

TWEAKABLE BLOCKCIPHERS SECURE AGAINST
GENERIC EXPONENTIAL ATTACKS

A Thesis

Presented to

The Faculty of the Department of Computer Science

The College of William and Mary in Virginia

In Partial Fulfillment

Of the Requirements for the Degree of

Master of Sciences

by

Elizabeth Ann Crump

2007

APPROVAL SHEET

This thesis is submitted in partial fulfillment of
the requirements for the degree of

Master of Science

Elizabeth A. Crump

Approved by the Committee, May 2007

Committee Chair

Assistant Professor Moses Liskov, Computer Science
The College of William & Mary

Assistant Professor Haining Wang, Computer Science
The College of William & Mary

Assistant Professor Qun Li, Computer Science
The College of William & Mary

To my best friend and my parents.

Table of Contents

Acknowledgments	vi
List of Figures	vii
Abstract	viii
1 Introduction	2
1.1 Our Work	5
1.1.1 Our Contributions	6
1.2 Background and Related Work	7
1.2.1 Tweakable Blockciphers	7
1.2.2 Feistel Blockciphers	8
1.2.3 Exponential Adversaries	10
1.2.4 Exponential Security of Feistel Ciphers	11
2 Definitions and Notation	15
2.1 Basic Notation	15

2.1.1	Adversaries	15
2.1.2	Negligible Functions	16
2.2	Definitions	16
2.3	Tweakable Blockcipher Notation	19
3	CPA-Secure Tweakable Blockciphers	23
4	CCA-Secure Tweakable Blockciphers	29
5	Longer Tweaks	32
5.1	General Lemmas	33
5.2	Longer Tweaks with CPA Security	34
5.2.1	Constructing F	35
5.2.1.1	Final Construction of F	41
5.2.2	Proving F 's correctness	41
5.3	Longer Tweaks with CCA security	51
5.4	Minimality of F	52
6	Conclusion	55
	Bibliography	57
	Vita	60

ACKNOWLEDGMENTS

I would like to thank my advisor, Moses Liskov, for his help and guidance that he has patiently given me. I would also like to thank David Goldenberg and Hakan Seyalioglu for sharing the crypto experience with me. Lastly, I would like to thank Adam Schwartz for his hard work pointing out my comma splices and run on sentences.

List of Figures

1.1	A 4-round Feistel blockcipher.	9
2.1	An illustration of Λ_4 ; the locations at which to XOR a tweak of length $ M /2$ for 4-round Feistel blockcipher.	19
3.1	An illustration of $\mathcal{BC}(7, \mathcal{R}_3 + \mathcal{L}_3)$; a seven-round exponentially CPA-secure tweakable Feistel blockcipher.	28
4.1	An illustration of $\mathcal{BC}(10, \mathcal{L}_3 + \mathcal{R}_3 + \mathcal{L}_7 + \mathcal{R}_7)$; a ten-round exponentially CCA-secure tweakable Feistel blockcipher.	31
5.1	An illustration of F which contains six half blocks of tweak ($t = 6$).	54

ABSTRACT

A blockcipher is a triple of algorithms, (G, E, D) where the key generation algorithm G on input 1^k produces a key; the encryption algorithm E takes two inputs, a key and a message, and produces a ciphertext; and the decryption algorithm D reverses the process. A blockcipher is considered secure if it is indistinguishable from a random permutation. A tweakable blockcipher is a blockcipher with an additional input, a tweak. The tweak is not meant to be kept secret, and in fact may be public knowledge, but creates variability within the cipher. A tweakable blockcipher is considered secure if it is indistinguishable from a family of random permutations indexed by the tweak.

Tweakable blockciphers were first formalized by Liskov, Rivest and Wagner, who constructed tweakable blockciphers directly from blockciphers [13]. Crump, Goldenberg, Hohenberger, Liskov, and Seyalioglu showed that tweakable blockciphers can be constructed directly from pseudorandom functions using a Feistel model [7]. Tweakable blockciphers have only been shown to be secure against polynomial-time adversaries, whereas the security of regular blockciphers has been proven against adversaries capable of launching generic attacks with certain specific exponential bounds. We analyze tweakable blockciphers in a comparable model, and present constructions that achieve a level of security equivalent to the best proven level of security blockciphers have attained.

Specifically, we prove that a tweak can be securely added to a seven-round Feistel construction for chosen-plaintext security, and that this construction is round optimal. We also prove that a tweak can be added to a ten-round Feistel construction for chosen-ciphertext security. In addition, we construct tweakable blockciphers that allow for longer tweak lengths; a tweak longer than the minimal size can be thought of as multiple tweaks. We prove that six rounds plus one round per tweak is sufficient for chosen-plaintext security, and eight rounds plus two rounds per tweak is sufficient for chosen-ciphertext security.

TWEAKABLE BLOCKCIPHERS SECURE AGAINST
GENERIC EXPONENTIAL ATTACKS

Chapter 1

Introduction

Symmetric cryptography is a class of cryptography where a shared private key is used for both encryption and decryption. Blockciphers are symmetric ciphers which operate on a fixed-length string of bits, which are known as blocks. Specifically a blockcipher is a function E (the encryption algorithm) which takes a key K , where $K \in \{0, 1\}^k$ and a message M , and produces a ciphertext C , where $M, C \in \{0, 1\}^n$. More formally a blockcipher's signature is:

$$E : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^n.$$

For each key K , E_K is a permutation on the domain $\{0, 1\}^n$. Since E_K is a permutation, for every ciphertext C there is exactly one message M such that $E_K(M) = C$. E_K also has an inverse D_K (the decryption algorithm) such that for every message M , there exist a single ciphertext C such that $D_K(C) = M$. Thus for all messages and for all ciphertexts $D_K(E_K(M)) = M$

and $E_K(D_K(C)) = C$.

Theoretically, a blockcipher is secure if it is indistinguishable from a random permutation. However in practice, a blockcipher security is based on its key size. If the best known attack is better than a brute force attack over the key space, a blockcipher is sometimes considered to be insecure. For instance, some consider SHA-1 to be broken because it is possible to find a collision in better than brute force time, even though the best known attack still requires many operations (a brute force attack requires 2^{80} hash operations while the best known attack requires 2^{63} hash operations [31]).

Blockciphers, by design, only allow us to encrypt messages of size n . The most natural way to encrypt larger messages is to break a message M into m blocks of size n ; thus $M = (M_1, M_2 \dots M_m)$ and $|M| = mn$.¹ To encrypt a message, for all $1 \leq i \leq m$, the ciphertext corresponding to M_i is calculated: $E_K(M_i) = C_i$, where the final ciphertext $C = (C_1, C_2 \dots C_m)$.² However, it is obvious that there are major security concerns with this scheme: plaintext blocks that are equal encrypt to the same ciphertext. Obviously this is undesirable so in order to incorporate variability in the ciphertext we need to use modes of operations.

Modes of operations, such as Cipher Block Chaining (CBC) mode or Cipher

¹If the size of M is not a multiple of n then pad the last block in a deterministic way such that the last block is of size n .

²This scheme is known as Electronic Code Book mode, however it is not very useful in practice because equal blocks encode to the same ciphertexts, which is a major security concern.

Feedback (CFB) mode, allow us to encrypt messages of arbitrary length while adding variability to the ciphertext. In order to allow for randomness, modes use an initialization vector (also known as a nonce) as input, and this initialization vector creates randomization within the ciphertext. This vector is not meant to be kept secret (in most modes) however it is important that the same initialization vector is not reused with the same key.

Tweakable blockciphers are blockciphers with an extra input, a tweak. The purpose of a tweak is to allow for variability within the blockcipher and is not meant to be kept secret. Therefore, by design, a tweakable blockcipher inherently allows for randomness in the ciphertext. Changing the tweak should be an inexpensive operation, ideally only adding trivial cost. Tweakable blockciphers are considered secure if they are indistinguishable from a random permutation family.

Specifically a tweakable blockcipher is a function \tilde{E} which takes a key K , a tweak T and a message M , and produces a ciphertext C , where $K \in \{0, 1\}^k$, $T \in \{0, 1\}^t$ and $M, C \in \{0, 1\}^n$. The decryption algorithm, \tilde{D} takes as input K , T , and C and produces a message M . More formally a tweakable blockcipher's signature is:

$$\tilde{E} : \{0, 1\}^k \times \{0, 1\}^t \times \{0, 1\}^n \rightarrow \{0, 1\}^n.$$

The first blockcipher to allow for an auxiliary input, called the spice, was the Hasty Pudding Cipher created by Rich Schroepel [28]. Another cipher,

the Mercy Cipher created by Paul Crowley [6] also allows for an additional input, called the randomiser, which creates variability within the cipher. Tweakable blockciphers were thereafter formalized by Liskov, Rivest, and Wagner [13] who present two tweakable blockcipher constructions \tilde{E}_K derived from a regular blockcipher E_K .

1.1 Our Work

An important open problem proposed by Liskov, Rivest, and Wagner from Crypto 2002 was to incorporate tweaks into existing blockciphers or to create tweakable blockciphers directly [13]. Crump, Goldenberg, Hohenberger, Liskov, and Seyalioglu addressed their open problem by constructing tweakable blockciphers by XOR-ing a tweak into specific locations in a Feistel stream [7]. They believed that this is the most natural approach for adding tweaks to a Feistel blockcipher because it changes the cipher minimally. By XOR-ing the tweak into the dataflow instead of direct-cryptographically processing the tweak (e.g. hashing the tweak), the cost associated with changing the tweak value is minimal. We also believe that this is the most natural approach to directly construct tweakable blockciphers from Feistel ciphers, and this is the model that we will use in our constructions.

Although tweakable blockciphers have been formally studied since 2002, the security of tweakable blockciphers have not been proven to the same high security

level as regular blockciphers. Blockciphers have been proven to be secure against adversaries able to make generic attacks with specific exponential bounds. Thus, an important open problem is to construct tweakable blockciphers that are as secure as the strongest regular blockciphers.

1.1.1 Our Contributions

In this thesis, we construct tweakable blockciphers directly from Feistel ciphers. We prove that our constructions are secure against “exponential adversaries” – adversaries allowed unlimited computations, but bounded by an exponential number of queries. We prove that seven rounds are sufficient to construct a chosen plaintext secure (CPA) tweakable blockcipher. We then show that ten rounds are sufficient to construct a chosen ciphertext secure (CCA) tweakable blockcipher. We are the first to prove security of tweakable blockciphers against a computationally unbounded adversary allowed $q \ll 2^k$ queries, where k is half the input size. Our results match the best level of security proven for blockciphers [22].

We also explicitly address the problem of incorporating tweaks of arbitrary length into a tweakable blockcipher. This is an important problem because certain applications require different, specific tweak sizes. In many scenarios, it makes sense for the tweak size to be the same size as the input or output. In other applications, such as TAE mode, each tweak is designed to hold a variety of information such that each tweak is unique [13]. It is also attractive

to allow for longer tweaks in order to allow for larger quantities of information, as this was the motivation for Schroepel to allow spice values of 512 bits in the Hasty Pudding Cipher [28]. We prove that six rounds plus one round per tweak is sufficient for CPA security, and eight rounds plus two rounds per tweak is sufficient for CCA security.

