $\mathsf{Rand}^{n \to n}$. Let h_1, h_2 be ϵ_1-bisymmetric ϵ_2-almost Δ-universal hash functions, keyed independently of each other and of f, from a keyed function family H with domain and range I_n. Let $\mathcal{B}$ be the family of permutations on I_{2n} consisting* *of permutations of the form* $P = \Psi(h_1, f, f, h_2)$. *Then* $\mathsf{Adv}^{\mathsf{sprp}}_{\mathcal{B}}(q,t) \leq q^2\epsilon_1 + \binom{q}{2}\left(2\epsilon_2 + 2^{-2n}\right)$.

3 Three-Round Luby-Rackoff Ciphers

Various negative results (for example, [10], [15], and [17]) were known regarding conventional XOR-based three-round Luby-Rackoff ciphers. We explore the validity of these results in the more general context of Feistel networks over Abelian groups. Overall, we reach the same conclusions between XOR-based Feistel constructions and the non-XOR constructions. Notably, for the most part, the techniques we need to obtain the various attacks are subtly different.

3.1 Attacking $\Psi(f_1, f_2, f_1)$

Rueppel [15] shows that $\Psi(f, f, f)$, the three-round Feistel cipher in which all the round functions are identical, is not even pseudorandom when the operation in the Feistel ladder is the bit-wise exclusive-or. The idea behind his attack is that when we use the same function f in all three rounds and addition is performed in a group of characteristic 2, $\Psi(f, f, f)$ has certain involution-like properties. Rueppel left open the problem of generalizing his attack to Feistel ladders that operate over other algebraic structures.

When operations are performed over an arbitrary algebraic structure, the involution-like properties that Rueppel exploits in his original attack no longer seem to hold. It turns out that this cipher is still insecure, but requires a subtly different type of attack. In fact, the new attack is applicable not only to $\Psi(f, f, f)$ over any Abelian group but also applies to $\Psi(f_1, f_2, f_1)$ where f_1 and f_2 are picked independently (and hence applies to the case of $f_1 = f_2 = f$). We present this attack here.

```
1. Query for the encryption of $0 \cdot 0$ and call the result $T_1 \cdot V_1$.
2. Query for the encryption of $0 \cdot T_1$ and call the result $T_2 \cdot V_2$.
3. Query for the encryption of $(V_1 - V_2) \cdot T_2$ and call the result $T_3 \cdot V_3$.
**If** $T_3 == T_1 + T_2$ **then** return that the cipher is not random
**else** return that the cipher is random.
```

This attack allows the adversary to distinguish $\Psi(f_1, f_2, f_1)$ from a random permutation with very high probability. We omit the analysis since it is similar to the analysis for the next attack.

3.2 Attacking $\Psi(f^i, f^j, f^k)$ When k Is a Multiple of $i + j$

Another interesting class of ciphers is $\Psi(f^i, f^j, f^k)$ where f is a pseudorandom function, and f^i represents the i-fold composition of f with itself. Zheng, Matsumoto, and Imai [17] show how to break this class of ciphers for arbitrary i, j, k, when working over $\mathrm{GF}(2^n)^+$. The attack more heavily depends on the involutory properties of $\mathrm{GF}(2^n)^+$. When considering arbitrary finite groups, we know how to break the cipher only when i, j, and k satisfy certain relations; in particular, when k is a multiple of $i + j$. Our attack is much different than the original one. We are unable to produce an attack that works in all cases, nor are we able to prove security for most other cases. We describe the attack.

```
1. Let $\alpha = k/(i + j)$.
2. Query the encryption of $(0, 0)$; call the result $(T_1, V_1)$.
3. Query the encryption of $(0, T_1)$; call the result $(T_2, V_2)$.
4. Initialize $X_1 = 0$, and $X_2 = T_1$.
5. For $l = 3$ to $\alpha + 1$ do
    5a. Set $X_l = T_{l-1} - X_{l-1}$.
    5b. Query the encryption of $(0, X_l)$; call the result $(T_l, V_l)$.
6. Query the encryption of $(T_{\alpha+1} - X_{\alpha+1} - V_1, 0)$; call the result $(T_{\alpha+2}, V_{\alpha+2})$.
7. Query the encryption of $(-T_1, T_{\alpha+2})$; call the result $(T_{\alpha+3}, V_{\alpha+3})$.
If $T_{\alpha+3} = 2 \times T_{\alpha+2}$ then return that the cipher is not random
else return that the cipher is random.
```

The above attack gives the adversary a distinguishing advantage which is exponentially close to one. The analysis follows. It is not hard to see that if the cipher were truly random, then the attack algorithm would return the correct answer with probability extremely close to 1. We now claim that if the cipher is of the form $\Psi(f^i, f^j, f^k)$ with $k = \alpha(i + j)$ then the attack will always output that the cipher is not random. The proof follows the same notation in the attack; in particular we let $X_1 = 0$, and $X_l = T_{l-1} - X_{l-1}$, for all $l > 1$. In other words, X_l is the alternating telescoping sum of all the previous T_i. First, consider what happens when we encrypt $0 \cdot 0$: $S_1 = f^i(0); T_1 = f^{i+j}(0); V_1 = f^i(0) + f^{i+j+k}(0)$.

