Provable Secure Scalable Block Ciphers

Dissertation
zur Erlangung des Grades
eines Doktors der Naturwissenschaften

dem Fachbereich 6 (Mathematik)
der Universität Duisburg-Essen
vorgelegt von

Lenka Fibíková
aus Bratislava (Slowakische Republik)

im Juli 2003

Die Disputation fand am 29. Oktober 2003 statt.

Vorsitzender:
Prof. Dr. Jürgen Herzog (Universität Duisburg-Essen, Deutschland)

Gutachter:
Prof Dr. Trung van Tran (Universität Duisburg-Essen, Deutschland)
Prof Dr. Spyros S. Magliveras (Florida Atlantic University, USA)

Zusammenfassung

Beweisbare Sicherheit und Skalierbarkeit sind zwei wünschenswerte Eigenschaften einer Blockchiffre. Die erste garantiert, dass die Chiffre unsere Erwartung der Sicherheit verschlüsselter Daten erfüllt. Die Skalierbarkeit macht die Nutzung der Chiffre einfacher und ermöglicht Anpassung des Sicherheitsniveaus an die gegenwärtigen Anforderungen durch Änderung gewisser Parameter. In dieser Dissertation untersuchen wir die beweisbare Sicherheit von drei skalierbaren Blockchiffren (Feistel-Chiffre, TST und IDEA) im Random Oracle Model.

Die Dissertation besteht aus zwei Teilen. Der erste Teil ist eine Einführung in die Theorie der beweisbaren Sicherheit. Bis jetzt existieren in der Literatur nur wenige Artikel, welche sich mit der Theorie der beweisbaren Sicherheit von Verschlüsselungsverfahren befassen. Zusätzlich verwenden sie unterschiedliche Sicherheitsmodelle und Begriffe der Ununterscheidbarkeit. Im ersten Teil der Dissertation wird die Theorie der beweisbaren Sicherheit vereinheitlicht und vervollständigt. Dabei werden das uniforme Modell und Vaudenays Begriff von Ununterscheidbarkeit verwendet. Wir illustrieren die Methoden am Beispiel der Analyse des asymmetrischen Feistel-Netzwerks. Im zweiten Teil wird diese Theorie angewandt, um die Sicherheit von zwei anderen skalierbaren Blockchiffren, nämlich TST und IDEA, zu analysieren.

Das erste Kapitel des ersten Teils (Kapitel 2) beschreibt die grundlegenden Begriffe der beweisbaren Sicherheit und führt das Sicherheitsmodell sowie die mathematischen Grundlagen ein. Das darauf folgende Kapitel diskutiert allgemeine Angriffe, nämlich die Known-Plaintext-Attack, die (Adaptive-)Chosen-Plaintext-Attack, die (Adaptive-)Chosen-Ciphertext-Attack und die (Adaptive-)Chosen-Plaintext-Ciphertext-Attack. Die Matrixnormen, die mit den individuellen Angriffen im Zusammenhang stehen, und die, die zu den oberen Grenzwerten der “Advantage” der Angriffe führen, werden hergeleitet. Die Beweise der Sicherheit für einige der Angriffe werden am Beispiel des asymmetrischen Feistel-Netzwerks illustriert.

Es wird oft versucht, einen iterativen Angriff durchzuführen, indem eine einfache Angriffsmethode auf eine Blockchiffre mehrfach angewandt wird. Ein iterativer Angriff ist offensichtlich stärker als ein Einfacher. Eine andere Möglichkeit, einen stärkeren Angriff zu erreichen, ist, mehrere einfache Angriffe nacheinander durchzuführen. Dabei ist die natürliche Frage, welcher Anteil der Advantage in dieser Weise erhöht werden könnte. In Kapitel 4 überprüfen wir die kombinierten Angriffe und leiten die oberen Schranken ihrer Advantage her. Die bekanntesten iterativen Angriffe sind differentielle und lineare Kryptoanalyse. Aufgrund ihrer großen Bedeutung werden sie getrennt behandelt.

Da die Blockgröße der Blockchiffre viel kürzer als die zu verschlüsselnden Nachrichten ist, werden Methoden benötigt, welche lange Daten zu verarbeiten ermöglichen. Einige solche Methoden wurden im NIST-FIPS 81 Standard vorgeschlagen. Das letzte Kapitel des ersten Teils analysiert diese Methoden, sowie eine modifizierte Methode von Diffie, und evaluiert deren Sicherheit.

Im zweiten Teil der Dissertation wird die Sicherheit der skalierbaren Blockchiffren behandelt. Es werden zwei Methoden der Skalierbarkeit — Skalierbarkeit der Chiffre durch Anpassung der Primitive und Skalierbarkeit durch Anpassung der Struktur — eingehend untersucht, und dabei wird die Sicherheit von zwei skalierbaren Verfahren, TST und IDEA, hergeleitet.

Das erste skalierbare Verfahren TST wurde in [8] eingeführt. Es basiert im wesentlichen auf einem modifizierten asymmetrischen Feistel-Netzwerk. Da die Sicherheit des asymmetrischen Feistel-Netzwerkes im ersten Teil bereits behandelt wurde, besprechen wir hier kurz die Auswirkung von Änderungen der Struktur auf die Sicherheit des gesamten TST-Verfahrens und fokussieren uns dann auf die Sicherheit seiner Primitiven und auf deren Beitrag zur Sicherheit der Chiffre. Genauer zeigen wir, dass eine in TST verwendete Hashfunktion schwach ist, und dass das Hinzufügen einer anderen Funktion in das Feistel-Netzwerk nicht den genügenden Ausgleich für die Schwäche liefert. Weiter zeigen wir, dass wenn eine gute Hashfunktion verwendet wird, die P-Box in TST nicht erheblich zur Sicherheit beiträgt, und damit entfernt werden kann. In dieser Weise wird das Schema vereinfacht. Anschließend analysieren wir andere Hashfunktionen und deren Anwendung im Verfahren. Wir zeigen, wie die beste Hashfunktion im Hinblick auf die Sicherheit des gesamten Verfahrens gewählt werden kann.

Die IDEA-Chiffre ist eine der bekanntesten Chiffren, die nicht auf einem Feistel-Netzwerk basieren. Jedoch ist die Skalierbarkeit ihrer Primitiven begrenzt, so dass die Blockgröße der Chiffre nicht 64 Bits übersteigen kann, was gegenwärtigen Anforderungen nicht gerecht wird. In Kapitel 8 untersuchen wir zuerst die Sicherheit des IDEA-Verfahrens, dann zeigen wir, wie aus der IDEA-Chiffre neue skalierbare Chiffren konstruiert werden können. Wir stellen zwei skalierbare Verfahren vor: das erste hat eine parallele Struktur unter Verwendung des zugrundeliegenden IDEA-Verfahrens im größtmöglichen Umfang, das zweite ist seriell und verwendet das IDEA-Verfahren nur einmal pro Runde. Wir evaluieren die Zahl der Runden, die notwendig sind, um Pseudorandomness und Super-Pseudorandomness sicherzustellen.

Die Dissertation schließt mit vier Anhängen. Die ersten Zwei listen die verwendeten Symbole und die Akronyme, der Dritte gibt Eigenschaften der verwendeten Matrixnormen an, und der Letzte enthält eine Anzahl von Lemmas, die in mehreren Beweisen eingesetzt werden.

Ich habe diese Arbeit selbständig verfasst und dabei keine anderen als die in der Literaturliste aufgeführten Hilfsmittel benutzt.

Lenka Fibíková
Essen, Juli 2003

Acknowledgements

First of all, I would like to thank my supervisor Prof. Tran van Trung for the possibility to work on my PhD thesis at the Institute for Experimental Mathematics of the University of Essen (now University Duisburg-Essen). I would further like to thank Prof. Han Vinck, the head of the Digital Communication Group, for the pleasant atmosphere he was able to create in our working group. I am thankful to both of them for the possibility to work on interesting and challenging projects during my study here which gave me the opportunity to apply my theoretical knowledge to practical security problems.

This thesis would not be as it is without three people: Valér Čanda, one of the authors of TST, who gave me insight to the cipher for its further analysis and later motivated my research on IDEA, Jozef Vyskoč with his numerous remarks and suggestions, and Oliver Meili with his invaluable and patient help at the finalization of the thesis.

Finally, I would like to thank my colleagues from our group — Lejla Batina, Valér Čanda, Jürgen Häring, Yuan Luo, Sosina Martirosyan, Oliver Meili, Chaichana Mitrpant, and Tadashi Wadayama — for their support, friendship, and contribution to the pleasant working environment in the group.

Contents

Chapter 1

Introduction

1.1 Short History of Provable Security

Before the second world war, security of encryption was usually based on secrecy of the ciphers. The rapid development of communication and spying technologies in the last century caused extensive research of ciphers with keys.

As early as 1949, theoretical security of cryptosystems was studied by Shannon [20]. He worked within an ideal model assuming that cryptanalysts have unlimited time and computation power available, and calculated the amount of information one may get about a plaintext from a ciphertext depending on its length and on the statistical structure of the plaintext language. He introduced the notion of perfect secrecy, and proved that the Vernam cipher (one-time pad) has this property. Except that the model is very restrictive, the (information-theoretic) approach led already for very simple ciphers, like a simple substitution, to complicated results, and thus it is very difficult to apply.

Absence of a practical method for evaluation of security of cryptosystems caused ad-hoc design of cryptosystems based on good ideas of their authors. Moreover, cryptosystems were considered to be secure until they were broken. This turned into a "ball game" between cryptographers and cryptanalysts — the first designed a cipher, the second found a way to break it, then the first tried to improve it or designed another one and so forth in a never-ending cycle. At the best some consensus emerged that good cryptosystems exhibit some common properties, like the avalanche criteria etc.