1.2 Background and Related Work

In this section we discuss tweakable blockciphers, Feistel blockciphers, and exponential adversaries.

1.2.1 Tweakable Blockciphers

The first blockcipher to allow for an auxiliary input, called the spice, was the Hasty Pudding Cipher created by Rich Schroepel [28]. Another cipher, the Mercy Cipher created by Paul Crowley [6] also allows for an additional input, called the randomiser, which creates variability within the cipher. Tweakable blockciphers were thereafter formalized by Liskov, Rivest, and Wagner [13] who present two tweakable blockcipher constructions \tilde{E}_K from a regular blockcipher E_K :

$$\tilde{E}_K(T, M) = E_K(T \oplus E_K(M))$$

and

$$\tilde{E}_{K,h}(T, m) = E_k(M \oplus h(T)) \oplus h(T)$$

where K is the key, T is the tweak, M is the message, and h is an ϵ -almost 2-xor-universal hash function. All subsequent constructions of tweakable blockcipher have been created in this model, where a tweakable blockcipher is created using a regular blockcipher as a primitive [10], [10], [26], [5].

Tweakable blockciphers are important primitives which have many practical applications. Liskov, Rivest and Wagner show that tweakable blockciphers can be used to implement secure symmetric encryption [13]. Halevi and Rogaway show that tweakable blockciphers have immediate applications to disk encryption, where the tweak is set to the memory address of an encrypted block [10],[11]. Thus two encrypted blocks storing the same data look completely different, even though the decryption of the blocks remains straightforward. Additionally, Rogaway developed XEX mode which creates a tweakable blockcipher using a regular blockcipher [26]. In fact, XTS-AES (AES in XEX mode with ciphertext stealing³) is currently being considered by SISWG (Security in Storage Working Group) for the proposed IEEE disk encryption standard P1619 [30].

Tweakable blockciphers have also been studied in a variety of other contexts including the security against key related attacks [1], the security of tweakable modes [12], [16], efficiency [3], and other general constructions [5].

1.2.2 Feistel Blockciphers

³Ciphertext stealing is a method for using modes of operations for encrypting messages that are not evenly divisible into blocks without expanding the ciphertext.

Since their introduction almost thirty-five years ago, Feistel ciphers [9], also known as Feistel networks, have become the most actively studied class of blockciphers. The formula for the Feistel blockcipher on input $M = (L^0, R^0)$ is:

$$L^{i+1} = R^i$$

$$R^{i+1} = f_{i+1}(R^i) \oplus L^i$$

where the output after n rounds is (L^n, R^n) , and each f_i is a pseudorandom function specified by the key.

In their famous paper, Luby and Rackoff showed that a three-round Feistel construction is CPA secure and a four-round Feistel construction is CCA secure against poly-

mial adversaries [14]. A four-round Feistel cipher is illustrated in Figure 1.1.

Lucks described an optimization for the CPA secure three-round Feistel construction by replacing the first round with a universal hash function [15]. Shortly thereafter, Naor and Reingold provided optimizations for the strongly secure four-round cipher, replacing both the first and last rounds with a more general

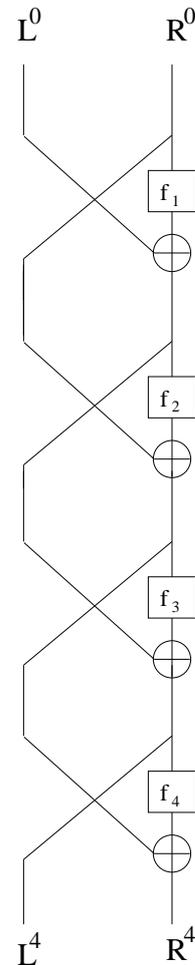


Figure 1.1: A 4-round Feistel blockcipher.

type of function [17]. In 2001, Ramzan [24] formally studied many variations on the Feistel construction. Most recently, Dodis and Puniya presented results about Feistel networks, including a combinatorial understanding of these constructions when the round functions are unpredictable rather than pseudo-random [8].

Many common blockciphers are constructed in a Feistel model and include: DES [18], RC6 [25], Mars [4], Blowfish [27], and Lucifer [29].

1.2.3 Exponential Adversaries

Thus far, the security of tweakable blockciphers have only been proven against polynomial adversaries. An adversary A is defined to be polynomial if A runs in polynomial time. More formally, let P be the set of all polynomial functions and let t be a function that outputs runtime. A is said to be polynomial if $\exists p \in P$ such that $t(A) \leq p(k)$, where k is the security parameter.

However, it is important to prove that tweakable blockciphers are as secure as possible. Therefore security is sometimes proven against a stronger adversary. We define an “exponential adversary”⁴ A' as an adversary that is allowed an unlimited number of computations but is limited only to an exponential number

⁴An adversary A is defined to be exponential if A runs in exponential time. More formally, let E be the set of all exponential functions and let t be a function that outputs runtime. A is said to be exponential if $\exists e \in E$ such that $t(A) \leq e(k)$, where k is the security parameter. However in this thesis, we are using the term *exponential adversary* to describe an adversary that is unbounded in the number of computations but limited to an exponential number of queries, specifically $q \ll 2^{|M|/2}$ queries where $|M|$ is the size of the message.

of oracle queries. Let us assume that the input size (the size of the message) is $2k$, then A' is allowed $q \ll 2^k$ oracle queries.

1.2.4 Exponential Security of Feistel Ciphers

The exponential security of Feistel blockciphers have been formally studied by Jacques Patarin [19], [20], [21], [22], [23]. Much of this thesis is based on Patarin's work where he proved that against exponential adversaries [21]:

- a four-round Feistel construction is secure against known plaintext attacks,
- a seven-round Feistel construction is secure against chosen plaintext attacks,
- and a ten-round construction is secure against chosen ciphertext attacks.⁵

First he proved that four-rounds is KPA secure against an adversary that is allowed an unlimited number of computations but limited to $q \ll 2^k$ queries. Not only did he show that a four-round Feistel cipher is KPA secure, but he also proved that it is secure against an adversary unable to make repeated queries, or unable to more than $O(k)$ queries where the right half collide⁶. More formally, he showed that

⁵Patarin then later showed that a five-round Feistel cipher is both CPA and CCA secure against an exponential adversary, and also proved that this construction is round optimal [22]. However, in this research, we use the proof model of his earlier work.

⁶In this thesis, collide means that two different queries are equal on the right half (or left half), and a full collision implies that two different queries are equal.

Theorem 1.1 (Patarin’s KPA Secure Constructions) *Let E be a four-round Feistel cipher from $2k$ bits to $2k$ bits. Given $q \ll 2^k$ inputs to E and $\nu \in \text{negl}$, if all inputs*

1. *are distinct with probability $1 - \nu(k)$*
2. *and the probability of having $l > O(k)$ queries such that $R_{0_1} = R_{0_2} = R_{0_3} = \dots R_{0_l}$ is less than $\nu(k)$*

then the output of E is indistinguishable from a random for $q \ll 2^k$ input queries.

Using Theorem 1.1 Patarin proved that a seven-round construction is CPA secure by breaking the construction into two constructions, F and E , where F is the first three rounds and E is the last four rounds.

Patarin proved that given $q \ll 2^k$ chosen inputs to F , the outputs of F were distinct with probability $1 - \nu(k)$ where $\nu \in \text{negl}$, and that the probability that the right half did not collide more than $O(k)$ times was also $1 - \nu(k)$. Since the output of F has the needed properties enumerated in Theorem 1.1 then $E \circ F$ is chosen ciphertext secure. More formally, he proved the following theorem:

Theorem 1.2 (Patarin’s CPA Secure Constructions) *Let F be a function from $2k$ bits to $2k$ bits. If F has the property that for $q \ll 2^k$ queries, the probability of having $l > O(k)$ indices such that $R_{i_1} = R_{i_2} = R_{i_3} = \dots R_{i_l}$ is $\nu(k)$ where $\nu \in \text{negl}$, (where R_{i_j} is the right half of the j 'th output of F), and on distinct inputs to F , a full collision occurs with probability less than $\nu(k)$,*

then $E \circ F$ (where E is a four-round Feistel function) is indistinguishable from random for $q \ll 2^k$ input queries.

In order to prove that a ten-round Feistel construction is CCA secure, Patarin divided the ten-round construction into three different constructions, F , E and F' where F is the first three rounds, E is the middle four rounds, and F' is the last three rounds. He showed that ten-rounds was CCA secure using by proving that the outputs of F and F'^{-1} (since F' is used for chosen ciphertext queries we must consider the inverse of F') are distinct with probability $1 - \nu(k)$ where $\nu \in \text{negl}$, and that the probability that the right half of the outputs do not collide more than $O(k)$ times is also $1 - \nu(k)$. More formally,

Theorem 1.3 (Patarin's CCA Secure Constructions) *Let F and F' be functions from $2k$ bits to $2k$ bits. If F and F'^{-1} each have the property that for $q \ll 2^k$ queries, the probability of having $l > O(k)$ indices such that $R_{i_1} = R_{i_2} = R_{i_3} = \dots R_{i_l}$ is $\nu(k)$ where $\nu \in \text{negl}$, (where R_{i_j} is the right half of the j 'th output of F or F'^{-1}), and on distinct inputs to F (and F'^{-1}), a full collision occurs with probability less than $\nu(k)$, then $F' \circ E \circ F$, (where E is a four-round Feistel function), is indistinguishable from random against chosen-ciphertext attack for $q \ll 2^k$ queries.*

Using Theorem 1.1 and subsequently Theorem 1.2 and Theorem 1.3 we are able to prove that our seven-round tweakable blockcipher construction is CPA-secure, our ten-round tweakable blockcipher construction is CCA-secure. We are

also able to use these theorems to extend our tweakable blockcipher constructions to allow for longer tweaks.

When proving the security of specific constructions against generic attacks it is a standard assumption to treat pseudorandom permutations or functions as random ones [2], [3]. Specifically when proving the security against general attacks, the inner primitives are treated as a blackbox and the constructions are proven secure assuming that the inner primitives are secure. When proving the security of his Feistel constructions against exponential adversaries, Patarin assumed that the inner round functions were random instead of pseudorandom ones [19], [20], [21], [22], [23]. In our tweakable Feistel constructions, we also assume that the inner round functions are actually random.

Chapter 2

Definitions and Notation

In this section we discuss important definitions, and establish notation which are used throughout this thesis.

2.1 Basic Notation

2.1.1 Adversaries

All adversaries in this thesis are assumed to be an exponential adversary unless otherwise explicitly stated. An adversary A that is given access to an oracle \mathcal{O} is written as $A^{\mathcal{O}}$; an adversary given access to n oracles, $\mathcal{O}_1, \mathcal{O}_2, \dots, \mathcal{O}_n$, is written as $A^{\mathcal{O}_1, \mathcal{O}_2, \dots, \mathcal{O}_n}$.

2.1.2 Negligible Functions

The expression $\nu(k)$ where $\nu \in \text{negl}$ is used to denote a function ν that is negligible in k , i.e. for any positive polynomial p and sufficiently large k , $\nu(k) < o(1/p(k))$.