Next, consider what happens when we encrypt $0 \cdot T_1$: $S_2 = f^i(T_1); T_2 = f^{i+j}(T_1) + T_1; V_2 = f^i(T_1) + f^k(f^{i+j}(T_1) + T_1)$. Thus, $T_2 - T_1 = f^{i+j}(T_1) = f^{2(i+j)}(0)$. In general, when we encrypt $0 \cdot x$: $S = f^i(x); T = f^{i+j}(x) + x; V = S + f^k(f^{i+j}(x) + x)$. Thus, $T - x = f^{i+j}(x)$. Using this observation, when we set $x = X_{\alpha+1}$, we see: $T_{\alpha+1} - X_{\alpha+1} = f^{(i+j)(\alpha+1)}(0) = f^{(i+j)\alpha+(i+j)}(0) = f^{i+j+k}(0)$. Thus $T_{\alpha+1} - X_{\alpha+1} - V_1 = -f^i(0)$. During query $\alpha + 2$ this value is on the left hand side of the encryption, and 0 is on the right hand side. The result of this encryption is: $S_{\alpha+2} = 0; T_{\alpha+2} = f^j(0); V_{\alpha+2} = f^{k+j}(0)$. Finally, when we encrypt $(-T_1, T_{\alpha+2})$, we get: $S_{\alpha+3} = 0; T_{\alpha+3} = f^j(0) + f^j(S_{\alpha+3}); V_{\alpha+3} = f^k(f^j(0) + T_{\alpha+2})$. Which implies that $T_{\alpha+3} = f^j(0) + f^j(0) = T_{\alpha+2} + T_{\alpha+2}$.

3.3 Attacking the Super Pseudorandom Property of $\Psi(f_1, f_2, f_3)$

Luby and Rackoff show that the three-round variant of their cipher $\Psi(f_1, f_2, f_3)$ is pseudorandom, but not super pseudorandom [10]. In particular, one can distinguish the cipher from random with high probability by making two plaintext queries, and one ciphertext query. We generalize their attack to work when the operation in the Feistel ladder is addition over an arbitrary Abelian group G. We stress that the cipher is still pseudorandom over these other algebraic structures – it is just not super pseudorandom. We describe the attack.

1. Choose a random plaintext: $(L_1, R_1) \overset{R}{\leftarrow} G \times G$.
2. Query for the encryption of $L_1 \cdot R_1$ and call the result $T_1 \cdot V_1$.
3. Choose a value L_2 at random: $L_2 \overset{R}{\leftarrow} G$.
4. If $L_2 = L_1$ repeat the above step.
5. Query for the encryption of $L_2 \cdot R_1$ and call the result $T_2 \cdot V_2$.
6. Query for the decryption of $T_2 \cdot (V_2 + L_1 - L_2)$, and call the result $L_3 \cdot R_3$.

If $R_3 = T_2 + R_1 - T_1$ **then** return that the cipher is not super pseudorandom
else return that the cipher is super pseudorandom.

If the cipher is a three-round Luby-Rackoff cipher, then the above test is always correct. On the other hand, if the cipher is a truly random permutation, the above test is wrong with negligibly small probability. We omit the analysis since it is similar to the analysis of the previous attack.

One can see that our attack makes use of the fact that the underlying group is Abelian. We are unable to develop an attack that works for non-Abelian groups, in general. At the same time, we note that in certain cases, there are non-Abelian groups which are still very "commutative." Consider, for example, the dihedral group D_{2n}, which represents the group of symmetries (rotations and flips) of a regular n-gon [8]. In this case, any two rotations with no flips commute. Thus, if two elements are picked uniformly at random from D_{2n}, they commute with probability at least $1/4$. In fact, the probability is higher since other randomly chosen pairs of elements commute; for example, every element commutes with itself and the identity. The above attack thus works for Dihedral groups, though the success probability diminishes by a constant factor.

4 Four-Round Luby Rackoff Ciphers

In this section we construct a four-round Luby-Rackoff cipher that is secure when the underlying operation is addition in certain algebraic structures, but is broken easily when addition is performed in $\mathrm{GF}(2^n)^+$.

We describe our construction. Let G be a group (with binary operation '+'). Let f be a function drawn from a family $\mathcal{F}$, with domain and range G. Let h be drawn from an ϵ_1-monosymmetric ϵ_2-almost Δ-universal family of hash functions. *Then, our construction is $\Psi(h, f, f, h)$ where addition in the underlying Feistel ladder is performed in the group G.* Note that $\Psi(h, f, f, h)$ can be viewed as a permutation on $G \times G$. The security of this construction can be related in a precise manner to the parameters ϵ_1 and ϵ_2, and the pseudorandomness of the function family $\mathcal{F}$. It turns out that if this group G has characteristic 2, then $\epsilon_1 = 1$, and the cipher can easily be broken. On the other hand, for various other groups, the construction is provably secure. We now make some important observations about this construction:

– The hash functions in the first and fourth rounds have the same randomly chosen key. Alternatively, we can replace h by f_2, where f_2 is a pseudorandom function keyed independently of f. Hence we also obtain the new result that $\Psi(f_2, f, f, f_2)$ is strongly pseudorandom as a corollary.

- The pseudorandom functions in the second and third rounds have the *same* randomly chosen key (though this key should be chosen independently from the key for the hash functions).
- If addition is performed over a group of characteristic 2 (e.g. $G = \mathrm{GF}(2^n)^+$), then this cipher has involution-like properties, so it can easily be distinguished from random.

The cipher $\Psi(h, f, f, h)$ is very efficient: only two calls to the *same* pseudorandom function are made, the universal hash functions operate on half the input block, and the *same* universal hash function is used in the first and fourth rounds, which saves additional key material. We cite our main theorem:

Theorem 4. *Let G be a group, and let f be a function chosen from $\mathsf{Rand}^{G \to G}$. Let $h \in H$ be an ϵ_1-monosymmetric ϵ_2-almost Δ-universal hash function over the group G. Let $\mathcal{P}$ be the family of permutations on $G \times G$ consisting of permutations of the form $P = \Psi(h, f, f, h)$. Then:* $\mathsf{Adv}^{\mathsf{sprp}}_{\mathcal{P}}(q, t) \le q^2\epsilon_1 + \binom{q}{2}\left(2\epsilon_2 + |G|^{-2}\right)$.