A turn arrived in 1984 when Goldwasser and Micali introduced the idea of *provable security* [11]. They developed it in the context of asymmetric encryption, but it soon spread to other areas of cryptography. The first step of proving a cryptographic scheme to be secure is to create a formal adversarial model and define what it means for a scheme to be secure. Given this basis, a particular scheme can be analyzed in terms of meeting the definition. The idea of provable security had great potential, but it did not have an actual impact on cryptography for several reasons: First, the proposed method actually did not prove anything, the idea was to reduce the problem of security of a scheme to the problem of security of the underlying cryptographic primitives. However, in symmetric cryptography, block ciphers are the most used primitives, and there was no tool for proving their security. Second, using public-key primitives leads to inefficient constructions. And at last, the method only has asymptotic results about the infeasibility of breaking a scheme in polynomial time; it does not quantify *how much* a scheme is secure. [1]

In 1988, Luby and Rackoff published their famous article [14], in which they proved security of the idealized Feistel scheme. They used the model of indistinguishability, also known as the Turing test. There were later several generalizations and applications of their results. One of the most significant was the simplification of the proof by Ueli M. Maurer [16]. While Luby and Rackoff worked in the non-uniform model using Boolean circuits for the definition of distinguishers and the complexity-theoretic approach for evaluation of the complexity of a successful distinguisher, Maurer made use of the uniform model and probability theory to evaluate the probability of success of a distinguisher. The Maurer's approach brought much simplification to provable security. It was later shown in [12] that the non-uniform model is stronger than the uniform one, however, there is no practical example of a security theorem proved in the non-uniform model, which cannot be proved in the uniform one.

In the 90's, the technique of the provable security was improved by Bellare and Rogaway. They introduced the practice-oriented provable security, also called concrete security, and formalized the *random oracle model* (it was previously used intuitively by different authors, also by Luby and Rackoff), and defined several notions of indistinguishability. The concept of the random oracle model is to build a secure system in two steps: First, one designs an idealized system, where all underlying cryptographic primitives are substituted by perfect random functions and permutations (i.e. random oracles), and proves security of this idealized system. Next, one replaces the random oracles by “good” cryptographic functions and permutations, and obtains an implementation of the ideal system in the “real world”, where random oracles do not exist. Since the security of the primitives is not evaluated, the second substitution decreases credibility of the whole system; however, this method brought efficiency into provable security and some security guarantees — although not at the same level as in the standard provable security approach, but superior to those provided by totally ad-hoc system design. Furthermore, this approach is non-asymptotic, based on the idea to treat cryptographic primitives as finite pseudo-random function families (since the key randomizes the behavior of the primitive), and enables one to quantify security in term of how much computation of the adversary the scheme can withstand in a certain type of attack. It also enables to compare different schemes, rather than simply stating whether they are secure or non-secure.

This method was further elaborated by Vaudenay to the decorrelation theory. His original goal was to design a tool for proving security against differential and linear cryptanalysis, however, it turned out that it is more general and can be used for different types of attacks [21]. The main idea is to construct a few primitives with small distance from a perfect random function/permutation (decorrelation modules) and to plug them into any regular cryptographic scheme. The distance of a function/permutation from a perfect random function/permutation is measured by norms on matrix sets since matrices are used to represent the probability that a particular input is mapped to a particular output. Different types of attacks induce application of different types of norms. Generally, the stronger the attack, the more sophisticated the matrix used to measure the distance has to be.

The random oracle model is very strong tool, and may be used to evaluate security of the main scheme; the construction of the basic decorrelation modules eventually eliminates the loose end of the random oracle model. Thus, by choosing a particular norm and corresponding decorrelation modules, and by plugging them into a cryptographic scheme proved to be secure in the random oracle model, one can obtain a system secure against the particular type of attack with no further conditions.

The decorrelation theory is a nice tool for designing provable secure ciphers and cryptographic functions; however, its use for proving security of an existing cipher has a shortcoming: When one proves that a cipher is secure in this model, then it is secure in a very strong notion of not being distinguishable from a perfect random function. On the other hand, when one proves that a cipher is not secure in this model, it says only that an attacker has some (eventually high) probability to distinguish that it is not a perfectly random output; it does not say anything about how secure the cipher is against weaker attacks which are more advantageous for an attacker, like for example finding an encryption key, or being able to encrypt/decrypt a different message.

1.2 Scalability of Encryption Algorithms

One of the most difficult tasks in the conventional cryptography has always been the distribution of secret keys among all parties involved in secret communication. The solution was usually based on the personal exchange of the keys or on their distribution through a trusted third party. For obvious reasons, neither of these methods was satisfying: One had either to meet the other party personally, what may have been a problem at distant communication, or both had to entrust their secrets to someone else. The idea of the asymmetric cryptography therefore meant a big turnover: It has enabled one to communicate privately without the necessity of any mediator.

The main problem of asymmetric ciphers is often seen in their slowness. But they have another drawback which comes from their very substance and cannot be eliminated by any fast hardware or efficient algorithm: If the plaintext message space is small, the exhaustive search through all possible plaintexts unfolds the encrypted message in a reasonable time. An attacker does not need to recover any key, since anyone may have access to the encryption (public) key. With the symmetric ciphers, the combination of the unknown message and the unknown key makes the space to be searched significantly larger, and by enlargement of these parameters it may be theoretically made as large as one wishes. Therefore, the conjunction of a symmetric and an asymmetric cipher into a so called hybrid system creates a scheme which (with respect to security properties) makes use of the best of both of them: Messages are encrypted with a random session key, which takes advantage of the speed of the symmetric cipher, and of using the secret key only once; and the asymmetric cipher is used to encrypt just a random string, the session key, chosen from a sufficiently large key space.

Naturally, using the hybrid schemes requires both of the ciphers to provide equal, or at least comparable, level of security. Otherwise, attacks may be aimed against the weaker one. Consequently, the whole system is only as secure as the weaker part. Here, flexibility plays an important role. Particularly, unlike the asymmetric algorithms which are scalable by nature, symmetric ciphers have usually a fixed key and block size. (Although it is possible to use a stream cipher for the symmetric encryption in hybrid systems, block ciphers are commonly used.) In asymmetric ciphers, the size of the plaintext block is given by a key parameter, e.g. by the size of the underlying finite field or group. By contrast, the block and key size of a block cipher may not necessarily be equal, but the security of the system depends also on their mutual rate. Therefore, we assume both the block and key size as essential parameters of block ciphers.

Since the minimal size of the security parameters necessary to ensure the appropriate level of security grows with technology progress, after some time it must happen that a symmetric algorithm with a fixed key and block size does not provide sufficient security any more and has to be replaced. It means replacing/reprogramming at least part of the cryptographic system. Therefore, the wide application of the hybrid schemes naturally calls for symmetric ciphers scalable to a similar extent as asymmetric ones. Changing view and increased preferences given to the scalability have been nicely illustrated with the requirements for AES, where the proposed architecture of the cipher was required to provide three different key sizes.

There is not much research done in the field of scalable block ciphers. Some scalable Feistel-like structures were studied in [27], and [22]. In the first case using the non-uniform model of Luby and Rackoff, in the other one in the context of the AES candidates. Some research oriented to scalability was done by Valér Čanda and Tran van Trung, and led to a design of a practical scalable cipher TST based on an unbalanced Feistel network [8].

1.3 Thesis Overview

The thesis is divided into two parts. The first part provides a general introduction to provable security. There are several papers devoted to provable security, however using different models of security and different notions of indistinguishability. In the first part, we unify and complete the theory of provable security using the uniform model and Vaudenay's notion of indistinguishability. We illustrate the methods on the example of analysis of the unbalanced Feistel networks. In the second part, we use the theory to analyze security of two scalable block ciphers: TST and IDEA.

The first chapter of the first part (Chapter 2) explains the basic terms of provable security, and introduces the security model and mathematical tools used in the rest of the thesis. The next chapter is devoted to the general attacks, namely to the known plaintext attack, the (adaptive) chosen plaintext attack, the (adaptive) chosen ciphertext attack, and the (adaptive) chosen plaintext-ciphertext attack. The matrix norms associated with the individual attacks and upper bounds on the success of the attacks are derived. The security proofs for some of the attacks are illustrated on example of the unbalanced Feistel scheme.

Having a simple attack against a cipher, one may try to repeat it several times with hope that the resulting iterated attack will be stronger than the underlying one. Another possibility is to perform several distinct attacks in sequence. The natural question is, how much improvement one can get in this way. In Chapter 4, we examine the composed attacks and derive the upper bounds on their success depending on the strength of the underlying attacks. The best known attacks of the iterative type are differential and linear cryptanalysis. Because of their popularity, we discuss them separately.

Since the block size of block ciphers is usually much shorter than the expected messages to be encrypted, one needs a method how to handle the long messages. Several modes of operations were suggested in the NIST FIPS 81 standard. The last chapter of the first section is devoted to these modes and one modification proposed by Diffie, and their security is evaluated.

In the second part of the thesis, we study security of scalable block ciphers. We address two methods of scalability: scalability through primitives of the cipher and scalability through the structure, and prove security of two scalable schemes: TST and IDEA.

The first one, TST, is a scalable symmetric scheme introduced in [8]. It is based on the unbalanced Feistel scheme with some modifications. Since the security of the unbalanced Feistel network is studied in the first part of the thesis, we only shortly discuss influence of the modifications to the security of the TST scheme and focus on the security of its primitives and on their contribution to the overall security of the cipher. Namely, we show that a hash function used in the TST scheme is weak and even addition of another primitive to the unbalanced Feistel scheme does not provide sufficient compensation for the weakness. Further on, we show that if a good (strong) hash function is used in the scheme, then the new primitive does not significantly contribute to the security, and may be removed thus simplifying the scheme. Subsequently, we will investigate other hash functions and their use in the scheme and show how one can select the best one with respect to security of the overall scheme.

The IDEA cipher is one of the most popular ciphers not based on the Feistel scheme. However, scalability of its underlying primitive (the round function) is limited so that the block size of the cipher cannot exceed 64 bits, which does not satisfy current requirements. In Chapter 8 we first discuss security of the basic scheme of IDEA, and then we show how to use it in order to build a scalable IDEA scheme. We introduce two scalable schemes — the first one has a parallel structure using the underlying primitive in the greatest possible extent, the other one is non-parallel using it only once per round — and evaluate the number of rounds necessary to ensure their pseudorandomness and super-pseudorandomness.

The thesis concludes with four appendices. The first two lists the used symbols and acronyms, the third one proves properties of the matrix norms associated with the general attacks, and the last one gives some simple lemmas employed in several proves.

Part I

Provable Security in Symmetric Key Cryptography

Chapter 2

Security Model

In this part of the thesis, we present foundations of provable security. It is mainly based on the work of Vaudenay and Bellare. We unify and complete published results, which use different models of security, into an integrated theory. For being able to track the original sources, we refer to them at the theorems taken from different articles, as well as at the theorems using methods introduced there. In the first case, we put the reference at the begin of the theorem, in the other one at the begin of the proof.