The expression $q \ll 2^k$ is equivalent to $\frac{q}{2^k} < \nu(k)$ where $\nu \in \text{negl}$.

2.2 Definitions

A *tweakable blockcipher* is a triple of algorithms $(\tilde{G}, \tilde{E}, \tilde{D})$ for key generation, encryption, and decryption, respectively. We restrict our attention to tweakable blockciphers where $\tilde{G}(\cdot)$, $\tilde{E}_K(\cdot, \cdot)$, and $\tilde{D}_K(\cdot, \cdot)$ are all efficiently computable algorithms and where the correctness property holds; that is, for all messages M , all tweaks T , and for all keys $K \in \tilde{G}(1^k)$, $\tilde{D}_K(\tilde{E}_K(M, T), T) = M$. We also generally assume that $\tilde{G}(1^k)$ draws keys uniformly at random from $\{0, 1\}^{p(k)}$ for some polynomial p .

In this thesis, we assume that the size of the tweak is half the size of the message unless explicitly stated otherwise.

Security is defined in terms of an exponential adversary, where an exponential adversary is allowed an unlimited number of computations, but is bounded by an exponential number of queries. More formally,

Definition 2.1 *An adversary A is exponential if it is allowed an unlimited number of computations but bounded by q queries, where for all messages M ,*

$|M| = 2k$ and $q \ll 2^k$.

The security of tweakable blockciphers can be defined over the maximum advantage that an adversary can obtain with access to an unknown oracle which returns encryption queries from either a random permutation family or a tweakable blockcipher; we define this advantage as ADV-TBC_K .

Definition 2.2 *Over all adversaries with access to an encryption oracle, the maximum advantage is defined as:*

$$\text{ADV-TBC}_K(\tilde{E}, \tilde{D}, q, t) = \max_{\mathcal{A}} : |\Pr[\mathcal{A}^{\tilde{E}_K(\cdot, \cdot)}(1^k) = 1] - \Pr[\mathcal{A}^{\Pi(\cdot, \cdot)}(1^k) = 1]|$$

where (1) for all k , K is generated by $\tilde{G}(1^k)$, (2) $\Pi(\cdot, \cdot)$ is a random permutation family parameterized by its second input, and (3) \mathcal{A} is allowed to run for t steps and make at most q oracle queries.

Stronger security of tweakable blockciphers can also be defined over the maximum advantage that an adversary can obtain with access to an unknown oracle which returns encryption and decryption queries from either a random permutation family or a tweakable blockcipher; we define this advantage as ADV-STBC_K .

Definition 2.3 *Over all adversaries with access to an encryption and decryption oracle, the maximum advantage is defined as:*

$$\text{ADV-STBC}_K(\tilde{E}, \tilde{D}, q, t) = \max_{\mathcal{A}} : |\Pr[\mathcal{A}^{\tilde{E}_K(\cdot, \cdot), \tilde{D}_K(\cdot, \cdot)}(1^k) = 1] - \Pr[\mathcal{A}^{\Pi(\cdot, \cdot), \Pi^{-1}(\cdot, \cdot)}(1^k) = 1]|$$

where (1) for all k , K is generated by $\tilde{G}(1^k)$, (2) Π, Π^{-1} is a random permutation family and its inverse, and (3) \mathcal{A} is allowed to run for t steps and make at most q oracle queries.

We have two notions of security: (1) chosen-plaintext secure (CPA) and (2) chosen-ciphertext secure (CCA). Both levels of security are defined in terms of an exponential adversary. We first define CPA security:

Definition 2.4 *A tweakable blockcipher is CPA exponentially secure if for all k and t , $\text{ADV-TBC}_K(\tilde{E}, \tilde{D}, q, t)$ is negligible in k , where q is some exponential function of k . More formally, a tweakable blockcipher is CPA secure if $\forall k, \forall t, \forall q \ll 2^k, \text{ADV-TBC}_K(\tilde{E}, \tilde{D}, q, t) < \nu(k)$ where $\nu \in \text{negl}$.*

We define CCA security in the same manner.

Definition 2.5 *A tweakable blockcipher is CCA exponentially secure if for all k and t , $\text{ADV-STBC}_K(\tilde{E}, \tilde{D}, q, t)$ is negligible in k , where q is some exponential function of k . More formally, a tweakable blockcipher is CCA secure if $\forall k, \forall t, \forall q \ll 2^k, \text{ADV-STBC}_K(\tilde{E}, \tilde{D}, q, t) < \nu(k)$ where $\nu \in \text{negl}$.*

2.3 Tweakable Blockcipher Notation

Let us establish some notation so that we can

discuss how to add tweaks to a Feistel cipher.

We use the same notation used by Crump et

al. [7]. Let

- n be the number of random functions used in Feistel cipher, also referred to as the number of rounds.
- M be a message where $M = (L^0, R^0)$ such that L^0 is the left half of the input, and R^0 is the right half. The size of M is $2k$; in other words $|M| = 2k$, while the size of both L^0 and R^0 is k .
- C be the ciphertext where $C = (L^n, R^n)$.

Unless otherwise specified, each tweak we refer to are a *half-block* in length; that is, on input M of size $2k$, the tweak is of size k . As we will later see, a blockcipher may allow for longer tweaks.

We can think of these as “multiple tweaks,” as conceptually, the longer tweak can be thought of as being composed of multiple tweaks, each of the same size.

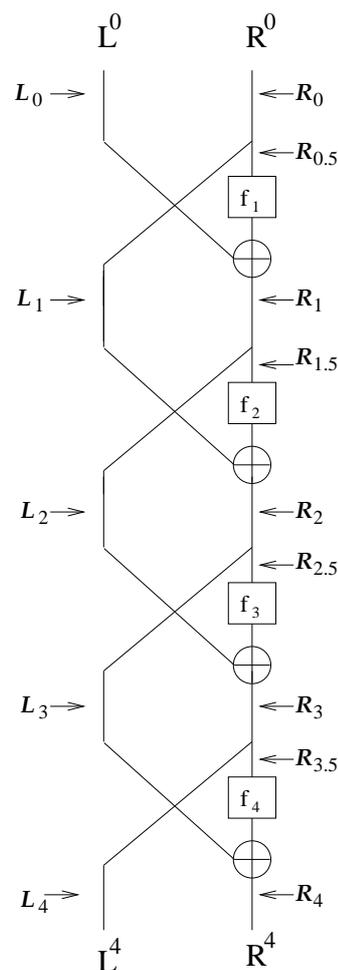


Figure 2.1: An illustration of Λ_4 ; the locations at which to XOR a tweak of length $|M|/2$ for 4-round Feistel blockcipher.

For an n -round Feistel cipher, a tweak can conceivably be XOR-ed in at any of the following unique locations: $\mathcal{L}_0, \mathcal{L}_1, \dots, \mathcal{L}_n, \mathcal{R}_0, \mathcal{R}_{0.5}, \mathcal{R}_1, \dots, \mathcal{R}_{n-0.5}, \mathcal{R}_n$. Let this set of all of these locations be denoted by Λ_n . We illustrate Λ_3 (the set of all possible tweak locations in 3-rounds) in Figure 2.1.

Let T^λ be the XOR of all the tweaks used at location $\lambda \in \Lambda_n$. The formula for our construction is:

$$\begin{aligned} L^{i+1} &= R^i \oplus T^{\mathcal{R}_i} \\ R^{i+1} &= f_{i+1}(R^i \oplus T^{\mathcal{R}_i} \oplus T^{\mathcal{R}_{i+0.5}}) \oplus L^i \oplus T^{\mathcal{L}_i} \end{aligned}$$

Let “ $\mathcal{BC}(n, \lambda)$ ” refer to the tweakable blockcipher construction where the number of Feistel rounds is n and a tweak T^λ is XOR-ed in at some location $\lambda \in \Lambda_n$. To denote adding t different tweaks, we write “ $\mathcal{BC}(n, \lambda_1, \dots, \lambda_t)$ ”, where $T^{\lambda_i} = T_i$ is the tweak for location λ_i and different locations each have their own independent tweak. Thus, in such a construction, the tweak size is tk .

We might also want to denote adding the *same* tweak value at multiple locations. We write this as “ $\mathcal{BC}(n, \lambda_1 + \lambda_2)$ ”, where the implication of using the *compound* location $\lambda_1 + \lambda_2$ is that $T^{\lambda_1} = T^{\lambda_2}$. Of course, we may also consider a construction with multiple tweaks, each of which may be a compound location; we use the obvious notation for this. We use the symbol Γ to denote a (possibly) compound tweak location.

In Λ_n , we have listed all tweaks at “.5” locations, that is $\mathcal{R}_{l+0.5}$ for some l . However, as Crump et al. proved, we do not have to consider these locations [7].

Lemma 2.1 *For all m , $\mathcal{R}_{m+0.5}$ is equivalent to $\mathcal{R}_m + \mathcal{L}_{m+1}$.*

Proof:

Suppose we start with inputs L^m and R^m . From the formulas,

$$\begin{aligned} L^{m+1} &= R^m \oplus T^{\mathcal{R}_m} \\ R^{m+1} &= f_{m+1}(R^m \oplus T^{\mathcal{R}_m} \oplus T^{\mathcal{R}_{m+0.5}}) \oplus L^m \oplus T^{\mathcal{L}_m} \\ L^{m+2} &= f_{m+1}(R^m \oplus T^{\mathcal{R}_m} \oplus T^{\mathcal{R}_{m+0.5}}) \oplus L^m \oplus T^{\mathcal{L}_m} \oplus T^{\mathcal{R}_{m+1}} \\ R^{m+2} &= f_{m+2}(L^{m+2} \oplus T^{\mathcal{R}_{m+1.5}}) \oplus R^m \oplus T^{\mathcal{R}_m} \oplus T^{\mathcal{L}_{m+1}} \end{aligned}$$

Note that in both the formula for L^{m+2} and for R^{m+2} , $T^{\mathcal{R}_{m+0.5}}$ always appears in an \oplus with $T^{\mathcal{R}_m}$, and the only other place $T_{\mathcal{R}_m}$ appears is with $T^{\mathcal{L}_{m+1}}$ (canceling each other out). Thus, using $\mathcal{R}_{m+0.5}$ as a tweak location is the same as using the combination of \mathcal{R}_m and \mathcal{L}_{m+1} instead. ■

Another simple observation from Crump et al. [7] is that adding a tweak at location \mathcal{L}_m is equivalent to adding a tweak to location \mathcal{R}_{m+1} .

Lemma 2.2 *For all $0 \leq m < n$, \mathcal{L}_m is equivalent to \mathcal{R}_{m+1} .*

Proof: Suppose we start with inputs L^m and R^m . From the formulas,

$$L^{m+1} = R^m \oplus T^{\mathcal{R}_m}$$

$$R^{m+1} = f_{m+1}(R^m \oplus T^{\mathcal{R}_m} \oplus T^{\mathcal{R}_{m+0.5}}) \oplus L^m$$

From the formulas, L^{m+1} is not affected by any tweaks at either \mathcal{L}_m or \mathcal{R}_{m+1} ; R^{m+1} is affected by both \mathcal{L}_m and \mathcal{R}_{m+1} . Therefore applying a tweak in either location has the same effect on L^{m+1} and R^{m+1} .