We remark that although Δ-universal hash functions are traditionally defined over Abelian groups one could easily extend the definition to hold over non-Abelian groups, and our above result would continue to hold. Also, by modifying the appropriate definitions we can model cases in which each individual round involves a possibly different group operation. Using our techniques in this general model we can easily prove, for example, that $\Psi(h, f, f, h)$ is secure if the second and third rounds involve exclusive-or, but the first and fourth perform addition in certain other groups (for example the integers modulo 2^n). This result is surprising since if we simply change two operations by making the first and fourth round use exclusive-or, then the cipher can be distinguished from random using only two queries.

The proof of theorem 4 follows the framework of Naor and Reingold [11]. We start by recasting the original setting into the more general context of arbitrary finite groups. As usual, our adversary $\mathcal{A}$ is modeled as a program for a random access machine that gets black-box access to either a permutation uniformly sampled from $\mathsf{Perm}^{G \times G}$ or one sampled from the set of ciphers $\mathcal{P} = \Psi(h, f, f, h)$. As was done previously, the adversary will have access to two oracles – one for computing each direction of the permutation. We denote a query for the forward direction of the permutation by $(+, x)$. Such a query asks to obtain the value $\mathcal{P}(x)$. Similarly, we denote query in the reverse direction by $(-, y)$. Such a query asks to obtain the value for $\mathcal{P}^{-1}(y)$. Like before, we write $L \cdot R$ to denote the left and right halves of the plaintext respectively, and we write $V \cdot W$ to denote the left and right halves of the ciphertext respectively. In this case, however, L and R are each elements of the group G, and we can think of $L \cdot R$ as an element of $G \times G$. Also, in the following proof we make the standard assumption that the adversary $\mathcal{A}$ is deterministic. Under this assumption, the i^{th} query made by $\mathcal{A}$ can be determined from the first $i - 1$ query-answer pairs in $\mathcal{A}$'s transcript. We consider the notion of a function C which can determine the adversary's next query given the previous queries and answers as well as the notion of which transcripts can possibly be generated.

Definition 6. *Let $C_{\mathcal{A}}[\{\langle x_1, y_1\rangle, \ldots, \langle x_{i-1}, y_{i-1}\rangle\}]$ denote the i^{th} query $\mathcal{A}$ makes as a function of the first $i-1$ query-answer pairs in $\mathcal{A}$'s transcript. The output of $\mathcal{A}$ as a function of its transcript is denoted by $C_{\mathcal{A}}[\{\langle x_1, y_1\rangle, \ldots, \langle x_q, y_q\rangle\}]$.*

Definition 7. *Let σ be a sequence $\{\langle x_1, y_1\rangle, \ldots, \langle x_q, y_q\rangle\}$, where for $1 \leq i \leq q$ we have that $\langle x_1, y_1\rangle \in (G \times G) \times (G \times G)$. Then, σ is a consistent $\mathcal{A}$-transcript if for every $1 \leq i \leq q : C_{\mathcal{A}}[\{\langle x_1, y_1\rangle, \ldots, \langle x_{i-1}, y_{i-1}\rangle\}] \in \{(+, x_i), (-, y_i)\}$.*

We now consider a special random process for $\mathcal{A}$'s queries that will be useful to us. This random process is analogous to the one given in the security proof of the Naor-Reingold construction [11].

Definition 8. *The random process $\tilde{R}$ answers the i^{th} query of $\mathcal{A}$ as follows:*

- *If $\mathcal{A}$'s query is $(+, x_i)$ and for some $1 \leq j < i$ the j^{th} query-answer pair is $\langle x_i, y_i\rangle$, then $\tilde{R}$ answers with y_i. If more than one such query-answer pair exists, we pick the one with the smallest index.*
- *If $\mathcal{A}$'s query is $(-, y_i)$ and for some $1 \leq j < i$ the j^{th} query-answer pair is $\langle x_i, y_i\rangle$, then $\tilde{R}$ answers with x_i. If more than one such query-answer pair exists, we pick the one with the smallest index.*
- *If neither of the above happens, then $\tilde{R}$ answers with a uniformly chosen pair $(g_1, g_2) \in G \times G$.*

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

Definition 9. *Let $\sigma = \langle (x_1, y_1), \ldots, (x_q, y_q)\rangle$ be any possible $\mathcal{A}$-transcript. We say that σ is inconsistent if for some $1 \leq j < i \leq q$ the corresponding query-answer pairs satisfy: $x_i = x_j$ and $y_i \neq y_j$, or $x_i \neq x_j$ and $y_i = y_j$.*

Fortunately, the process $\tilde{R}$ often "behaves" exactly like a permutation over $G \times G$. If $\mathcal{A}$ is given oracle access to either $\tilde{R}$ or $\mathsf{Perm}^{G\times G}$, then it will have a negligible advantage in distinguishing between the two. Naor and Reingold prove this when the group is $\mathrm{GF}(2^n)^+$. We generalize their proof to the case of any algebraic structure. Before proceeding, recall that we denote by the random variables $T_{\mathcal{P}}$, $T_{\mathsf{Perm}^{G\times G}}$, and $T_{\tilde{R}}$ the transcript seen by $\mathcal{A}$ when its oracle queries are answered by $\mathcal{P}$, $\mathsf{Perm}^{G\times G}$, and $\tilde{R}$ respectively.