In this chapter we introduce the model of security used in the following chapters to study the security of block ciphers. It is based on three general ideas: The first is the random oracle model which enables decomposition of the proof of security into smaller tasks. Next, the distinguishing approach to attacks that defines very strong notion of security based on rather simple method. And finally decorrelation, which provides a mathematical tool for the proofs.

2.1 Random Oracle Model

In our security model we will use oracles in two different meanings. The first one is the concept of the random oracle model (ROM). It was formalized by Bellare and Rogaway [3] and its principle is substitution of building blocks of a cryptographic scheme by **random oracles**, i.e. by oracles generating perfectly random outputs.

Cryptographic schemes are built on various pseudorandom functions and permutations, which do not solve any specific cryptographic problem, but have to be put together to create a scheme. These functions and permutations are called **atomic primitives** of the scheme. For example in a Feistel cipher (e.g. DES) the round functions are its atomic primitives. On a higher level, for modes of encryption operation (ECB, CBC, etc.), the whole block cipher may be considered to be an atomic primitive of the mode (see Chapter 5).

Analyzing security of a scheme together with the specific primitives causes two problems: First, the structure of the primitives brings much complexity into the analysis; second, every change in a primitive demands reevaluation of the whole scheme. The approach of the **random oracle model** is to build a system in two steps. The first step is to design an idealized scheme where all underlying cryptographic primitives are substituted by random oracles, and prove security of this idealized scheme. Next, the random oracles are replaced by “good” cryptographic functions and permutations. In this way, one obtains an implementation of the ideal system in the “real world”, where random oracles do not exist.

> **Notation:** Since we will use also another type of oracles (see Section 2.2) in our security proofs, in the following we will use the term “perfect random function” or “perfect random permutation” instead of the “random oracle”. Further, the random oracles will always be marked with a star, usually writing F^* for a perfect random function and C^* for a perfect random permutation (perfect cipher).

2.2 Distinguishers

Using the ROM, we will work in a rather strict model — the attacker’s goal will be to distinguish a cryptographic scheme from a perfect random function in the following game: The attacker has access to an oracle which implements either the cryptographic scheme or a perfect random function — it is not known to the attacker which one, but it is known that the oracle implements the same function during the whole game. Querying the oracle a limited number of times and using its answers (and unlimited computation power), the attacker has to decide which function the oracle implements, and output 1 ("accept") if it is the encryption scheme, or 0 ("reject") if it is a perfect random function.

Since the goal of the attacker is to *distinguish* two functions, it is called a **distinguisher** [21]. A distinguisher which may query the oracle up to d times is called a ***d*-limited distinguisher** [21]. In general, the goal of a distinguisher can be to distinguish any two fixed random functions from each other.

Different types of distinguishers may be defined, depending on the type of attack they perform. For example:

- a known-plaintext-attack (KPA) distinguisher may query the oracle only with a predefined set of plaintexts and gets its ciphertexts;
- a chosen-plaintext-attack (CPA) distinguisher queries the oracle by any set of plaintexts chosen in advance;
- an adaptive-chosen-plaintext-attack (ACPA) distinguisher may choose any set of plaintexts adaptively depending on all previous answers of the oracle.

Similar distinguishers can be defined for attacks which query the oracle with ciphertexts and get plaintexts (ciphertext-attack distinguishers), as well as attacks which combine choosing plaintexts and ciphertexts (plaintext-ciphertext-attack distinguishers). The adaptive-chosen-plaintext-ciphertext-attack (ACPCA) distinguishers are the most powerful ones.

The success of a distinguisher D to distinguish a random function F from a perfect random function F^* is determined by two probabilities:

- probability of answering "accept" when the oracle implements F (a correct answer) — p_0, and
- probability of answering "accept" when the oracle implements F^* (an incorrect answer) — p_1.

The overall ability of a distinguisher D implementing an attack ATK using at most d oracle queries to distinguish the two functions is measured by the **advantage** defined as $Adv_D^{\mathrm{ATK}(d)}(F, F^*) = |p_0 - p_1|$. It is a value in the interval $\langle 0, 1\rangle$ expressing the probability that the distinguisher is able to distinguish the two functions from each other — a high advantage implies that it can distinguish them with high probability, and vice versa, if the advantage is small, the probability that it distinguishes them is small.

> **Notation:** The advantage of the best distinguisher between a random function and a perfect random function for a particular class of attacks will be denoted by $AdvF^{\mathrm{ATK}(d)}(F) = \max_D\{Adv_D^{\mathrm{ATK}(d)}(F, F^*)\}$, and similarly $AdvC^{\mathrm{ATK}(d)}(C) = \max_D\{Adv_D^{\mathrm{ATK}(d)}(C, C^*)\}$ will denote the advantage of the best distinguisher between a random permutation and a perfect random permutation.

Example 2.2.1 *Consider a simple distinguisher for a function F, which always returns "accept". Whenever the oracle implements the function F, the distinguisher "accepts", and thus $p_0 = 1$. When a perfect random function is implemented, it also always accepts, and therefore $p_1 = 1$ too. Hence, the advantage of this distinguisher is $|1 - 1| = 0$, which means that the distinguisher cannot distinguish the two functions at all.*

The example above shows that the attacker has to use a cleverer idea then always accepting in order to get nonzero probability of success. However, the goal of a designer of a cryptographic function is to make it difficult for an attacker to distinguish it from a perfect random one, even when the attacker has the very best idea how to attack the function. In other words, the advantage should be very small even for the best possible attack — in the ideal case equal to 0, which means that the attacker cannot do anything better than always accept.

Note that this model of security is very strong. When one proves that a cipher is secure in this model, then it looks like a perfect random function. On the other hand, when one proves that a cipher is not secure in this model, it says only that an attacker has some (eventually high) probability to distinguish that it is not a perfectly random output. This fact does not say anything about how secure the cipher is against weaker attacks which are more practical for an attacker, like for example finding an encryption key, or being able to encrypt/decrypt another message.

When we discuss the best d-limited distinguisher for a particular type of attacks, we will sometimes consider that $AdvF_D^{\mathrm{ATK}(d)}(F) = p_0 - p_1$. We may do this without loss of generality, because in the case that $p_0 - p_1 < 0$ for the best distinguisher D, we can construct another distinguisher D' which returns the inverse answers of D, i.e. it accepts whenever D rejects, and vice versa. For this distinguisher

$$p'_0 = \mathrm{Pr}_{D'}[\text{“accept”}|F] = \mathrm{Pr}_D[\text{“reject”}|F] = 1 - \mathrm{Pr}_D[\text{“accept”}|F] = 1 - p_0.$$

Similarly, $p'_1 = 1 - p_1$, and $AdvF_D^{\mathrm{ATK}(d)}(F) = p_1 - p_0 = p'_0 - p'_1 = AdvF_{D'}^{\mathrm{ATK}(d)}(F)$. Thus, there is another best distinguisher, for which $AdvF_{D'}^{\mathrm{ATK}(d)}(F) = p'_0 - p'_1$. By analogy, we may write that $AdvF_D^{\mathrm{ATK}(d)}(F) = p_1 - p_0$.

The following lemma makes use of this property.

Notation: Let A be a condition for the oracle queries and responses. $AdvF_D^{\mathrm{ATK}(d)}(F|A)$ will denote the advantage of the distinguisher D under the condition that the oracle queries and responses satisfy the condition A. For the advantage of a cipher we will write $AdvC_D^{\mathrm{ATK}(d)}(C|A)$

Lemma 2.2.2 *Let F be a random function, d an integer, and A a condition for the oracle queries and responses. Let* ATK *be a class of attacks. Then,*

$$AdvF^{\mathrm{ATK}(d)}(F) = AdvF^{\mathrm{ATK}(d)}(F|A) \cdot \mathit{Pr}[A] + AdvF^{\mathrm{ATK}(d)}(F|\neg A) \cdot \mathit{Pr}[\neg A].$$

Proof: Let D be the best d-limited distinguisher for the class of attacks ATK

$$\begin{aligned} AdvF_D^{\mathrm{ATK}(d)}(F) &= p_0 - p_1 = \mathrm{Pr}[\text{“accept”}|F] - \mathrm{Pr}[\text{“accept”}|F^*] \\ &= \mathrm{Pr}[\text{“accept”}|F \wedge A] \cdot \mathrm{Pr}[A] + \mathrm{Pr}[\text{“accept”}|F \wedge \neg A] \cdot \mathrm{Pr}[\neg A] \\ &\quad - \mathrm{Pr}[\text{“accept”}|F^* \wedge A] \cdot \mathrm{Pr}[A] - \mathrm{Pr}[\text{“accept”}|F^* \wedge \neg A] \cdot \mathrm{Pr}[\neg A] \\ &= (\mathrm{Pr}[\text{“accept”}|F \wedge A] - \mathrm{Pr}[\text{“accept”}|F^* \wedge A]) \cdot \mathrm{Pr}[A] \\ &\quad + (\mathrm{Pr}[\text{“accept”}|F \wedge \neg A] - \mathrm{Pr}[\text{“accept”}|F^* \wedge \neg A]) \cdot \mathrm{Pr}[\neg A] \\ &= AdvF_D^{\mathrm{ATK}(d)}(F|A) \cdot \mathrm{Pr}[A] + AdvF_D^{\mathrm{ATK}(d)}(F|\neg A) \cdot \mathrm{Pr}[\neg A] \end{aligned}$$

■

Now, we use this lemma to show how the overall advantage depends on the local quality of the function.

Theorem 2.2.3 *Let F be a random function, d an integer, and A a condition for the oracle queries and responses, under which output of the function F is undistinguishable from a perfectly random output, then for any class of attacks* ATK*:*

$$AdvF^{\mathrm{ATK}(d)}(F) \leq 1 - \mathit{Pr}[A],$$

where $\mathit{Pr}[A]$ is probability that queries of an execution of the attack satisfy the condition A.

Proof:

$$\begin{aligned} AdvF^{\mathrm{ATK}(d)}(F) &= AdvF^{\mathrm{ATK}(d)}(F|A) \cdot \mathrm{Pr}[A] + AdvF^{\mathrm{ATK}(d)}(F|\neg A) \cdot \mathrm{Pr}[\neg A] \\ &\leq 0 + \mathrm{Pr}[\neg A] = 1 - \mathrm{Pr}[A] \end{aligned}$$

■

The theorem states that if we are able to construct a function which is almost always ideal then the overall advantage will be small.