■

Since \mathcal{L}_m and \mathcal{R}_{m+1} are equivalent, we will sometimes use them interchangeably. From Lemmas 2.1 and 2.2 we are able to reduce the set of possibly tweakable locations which includes tweaks at $\mathcal{L}_n, \mathcal{R}_0, \dots, \mathcal{R}_n$ and all combinations thereof.

Chapter 3

CPA-Secure Tweakable

Blockciphers

Now that we have introduced all necessary definitions and notation, we can construct a CPA-secure tweakable blockcipher from a Feistel cipher. Crump et al. proved that all tweakable blockciphers constructed from a six-round Feistel cipher in our model is insecure against exponential adversaries [7]. Thus possible tweakable blockcipher constructions which are CPA-secure against an exponential adversary must include at least seven random functions. In fact, seven rounds is enough to create a secure tweakable blockcipher; $\mathcal{BC}(7, \mathcal{R}_3 + \mathcal{L}_3)$ is a secure construction and is illustrated in Figure 3.1.

Theorem 3.1 $\mathcal{BC}(7, \mathcal{R}_3 + \mathcal{L}_3)$ is CPA-secure for $q \ll 2^k$ queries.

Proof:

In order to prove that $\mathcal{BC}(7, \mathcal{R}_3 + \mathcal{L}_3)$ is a secure tweakable blockcipher we decompose our seven-round construction into two functions, F and E , where F is the first three rounds, including the XOR-ed tweak at both \mathcal{L}_3 and \mathcal{R}_3 ,¹ and E is the last four rounds. It is obvious that E is a regular four-round Feistel function. To prove that F has the properties enumerated in Theorem 1.2, we need to prove that the probability of two different outputs being equal is small and the probability that the right side of two different outputs being equal is also small. More formally, we need to prove lemma 3.1.

Lemma 3.1 *F is constructed such that for any two distinct queries, the probability of the outputs being equal is $O(2^{-2k})$ and the probability of the right halves of the outputs being equal is $O(2^{-k})$.*

Proof: We show here that given two queries, the probability of an equality in the right half of the output is at most 2^{-k+1} , and that the probability of both outputs being equal is at most 2^{-2k+1} .

We call two queries L^0, R^0, T and L'^0, R'^0, T' respectively. We also assume that these queries are distinct, that is either $L^0 \neq L'^0$, or $R^0 \neq R'^0$, or $T \neq T'$. For ease of notation, we define δR^i as $R^i \oplus R'^i$, and $\delta f_i(R^i) = f_i(R^i) \oplus f_i(R'^i)$; we also define δL^i and δT_i similarly.

Thus we need to prove that the probability that $\delta R^3 \oplus \delta T = 0$ (i.e. the right halves of the outputs are equal) is $O(2^{-k})$ and that the probability that

¹Although \mathcal{L}_3 is equivalent to \mathcal{R}_4 (by lemma 2.2), we can think of this construction as using \mathcal{L}_3 , so that we can conceptually split the function this way.

$\delta R^3 \oplus \delta T = 0$ and $\delta L^3 \oplus \delta T = 0$ (i.e. both outputs are equal) is $O(2^{-2k})$.

Since the queries are unique, we can divide this proof into three cases, $\delta R^0 \neq 0$, $\delta R^0 = 0$ but $\delta L^0 \neq 0$, and $\delta R^0 = \delta L^0 = 0$ and $\delta T \neq 0$.

Case 1: $\delta R^0 \neq 0$. In order for the right half of the outputs to be equal, $\delta R^3 \oplus \delta T = 0$, we know that $\delta f_1(R^0) = \delta L^0 \oplus \delta f_3(R^2) \oplus \delta T$. Since $\delta R^0 \neq 0$ and f_1 is a random function, $\delta f_1(R^0)$ is a random value. Therefore the probability that $\delta f_1(R^0) = \delta L^0 \oplus \delta f_3(R^2) \oplus \delta T$, which corresponds to the probability that the right half of any two outputs are equal, is 2^{-k} .

In order for $\delta L^3 \oplus \delta T = 0$, (i.e. the left halves of the outputs are equal), we know that $\delta f_2(L^0 \oplus f_1(R^0)) = \delta R^0 \oplus \delta T$. If $\delta L^0 \oplus \delta f_1(R^0) \neq 0$, $\delta f_2(L^0 \oplus f_1(R^0)) = \delta R^0 \oplus \delta T$ only occurs with probability 2^{-k} . Furthermore, given this and because $\delta L^0 \oplus \delta f_1(R^0) = \delta R^2$, the probability that $\delta f_3(R^2) = \delta f_1(R^0) \oplus \delta L^0 \oplus \delta T$ is 2^{-k} , and therefore, the probability of a full collision is 2^{-2k} .

However, $\delta L^0 \oplus \delta f_1(R^0) = 0$ occurs with probability 2^{-k} . In that case, in order to have $\delta L^3 \oplus \delta T = 0$, we must have $\delta T = \delta R^0$. If $\delta R^3 \oplus \delta T = 0$ as well, we know $\delta f_1(R^0) = \delta L^0 \oplus \delta f_3(R^2) \oplus \delta T$, but since $\delta L^0 = \delta f_1(R^0)$ in this case, this implies that $\delta f_3(R^2) = \delta T = \delta R^0 \neq 0$, yet, this can occur with probability at most 2^{-k} . Therefore, the probability of an overall collision is at most $2(2^{-2k}) = 2^{-2k+1}$.

Case 2: $\delta R^0 = 0$ and $\delta L^0 \neq 0$. In order for $\delta R^3 \oplus \delta T = 0$, $\delta f_3(R^2) = \delta f_1(R^0) \oplus \delta L^0 \oplus \delta T$ must hold. Note that $\delta R^2 = \delta R^0 \oplus \delta f_2(R^1) = \delta f_2(R^1)$, and

$\delta R^1 = \delta L^0 \oplus \delta f_1(R^0) = \delta L^0 \neq 0$. If $\delta R^2 \neq 0$, there is a collision on the right only with probability 2^{-k} . However, the probability that $\delta R^2 = 0$ is 2^{-k} , so the probability of a collision on the right is at most $2 \cdot 2^{-k}$.

In order for the $\delta L^3 \oplus \delta T = 0$ to be true, we must have $\delta f_2(L^0 \oplus f_1(R^0)) = \delta R^0 \oplus \delta T = \delta T$. Because $\delta L^0 \neq 0$ and $\delta f_1(R^0) = 0$, $\delta f_2(L^0 \oplus f_1(R^0))$ is random. Therefore the equation is true with probability 2^{-k} , therefore the probability of the left halves of the output being equal is 2^{-k} .

If the left halves are equal, we know that $\delta f_2(R^1) = \delta T$. Recall that $\delta R^2 = \delta f_2(R^1)$, so if $\delta T = 0$, then $\delta R^2 = 0$. Thus $\delta R^3 \oplus \delta T = \delta L^0 \neq 0$. However, if $\delta T \neq 0$ then the probability that $\delta R^3 = \delta T$ is at most 2^{-k} . Therefore, the overall probability of a collision in this case is at most 2^{-2k} .

Case 3: $\delta R^0 = 0$ and $\delta L^0 = 0$. This case is trivial. Since the message queries are equal, $\delta R^3 = \delta L^3 = 0$. However, $\delta T \neq 0$, therefore $\delta R^3 \oplus \delta T = \delta L^3 \oplus \delta T \neq 0$. Therefore the outputs are never equal in either half of the output.

Thus, the overall probability that two distinct queries will have the same output is at most $O(2^{-2k})$ and the probability that the right half of the outputs will be equal is at most $O(2^{-k})$. Thus, we have proven Lemma 3.1. ■

So long as the queries the adversary makes do not produce a full collision on F or a multi-collision on the right half of the output of F , the responses are indistinguishable from random. Therefore, the queries of the adversary are independent of the outputs of F so long as the required conditions hold. By

Lemma 3.1, the probability of an overall collision in $q \ll 2^k$ queries is $O(q^2 2^{-2k})$, which is negligible. Similarly, the probability of an l -way multi-collision on the right is $O(q^l 2^{-(l-1)k}) = O(2^k (q 2^{-k})^l)$, which is less than $O((q 2^{1-k})^l)$. Since $(q 2^{1-k})^l$ is negligible, the probability of an l -way multi collision on the right is negligible. Thus, F satisfies the necessary properties with all but a negligible probability, which completes our proof of Theorem 3.1. \blacksquare

Since we have proved that $\mathcal{BC}(7, \mathcal{R}_3 + \mathcal{L}_3)$ is a seven-round tweakable block-cipher that is CPA-secure against exponential adversaries, and Crump et al. showed that all constructions with six or fewer rounds are insecure against comparable adversaries, our construction is round optimal in our model.

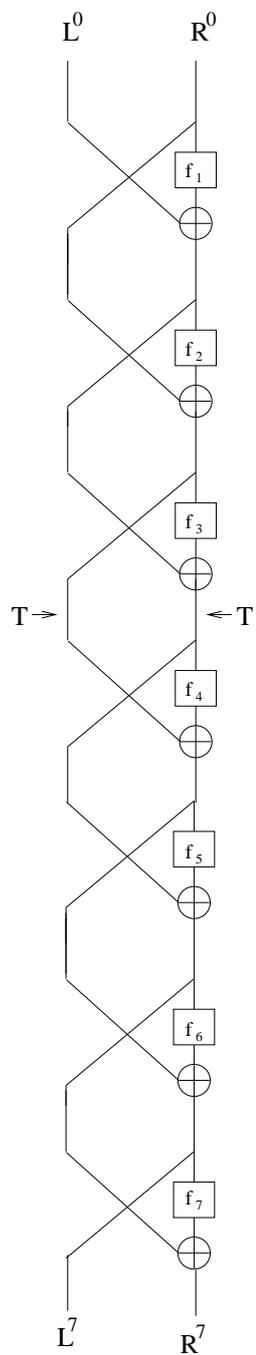


Figure 3.1: An illustration of $BC(7, \mathcal{R}_3 + \mathcal{L}_3)$; a seven-round exponentially CPA-secure tweakable Feistel blockcipher.

Chapter 4

CCA-Secure Tweakable Blockciphers

In this chapter we construct tweakable blockciphers which are CCA-secure against exponential adversaries. We prove that $\mathcal{BC}(10, \mathcal{L}_3 + \mathcal{R}_3 + \mathcal{L}_7 + \mathcal{R}_7)$ illustrated in Figure 4.1 is a ten round CCA-secure tweakable blockcipher.