Proposition 1. *Let $\mathcal{A}$ be a 2-oracle adversary restricted to making a total of at most q queries to its oracles. Then:* $\Pr_{\tilde{R}}[C_{\mathcal{A}}(T_{\tilde{R}}) = 1] - \Pr_{\mathsf{Perm}^{G\times G}}[C_{\mathcal{A}}(T_{\mathsf{Perm}^{G\times G}}) = 1] \leq \binom{q}{2} \cdot |G|^{-2}$.

Proof. First, let Con denote the event that $T_{\tilde{R}}$ is consistent, and let $\neg\mathsf{Con}$ denote the complement. For any possible and consistent $\mathcal{A}$-transcript σ we have that:

$$\Pr_{\mathsf{Perm}^{G\times G}}[T_{\mathsf{Perm}^{G\times G}} = \sigma] = \frac{(|G|-q)!}{|G|!} = \Pr_{\tilde{R}}[T_{\tilde{R}} = \sigma \mid \mathsf{Con}].$$

Thus $T_{\mathsf{Perm}^{G\times G}}$ and $T_{\tilde{R}}$ have the same distribution conditioned on the event Con. We now bound the probability of $\neg\mathsf{Con}$. Recall that $T_{\tilde{R}}$ is inconsistent if there exists an i and j with $1 \le j < i \le q$ for which $x_i = x_j$ and $y_i \neq y_j$, or $x_i \neq x_j$ and $y_i = y_j$. For a particular i and j this event happens with probability $|G|^{-2}$. So, $\Pr_{\tilde{R}}[\neg\mathsf{Con}] \le \binom{q}{2} \cdot |G|^{-2}$. We complete the proof via a standard argument: $\Pr_{\tilde{R}}[C_{\mathcal{A}}(T_{\tilde{R}}) = 1] - \Pr_{\mathsf{Perm}^{G\times G}}[C_{\mathcal{A}}(T_{\mathsf{Perm}^{G\times G}}) = 1] =$

$$\left(\Pr_{\tilde{R}}[T_{\tilde{R}} = \sigma \mid \mathsf{Con}] - \Pr_{\mathsf{Perm}^{G\times G}}[C_{\mathcal{A}}(T_{\mathsf{Perm}^{G\times G}}) = 1]\right) \cdot \Pr_{\tilde{R}}[\mathsf{Con}] +$$
$$\left(\Pr_{\tilde{R}}[T_{\tilde{R}} = \sigma \mid \mathsf{Con}] - \Pr_{\mathsf{Perm}^{G\times G}}[C_{\mathcal{A}}(T_{\mathsf{Perm}^{G\times G}}) = 1]\right) \cdot \Pr_{\tilde{R}}[\neg\mathsf{Con}]$$
$$\le \Pr_{\tilde{R}}[\neg\mathsf{Con}] \le \binom{q}{2} \cdot |G|^{-2}.$$

We now proceed to obtain a bound on the advantage $\mathcal{A}$ will have in distinguishing between $T_{\mathcal{P}}$ and $T_{\tilde{R}}$. It turns out that $T_{\mathcal{P}}$ and $T_{\tilde{R}}$ are identically distributed unless some event depending on the choice of h in $\mathcal{P}$ occurs. We call this event Bad and obtain a bound on the probability that it actually occurs. Intuitively, Bad occurs whenever the internal function f in $\mathcal{P}$ would be evaluated on the exact same point twice for two distinct oracle queries – that is, whenever there is an internal collision. We formalize this concept as follows.

Definition 10. *For every specific monosymmetric ϵ-almost-Δ-universal hash function h, define $\mathsf{Bad}(h)$ to be the set of all possible and consistent $\mathcal{A}$-transcripts $\sigma = \langle (L_1 \cdot R_1, V_1 \cdot W_1), \ldots, (L_q \cdot R_q, V_q \cdot W_q)\rangle$ satisfying:*

- *Event* $\mathsf{B1}$*: there exists $1 \le i < j \le q$ such that $h(R_i) + L_i = h(R_j) + L_j$, or*
- *Event* $\mathsf{B2}$*: there exists $1 \le i < j \le q$ such that $W_i - h(V_i) = W_j - h(V_j)$, or*
- *Event* $\mathsf{B3}$*: there exists $1 \le i, j \le q$ such that $h(R_i) + L_i = W_j - h(V_j)$.*

Proposition 2. *Let $h \in H$ be an ϵ_1-monosymmetric ϵ_2-almost-Δ-universal hash function. Then, for any possible and consistent $\mathcal{A}$-transcript $\sigma = \langle (L_1 \cdot R_1, V_1 \cdot W_1), \ldots, (L_q \cdot R_q, V_q \cdot W_q)\rangle$, we have that* $\Pr_h[\sigma \in \mathsf{Bad}(h)] \le 2\binom{q}{2}\cdot\epsilon_2 + q^2\cdot\epsilon_1$.

Proof. Recall that a transcript $\sigma \in \mathsf{Bad}(h)$ if event $\mathsf{B1}$, event $\mathsf{B2}$, or event $\mathsf{B3}$ occurs. We can determine an upper bound on the individual probabilities of each of these events separately using the fact that they are ϵ_1-monosymmetric, ϵ_2-almost-Δ-universal, and obtain an upper bound on the overall probability by taking the sum. □

The following key lemma for proving theorem 4 shows that the distribution of possible and consistent transcripts generated by $T_{\mathcal{P}}$ given that the bad conditions do not occur is identical to the distribution of possible and consistent transcripts generated by $T_{\tilde{R}}$. This lemma will be useful when we try to determine a bound on the advantage our adversary $\mathcal{A}$ will have when trying to distinguish between these two cases in general.