The advantage of a distinguisher expresses the probability that an attacker is able to distinguish between the random function and a perfect random one. However, often the goal of an attacker is considered to be the generation of a valid plaintext-ciphertext pair after seeing a particular number of plaintext-ciphertext pairs (a generating attack). The following theorem tell us how difficult it is to meet this task comparing to the distinguishing one.

Theorem 2.2.4 ([21]) *Let C be a cipher (a random permutation) on $\mathcal{M}$,* ATK *a class of attacks, and d an integer. If $AdvC^{\mathrm{ATK}(d)}(C) = a(d)$, then for any generating attack of the same type, which queries the oracle up to $d-1$ times and which issues a pair (x_d, y_d) such that $x_d \neq x_i$ (for all $i = 1, \ldots, d-1$), the probability that $C(x_d) = y_d$ is at most $\frac{1}{|\mathcal{M}|} + a(d)$.*

Proof: Let G be a generating attack which queries the oracle up to $d-1$ times and which issues a pair (x_d, y_d) such that $x_d \neq x_i$ for all $i = 1, \ldots, d-1$. Let p be the probability that this attack is successful.

We can construct a distinguisher D from the attack G. The following example is true for $\mathrm{ATK} = \mathrm{ACPA}$, but similar distinguishers can be defined for other types of attacks by modifying Steps 1–3.

DISTINGUISHER 2.1 ($G \longrightarrow D$): d-limited distinguisher for C

```
1. For $j = 1$ to $d - 1$ do
    1.1 Let $G$ choose a plaintext $x_j$.
    1.2 Query the oracle with $x_j$, and get $y_j = \tilde{C}(x_j)$, where $\tilde{C}$ is either $C$ or $C^*$.
2. Let $G$ calculate the pair $(x_d, y_d)$ depending on $(x_1, y_1), \dots (x_{d-1}, y_{d-1})$.
3. Query the oracle with $x_d$, and get $y'_j = \tilde{C}(x_j)$, where $\tilde{C}$ is either $C$ or $C^*$.
4. Outputs "accept" if and only if $y_d = y'_d$.
```

The probability that the distinguisher D outputs "accept" when the oracle implements C is the same as the probability, that the attack G is successful, i.e. p. The probability that the distinguisher outputs "accept" when the oracle implements a perfect cipher is $\frac{1}{|\mathcal{M}|}$, since the output of the oracle for x_d is random. Hence, $AdvC_D^{\mathrm{ATK}(d)}(C) = \left|p - \frac{1}{\mathcal{M}}\right| \leq a(d)$, and therefore $p \leq a(d) + \frac{1}{|\mathcal{M}|}$. ■

We will further study only distinguishing attacks.

2.3 Decorrelation

A **cryptographic function** can be seen as a map $F : \mathcal{K} \times \mathcal{M}_1 \rightarrow \mathcal{M}_2$, where $\mathcal{K}$ is the space of keys, $\mathcal{M}_1$ is the domain (set of plaintexts) and $\mathcal{M}_2$ is the range (set of ciphertexts). Fixing the key, we get a map with one input $F_K : \mathcal{M}_1 \rightarrow \mathcal{M}_2$ by setting $F_K(x) = F(K, x)$ for all $x \in \mathcal{M}_1$. This function takes a plaintext and returns a ciphertext corresponding to the fixed key and the plaintext. Using different keys we get a collection of maps — a **family of functions** (**function family**) [2] — where each map is associated with one key. In case that $\mathcal{M}_1 = \mathcal{M}_2$, the collection is called a **permutation family** [2]. If the key K is chosen uniformly at random from $\mathcal{K}$, F_K is a random instance of F, and F is called a **random function/permutation family** [2].

> **Notation:** For short, we will further omit the word "family", and call F a **random function/permutation**). When it is important that the function is a permutation, we will denote it by C rather than F.

A random function is **locally random of degree *d*** [16] if for every set of at most d inputs, the outputs of the random function are independent and uniformly distributed, i.e. the function behaves like a perfect random function as long as it is evaluated for at most d inputs. More formally, a random function $F : \mathcal{M}_1 \rightarrow \mathcal{M}_2$ is locally random of degree d if for all possible d-tuples of inputs $(x_1, \dots, x_d) \in \mathcal{M}_1^d$ and outputs $(y_1, \dots, y_d) \in \mathcal{M}_2^d$, $\Pr[F(x_1) = y_1, \dots, F(x_d) = y_d] = \Pr[F^*(x_1) = y_1, \dots, F^*(x_d) = y_d]$. Therefore, if an attacker is able to obtain maximally d input/output pairs of the function, he cannot distinguish whether he has outputs of the function or outputs of a perfectly random source.

The probabilities of all combinations of input and output d-tuples may be organized into a huge $|\mathcal{M}_1|^d \times |\mathcal{M}_2|^d$ matrix, denoted by $[F]^d$ and called the ***d*-wise distribution matrix** [21], i.e. $[F]^d_{X,Y} = \Pr[F(X) = Y]$, for all $X = (x_1, \dots, x_d) \in \mathcal{M}_1^d$ and $Y = (y_1, \dots, y_d) \in \mathcal{M}_2^d$. Entries of the matrix $[F]^d$ are thus real numbers from the interval $\langle 0, 1 \rangle$. Furthermore, if there is a pair of indices (i, j) such that $x_i = x_j$ and $y_i \neq y_j$ then $[F]^d_{X,Y} = 0$, since the same inputs have to be projected into the same output; and if F is a permutation, $[F]^d_{X,Y} = 0$ also when $y_i = y_j$ and $x_i \neq x_j$. Each row of the d-wise distribution matrix corresponds to the distribution of all output d-tuples $(F(x_1), F(x_2), \dots, F(x_d))$ over all possible d-tuples of $\mathcal{M}_2^d$ for a fixed multi-point $X = (x_1, \dots x_d)$. Therefore, $\sum_{Y \in \mathcal{M}_2^d} [F]^d_{X,Y} = 1$, for any $X \in \mathcal{M}_1^d$.

Let F^* be a perfect random function (a uniformly distributed random function) and C^* a perfect random permutation (a uniformly distributed random permutation). Then for any $X = (x_1, \dots x_d)$ with c pairwise different entries among x_i's and any $Y = (y_1, \dots y_d)$,

$$[F^*]^d_{X,Y} = \begin{cases} 1/|\mathcal{M}_2|^c & \text{if } \forall i, j : x_i = x_j \Rightarrow y_i = y_j \\ 0 & \text{otherwise} \end{cases}$$

and

$$[C^*]^d_{X,Y} = \begin{cases} 1/|\mathcal{M}_2|^{\underline{c}} & \text{if } \forall i, j : x_i = x_j \Leftrightarrow y_i = y_j \\ 0 & \text{otherwise} \end{cases}$$

The following lemmas show basic properties of d-wise distribution matrices:

Lemma 2.3.1 ([26]) *Let $F_1 : \mathcal{M}_1 \to \mathcal{M}_2$ and $F_2 : \mathcal{M}_0 \to \mathcal{M}_1$ be two independent random functions, and $F^* : \mathcal{M}_0 \to \mathcal{M}_2$, $F_1^* : \mathcal{M}_1 \to \mathcal{M}_2$, and $F_2^* : \mathcal{M}_0 \to \mathcal{M}_1$ be three independent perfect random functions. Then*

1. $[F_1 \circ F_2]^d = [F_2]^d \times [F_1]^d$

2. $[F_2]^d \times [F_1^*]^d = [F_2^*]^d \times [F_1]^d = [F^*]^d$

3. $[F_1 \circ F_2]^d - [F^*]^d = ([F_2]^d - [F_2^*]^d) \times ([F_1]^d - [F_1^*]^d)$

Proof:

1. $\forall X, Y : [F_1 \circ F_2]^d_{X,Y} = \Pr[(F_1 \circ F_2)(X) = Y] = \sum_Z \Pr[F_2(X) = Z \wedge F_1(Z) = Y] = \sum_Z \Pr[F_2(X) = Z] \cdot \Pr[F_1(Z) = Y] = \sum_Z [F_2]^d_{X,Z} \cdot [F_1]^d_{Z,Y} = ([F_2]^d \times [F_1]^d)_{X,Y}$

2. i. Let $x_1, ..., x_d$ be pairwise different. Then $[F^*]^d_{X,Y} = 1/|\mathcal{M}_2|^d$ for any Y and thus

$$([F_2]^d \times [F_1^*]^d)_{X,Y} = \sum_Z [F_2]^d_{X,Z} \cdot [F_1^*]^d_{Z,Y} = \frac{1}{|\mathcal{M}_2|^d} \sum_Z [F_2]^d_{X,Z} = \frac{1}{|\mathcal{M}_2|^d} = [F^*]^d_{X,Y}$$

ii. Let there is a pair (a, b) such that $x_a = x_b$. Then for all $Y = (y_1, \ldots, y_d)$ with $y_a \neq y_b$ $[F_1 \circ F_2]^d_{X,Y} = [F^*]^d_{X,Y} = 0$. Hence,

$$\begin{aligned} ([F_2]^d \times [F_1^*]^d)_{X,Y} &= \sum_{\substack{Z \\ z_a = z_b}} [F_2]^d_{X,Z} \cdot [F_1^*]^d_{Z,Y} \\ &= \sum_{\substack{Z \\ z_a = z_b}} \Pr[F_2(x_i) = z_i \wedge F_1^*(z_i) = y_i \mid \forall i \in \{1, \ldots, d\}] \\ &= \sum_{\substack{Z \\ z_a = z_b}} \Pr[F_2(x_i) = z_i \wedge F_1^*(z_i) = y_i \mid \forall i \in \{1, \ldots, d\} \setminus \{b\}] \\ &= \sum_{Z'} [F_2]^{d-1}_{X',Z'} \cdot [F_1^*]^{d-1}_{Z',Y'} = ([F_2]^{d-1} \times [F_1^*]^{d-1})_{X',Y'} \end{aligned}$$

where X', Z', Y' arise from X, Z, Y by dropping the b-th coordinate. Following this method, we can eliminate all equal pairs and then apply the step i.