Theorem 4.1 *$\mathcal{BC}(10, \mathcal{L}_3 + \mathcal{R}_3 + \mathcal{L}_7 + \mathcal{R}_7)$ is CCA-secure for $q \ll 2^k$ queries.*

Proof: In order to prove that $\mathcal{BC}(10, \mathcal{L}_3 + \mathcal{R}_3 + \mathcal{L}_7 + \mathcal{R}_7)$ is CCA-secure against exponential adversaries, we use Theorem 1.3. In our construction, the first three rounds, including the tweaks at \mathcal{L}_3 and \mathcal{R}_3 , form F , and the last three rounds, including the tweaks at \mathcal{L}_7 and \mathcal{R}_7 , form F' . Thus $\mathcal{BC}(10, \mathcal{L}_3 + \mathcal{R}_3 + \mathcal{L}_7 + \mathcal{R}_7) = F' \circ E \circ F$. F'^{-1} is identical to F , except with distinct random round functions. Both F and F'^{-1} meet the properties of Theorem 1.3, as we have shown in

our proof of Lemma 3.1. Since F , E , and F' have the properties given in Theorem 1.3, $\mathcal{BC}(10, \mathcal{L}_3 + \mathcal{R}_3 + \mathcal{L}_7 + \mathcal{R}_7)$ is CCA-secure against an exponential adversary bounded by $q \ll 2^k$ queries. ■

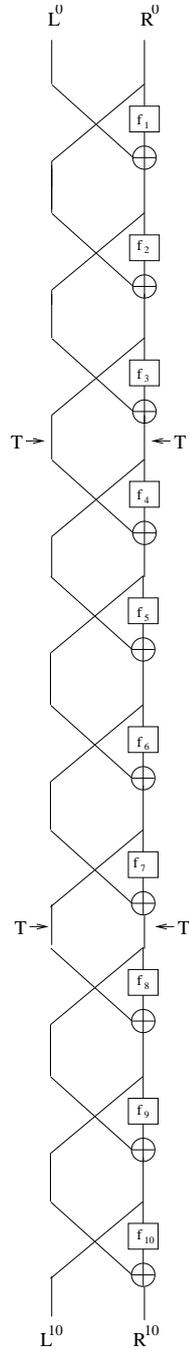


Figure 4.1: An illustration of $BC(10, \mathcal{L}_3 + \mathcal{R}_3 + \mathcal{L}_7 + \mathcal{R}_7)$; a ten-round exponentially CCA-secure tweakable Feistel blockcipher.

Chapter 5

Longer Tweaks

In this chapter, we demonstrate that secure (both CPA-secure and CCA-secure) tweakable blockciphers exist with arbitrary tweak length at the cost of one additional round per half-block of tweak, and that the constructions we give for CPA security are round-optimal, where the security is proved against an exponential adversary. We prove in Section 5.2 that CPA-secure tweakable blockciphers exist with arbitrary tweak length at the cost of one additional round per half-block of tweak. We then prove that this construction is round-optimal in our model. We then demonstrate in Section 5.3 that CCA-secure tweakable blockciphers exist with arbitrary tweak length at the cost of two additional rounds per half-block of tweak. (The optimality of this construction is an open problem.)

First, we discuss several general lemmas about the security multiple tweak and compound tweak locations first stated by Crump et al. [7] in Section 5.1.

5.1 General Lemmas

In order to prove security for longer tweaks, we need to adopt additional notation. The notation used in this section was adopted by Crump et al. [7]. Let Λ_n^* be the set of all compound tweak locations¹ over Λ_n .

In this section, we review general lemmas from Crump et al. that can be used for both CPA and CCA security.

Lemma 5.1 *For all n and $\Gamma_1, \dots, \Gamma_t \in \Lambda_n^*$, if $\mathcal{BC}(n, \Gamma_1, \dots, \Gamma_t)$ is secure, then for all $i = 1$ to t , $\mathcal{BC}(n, \Gamma_i)$ is secure.*

Proof: We prove this contrapositively; let $j \in [1, t]$ be such that $\mathcal{BC}(n, \Gamma_j)$ is insecure. We can attack $\mathcal{BC}(n, \Gamma_1, \dots, \Gamma_t)$ by following the attack on $\mathcal{BC}(n, \Gamma_j)$, but setting all tweaks other than T_j to 0^k . ■

We can define $\Gamma = \sum_{i \in S_\Gamma} \lambda_i$, where S_Γ is the set of locations used in Γ . If we do so, then clearly $\Gamma + \Gamma' = \sum_{i \in S_\Gamma \Delta S_{\Gamma'}} \lambda_i$ where Δ represents symmetric difference. Crump et al. also proved a generalization of Lemma 5.1.

Lemma 5.2 *For all n and $\Gamma_1, \dots, \Gamma_t \in \Lambda_n^*$, if $\mathcal{BC}(n, \Gamma_1, \dots, \Gamma_t)$ is secure, then for all $\emptyset \neq S \subset \{1, \dots, t\}$, $\mathcal{BC}(n, \sum_{i \in S} \Gamma_i)$ is secure.*

¹A compound tweak location is when the same tweak is XOR-ed in multiple locations in the Feistel stream.

Proof: If not, let S be such that $\mathcal{BC}(n, \sum_{i \in S} \Gamma_i)$ is insecure. We can attack $\mathcal{BC}(n, \Gamma_1, \dots, \Gamma_t)$ by following the attack on $\mathcal{BC}(n, \sum_{i \in S} \Gamma_i)$ by setting all tweaks T_j for $j \notin S$ equal to 0, and all tweaks T_i for $i \in S$ equal to each other. ■

Another observation from Crump et al. is that if using the same tweak in multiple locations, then at least one location must be secure.

Lemma 5.3 (Combinations With The Same Tweak) *For all n and $\lambda_1, \dots, \lambda_r \in \Lambda_n$, $\mathcal{BC}(n, \lambda_1 + \dots + \lambda_r)$ is secure, then $\lambda_i \in \{\mathcal{R}_2, \dots, \mathcal{R}_{n-2}\}$ for some $1 \leq i \leq r$.*

Proof: Since without loss of generality, all λ_i are in $\{\mathcal{R}_2, \dots, \mathcal{R}_{n-1}\}$, the only way the condition is not met is if all λ_i are \mathcal{R}_{n-1} . If r is even, the construction is equivalent to $\mathcal{BC}(n, \emptyset)$, while if r is odd, the construction is equivalent to $\mathcal{BC}(n, \mathcal{R}_{n-1})$, both of which are insecure. ■

These three lemmas apply for any type of security.

5.2 Longer Tweaks with CPA Security

In this section we construct a Feistel-based tweakable blockcipher CPA secure against an exponential adversary with a fixed arbitrary sized tweak. For t half-blocks of tweak, we construct a Feistel-based tweakable blockcipher in $t + 6$ rounds that is CPA secure.

We construct a tweakable blockcipher with longer tweaks similarly to the way we constructed tweakable blockciphers in Chapter 3 with “normal” sized tweaks.

We break our tweaked Feistel cipher into two ciphers, F and E , where F is a $t+2$ round function which includes all tweaks XOR-ed into the datastream² and meets the properties required from Theorem 1.2 and E is a regular four-round Feistel cipher.

5.2.1 Constructing F

The construction of F is not as simple as we would have liked, and therefore we give the intuition of how to construct F so that it meets the properties of Theorem 1.2 and why simpler attempts fail at meeting these required properties.

Remember, we must construct F such that with $q \ll 2^k$ distinct queries

1. the probability that any two outputs collide is $\nu(k)$ where $\nu \in \text{negl}$ and
2. the probability that there are $l > O(k)$ outputs such that the right halves collide is $\nu(k)$ where $\nu \in \text{negl}$.

For all possible constructions of F , we start adding tweaks at location \mathcal{L}_{i+2} since tweaks included at locations involving only at \mathcal{L}_1 and/or \mathcal{L}_2 allow for a full collision attack, as we will explain in Section 5.4. Therefore \mathcal{L}_1 and/or \mathcal{L}_2 are not useful as the sole location of a tweak.

All tweaks are included on the left only for simplicity of presentation: apart from the tweak included at \mathcal{L}_{t+2} , all tweaks could be included on the right instead.

²For example, in Chapter 3, when there was only 1 tweak, F was three rounds.

First Attempt Constructing F

The simplest construction of F is for all $1 \leq i \leq t$ include tweak T_i at location \mathcal{L}_{i+2} .

This construction obviously does not meet the requirements of Theorem 1.2 because the right half of the output is not affected by the last tweak. Thus it is easy to force collisions on the right by querying the cipher with the same message and the same T_i for all $1 \leq i \leq t - 1$ but vary T_t . Because T_t never effects the right side of F 's output, all queries will collide on the right. Since F fails at meeting the requirements, we need to consider a different F construction.

Second Attempt Constructing F

We constructed F in Chapter 3 so that it was able to meet the requirements for one tweak. Intuitively, we should build on that construction, to allow for multiple tweaks; we construct F such that for all $1 \leq i \leq t$ include tweak T_i at location \mathcal{L}_{i+2} , and also at the following location:

- If $i = t$, then also include T_i at \mathcal{R}_{t+2} .

Since this construction is similar to F constructed for a single tweak in Chapter 3, it would appear to be a likely candidate for a “good” F . However, if equal tweaks appeared at both L_i and L_{i+2} , by Lemma 2.1 that would be equivalent to having only one tweak at $R_{i+0.5}$. Crump et al. proved that having a tweak only appear $R_{i+0.5}$ for any i is insecure [7]. In order to attack this

construction, we query F with $2^{k/2}$ different messages, where L^0 and R^0 remains constant, for any constant $1 \leq c \leq t$ let $T_c = T_{c-2}$, and for all $1 \leq i \leq t$ such that $i \neq c$ and $i \neq c - 2$, fix T_i . Therefore, all queries are identical except at locations \mathcal{L}_{c-2} and \mathcal{L}_c which is equivalent to all queries being identical except at location $R_{c-1.5}$ (from Crump et al.). Thus, the internal values stay constant until the input to f_{c-1} . Since we have made $2^{k/2}$ queries to an ideal round function, by the birthday bound, we can expect with non-negligible probability to get a collision on the output of f_{c-1} for two different queries. If we get such a collision, the entire output ciphertext will collide. Therefore this F construction does not meet the properties, and we need to consider a different F construction.

Since the previous construction didn't meet the required properties, we might also consider extending our construction of F from Chapter 3 such that for all $1 \leq i \leq t$, add a new round and XOR T_i into the stream at both \mathcal{L}_{i+2} and \mathcal{R}_{i+2} . Thus the construction of F for all $1 \leq i \leq t$ includes tweak T_i at location \mathcal{L}_{i+2} . and

- for all i include T_i at \mathcal{R}_{i+2} .

This construction is also insecure, because for any two queries, $(L^0, R^0, T_1, \dots, T_t)$ and $(L'^0, R'^0, T'_1, \dots, T'_t)$, such that $L^0 = L'^0$, $R^0 = R'^0$, for all $1 \leq i \leq t - 2$ $T_i = T'_i$, and $T_{t-1} \oplus T_t = T'_{t-1} \oplus T'_t$, the probability to get a full collision is

$O(2^{-k})$. Therefore, by the birthday bound, with $q \ll 2^k$ queries we can expect full collisions with non-negligible probability.

Third Attempt Constructing F

Since simpler constructions did not meet the requirements, we considered adding two new rounds for each tweak. Let F be a construction such that for all $1 \leq i \leq t$,

- T_i is included at \mathcal{R}_{1+2*i} .
- T_i is included at \mathcal{L}_{1+2*i} .

This construction is the first construction to seemingly meet all of the requirements from Theorem 1.2; however this construction requires $2t + 1$ rounds which becomes inefficient very quickly and is far from optimal. Therefore, we need to find another way to construct F .