Lemma 1. *Let σ be any possible and consistent $\mathcal{A}-$transcript, then* $\Pr_{\mathcal{P}}[T_{\mathcal{P}} = \sigma | \sigma \notin \mathsf{Bad}(h)] = \Pr_{\tilde{R}}[T_{\tilde{R}} = \sigma]$.

Proof. First observe that $\Pr_{\tilde{R}}[T_{\tilde{R}} = \sigma] = |G|^{-2q}$. This equality follows since $\tilde{R}$ picks elements from $G \times G$. Thus for a given fixed transcript entry, the probability that $\tilde{R}$ could generate it is $1/|G \times G|$ which equals $|G|^{-2}$. Now, for q consistent transcript entries, $\tilde{R}$ would generate them by picking q elements independently from $G \times G$, which gives us the desired probability of $|G|^{-2q}$.

Since σ is a possible $\mathcal{A}$-transcript, it follows that $T_{\mathcal{P}} = \sigma$ if and only if $V_i \cdot W_i = P(L_i \cdot R_i)$ for all $1 \leq i \leq q$. Next, suppose h is an ϵ_1-monosymmetric ϵ_2-almost-Δ-universal hash function for which $\sigma \notin \mathsf{Bad}(h)$. Now, we know that $L_i \cdot R_i$ and $V_i \cdot W_i$ must satisfy the following series of equations:

$$\begin{aligned} S_i &= L_i + h(R_i); \\ T_i &= R_i + f(S_i); \\ V_i &= S_i + f(T_i); \\ W_i &= T_i + h(V_i). \end{aligned}$$

So, in particular $(V_i, W_i) = P(L_i, R_i) \Leftrightarrow f(S_i) = T_i - R_i$ and $f(T_i) = V_i - S_i$. Now observe that for all $1 \leq i < j \leq q$, $S_i \neq S_j$ and $T_i \neq T_j$ (otherwise $\sigma \in \mathsf{Bad}(h)$). Similarly, for all $1 < i, j < q$, $S_i \neq T_j$. So, if $\sigma \notin \mathsf{Bad}(h)$ all the inputs to f are distinct. Since f is a random function, for every specific choice of h such that $\sigma \notin \mathsf{Bad}(h)$ the probability that $T_{\mathcal{P}} = \sigma$ is exactly $|G|^{-2q}$. Therefore: $\Pr_{\mathcal{P}}[T_{\mathcal{P}} = \sigma | \sigma \notin \mathsf{Bad}(h)] = |G|^{-2q}$. □

To complete the proof we use the above lemma as well as propositions 1 and 2 in a probability argument. Letting Γ be the set of all possible and consistent transcripts σ such that $C_{\mathcal{A}}(\sigma) = 1$:

$$\begin{aligned} &\Pr_{\mathcal{P}}[\mathcal{A}^{\mathcal{P},\mathcal{P}^{-1}} = 1] - \Pr_{R}[\mathcal{A}^{R,R^{-1}} = 1] \\ &= \Pr_{\mathcal{P}}[C_{\mathcal{A}}(T_{\mathcal{P}}) = 1] - \Pr_{R}[C_{\mathcal{A}}(T_R) = 1] \\ &\leq \Pr_{\mathcal{P}}[C_{\mathcal{A}}(T_{\mathcal{P}}) = 1] - \Pr_{\tilde{R}}[C_{\mathcal{A}}(T_{\tilde{R}}) = 1] + \binom{q}{2} \cdot |G|^{-2} \end{aligned}$$

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 $\Pr_{\mathcal{P}}[C_{\mathcal{A}}(T_{\mathcal{P}}) = 1] - \Pr_{\tilde{R}}[C_{\mathcal{A}}(T_{\tilde{R}}) = 1] =$

$$\begin{aligned} &\sum_{\sigma \in \mathcal{T}} \Pr_{\mathcal{P}}[C_{\mathcal{A}}(\sigma) = 1] \cdot \Pr_{\mathcal{P}}[T_{\mathcal{P}} = \sigma] - \sum_{\sigma \in \mathcal{T}} \Pr_{\tilde{R}}[C_{\mathcal{A}}(\sigma) = 1] \cdot \Pr_{\tilde{R}}[T_{\tilde{R}} = \sigma] \\ &= \sum_{\sigma \in \Gamma} (\Pr_{\mathcal{P}}[T_{\mathcal{P}} = \sigma] - \Pr_{\tilde{R}}[T_{\tilde{R}} = \sigma]) + \sum_{\sigma \in \Delta} (\Pr_{\mathcal{P}}[T_{\mathcal{P}} = \sigma] - \Pr_{\tilde{R}}[T_{\tilde{R}} = \sigma]) \\ &\leq \sum_{\sigma \in \Gamma} (\Pr_{\mathcal{P}}[T_{\mathcal{P}} = \sigma] - \Pr_{\tilde{R}}[T_{\tilde{R}} = \sigma]) \end{aligned}$$