3. $([F_2]^d - [F_2^*]^d) \times ([F_1]^d - [F_1^*]^d) = [F_2]^d \times [F_1]^d - [F_2]^d \times [F_1^*]^d - [F_2^*]^d \times [F_1]^d + [F_2^*]^d \times [F_1^*]^d = [F_2]^d \times [F_1]^d - [F^*]^d - [F^*]^d + [F^*]^d = [F_1 \circ F_2]^d - [F^*]^d$

■

Lemma 2.3.2 *Let C_1 and C_2 be two independent random permutations and C^* a perfect random permutation on a set $\mathcal{M}$. Then*

1. $[C_1 \circ C_2]^d = [C_2]^d \times [C_1]^d$

2. $[C_1]^d \times [C^*]^d = [C^*]^d \times [C_1]^d = [C^*]^d$

3. $[C_1 \circ C_2]^d - [C^*]^d = ([C_2]^d - [C^*]^d) \times ([C_1]^d - [C^*]^d)$

Proof: The proof is similar to the one of the previous lemma and is omitted. ■

Lemma 2.3.3 *Let C be a random permutation on a set $\mathcal{M}$. Then*

$$[C^{-1}]^d = \left[[C]^d\right]^T$$

Proof: $\forall X, Y :\ [C^{-1}]^d_{X,Y} = \Pr[C^{-1}(X) = Y] = \Pr[C(Y) = X] = [C]^d_{Y,X} = \left[[C]^d\right]^T_{X,Y}$ ■

Similarity of two random functions may be measured by distance of their distribution matrices. Given two random functions F and G from a set $\mathcal{M}_1$ to a set $\mathcal{M}_2$, an integer d, and a distance D over the space of all $\left|\mathcal{M}_1^d\right| \times \left|\mathcal{M}_2^d\right|$ matrices, the ***d*-wise decorrelation *D*-distance** [21] between F and G is $D([F]^d, [G]^d)$.

A decorrelation distance of zero means that for any multi-point $X = (x_1, \dots, x_d)$ the d-tuples of output values $(F(x_1), F(x_2), \dots, F(x_d))$ and $(G(x_1), G(x_2), \dots, G(x_d))$ have the same distribution.

The goal of cryptographers is to construct functions, which look random. In other words, they try to construct functions with minimal distance from a perfect random one. The decorrelation D-distance between a random function F and a perfect random function F^* is called the **d-wise decorrelation D-bias** [21].

> **Notation:** The decorrelation bias of a function F will be denoted by $DecF^d_D$, i.e. $DecF^d_D(F) = D([F]^d, [F^*]^d)$. If it is important that the distance is measured between a random permutation and a perfect cipher, we will denote it by $DecC^d_D(C) = D([C]^d, [C^*]^d)$.

A random function (permutation) with zero d-wise decorrelation distance from a perfect random one is said to have **perfect d-wise decorrelation**. It means, that for any pairwise different $x_1, \dots x_d$ the random variable $(F(x_1)), \dots, F(x_d))$ $((C(x_1)), \dots, C(x_d)))$ is uniformly distributed among all $(y_1, \dots, y_d)$ (among all pairwise different $(y_1, \dots, y_d)$).

We present here some examples of ciphers with a perfect decorrelation of some order.

Example 2.3.4 *[21] The Vernam Cipher has perfect* 1*-wise decorrelation.*

Proof: The Vernam Cipher is a cipher on a set $\mathcal{M}$ defined as $C(x) = x + K$, where K is uniformly distributed on $\mathcal{M}$. For all $x, y \in \mathcal{M}$ it holds:
$\Pr[C(x) = y] = \Pr[x + K = y] = \Pr[K = y - x] = 1/|\mathcal{M}| = \Pr[C^*(x) = y]$ ■

Note that although the 1-wise decorrelation of the Vernam Cipher is perfect, its 2-wise decorrelation is far from the ideal: Since $[C]^2_{X,Y} = \Pr[x_1 + K = y_1 \wedge x_2 + K = y_2] = \Pr[x_1 - x_2 = y_1 - y_2]$, for each fixed pair $X = (x_1, x_2)$, most of the values $[C]^2_{X,Y}$ are zero. More precisely, in each line (x_1, x_2) of the distribution matrix, only $|\mathcal{M}|$ entries of all $|\mathcal{M}|^2$ are nonzero, and for those $[C]^2_{X,Y} = 1/|\mathcal{M}|$, while $[C^*]^2_{X,Y} = 1/|\mathcal{M}|^2$ whenever $x_1 \neq x_2$ and $y_1 \neq y_2$.

The next example is a generalization of the Vernam cipher using two different keys:

Example 2.3.5 *[21] Let $C : \mathcal{M} \to \mathcal{M}$ be defined as follows: $C(x) = ax + b$, where $K = (a, b)$ is a key uniformly distributed in $\mathcal{M}^+ \times \mathcal{M}$. The cipher C has perfect* 2*-wise decorrelation.*

Proof: Let $X = (x_1, x_2), Y = (y_1, y_2) \in \mathcal{M}^2$.

1. Let $x_1 = x_2$ and $y_1 = y_2$ (1-wise decorrelation). Then

$$\Pr[C(X) = Y] = \Pr[ax_1 + b = y_1] = \frac{|\{(a, b) : b = y_1 - ax_1, a \in \mathcal{M}^+, b \in \mathcal{M}\}|}{|\mathcal{M}^+ \times \mathcal{M}|}$$
$$= \frac{|\mathcal{M}^+|}{|\mathcal{M}^+| \cdot |\mathcal{M}|} = \frac{1}{|\mathcal{M}|} = \Pr[C^*(X) = Y]$$

2. Let $x_1 = x_2$ and $y_1 \neq y_2$. Then $\Pr[C(X) = Y] = 0 = \Pr[C^*(X) = Y]$.
3. Let $x_1 \neq x_2$ and $y_1 = y_2$. Then
$\Pr[C(X) = Y] = \Pr[ax_1 + b = y_1 \wedge ax_2 + b = y_1] = 0 = \Pr[C^*(X) = Y]$
4. Let $x_1 \neq x_2$ and $y_1 \neq y_2$. Then

$$\begin{aligned}\Pr[C(X) = Y] &= \Pr[ax_1 + b = y_1 \wedge ax_2 + b = y_2] \\ &= \Pr\left[a = \frac{y_1 - y_2}{x_1 - x_2} \wedge b = y_2 - x_2\frac{y_1 - y_2}{x_1 - x_2}\right] \\ &= \frac{1}{|\mathcal{M}^+|} \cdot \frac{1}{|\mathcal{M}|} = \frac{1}{(|\mathcal{M}| - 1)|\mathcal{M}|} = \Pr[C^*(X) = Y]\end{aligned}$$

■

Example 2.3.6 *[21] Let $C : \mathcal{M} \to \mathcal{M}$ be defined as follows: $C(x) = a + b/(c + x)$, where $K = (a, b, c)$ is a key uniformly distributed in $\mathcal{M} \times \mathcal{M}^+ \times \mathcal{M}$, and let $1/0 = 0$ by definition. The cipher C has perfect 1-wise, and almost perfect 2- and 3-wise decorrelation.*

Proof: Let $X = (x_1, x_2, x_3), Y = (y_1, y_2, y_3) \in \mathcal{M}^3$.

1. If there is a pair (i, j) such that $x_i = x_j$ and $y_i \neq y_j$ or $y_i = y_j$ and $x_i \neq x_j$ then Pr$[C(X) = Y] = 0 =$ Pr$[C^*(X) = Y]$. (If $x_i = x_j$ then $y_i = a + b/(c + x_i) = a + b/(c + x_j) = y_j$. If $y_i = y_j$ then $x_i = c + b/(a + y_i) = c + b/(a + y_j) = x_j$.)

2. If $x_1 = x_2 = x_3$ and $y_1 = y_2 = y_3$ (1-wise decorrelation). Then

$$\begin{aligned}\text{Pr}[C(X) = Y] &= \text{Pr}\left[a + \frac{b}{c + x_1} = y_1\right] \\ &= \frac{|\{(a, b, c) : b = (y_1 - a)(c + x_1), b \in \mathcal{M}^+, a, c \in \mathcal{M}\}|}{|\mathcal{M} \times \mathcal{M}^+ \times \mathcal{M}|} \\ &= \frac{|\mathcal{M}^+| \cdot |\mathcal{M}|}{|\mathcal{M}| \cdot |\mathcal{M}^+| \cdot |\mathcal{M}|} = \frac{1}{|\mathcal{M}|} = \text{Pr}[C^*(X) = Y]\end{aligned}$$

3. Let there be two distinct x_i's and y_i's (2-wise decorrelation). Without loss of generality, we may assume $x_1 \neq x_2$ and $y_1 \neq y_2$. Then

$$\begin{aligned}\text{Pr}[C(X) = Y] &= \text{Pr}\left[a + \frac{b}{c + x_1} = y_1 \wedge a + \frac{b}{c + x_2} = y_2\right] \\ &= \frac{|\{(a, b, c) : b = (y_1 - a)(c + x_1), a = \frac{y_1(x_1+c) - y_2(x_2+c)}{x_1 - x_2}, b \in \mathcal{M}^+, a, c \in \mathcal{M}\}|}{|\mathcal{M} \times \mathcal{M}^+ \times \mathcal{M}|} \\ &= \frac{|\mathcal{M}^+|}{|\mathcal{M}| \cdot |\mathcal{M}^+| \cdot |\mathcal{M}|} = \frac{1}{|\mathcal{M}|^2} = \text{Pr}[F^*(X) = Y]\end{aligned}$$

4. Let all x_i's and all y_i's be pairwise distinct (3-wise decorrelation). Then

$$\begin{aligned}\text{Pr}[C(X) = Y] &= \text{Pr}\left[a + \frac{b}{c + x_1} = y_1 \wedge a + \frac{b}{c + x_2} = y_2 \wedge a + \frac{b}{c + x_3} = y_3\right] \\ &= \frac{1}{|\mathcal{M}|(|\mathcal{M}| - 1)|\mathcal{M}|}\end{aligned}$$

The last two probabilities only slightly differ from Pr$[C^*(X) = Y]$ (Pr$[C^*(X) = Y] -$ Pr$[C(X) = Y] = \frac{1}{|\mathcal{M}| \cdot |\mathcal{M}|^{\underline{2}}}$ and $\frac{2}{|\mathcal{M}| \cdot |\mathcal{M}|^{\underline{3}}}$ respectively).