Fourth Attempt Constructing F

The problem with the second construction is that when two equal tweaks are used two rounds apart, there is an attack. A way to resolve this problem is to construct F such that two equal tweaks can never be placed two rounds apart. Thus a simple solution would be to construct F such that for all $1 \leq i \leq t$ use tweak T_i at the following location:

- If $i \equiv 0 \pmod{2}$ (i is even), then include T_i at \mathcal{L}_{2*i} .

- If $i \equiv 1 \pmod{2}$ (i is odd), then include T_i at \mathcal{L}_{2*i+1} .
- If $i = t$, then include T_i at \mathcal{R}_{i+2} .

This construction also seems to meet the requirements of Theorem 1.2, however on average this construction adds about two rounds per tweak, and like the previous construction is far too inefficient. Therefore we need to consider how to construct F so that it meets the required properties and is more efficient.

Fifth Attempt Constructing F

Thus far, we have been constructing F to avoid the following two attacks: an adversary can easily force a collision on the right half of the output because the right side isn't affected by the last tweak, or the adversary can force a full collisions because two different equal tweaks can be placed two rounds apart. In order to avoid the first attack, we can place the last tweak at location \mathcal{R}_{t+2} , but we have not yet constructed F to efficiently avoid the second attack.

However, we have not yet constructed F to include any tweaks at \mathcal{L}_1 or at \mathcal{L}_2 . Although these locations are not secure on their own, adding tweaks at \mathcal{L}_1 or \mathcal{L}_2 that are also included at other locations prevents the second attack (since tweaks will no longer be used *only* at a single location). We think of \mathcal{L}_1 and \mathcal{L}_2 as riders since every tweak used at these locations are also included elsewhere.

Therefore we construct F for all $1 \leq i \leq t$ to include tweak T_i at location \mathcal{L}_{i+2} , and also at the following locations:

- If $i \equiv 0 \pmod{4}$, then also include T_i at \mathcal{L}_2 .
- If $i \equiv 3 \pmod{4}$, then also include T_i at \mathcal{L}_1 .
- If $i = t$, then also include T_i at \mathcal{R}_{t+2} .

This construction avoids the equivalent tweak attack by adding a compound location at \mathcal{L}_1 and \mathcal{L}_2 . By adding the riders at \mathcal{L}_1 and \mathcal{L}_2 we avoid any attack that occurs when two tweaks are *only* used at two individual locations which are two rounds apart. Thus every other odd tweak, beginning with T_3 , is included at \mathcal{L}_1 (for $1 \leq i \leq t$, if $i \equiv 3 \pmod{4}$ use T_i at location \mathcal{L}_1) and every other even tweak, beginning with T_4 is included at \mathcal{L}_2 (for $1 \leq i \leq t$, if $i \equiv 0 \pmod{4}$ use T_i at location \mathcal{L}_2).

Unfortunately, this construction is also susceptible to attack if we aren't careful: if T_t is used at \mathcal{R}_{t+2} and not included in a rider, and T_{t-1} is used only at \mathcal{L}_{t+1} , by Lemma 2.2 the tweaks effectively occur at the same spot. Therefore, an adversary can easily force collisions on the right by querying with identical messages and tweaks except for tweaks T_{t-1} and T_t . If an adversary varies T_{t-1} and T_t such that $T_{t-1} \oplus T_t = c$ for some constant c , all outputs of F will collide on the right. Therefore, in some cases, namely when T_{t-1} and T_t are not included in riders, an adversary is able to force collisions on the right.

5.2.1.1 Final Construction of F

In order to avoid the attack that the fifth construction of F is susceptible to, we adjust the only case when there is a problem, namely when $t \equiv 2 \pmod{4}$. We construct F for all $1 \leq i \leq t$ to include tweak T_i at location \mathcal{L}_{i+2} , and also at the following locations:

- If $i \equiv 0 \pmod{4}$, then also include T_i at \mathcal{L}_2 .
- If $i \equiv 3 \pmod{4}$, then also include T_i at \mathcal{L}_1 .
- If $i = t$, then also include T_i at \mathcal{R}_{t+2} .
- If $i = t$ and $t \equiv 2 \pmod{4}$, then also include T_t at \mathcal{L}_1 .

This construction of F requires only $t+2$ rounds, which is much smaller than other possibly secure constructions. F with six half blocks of tweak is illustrated in Figure 5.1.

In the next section we prove that F meets the properties of Theorem 1.2 and the minimality of F is proven in Section 5.4.

5.2.2 Proving F 's correctness

In this section we prove that F has the properties needed from Theorem 1.2; i.e. given unique queries that it is “hard” to get collisions on the right half of the output, and that it is “even harder” to get full collisions on the output.

In order to prove that F has the required properties, we introduce the following additional notation:

- Let T_0 be $\oplus_{i \equiv 0 \pmod 4} T_i$.
- If $t \equiv 2 \pmod 4$ then define T_{-1} to be $T_t \oplus_{i \equiv 3 \pmod 4} T_i$;
otherwise, define T_{-1} to be $\oplus_{i \equiv 3 \pmod 4} T_i$.
- Let $T_{ev} = \oplus_{i=0}^{\lfloor t/2 \rfloor} T_{2i}$.
- Let $T_{od} = \oplus_{i=-1}^{\lfloor (t+1)/2 \rfloor} T_{2i+1}$.
- Let T_{te} be T_t if t is even, and 0 otherwise.
- Let T_{to} be T_t if t is odd, and 0 otherwise.

We are also reusing the δ notation defined in Chapter 3 where δR^i as $R^i \oplus R^i$, $\delta f_i(R^i) = f_i(R^i) \oplus f_i(R^i)$, and δL^i and δT_i is defined similarly..

First, we focus on the probability of a full collision on the outputs of F given two distinct queries.

Theorem 5.1 *On any pair of distinct inputs, the probability that F will produce the same output on each is $O(2^{-2k})$.*

Proof: In order for two unique queries to yield a full collision of F , we must have the following two equations:

$$\begin{aligned}
0 &= \delta R^0 \oplus \delta T_{te} \oplus \delta T_{od} \oplus \delta f_2(R^1) \oplus \delta f_4(R^3) \oplus \dots \oplus \delta f_{2^{\lfloor t/2 \rfloor + 2}}(R^{2^{\lfloor t/2 \rfloor + 1}}) \\
0 &= \delta L^0 \oplus \delta T_{to} \oplus \delta T_{ev} \oplus \delta f_1(R^0) \oplus \delta f_3(R^2) \oplus \dots \oplus \delta f_{2^{\lfloor (t+1)/2 \rfloor + 1}}(R^{2^{\lfloor (t+1)/2 \rfloor}})
\end{aligned}$$

where the first equation corresponds to the left side of the output if there are an odd number of tweaks and the right side of the output if there are an even number of tweaks; the second equation corresponds to the left side of the output if there are an even number of tweaks and the right side of the output if there are an odd number of tweaks.

Note that for every $0 \leq i \leq t - 1$ R^i is involved in one of the two equations above; If i is odd, R^i is involved in the first equation, while if i is even R^i is involved in the second equation. We break the proof into the following three cases:

Case 1: There is an even $i < t + 2$ such that $\delta R^i \neq 0$, and there is an odd $j < t + 2$ such that $\delta R^j \neq 0$. In this case, the probability that the first equation is true is 2^{-k} and the probability that the second equation is true is 2^{-k} . Therefore the probability that both equations are true is 2^{-2k} .

Case 2: For all $i < t + 2$, $\delta R^i = 0$. If this is the case, it is easy to see that $\delta R^0 = 0$, and $\delta L^0 = 0$ (since $\delta R^1 = \delta f_1(R^0) \oplus \delta L^0$). Furthermore, for each

$$1 \leq i \leq t + 1, 0 = \delta R^{i+1} = \delta f_{i+1}(R^i) \oplus \delta L^i = 0 \oplus \delta L^i = \delta R^{i-1} \oplus \delta T_{i-2} = \delta T_{i-2}.$$

Therefore, all the tweak values must also remain constant up to T_{t-1} . But if both L^0 and R^0 are the same, and all the tweak values up to T_{t-1} are the same, then the difference on the right of the output will be the difference in T_t , so if there is a collision, $\delta T_t = 0$. Therefore, both queries are the same. Thus the probability of two distinct queries leading to this case is 0.

Case 3: Either for all odd $i < t + 2$ or for all even $i < t + 2$, $\delta R^i = 0$, but there is some $j < t$ such that $\delta R^j \neq 0$. This covers all remaining cases, but this case is the hardest one to prove. A priori, one of the two equations may be true with probability 1, while the other equation is true with probability 2^{-k} . However, as we will prove in Lemma 5.4, a full collision can only occur with very low probability or a full collision only occurs when two inputs were identical.

Lemma 5.4 *In two distinct queries to F for which either for all odd $i < t + 2$ or for all even $i < t + 2$, $\delta R^i = 0$, but $\exists j < t$ such that $\delta R^j \neq 0$ the probability of a full collision on F is $O(2^{-2k})$.*

Proof: Let j be such that $\delta R^j \neq 0$ but for all $i > j$ of the same parity (even or odd), $\delta R^i = 0$. Suppose, without loss of generality, that j is even (the case when j is odd is very similar). In other words, we are assuming that given two different inputs and for some even j , $\delta R^j \neq 0$ but for all $i < j$ and for all odd $k < t + 2$, $\delta R^i = \delta R^k = 0$.

Since for all odd $i < t + 2$, $\delta R^i = 0$, we learn that if $i > 2$ is odd, then

$$\begin{aligned}
0 &= \delta R^i \\
&= \delta f_i(R^{i-1}) \oplus \delta L^{i-1} \\
&= \delta f_i(R^{i-1}) \oplus \delta T_{i-3} \oplus \delta R^{i-2} \\
&= \delta f_i(R^{i-1}) \oplus \delta T_{i-3}
\end{aligned}$$

Therefore, $\delta f_i(R^{i-1}) = \delta T_{i-3}$. Let a “rare event” be an event which happens with probability 2^{-k} . If $i - 1 < j$, then $\delta f_i(R^{i-1}) = \delta T_{i-3}$, we either must have a rare event, specifically that $\delta R^{i-1} \neq 0$ but the random function outputs a pre-specified difference, or $\delta R^{i-1} = 0$. If $i - 1 > j$, then by our choice of j , $\delta R^{i-1} = 0$. Therefore, for all even $i - 1 \neq j$ and $i - 1 \neq 0$, $\delta R^{i-1} = 0$, or two rare events must occur. We also know that $\delta R^0 = 0$, because

$$\begin{aligned}
0 &= \delta R^1 \\
&= \delta f_1(R^0) \oplus \delta L^0,
\end{aligned}$$

so $\delta f_1(R^0) = \delta L^0$. Again, this can only occur without a rare event if $\delta R^0 = 0$. If none of these rare events occur, then $\delta L^0 = 0$, and $\delta T_{i-3} = 0$ for all odd $2 \leq i < t + 2$ other than $j - 2$.

Furthermore, if i is even such that $\delta R^i = 0$ and $\delta R^{i-2} = 0$, then we can conclude that $\delta T_{i-3} = \delta f_i(R^{i-1}) \oplus \delta R^{i-2} = 0$, by the above deduction, and

because $i - 1$ is odd.