The last expression follows from the previous since for any *inconsistent* transcript σ, $\Pr_{\mathcal{P}}[T_{\mathcal{P}} = \sigma] = 0$. To bound the above expression observe that it equals:

$$\sum_{\sigma \in \Gamma} (\Pr_{\mathcal{P}}[T_{\mathcal{P}} = \sigma | \sigma \in \mathsf{Bad}(h)] - Pr_{\tilde{R}}[T_{\tilde{R}} = \sigma]) \cdot \Pr_{\mathcal{P}}[\sigma \in \mathsf{Bad}(h))]$$
$$+ \sum_{\sigma \in \Gamma} (\Pr_{\mathcal{P}}[T_{\mathcal{P}} = \sigma | \sigma \notin \mathsf{Bad}(h)] - \Pr_{\tilde{R}}[T_{\tilde{R}} = \sigma]) \cdot \Pr_{\mathcal{P}}[\sigma \notin \mathsf{Bad}(h)]$$

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:

$$\sum_{\sigma \in \Gamma} (\Pr_{\mathcal{P}}[T_{\mathcal{P}} = \sigma | \sigma \in \mathsf{Bad}(h))] - \Pr_{\tilde{R}}[T_{\tilde{R}} = \sigma]) \cdot \Pr_{\mathcal{P}}[\sigma \in \mathsf{Bad}(h))]$$
$$\leq \max_{\sigma} \Pr_{\mathcal{P}}[\sigma \in \mathsf{Bad}(h)] \times$$
$$\max \left\{ \sum_{\sigma \in \Gamma} (\Pr_{\mathcal{P}}[T_{\mathcal{P}} = \sigma | \sigma \in \mathsf{Bad}(h)], \sum_{\sigma \in \Gamma} \Pr_{\tilde{R}}[T_{\tilde{R}} = \sigma]) \right\}.$$

Note that the last two sums of probabilities are both between 0 and 1, so the above expression is bounded by $\max_{\sigma} \Pr_{\mathcal{P}}[\sigma \in \mathsf{Bad}(h)]$, which is, by Proposition 2, bounded by $2\binom{q}{2} \cdot \epsilon_2 + q^2 \cdot \epsilon_1$. We combine the above computations to complete the proof:

$$\Pr_{\mathcal{P}}[\mathcal{A}^{\mathcal{P},\mathcal{P}^{-1}} = 1] - \Pr_{R}[\mathcal{A}^{R,R^{-1}} = 1] \leq 2\binom{q}{2} \cdot \epsilon_2 + q^2 \cdot \epsilon_1 + \binom{q}{2} \cdot |G|^{-2}.$$

5 Monosymmetric Δ-Universal Hash Functions

The security of the construction in the previous section rests upon the pseudorandomness of the round function f, and the parameters ϵ_1, ϵ_2 associated with the hash function h. This section focuses on constructing monosymmetric universal hash functions and argues that they are necessary for our constructions.

Constructions. We initially demonstrate that over certain algebraic structures, small values of ϵ_1, ϵ_2 are easy to attain. Next, we show that in other groups, the value of ϵ_1 will always be quite large. Our first example concerns the family $\mathsf{Rand}^{G \to G}$, for a group G.

Lemma 2. *Suppose G has no element of order 2. Then, the set of all possible functions $\mathsf{Rand}^{G \to G}$ is $1/|G|$-monosymmetric $1/|G|$-almost-Δ-universal.*

Proof. First we show that $\mathsf{Rand}^{G \to G}$ is $1/|G|$-almost-Δ-universal. Consider three values $x, y, \delta \in G$, with $x \neq y$. The values $f(x)$ and $f(y)$ are uniformly distributed when $f \overset{R}{\leftarrow} \mathsf{Rand}^{G \to G}$. Thus, their difference is uniformly distributed, and takes on any value in the range G with equal probability. Consequently, $\forall x \neq y \in G, \forall \delta \in G : \Pr[f \overset{R}{\leftarrow} \mathsf{Rand}^{G \to G} : f(x) - f(y) = \delta] = 1/|G|$.

Now we examine the monosymmetric property. Again, pick values $x, y, \delta \in G$. There are two cases. If $x \neq y$, then using an argument very similar to the one above, we get: $\forall x \neq y \in G, \forall \delta \in G : \Pr[f \overset{R}{\leftarrow} \mathsf{Rand}^{G \to G} : f(x) + f(y) = \delta] = 1/|G|$. The only remaining case is when $x = y$. Then, $f(x) + f(y) = 2f(x)$ and we are then left with an equation of the form $2f(x) = \delta$. We claim that at most one value of $f(x)$ that satisfies this equation. If this claim were true, then we are done since $f(x)$ is uniformly distributed over the choice of f. We now prove the claim by contradiction. Suppose that there are two values x_1, x_2 such that $2f(x_1) = 2f(x_2) = \delta$, but $f(x_1) \neq f(x_2)$. It follows that $2(f(x_1) - f(x_2)) = 0$. However, $f(x_1) - f(x_2) \in G$, and $f(x_1) - f(x_2) \neq 0$ contradicting the assumption that G does not contain any element of order 2. □

Unfortunately, it is difficult to efficiently sample from $\mathsf{Rand}^{G \to G}$ because it contains $|G|^{|G|}$ elements, which is quite large for our purposes. If one looks at the proof of the above lemma, it is not hard to see that we did not need truly random functions. Instead, strongly universal families of functions suffice since they appear random whenever one considers only two input / output pairs. Therefore:

Corollary 1. *Suppose G has no element of order 2. Then any strongly universal family of functions is $1/|G|$-monosymmetric $1/|G|$-almost-Δ-universal.*

Example. If G is the additive group attached to a finite field F, then one such family of strongly universal hash functions is the linear congruential hash family: $h_{a,b}(x) = ax + b$ where arithmetic is performed in F.