■

The distance between two matrices may be measured by norms on matrix sets. A mapping from a set of matrices $\mathcal{A}$ to the set of real numbers is a **norm** if the following properties hold for all matrices $A, B \in \mathcal{A}$, for which the according operations make sense:

1. $||A|| = 0$ if and only if $A = 0$,

2. $||u \cdot A|| = |u| \cdot ||A||$, for any real number u,

3. $||A + B|| \leq ||A|| + ||B||$.

A norm is a **matrix norm** [26], if

4. $||A \times B|| \leq ||A|| \cdot ||B||$

For examples of different norms, see Appendix C. Given a norm $|| \cdot ||$, the distance between two functions F and G can be defined as $D_{||\cdot||}(F, G) = ||[F]^d - [G]^d||$.

Using matrix norms has the advantage of being able to determine the decorrelation of a composite function from the decorrelation of its components, as the following theorem shows.

Theorem 2.3.7 ([25]) *Let $|| \cdot ||$ be a matrix norm. Then for any independent random functions F_1 and F_3 from $\mathcal{M}_1$ to $\mathcal{M}_2$, and F_2 and F_4 from $\mathcal{M}_0$ to $\mathcal{M}_1$, the following properties hold:*

1. $DecF^d_{||\cdot||}(F_1 \circ F_2) \leq DecF^d_{||\cdot||}(F_1) \cdot DecF^d_{||\cdot||}(F_2)$

2. $\|[F_1 \circ F_2]^d - [F_1 \circ F_4]^d\| \leq DecF^d_{\|\cdot\|}(F_1) \cdot \|[F_2]^d - [F_4]^d\|$

3. $\|[F_1 \circ F_2]^d - [F_3 \circ F_4]^d\| \leq DecF^d_{\|\cdot\|}(F_1) \cdot \|[F_2]^d - [F_4]^d\| + DecF^d_{\|\cdot\|}(F_4) \cdot \|[F_1]^d - [F_3]^d\|$

Proof: Let $F^* : \mathcal{M}_0 \to \mathcal{M}_2$, $F_1^* : \mathcal{M}_1 \to \mathcal{M}_2$, and $F_2^* : \mathcal{M}_0 \to \mathcal{M}_1$ be three independent perfect random functions. Then

1. $DecF^d_{\|\cdot\|}(F_1 \circ F_2) = \|[F_1 \circ F_2]^d - [F^*]^d\| \overset{\text{Lemma 2.3.1}}{=} \|([F_1]^d - [F_1^*]^d) \times ([F_2]^d - [F_2^*]^d)\|$
   $\leq \|[F_1]^d - [F_1^*]^d\| \cdot \|[F_2]^d - [F_2^*]^d\| = DecF^d_{\|\cdot\|}(F_1) \cdot DecF^d_{\|\cdot\|}(F_2)$

2. $DecF^d_{\|\cdot\|}(F_1) \cdot \|[F_2]^d - [F_4]^d\| = \|[F_1]^d - [F_1^*]^d\| \cdot \|[F_2]^d - [F_4]^d\| \geq$
   $\|([F_1]^d - [F_1^*]^d) \times ([F_2]^d - [F_4]^d)\| =$
   $\|[F_1]^d \times [F_2]^d - [F_1]^d \times [F_4]^d - [F_1^*]^d \times [F_2]^d + [F_1^*]^d \times [F_4]^d\| \overset{\text{Lemma 2.3.1}}{=}$
   $\|[F_1]^d \times [F_2]^d - [F_1]^d \times [F_4]^d - [F^*]^d + [F^*]^d\| = \|[F_1 \circ F_2]^d - [F_1 \circ F_4]^d\|$

3. $DecF^d_{\|\cdot\|}(F_1) \cdot \|[F_2]^d - [F_4]^d\| + DecF^d_{\|\cdot\|}(F_4) \cdot \|[F_1]^d - [F_3]^d\| =$
   $\|[F_1]^d - [F_1^*]^d\| \cdot \|[F_2]^d - [F_4]^d\| + \|[F_4]^d - [F_2^*]^d\| \cdot \|[F_1]^d - [F_3]^d\| \geq$
   $\|([F_1]^d - [F_1^*]^d) \times ([F_2]^d - [F_4]^d) + ([F_1]^d - [F_3]^d) \times ([F_4]^d - [F_2^*]^d)\| =$
   $\|[F_1 \circ F_2]^d - [F_1 \circ F_4]^d + [F_1 \circ F_4]^d - [F_3 \circ F_4]^d\| = \|[F_1 \circ F_2]^d - [F_3 \circ F_4]^d\|$

■

A similar theorem holds also for composite permutations:

Theorem 2.3.8 *Let $\|\cdot\|$ be a matrix norm. Then for any independent random permutations C_1, C_2, C_3, and C_4 on $\mathcal{M}$, the following properties hold:*

1. $DecC^d_{\|\cdot\|}(C_1 \circ C_2) \leq DecC^d_{\|\cdot\|}(C_1) \cdot DecC^d_{\|\cdot\|}(C_2)$

2. $\|[C_1 \circ C_2]^d - [C_1 \circ C_4]^d\| \leq DecC^d_{\|\cdot\|}(C_1) \cdot \|[C_2]^d - [C_4]^d\|$

3. $\|[C_1 \circ C_2]^d - [C_3 \circ C_4]^d\| \leq DecC^d_{\|\cdot\|}(C_1) \cdot \|[C_2]^d - [C_4]^d\| + DecC^d_{\|\cdot\|}(C_4) \cdot \|[C_1]^d - [C_3]^d\|$

Proof: The proof is similar to the one of the previous theorem and is omitted. ■

2.4 Proof of Security

An implementation of a cryptographic scheme Ω may be considered to be a random function (or permutation) which calls another random functions and permutations — its primitives. We will denote this by $F = \Omega[F_1, \ldots, F_r, C_1, \ldots, C_s]$, or shortly $F = \Omega[F_{1,\ldots,r}, C_{1,\ldots,s}]$. The proof of security of the function F using the decorrelation theory consists of the following five steps:

1. *Definition of a class of attacks the cipher should be secure against.*

   Different types of attacks (distinguishers) can be defined. Besides the general attacks mentioned in the previous section, it is possible to define special subclasses of attacks (differential cryptanalysis, linear cryptanalysis, etc.) in order to get a more accurate evaluation of security.

   Generally, one has to keep in mind that it may not be enough to consider only one type of attack when dealing with composite ciphers. Particularly, it is necessary to distinguish between attacks against the scheme as a whole and attacks against its primitives (functions). For example, in Feistel networks (see page 71) we never need to calculate inverse of the round functions (neither during the decryption). Thus, studying a ciphertext attack against a Feistel network, we get a plaintext attack related to the underlying round functions. Table 2.1 shows what type of attack applies for underlying functions when a particular attack against the cipher is regarded.

   > **Notation:** Since the induced attack is independent from the function F — it depends only on its usage in the particular attack — ATK^+ will denote the induced attack when only F is used, ATK^- when only F^{-1} is used, and $\text{ATK}^\pm$ when both F and F^{-1} are used when attacking the scheme. On the other hand, when the usage of the function is not determined, we will write ATK_F.

Attack against $C[F]$ (ATK)	Calculation of F in the attack ATK	Induced attack against F ATK$_F$
KPA	any	KPA
(A)CPA	F	(A)CPA
(A)CPA	F, F^{-1}	(A)CPCA
(A)CCA	F	(A)CPA
(A)CCA	F^{-1}	(A)CCA
(A)CCA	F, F^{-1}	(A)CPCA
(A)CPCA	F	(A)CPA
(A)CPCA	F, F^{-1}	(A)CPCA

Table 2.1: Induced attacks against F in $C[F]$

2. *Identification of a norm corresponding with the defined class of attacks.*

   The definition of a particular class of attacks usually induces a special norm, which measures the advantage of distinguishers realizing an attack from this class (see Chapter 3). In other words, there is a norm $\|\cdot\|$ such that for any random function F, $AdvF^{\mathrm{ATK}(d)}(F) \le f(DecF^d_{\|\cdot\|}(F))$ for some function f.

3. *Evaluation of the advantage of the individual cryptographic primitives of the scheme.*

   The decorrelation of the functions $F_1, \dots, F_r$, and permutations $C_1, \dots, C_s$ is calculated separately. In more complex scheme, some of them may be decomposed in a similar way as the main scheme; then the advantage may be calculated recursively.

4. *Evaluation of the advantage of the scheme in the random oracle model.*

   The primitives of the schemes are substituted by perfect random functions $F_1^*, \dots, F_r^*$ and $C_1^*, \dots, C_s^*$ permutations, and the advantage of $\Omega[F_1^*, \dots, F_r^*, C_1^*, \dots, C_s^*]$ is calculated. This is usually a much easier task than the evaluation of the advantage of the whole scheme at once. However, Steps 3 and 4 are the most difficult, since there is no universal method for the evaluation.

5. *Proof of how the decorrelation of the underlying primitives propagates to the scheme.*

   Here, the results of Steps 2 and 3 are combined into the final advantage. The following theorems show how the advantage of composite schemes depends on the advantage of their primitives, if the function f is defined to be $f(x) = kx$ for a fixed constant k.

Theorem 2.4.1 *Let $F_1, \dots, F_r$ be r independent random functions, and $C_1, \dots, C_s$ be s independent random permutations, which are used in order to define a function $F = \Omega[F_1, \dots, F_r, C_1, \dots, C_s]$. Let the computation of the function F require a_i computations of the function F_i ($i = 1, \dots, r$), and b_i computations of the permutation C_i ($i = 1, \dots, s$). Let d be an integer, and $F' = \Omega[F_1^*, \dots, F_r^*, C_1^*, \dots, C_s^*]$, where $F_1^*, \dots, F_r^*$ are independent perfect random functions, and $C_1^*, \dots, C_s^*$ are independent perfect ciphers. If a class of attacks* ATK *is associated with a matrix norm $\|\cdot\|$ such that $AdvF^{\mathrm{ATK}(d)}(F) = k \cdot DecF^d_{\|\cdot\|}(F)$, then*

$$AdvF^{\mathrm{ATK}(d)}(F) \le AdvF^{\mathrm{ATK}(d)}(F') + \sum_{i=1}^{r} AdvF^{\mathrm{ATK}_{F_i}(a_i d)}(F_i) + \sum_{i=1}^{s} AdvC^{\mathrm{ATK}_{C_i}(b_i d)}(C_i).$$

Proof: [Similar theorems for ACPA and ACPCA only can be found in [25].]