Thus we have learned that for most i , $\delta T_i = 0$. The exceptions are for $i \in \{j - 3, j - 2, j - 1, t - 1, t\}$. For those, we still know something:

- Since $\delta R^j = \delta f_j(R^{j-1}) \oplus \delta R^{j-2} \oplus \delta T_{j-3}$, and $\delta R^{j-1} = \delta R^{j-2} = 0$, we know $\delta T_{j-3} = \delta R^j$.
- Since $0 = \delta R^{j+1} = \delta f_{j+1}(R^j) \oplus \delta R^{j-1} \oplus \delta T_{j-2}$, we know $\delta T_{j-2} = \delta f_{j+1}(R^j)$.
- Since $0 = \delta R^{j+2} = \delta f_{j+2}(R^{j+1}) \oplus \delta R^j \oplus \delta T_{j-1}$, we know $\delta T_{j-1} = \delta R^j$.

We also know that $\delta L^0 = 0$ since $\delta L^0 = \delta R^1 \oplus \delta f_1(R^0)$ (except if $j = 0$, which we will handle as a special case.)

Note that because the two output halves are equal, this lets us conclude that $\delta T_t = 0$ and that $\delta T_{t-1} = 0$. Recall that:

$$\begin{aligned}
0 &= \delta R^0 \oplus \delta T_{te} \oplus \delta T_{od} \oplus \delta f_2(R^1) \oplus \delta f_4(R^3) \oplus \dots \oplus \delta f_{2^{\lfloor t/2 \rfloor + 2}}(R^{2^{\lfloor t/2 \rfloor + 1}}) \\
0 &= \delta L^0 \oplus \delta T_{to} \oplus \delta T_{ev} \oplus \delta f_1(R^0) \oplus \delta f_3(R^2) \oplus \dots \oplus \delta f_{2^{\lfloor (t+1)/2 \rfloor + 1}}(R^{2^{\lfloor (t+1)/2 \rfloor}})
\end{aligned}$$

In one of these two equations, both δT_t and δT_{t-1} appear (the one for which T_{te} or T_{to} is nonzero); in the other, only δT_t does. All other terms come out to zero; in one equation, only δT_{j-3} and δT_{j-1} are not guaranteed to be 0, but these are equal to each other. In the other, δT_{j-2} is not guaranteed to be 0, and neither is $f_{j+1}(R^j)$, which appears in the same equation, but $\delta T_{j-2} = f_{j+1}(R^j)$,

and therefore XOR to 0. Thus, from the equation in which T_t appears alone, we conclude $\delta T_t = 0$, and from the other one we then conclude that $\delta T_{t-1} = 0$.

We now break the proof into five cases: $j \geq 4$, $j = 3$, $j = 2$, $j = 1$ and $j = 0$.

case i: $j \geq 4$. If we assume that $j \geq 4$ then we know that both $\delta T_0 = 0$ and $\delta T_{-1} = 0$. One of those two terms includes *one* of T_{j-3} and T_{j-1} , but not both, and all other tweaks included must remain unchanged. Therefore, both δT_{j-3} and δT_{j-1} are 0. But we know that $0 \neq \delta R^j = \delta T_{j-1}$, so this is a contradiction; thus $j < 4$.

case ii: $j = 3$. If $j = 3$, then $\delta T_0 = \delta T_2$, but all other even-numbered tweaks are unchanged. Since T_2 is not included in T_0 , we note that all the other terms in T_0 are known to have no difference between the two queries. Therefore, $\delta T_0 = 0$ and so $\delta T_2 = 0$, which then implies that $\delta R^3 = 0$, which is a contradiction; thus $j < 3$.

case iii: $j = 2$. If $j = 2$, then $\delta T_{-1} = \delta T_1$, but all other odd-numbered tweaks are unchanged. Since T_1 is not part of T_{-1} , we can conclude that $\delta T_{-1} = 0$, which is a contradiction; thus $j < 2$.

case iv: $j = 1$. If $j = 1$, then we can conclude that $\delta L^0 = \delta T_0$, via a similar deduction. Because all even-numbered tweaks are 0, we get $\delta T_0 = 0 = \delta L^0$. Since $\delta R^0 = 0$, we know that $\delta R^1 = \delta L^0 \oplus \delta f_1(R^0) = 0$, which is a contradiction; thus $j < 1$.

case v: $j = 0$. If $j = 0$, we know that $\delta T = 0$ for all i . Since $0 = \delta R^2 = \delta f_2(R^1) \oplus \delta R^0 \oplus \delta T_{-1}$ and $\delta T_{-1} = \delta R^1 = 0$, we get that $\delta R^0 = 0$, which is a contradiction.

Therefore, if two distinct queries are such that either for all odd $i < t + 2$ or for all even $i < t + 2$, $\delta R^i = 0$, but there is some $j < t + 2$ such that $\delta R^j \neq 0$ then at least two rare events must occur in order for an overall collision to occur: therefore, the probability of a collision in this case is at most $O(2^{-2k})$. This completes the proof of Lemma 5.4. ■

Since in cases 1 and 3, the probability of a collision is at most $O(2^{-2k})$, and the probability of a collision in case 2 is 0, the overall probability of a full collision given two unique queries is at most $O(2^{-2k})$. This completes the proof of Theorem 5.1 ■

Now that we have proven that for any pair of distinct queries F does not allow full collisions with probability greater than $O(2^{-2k})$, we must also prove that for any pair of distinct queries, the probability of a collision on the right is at most $O(2^{-k})$.

Theorem 5.2 *On any pair of distinct inputs, the probability that F will produce the same output on the right in each is $O(2^{-k})$.*

Proof: Assume without loss of generality that t is odd; if not, the proof is similar. If t is odd, then whenever two queries lead to a collision on the right,

we have

$$\delta L^0 \oplus \delta T_t \oplus \delta T_{ev} \oplus \delta f_1(R^0) \oplus \dots \oplus \delta f_{t+2}(R^{t+1}) = 0$$

If for some even $i < t + 2$, $\delta R^i \neq 0$, then the probability of a collision occurring is 2^{-k} .

Therefore we can assume that for all even $i < t + 2$, $\delta R^i = 0$. If $2 \leq i < t + 2$ is even, then $0 = \delta R^i = \delta f_i(R^{i-1}) \oplus \delta L^{i-1} \oplus \delta T_{i-3} = \delta f_i(R^{i-1}) \oplus \delta T_{i-3} \oplus \delta R^{i-2}$, so $\delta f_i(R^{i-1}) = \delta T_{i-3}$, since $\delta R^{i-2} = 0$. Thus, either $\delta R^{i-1} = 0$ or this equation is true with probability 2^{-k} . Therefore we can assume in all such cases, $\delta R^{i-1} = 0$;

Since we can assume that $\delta R^{i-1} = 0$ then $\delta f_i(R^{i-1}) = 0$, so $\delta T_{i-3} = 0$. Thus we know that all the odd-numbered tweaks up to T_{t-2} do not change between the two queries, including the value T_0 . But similarly, if $1 \leq i < t + 1$ is odd then $\delta T_{i-3} = 0$.

If a collision occurs, we also have that $\delta T_t = \delta R^{t+2} = \delta f_t(R^{t+1}) \oplus \delta R^t \oplus \delta T_{t-1} = \delta T_{t-1}$.

We have been able to deduce that $\delta T_i = 0$ for $-1 \leq i \leq t - 2$. We have included T_t in T_0 and T_{-1} in such a way that regardless of $t \bmod 4$, T_t is involved in one that T_{t-1} is *not* involved in, or vice-versa. Since $\delta T_0 = \delta T_{-1}$, this allows us to conclude that both δT_t and δT_{t-1} are 0. Thus proving that the two queries are actually the same.

Therefore, assuming that the two queries are distinct at least one rare event must occur in order to produce a collision on the right half of the output. There-

fore, the probability of such a collision is at most $O(2^{-k})$. ■

Theorem 5.1 and Theorem 5.2 allow us to prove our constructed F has the properties enumerated in Theorem 1.2.

Theorem 5.3 *F is a function such that for $q \ll 2^k$ queries, the probability of having $l = O(k)$ indices such that $R_{i_1} = R_{i_2} = R_{i_3} = \dots R_{i_l}$ is negligible, (where R_{i_j} is the right half of the j 'th output of F), and on q distinct inputs F has only a negligible probability of a full collision on its outputs.*

Proof: The proof follows from Theorem 5.1 and Theorem 5.2. We can once again apply the principle of deferred decisions, and furthermore, $q \ll 2^k$ queries will not allow a non-negligible probability for the failure of either condition. The reasoning is parallel to that given in the proof of Theorem 3.1. ■

From Theorem 5.3 we can finally prove that CPA-secure tweakable blockciphers that allow for tweaks of arbitrary length exist. Let F be the function described in the final construction of this section and let E be a regular four-round Feistel cipher, then $E \circ F$ is a tweakable blockcipher CPA secure against exponential adversaries that allows for tweaks of arbitrary size.

Theorem 5.4 *$E \circ F$ is a tweakable blockcipher with t tweaks that is secure against any exponential adversary with at most $q \ll 2^k$ queries, where E is a four-round Feistel cipher and F is a $t+2$ -round tweaked Feistel cipher described in Section 5.2.1.1.*

This follows from Theorem 5.3 and Theorem 1.2. Note that $E \circ F$ requires a total of $t + 6$ rounds.

5.3 Longer Tweaks with CCA security

In this section, we prove the existence of CCA-secure tweakable blockcipher that allow for arbitrary length tweaks. Let F be a $t + 2$ round tweaked Feistel construction described in Section 5.2.1.1. Let F' be a $t + 2$ round tweaked Feistel construction which is equivalent to the inverse of F except that the random round functions used in F' are newly chosen at random. Let E be a regular four-round Feistel cipher. We prove that $F' \circ E \circ F$ is CCA-secure against exponential adversaries.

Theorem 5.5 *$F' \circ E \circ F$ is a tweakable blockcipher with t tweaks that is CCA-secure against any unbounded adversary with at most $q \ll 2^k$ queries, where E is a four-round Feistel cipher, F' is the inverse of the F described above, with new independent round functions.*

Proof: The proof follows from Theorem 5.3 and Theorem 1.3. ■

Notice that our CCA-secure construction, $F' \circ E \circ F$ requires $2(t + 2) + 4 = 2t + 8$ rounds.

5.4 Minimality of F

In this section, we show that F is minimal in terms of the number of rounds needed in order to meet the properties required by Theorem 1.2.

Lemma 5.5 *If F is a Feistel-based blockcipher incorporating t tweaks, and F has $n < t + 2$ rounds, then certain pairs of queries can lead to an overall collision on the output of F with probability $O(2^{-k})$.*

Proof: Without loss of generality, the location for each tweak can be expressed in terms of compound locations based on the locations $\mathcal{L}_1, \dots, \mathcal{L}_n$ (since \mathcal{L}_0 and \mathcal{R}_0 can be simulated away). Let $\Gamma_1, \dots, \Gamma_t$ be the compound locations for T_1, \dots, T_t , respectively. Let Γ'_i be defined as the portion of Γ made up of only $\mathcal{L}_3, \dots, \mathcal{L}_n$, for each i .