Since finite fields, of characteristic greater than 2 with size p^k exist for any odd prime p and any natural number k, we can construct good monosymmetric Δ-universal hash function families for sets of these size. We now show how to construct reasonably good families of such hash functions for sets of *any* size.

Lemma 3. *Let m be a natural number, and let G be the cyclic group $\mathcal{Z}_m$. Let p be the smallest prime such that $m \leq p$. Let H be any ϵ_1-monosymmetric ϵ_2-almost-Δ-universal hash function family over the additive group attached to the finite field F_p. Consider the family of functions H', which is defined as follows: $H' = \{h'_a : \mathcal{Z}_m \to \mathcal{Z}_m \mid a \in \mathsf{Keys}(H)\}$, where the functions h'_a are defined as: $h'_a(x) = h_a(x) \bmod m$. Here h_a is chosen from H according to key a. We are also using the natural representation of elements in $\mathcal{Z}_m$ and $\mathcal{Z}_p$ whereby we utilize the smallest non-negative integer. Then H' is $4\epsilon_1$-monosymmetric $4\epsilon_2$-almost-Δ-universal over the group $\mathcal{Z}_m$.*

Proof. We first start with Bertrand's postulate, which states that for each integer $m \geq 2$, there is a prime number p with $m \leq p < 2m$. First, we examine the Δ-universal property. Let $x, y, \delta \in \mathcal{Z}_m$ be chosen arbitrarily, with $x \neq y$. Let a be a key such that $h'_a(x) - h'_a(y) = \delta$. Equivalently, $(h_a(x) - h_a(y)) \bmod m = \delta$.

Now, since H operates over the additive group attached to the finite field F_p, it follows that $h_a(x), h_a(y) \in \{0, \ldots, p-1\}$. Combining this fact with the previous observation, we see $h_a(x) - h_a(y) \in \{\delta, \delta + m, \delta - m, \delta + 2m, \delta - 2m, \ldots\}$.

Since the set of possible differences is finite there must be some minimum element which we can denote by l_0. It must be of the form $l_0 = \delta + im$, for some integer i. Consider values of the form $l_r = \delta + (i+r)m$, where $r \geq 4$. Observe that $l_r - l_0 = rm \geq 4m > 2p$. Since there are at most $2p$ values in the range, l_r cannot appear as the difference of h applied to two inputs for $r \geq 4$. Thus, $\Pr[a \overset{R}{\leftarrow} \mathsf{Keys}(H) : h'_a(x) - h'_a(y) = \delta] \leq \Pr[a \overset{R}{\leftarrow} \mathsf{Keys}(H') : h_a(x) - h_a(y) \in \{l_0, l_1, l_2, l_3\}] \leq 4\epsilon_2$ which follows by applying the union bound and observing that h is ϵ_2-almost-Δ-universal. With the same technique we can show that H' is $4\epsilon_1$-monosymmetric. $\square$

In specific cases we can get tighter bounds by exploiting either the algebraic structure of the hash function itself or the relationship between p and m (for example, if $m < p \leq 3m/2$, then we can achieve values of $3\epsilon_1$ and $3\epsilon_2$). For the case $m = 2^n$, these functions are interesting since addition modulo 2^n is easily implemented on most processors. Therefore, our constructions have practical implications. Another very efficient family of monosymmetric ϵ-almost-Δ-universal hash functions for which ϵ is small ($=2/2^n$) is the square hash family [5].

We now consider groups for which no good families of monosymmetric-Δ universal hash functions exist. The most striking example occurs in the additive group of the Galois Field of order 2^n, $\mathrm{GF}(2^n)^+$.

Example. If H is *any* family of functions whose range is $\mathrm{GF}(2^n)^+$ (i.e. characteristic 2), then $\Pr_{h \in H}[h(x) + h(y) = \delta] = 1$ whenever $x = y$ and $\delta = 0$. Thus, it is not possible to get a value of ϵ_1 smaller than 1 for the monosymmetry property.

Another case in which a group may not possess good monosymmetry properties is the *multiplicative* group modulo m, $\mathcal{Z}_m^*$. If we set $x = y$, then the expression $\Pr_{h \in H}[h(x) \cdot h(y) = \delta]$ may be high if δ has many square roots. For example, if the prime factorization of m consists of k distinct odd primes $p_1, \ldots, p_k$ then it follows from the Chinese Remainder Theorem that certain elements of $\mathcal{Z}_m^*$ may have up to 2^k square roots.

Necessity. We give evidence that our minimal-key construction is fairly optimal and that the monosymmetry property is needed. Specifically, we show that $\Psi(h_1, f, f, h_2)$ is not necessarily secure if h_1 and h_2 are independently keyed Δ-universal hash functions that do not satisfy the additional monosymmetry property. Note that in this case, we consider a cipher for which h_1 and h_2 may be different hash functions, so our attack is more general. The attack also works if they are the same hash function. Patel, Ramzan, and Sundaram [14] show that this particular cipher can be broken when operations were performed in $\mathrm{GF}(2^n)^+$; we now extend the result to the case when operations are performed over arbitrary *finite fields* and resort to different techniques to do so. We note that this result is shown only for finite fields. Extending it to hold for arbitrary groups is left as an open problem.