Since the class of attacks ATK is associated with the norm $\|\cdot\|$, the best distinguisher has advantage $AdvF^{\mathrm{ATK}(d)}(F) = k \cdot DecF^d_{\|\cdot\|}(F) = k \cdot \|[F]^d - [F^*]\|$, where F^* is a perfect random function, independent from all other perfect random functions $(F_1^*, \dots, F_r^*)$.

- $DecF^d_{\|\cdot\|}(F) = \|\, [\Omega[F_{1,\dots,r}, C_{1,\dots,s}]]^d - [F^*]^d \| \le$
  $\|\, [\Omega[F_{1,\dots,r}, C_{1,\dots,s}]]^d - \left[\Omega[F^*_{1,\dots,r}, C^*_{1,\dots,s}]\right]^d \| + \|\, \left[\Omega[F^*_{1,\dots,r}, C^*_{1,\dots,s}]\right]^d - [F^*]^d \|$
- The second term $\|\, \left[\Omega[F^*_{1,\dots,r}, C^*_{1,\dots,s}]\right]^d - [F^*]^d \|$ corresponds to the best d-limited distinguisher D between F' and F^* for the class of attacks ATK. Hence,

$$\|\, \left[\Omega[F^*_{1,\dots,r}, C^*_{1,\dots,s}]\right]^d - [F^*]^d \| = \frac{1}{k}\, AdvF^{\mathrm{ATK}(d)}(F')$$

- $\| [\Omega[F_{1,\dots,r}, C_{1,\dots,s}]]^d - \left[\Omega[F^*_{1,\dots,r}, C^*_{1,\dots,s}]\right]^d \| \le$
  $\| [\Omega[F_{1,\dots,r}, C_{1,\dots,s}]]^d - \left[\Omega[F_1, F^*_{2,\dots,r}, C^*_{1,\dots,s}]\right]^d \| +$
  $\| \left[\Omega[F_1, F^*_{2,\dots,r}, C^*_{1,\dots,s}]\right]^d - \left[\Omega[F^*_{1,\dots,r}, C^*_{1,\dots,s}]\right]^d \| \le \cdots \le$
  $\sum_{i=1}^{r} \| \left[\Omega[F_{1,\dots,i}, F^*_{i+1,\dots,r}, C^*_{1,\dots,s}]\right]^d - \left[\Omega[F_{1,\dots,i-1}, F^*_{i,\dots,r}, C^*_{1,\dots,s}]\right]^d \| +$
  $\sum_{i=1}^{s} \| \left[\Omega[F_{1,\dots,r}, C_{1,\dots,i}, C^*_{i+1,\dots,s}]\right]^d - \left[\Omega[F_{1,\dots,r}, C_{1,\dots,i-1}, C^*_{i,\dots,s}]\right]^d \|$
- Each term $\| \left[\Omega[F_{1,\dots,i}, F^*_{i+1,\dots,r}, C^*_{1,\dots,s}]\right]^d - \left[\Omega[F_{1,\dots,i-1}, F^*_{i,\dots,r}, C^*_{1,\dots,s}]\right]^d \|$ corresponds to the best d-limited distinguisher D_i between $\Omega[F_{1,\dots,i}, F^*_{i+1,\dots,r}, C^*_{1,\dots,s}]$ and $\Omega[F_{1,\dots,i-1}, F^*_{i,\dots,r}, C^*_{1,\dots,s}]$ for the class of attacks ATK.

  The distinguisher D_i can be transformed into a $(a_i d)$-limited distinguisher D'_i between F_i and F^*_i for the class of attacks ATK_{F_i}. The following example shows the construction of D'_i if $\mathrm{ATK}_{F_i} = \mathrm{(A)CPA}$. Distinguishers for other attacks can be created in a similar way.

DISTINGUISHER 2.2 (D'_i): d-limited distinguisher for F_i

```
1. For $j = 1$ to $d$ do
    1.1 Let $D_i$ choose an input value $x_j$.
    1.2 Calculate subterms of $\Omega$ using the functions $F_1, \dots F_{i-1}$, $F^*_{i+1}, \dots, F^*_r$, and
        permutations $C^*_1, \dots, C^*_s$, and each time the subterm $F_i$ occurs, query the oracle
        implementing $F_i$ or $F^*_i$ with its subterm as the input. The distinguisher has to query
        the oracle $a_i$ times and at the end of the calculation it gets either
        $\Omega[F_{1,\dots,i-1}, F_i, F^*_{i+1,\dots,r}, C^*_{1,\dots,s}]$ or $\Omega[F_{1,\dots,i-1}, F^*_i, F^*_{i+1,\dots,r}, C^*_{1,\dots,s}]$
        depending on which function the oracle implements.
2. Get answer $a$ from the oracle $D_i$ for $(x_1, y_1), \dots, (x_d, y_d)$
3. Outputs "accept" if and only if $a$ is "accept".
```

From the definition of D'_i follows, that both D_i and D'_i have the same advantage. Since ATK depends on the norm $\| \cdot \|$,

$$\begin{aligned}
&\| \left[\Omega[F_{1,\dots,i}, F^*_{i+1,\dots,r}, C^*_{1,\dots,s}]\right]^d - \left[\Omega[F_{1,\dots,i-1}, F^*_{i,\dots,r}, C^*_{1,\dots,s}]\right]^d \| \\
&= \frac{1}{k}\, AdvF^{\mathrm{ATK}(d)}(\Omega[F_{1,\dots,i}, F^*_{i+1,\dots,r}, C^*_{1,\dots,s}], \Omega[F_{1,\dots,i-1}, F^*_{i,\dots,r}, C^*_{1,\dots,s}]) \\
&= \frac{1}{k}\, AdvF^{\mathrm{ATK}_{F_i}(a_i d)}_{D'_i}(F_i) \le \frac{1}{k}\, AdvF^{\mathrm{ATK}_{F_i}(a_i d)}(F_i)
\end{aligned}$$

Similarly, each term $\| \left[\Omega[F_{1,\dots,r}, C_{1,\dots,i}, C^*_{i+1,\dots,s}]\right]^d - \left[\Omega[F_{1,\dots,r}, C_{1,\dots,i-1}, C^*_{i,\dots,s}]\right]^d \|$ corresponds to the best d-limited distinguisher between $\Omega[F_{1,\dots,r}, C_{1,\dots,i}, C^*_{i+1,\dots,s}]$ and $\Omega[F_{1,\dots,r}, C_{1,\dots,i-1}, C^*_{i,\dots,s}]$ in the class of attacks ATK, and following the same steps, we get

$$\| \left[\Omega[F_{1,\dots,r}, C_{1,\dots,i}, C^*_{i+1,\dots,s}]\right]^d - \left[\Omega[F_{1,\dots,r}, C_{1,\dots,i-1}, C^*_{i,\dots,s}]\right]^d \| \le \frac{1}{k}\, AdvC^{\mathrm{ATK}_{C_i}(b_i d)}(C_i)$$

- Combining the previous results, we get:

$$DecF^d_{\|\cdot\|}(F) \le \frac{1}{k}\, AdvF^{\mathrm{ATK}(d)}(F') + \sum_{i=1}^{r} \frac{1}{k}\, AdvF^{\mathrm{ATK}_{F_i}(a_i d)}(F_i) + \sum_{i=1}^{s} \frac{1}{k}\, AdvC^{\mathrm{ATK}_{C_i}(b_i d)}(C_i)$$

and thus

$$\begin{aligned}
AdvF^{\mathrm{ATK}(d)}(F) &= k \cdot DecF^d_{\|\cdot\|}(F) \\
&\le AdvF^{\mathrm{ATK}(d)}(F') + \sum_{i=1}^{r} AdvF^{\mathrm{ATK}_{F_i}(a_i d)}(F_i) + \sum_{i=1}^{s} AdvC^{\mathrm{ATK}_{C_i}(b_i d)}(C_i)
\end{aligned}$$

■

A similar theorem holds also for permutations:

Theorem 2.4.2 *Let $F_1, \ldots, F_r$ be r independent random functions, and $C_1, \ldots, C_s$ s independent random permutations, which are used in order to define a permutation $C = \Omega[F_1, \ldots, F_r, C_1, \ldots, C_s]$. Let the computation of the permutation C require a_i computations of the function F_i ($i = 1, \ldots, r$), and b_i computations of the permutation C_i ($i = 1, \ldots, s$). Let d be an integer, and $C' = \Omega[F_1^*, \ldots, F_r^*, C_1^*, \ldots, C_s^*]$, where $F_1^*, \ldots F_r^*$ are independent perfect random functions, and $C_1^*, \ldots C_s^*$ are independent perfect ciphers. If a class of attacks* ATK *is associated with a matrix norm $\|\cdot\|$ such that $AdvC^{\mathrm{ATK}(d)}(C) = k \cdot DecC^d_{\|\cdot\|}(C)$, then*

$$AdvC^{\mathrm{ATK}(d)}(C) \leq AdvC^{\mathrm{ATK}(d)}(C') + \sum_{i=1}^{r} AdvF^{\mathrm{ATK}_{F_i}(a_i d)}(F_i) + \sum_{i=1}^{s} AdvC^{\mathrm{ATK}_{C_i}(b_i d)}(C_i).$$

Proof: The proof is similar to the one of the previous theorem and is omitted. ■

Another way to combine functions into a more complex scheme is a composition. The following theorem evaluates decorrelation of a composite function depending of decorrelation of its components.