Since $n < t + 2$, there are fewer than t locations in $\mathcal{L}_3, \dots, \mathcal{L}_n$. Therefore, there will be some linear dependency among the Γ' values, that is, there will be some i such that for some $S \subset \{1, \dots, n\}$ such that

$$0 = \sum_{j \in S} \Gamma'_j.$$

Therefore, by Lemma 5.2, such a construction is insecure; the compound location $\sum_{j \in S} \Gamma_j$ will consist of only locations from \mathcal{L}_1 and \mathcal{L}_2 .

Note that \mathcal{L}_1 and \mathcal{L}_2 on their own can be thought of as equivalent to $\mathcal{L}_1 + \mathcal{R}_0$ and $\mathcal{L}_2 + \mathcal{L}_0$, respectively. Those constructions fall to the attack given by Crump

et al. in Lemma 4.9 [7]. $\mathcal{L}_1 + \mathcal{L}_2$ can be thought of as equivalent to $\mathcal{L}_0 + \mathcal{L}_1 + \mathcal{L}_2$ which is the same as $\mathcal{R}_{1.5} + \mathcal{R}_2$, which falls to the attack given by Crump et al. in Corollary 4.10 [7]. ■

Therefore F (and F' presented in Section 5.3) is round-optimal in our model.

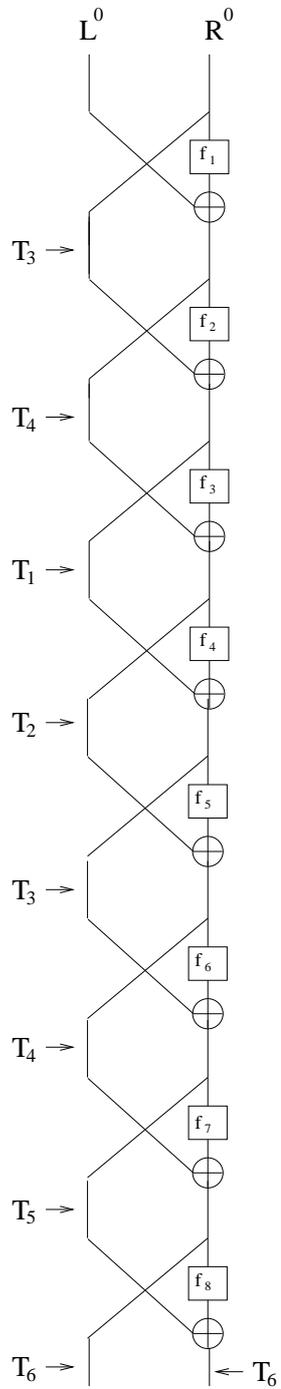


Figure 5.1: An illustration of F which contains six half blocks of tweak ($t = 6$).

Chapter 6

Conclusion

Crump et al. constructed tweakable blockciphers from Feistel ciphers by XOR-ing tweaks directly in the datastream and proved the security of their constructions against polynomial adversaries [7]. In this thesis, we extend their model and construct tweakable blockcipher from Feistel ciphers which are secure against adversaries allowed unlimited computations, but bounded by an exponential number of queries.

We proved that seven rounds are sufficient to construct a chosen plaintext secure (CPA) tweakable blockcipher. We then showed that ten rounds are sufficient to construct a chosen ciphertext secure (CCA) tweakable blockcipher. We are the first to prove the security of tweakable blockciphers against a computationally unbounded adversary allowed $q \ll 2^k$ queries, where k is half the input size. Our results match the best security results proven for regular blockciphers [22].

We also address the problem of incorporating tweaks of arbitrary length into a tweakable blockcipher. We proved that six rounds plus one round per tweak is sufficient for CPA security, and eight rounds plus two rounds per tweak is sufficient for CCA security.

Bibliography

- [1] MIHIR BELLARE AND TADAYOSHI KOHNO. A Theoretical Treatment of Related-Key Attacks: PKA-PRPs, RKA-PRFs, and Applications. In *Advances in Cryptology – EUROCRYPT '03*, E. Biham, editor, volume 2656 of LNCS, pages 491–506, 2003.
- [2] J. BLACK, P. ROGAWAY, AND T. SHRIMPTON. Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV. In *Advances in Cryptology—CRYPTO 2002*, Moti Yung, editor, Lecture Notes in Computer Science, pages 320–335. Springer-Verlag, 2002.
- [3] JOHN BLACK, MARTIN COCHRAN, AND THOMAS SHRIMPTON. On The Impossibility of Highly-Efficient Blockcipher-Based Hash Functions. In *Advances in Cryptology – Eurocrypt 2005*, volume 3494 of LNCS, pages 526–541. Springer Verlag, May 2005.
- [4] C. BURWICK, D. COPPERSMITH, E. D’AVIGNON, R. GENNARO, S. HALEVI, C. JUTLA, S. M. MATYAS JR., L. O’CONNOR, M. PEYRAVIAN, D. SAFFORD, AND N. ZUNIC. MARS - A Candidate Cipher for AES. In *NIST AES proposal*, June 1998.
- [5] DEBRUP CHAKRABORTY AND PALASH SARKAR. A General Construction of Tweakable Block Ciphers and Different Modes of Operation. In *Information Security and Cryptology*, volume 4318, pages 88–102. Springer-Verlag Berlin Heidelberg, 2006.
- [6] PAUL CROWLEY. Mercy: A fast large block cipher for disk sector encryption. In *Fast Software Encryption: 7th International Workshop*, volume 1978 of *Lecture Notes in Computer Science*, pages 49–63. Springer-Verlag, 2000. Also available at: www.ciphergoth.org/crypto/mercy.
- [7] ELIZABETH CRUMP, DAVID GOLDBERG, SUSAN HOHENBERGER, MOSES LISKOV, AND HAKAN SEYALIOGLU. On Tweaking Feistel Ciphers. Submitted Crypto 2007.

- [8] YEVGENIY DODIS AND PRASHANT PUNIYA. Feistel Networks made Public, and Applications. In *Advances in Cryptology – EUROCRYPT 2007, to appear*, 2007.
- [9] H. FEISTEL. Cryptography and Computer Privacy. *Scientific American*, pages 15–23, May 1973.
- [10] SHAI HALEVI AND PHILLIP ROGAWAY. A Tweakable Enciphering Mode. In *Advances in Cryptology – CRYPTO '03*, Dan Boneh, editor, volume 2729 of LNCS, pages 482–499, 2003.
- [11] SHAI HALEVI AND PHILLIP ROGAWAY. A Parallelizable Enciphering Mode. In *Topics in Cryptology – CT-RSA '04*, Tasuaki Okamoto, editor, volume 2964 of LNCS, pages 292–304, 2004.
- [12] ANTOINE JOUX. Cryptanalysis of the EMD Mode of Operation. In *Advances in Cryptology – EUROCRYPT '03*, E. Biham, editor, volume 2656 of LNCS, pages 1–16, 2003.
- [13] MOSES LISKOV, RONALD L. RIVEST, AND DAVID WAGNER. Tweakable Block Ciphers. In *Advances in Cryptology – CRYPTO '02*, Moti Yung, editor, volume 2442 of LNCS, pages 31–46, 2002.
- [14] MICHAEL LUBY AND CHARLES RACKOFF. How To Construct Pseudorandom Permutations from Pseudorandom Functions. *SIAM Journal of Computing*, 17(2):373–386, 1988.
- [15] STEFAN LUCKS. Faster Luby-Rackoff Ciphers. In *Fast Software Encryption*, Dieter Gollmann, editor, volume 1039 of LNCS, pages 189–203, 1996.
- [16] KAZUHIKO MINEMATSU. Improved Security Analysis of XEX and LRW Modes. In *Proceedings of 13th annual workshop on Selected Areas in Cryptography (SAC) 2006*, pages 92–109, August 2006 (preprint).
- [17] MONI NAOR AND OMER REINGOLD. On the Construction of Pseudorandom Permutations: Luby-Rackoff Revisited. *Journal of Cryptology*, 12(1):29–66, 1999.
- [18] Data Encryption Standard, FIPS-Pub.46. National Bureau of Standards, U.S. Department of Commerce, Washington D.C., January 1977.
- [19] JACQUES PATARIN. *Etude des Générateurs de Permutations Basés sur le Schéma du DES*. PhD thesis, Université Paris VI, November 1991.
- [20] JACQUES PATARIN. How to Construct Pseudorandom and Super Pseudorandom Permutations from one single Pseudorandom Function. In *Advances in Cryptology – EURO-CRYPT '92*, Rainer A. Rueppel, editor, volume 658 of LNCS, pages 256–266, 1992.

- [21] JACQUES PATARIN. Luby-Rackoff: 7 Rounds are Enough for $2^{n(1-\varepsilon)}$ Security. In *Advances in Cryptology – CRYPTO '03*, Dan Boneh, editor, volume 2729 of LNCS, pages 513–529, 2003.
- [22] JACQUES PATARIN. Security of Random Feistel Schemes with 5 or More Rounds. In *Advances in Cryptology – CRYPTO '04*, pages 106–122, 2004.
- [23] JACQUES PATARIN. On Linear Systems of Equations with Distinct Variables and Small Block Size. In *Proceedings of the 8th International Conference on Information Security and Cryptology (ICISC)*, volume 3935. Springer-Verlag, 2005.
- [24] ZULFIKAR RAMZAN. *A Study of Luby-Rackoff Ciphers*. PhD thesis, Massachusetts Institute of Technology (MIT), 2001.
- [25] R. RIVEST, M. ROBSHAW, R. SIDNEY, AND Y. YIN. The RC6 Block Cipher. Version 1.1, August 1998.
- [26] PHILLIP ROGAWAY. Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC. In *Advances in Cryptology - ASIACRYPT 2004*, Pil Joong Lee, editor, volume 3329, pages 16–31. Springer-Verlag Berlin Heidelberg, 2004.
- [27] B. SCHNEIER. Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish). In *Fast Software Encryption, Cambridge Security Workshop Proceedings*, pages 191–204. Springer-Verlag, 1994.
- [28] RICH SCHROEPPPEL. The Hasty Pudding Cipher. Available at <http://www.cs.arizona.edu/rcs/hpc/>, 1999.
- [29] A. SORKIN. LUCIFER: A Cryptographic Algorithm. In *Cryptologia*, pages 22–35, 1984.
- [30] Standard Architecture for Encrypted Shared Storage Media, IEEE Project 1619 (P1619). Available at <http://ieee-p1619.wetpaint.com/>.
- [31] XIAOYUN WANG, ANDREW YAO, AND FRANCES YAO. Improving the SHA-1 attack from 2^{69} to 2^{63} Operations. Rump Session Crypto 2005, 2005.

VITA

Elizabeth Ann Crump

Elizabeth Ann Crump was born on August 16th, 1983 in Waynesboro, Virginia to her parents James Edward Crump Jr. and Kathleen O'Connor Crump. She graduated magna cum laude from Robert E. Lee High School in 2001. In 2005, she earned her bachelors degree from the University of Mary Washington where she graduated magna cum laude with honors. She entered the department of computer science at the College of William and Mary in the fall of 2005 and is continuing to pursue her Ph.D. This thesis was defended on April 11th, 2007.