We describe the attack. Suppose that h_1 and h_2 are taken from the linear hash family. That is $h_1(x) = a_1 \cdot x$ and $h_2(x) = a_2 \cdot x$, where multiplication is performed with respect to the underlying finite field. This family is known to be Δ-universal. Pick values α, α' at random (with $\alpha \neq \alpha'$), and obtain both the encryption of $x = \alpha \cdot 0$ and the decryption of $0 \cdot \alpha$. Next, obtain the encryption of $\alpha' \cdot 0$ and the decryption of $0 \cdot \alpha'$. Working through the equations for the encryption of $x = \alpha \cdot 0$: $S_1 = h_1(R_1) + L_1 = h_1(0) + \alpha = 0 + \alpha = \alpha; T_1 = f(S_1) + R_1 = f(\alpha); V_1 = f(T_1) + S_1 = f^2(\alpha) + \alpha; W_1 = h_2(V_1) + T_1 = a_2 \cdot V_1 + f(\alpha)$. Now we work through the equations for *decrypting* $V_2 \cdot W_2 = 0 \cdot \alpha$: $T_2 = W_2 - h_2(V_2) = \alpha - 0 = \alpha; S_2 = V_2 - f(T_2) = 0 - f(\alpha) = -f(\alpha); R_2 = T_2 - f(S_2) = \alpha - f(-f(\alpha)); L_2 = S_2 - h_1(R_2) = -f(\alpha) - a_1 \cdot R_2$. Next, let $A_1 = L_2 + W_1$. Observe that $A_1 = L_2 + W_1 = a_2 \cdot V_1 - a_1 \cdot R_2$.

Now, we repeat the same process as above. In particular, we ask for the encryption of $L_3 \cdot R_3 = \alpha' \cdot 0$ and call the result $V_3 \cdot W_3$. We also ask for the decryption of $V_4 \cdot W_4 = 0 \cdot \alpha'$ and call the result $L_4 \cdot R_4$. Let A_2 denote $L_4 + W_3$. By an argument similar to the one given above, $A_2 = a_2 \cdot V_3 - a_1 \cdot R_4$. We now have a system of equations: $-a_1 \cdot R_2 + a_2 \cdot V_1 = A_1; -a_1 \cdot R_4 + a_2 \cdot V_3 = A_2$.

Since we know $R_2, R_4, V_1, V_3, A_1, A_2$ the only unknowns are a_1, a_2. With high probability, this system of equations has full rank, and we can solve for a_1 and a_2. If $h_1 = h_2 = h$, then $a_1 = a_2 = a$ and the function is an involution that can easily be distinguished from random. The above procedure allows us to compute the keys to the hash functions in the first and fourth rounds. Knowing these keys reduces the problem to distinguishing the two-round Luby-Rackoff cipher $\Psi(f, f)$ from random, which can easily be done in two queries [10].

6 Conclusion

This paper initiated a study of Luby-Rackoff ciphers over arbitrary finite algebraic structures. To our surprise, we discovered that certain Luby-Rackoff cipher constructions are secure when the Feistel operation is taken over particular groups but are insecure when operations are taken with respect to other groups. For example, when we replace bit-wise exclusive-or by addition modulo 2^n we turn an insecure cipher into a provably secure one. Precisely, we prove that $\Psi(h, f, f, h)$ is a super pseudorandom permutation where h is a mono-symmetric hash function and that such hash functions do not apply to XOR-based groups. We also gave attacks on various well-known three-round constructions over general Abelian groups. We proved that:

- $\Psi(f_1, f_2, f_1)$ and consequently $\Psi(f, f, f)$ are not pseudorandom.
- $\Psi(f^i, f^j, f^k)$ is not pseudorandom, when k is a multiple of $i + j$.
- $\Psi(f_1, f_2, f_3)$ is not super pseudorandom.

Our results spawn new areas for research and motivate a need to re-examine the literature on Luby-Rackoff ciphers to determine the extent to which the old results hold when we look at arbitrary finite algebraic structures. More generally, this work opens up the possibility of security advantages with non-XOR operations for almost any cryptographic primitive and not just block ciphers.

References

1. M. Bellare, J. Kilian, and P. 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.
2. E. R. Berlekamp. Factoring polynomials over large finite fields. *Mathematics of Computation*, 24:713–735, 1970.
3. E. Biham and A. Shamir. *Differential Cryptanalysis of the Data Encryption Standard.* Springer Verlag, 1993. ISBN: 0-387-97930-1, 3-540-97930.
4. G. Carter, E. Dawson, and L. Nielsen. DESV: A Latin Square variation of DES. In *Proceeding of Workshop on Selected Areas of Cryptography*, 1995.
5. 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.
6. M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman, 1979.
7. O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, October 1986.
8. I. N. Herstein. *Topics in Algebra.* Blaisdell Publishing Company, 1964.
9. R. Karp. Reducibility among combinatorial problems. in Complexity of Computer Computations, 1972.
10. M. Luby and C. Rackoff. How to construct pseudorandom permutations and pseudorandom functions. *SIAM J. Computing*, 17(2):373–386, April 1988.
11. 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.*
12. J. Patarin. New results on pseudorandom permutation generators based on the DES scheme. In *Proc. CRYPTO 91*, Lecture Notes in Computer Science. Springer-Verlag, 1991.
13. J. Patarin. Improved security bounds for pseudorandom permutations. In *4th ACM Conference on Computer and Communications Security*, pages 140–150, 1997.
14. 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.
15. R. A. Rueppel. On the security of Schnorr's pseudo random generator. In *Proc. EUROCRYPT 89*, Lecture Notes in Computer Science. Springer-Verlag, 1989.
16. D. R. Stinson. Comments on definitions of universal hash families, August 2000. Available from: `http://cacr.math.uwaterloo.ca/~dstinson/`.
17. Y. Zheng, T. Matsumoto, and H. Imai. Impossibility and optimality results on constructing pseudorandom permutations. In *Proc. EUROCRYPT 89*, Lecture Notes in Computer Science. Springer-Verlag, 1989.