Theorem 2.4.3 *Let $F_1, \ldots, F_r$ be r independent random functions such that $F_i : \mathcal{M}_i \to \mathcal{M}_{i-1}$, $F = F_1 \circ \ldots \circ F_r$, and d be an integer. If a class of attacks* ATK *is associated with a matrix norm $\|\cdot\|$ such that $AdvF^{\mathrm{ATK}(d)}(F) = k \cdot DecF^d_{\|\cdot\|}(F)$, then*

$$AdvF^{\mathrm{ATK}(d)}(F) \leq k^{1-r} \prod_{i=1}^{r} AdvF^{\mathrm{ATK}(d)}(F_i).$$

Proof: From Theorem 2.3.7:

$$\begin{aligned} AdvF^{\mathrm{ATK}(d)}(F) &= k \cdot DecF^d_{\|\cdot\|}(F_1 \circ \ldots \circ F_r) \leq k \prod_{i=1}^{r} DecF^d_{\|\cdot\|}(F_i) \\ &= k \prod_{i=1}^{r} k^{-1} AdvF^{\mathrm{ATK}(d)}(F_i) = k^{1-r} \prod_{i=1}^{r} AdvF^{\mathrm{ATK}(d)}(F_i) \end{aligned}$$

■

Again, a similar theorem holds also for permutations.

Theorem 2.4.4 *Let $C_1, \ldots, C_r$ be r independent random permutations on $\mathcal{M}$, $C = C_1 \circ \ldots \circ C_r$, and d be an integer. If a class of attacks* ATK *is associated with a matrix norm $\|\cdot\|$ such that $AdvC^{\mathrm{ATK}(d)}(C) = k \cdot DecC^d_{\|\cdot\|}(C)$, then*

$$AdvC^{\mathrm{ATK}(d)}(C) \leq k^{1-r} \prod_{i=1}^{r} AdvC^{\mathrm{ATK}(d)}(C_i).$$

Proof: The proof is similar to the one of the previous theorem and is omitted. ■

Note that the bounds given by Theorems 2.4.1–2.4.4 are not tight.

Splitting the proof into the steps described above enables one to separate the design of a cipher into several independent tasks. It also enables quantification of security in terms of how much computation of the adversary the cipher can withstand in a certain type of attack, and comparison of different ciphers as more or less secure than others.

Chapter 3

General Attacks

In this chapter we discuss the following classes of attacks:

- known plaintext attack,
- chosen plaintext attack,
- chosen ciphertext attack,
- chosen plaintext-ciphertext attack,
- adaptive chosen plaintext attack,
- adaptive chosen ciphertext attack,
- adaptive chosen plaintext-ciphertext attack,

For each class of attacks, we derive a norm which determines the advantage of the best distinguisher from the class (Step 2 of the design methodology introduced in Section 2.4)), and show how the advantage depends on the norm. We also evaluate advantage of the attacks against mixed unbalanced Feistel networks (for more details about unbalanced Feistel networks see page 71) in the random oracle model (Step 4 of the design methodology).

Applying any attack, an attacker has a possibility to obtain d plaintext-ciphertext pairs from an oracle and has to decide whether the oracle implements a particular random function or a perfect random one. The individual attacks differ in the way how the attacker obtains the pairs. In general, the distinguisher constructs (en bloc, or adaptively, depending on the type of attack it performs) a sequence of queries $Q = (q_1, \ldots, q_d)$, and gets a sequence of responses $R = (r_1, \ldots, r_d)$ from the oracle. To be able to handle any attack, we can define the query as a pair $q_k = (0, x_k)$ if the distinguisher queries with a plaintext — in this case it gets a ciphertext $r_k = y_k$ as the response from the oracle; or a pair $q_k = (1, y_k)$ if the distinguisher queries with a ciphertext, and gets a plaintext $r_k = x_k$ as the response. Note that plaintext attacks always query with $q_k = (0, x_k)$; ciphertext attacks always query with $q_k = (1, y_k)$; only plaintext-ciphertext attacks may use both types of queries.

> **Notation:** For simplicity, a **trace** τ will denote the sequence of all queries and responses $\tau = (Q, R)$ occurred during the attack; $X_\tau = (x_1, \ldots, x_d)$ the sequence of plaintexts, and $Y_\tau = (y_1, \ldots, y_d)$ the sequence of ciphertexts which the distinguisher can extract from τ.

Remember that the only difference between an adaptive attack and its non-adaptive form is that the non-adaptive attack has to choose all queries before it can access the oracle, while the adaptive one can adapt its choice at each query based on its previous queries and responses from the oracle.

We will further assume that queries to the oracle are always pairwise different. If there is an oracle D which chooses an entry more than once, we can construct another oracle D', which replaces repeated queries with some new values, but returns the same answers as D — i.e. it ignores the new responses. The distinguisher D' has thus the same advantage as D, and satisfies the requirements that it queries with different entries.

3.1 General Bounds

Consider a random function F, and a perfect random function F^*, each from a set $\mathcal{M}_1$ to a set $\mathcal{M}_2$. During the attack, a d-limited distinguisher D between F and F^* obtains a d-tuple of plaintexts X_τ and a d-tuple of ciphertexts Y_τ, and it has to decide which function (F or F^*) was implemented. It outputs “accept” when it concludes it was F, or "reject" otherwise. Let $\mathcal{A} \subseteq \mathcal{M}_1^d \times \mathcal{M}_2^d$ be the set of all pairs (X_τ, Y_τ), for which the distinguisher outputs "accept". The probability that it outputs "accept" when the oracle implements the function F is

$$p_D = \sum_{\tau=(Q,R)} 1_{(X_\tau,Y_\tau)\in\mathcal{A}} \Pr[Q] \Pr[F(X_\tau) = Y_\tau] = \sum_{\tau=(Q,R)} 1_{(X_\tau,Y_\tau)\in\mathcal{A}} \Pr[Q] \cdot [F]^d_{X_\tau,Y_\tau},$$

where $\Pr[Q]$ is the probability that the distinguisher queries with Q, and $\Pr[F(X_\tau) = Y_\tau]$ is the probability that it gets R as the response. Similarly, the probability that the distinguisher D outputs "accept" when the oracle implements a perfect random function F^* is

$$p^*_D = \sum_{\tau=(Q,R)} 1_{(X_\tau,Y_\tau)\in\mathcal{A}} \Pr[Q] \Pr[F^*(X_\tau) = Y_\tau] = \sum_{\tau=(Q,R)} 1_{(X_\tau,Y_\tau)\in\mathcal{A}} \Pr[Q] \cdot [F^*]^d_{X_\tau,Y_\tau},$$

and the advantage of D is $AdvF_D^{\mathrm{ATK}(d)}(F) = |p^*_D - p_D|$.

The following theorems upper-bound the advantage of a distinguisher realizing an attack against F. First, we consider only specialized classes of attacks, and then we introduce a general norm limiting the advantage of any distinguisher.

Theorem 3.1.1 *Let F be a random function from a set $\mathcal{M}_1$ to a set $\mathcal{M}_2$. Let $\mathcal{X}$ be the subset of $\mathcal{M}_1^d$ of all $(x_1, \ldots, x_d)$ with pairwise different entries. Let F^* be a perfect random function from $\mathcal{M}_1$ to $\mathcal{M}_2$, $p_0 = 1/|\mathcal{M}_2|^d$, and d an integer. If there is a subset $\mathcal{Y} \subseteq \mathcal{M}_2^d$, and two positive real values ε_1 and ε_2 such that:*

- $|\mathcal{Y}|p_0 \geq 1 - \varepsilon_1$ *(i.e. there are almost all d-tuples over $\mathcal{M}_2$ in the set $\mathcal{Y}$), and*
- $\forall X \in \mathcal{X},\ \forall Y \in \mathcal{Y} : [F]^d_{X,Y} \geq [F^*]^d_{X,Y}(1 - \varepsilon_2)$ *(i.e.$[F]^d_{X,Y}$ is close to $[F^*]^d_{X,Y}$ for all $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$),*

then for any class of plaintext attacks ATK,

$$AdvF^{\mathrm{ATK}(d)}(F) \leq \varepsilon_1 + \varepsilon_2.$$

Proof: [This is an extension of Lemma 4 in [22].] Let D be the best d-limited distinguisher between F and F^*. Let $\mathcal{A}$ be the set of all pairs (X, Y) such that the distinguisher outputs "accept". We may assume that all queries X which occur in $\mathcal{A}$ have pairwise different entries.

$$\begin{aligned}
AdvF_D^{\mathrm{ATK}(d)}(F) = p^*_D - p_D &= \sum_{\tau=(Q,R)} 1_{(X_\tau,Y_\tau)\in\mathcal{A}} \Pr[X_\tau] \cdot \left([F^*]^d_{X_\tau,Y_\tau} - [F]^d_{X_\tau,Y_\tau}\right) \\
&= \sum_{\substack{\tau=(Q,R)\\(X_\tau,Y_\tau)\in\mathcal{A}\\Y_\tau\in\mathcal{Y}}} \Pr[X_\tau] \cdot \left([F^*]^d_{X_\tau,Y_\tau} - [F]^d_{X_\tau,Y_\tau}\right) \\
&\quad + \sum_{\substack{\tau=(Q,R)\\(X_\tau,Y_\tau)\in\mathcal{A}\\Y_\tau\notin\mathcal{Y}}} \Pr[X_\tau] \cdot \left([F^*]^d_{X_\tau,Y_\tau} - [F]^d_{X_\tau,Y_\tau}\right) \\
&\leq \sum_{\substack{\tau=(Q,R)\\(X_\tau,Y_\tau)\in\mathcal{A}\\Y_\tau\in\mathcal{Y}}} \Pr[X_\tau] \cdot (p_0 - p_0(1 - \varepsilon_2)) + \sum_{\substack{\tau=(Q,R)\\(X_\tau,Y_\tau)\in\mathcal{A}\\Y_\tau\notin\mathcal{Y}}} \Pr[X_\tau] \cdot p_0 \\
&\leq \varepsilon_2 p_0 |\mathcal{Y}| + p_0 |\overline{\mathcal{Y}}| \leq \varepsilon_2 + \varepsilon_1
\end{aligned}$$

■

Corollary 3.1.2 *Let F be a random function from a set $\mathcal{M}_1$ to a set $\mathcal{M}_2$. Let $\mathcal{X}$ be the subset of $\mathcal{M}_1^d$ of all $(x_1, \ldots, x_d)$ with pairwise different entries. Let F^* be a perfect random function from $\mathcal{M}_1$ to $\mathcal{M}_2$. Let $p_0 = 1/|\mathcal{M}_2|^d$, and d an integer. If there is a subset $\mathcal{Y} \subseteq \mathcal{M}_2^d$, a condition A, and three positive real values ε_1, ε_2, and ε_3 such that:*

- $|\mathcal{Y}|p_0 \geq 1 - \varepsilon_1$,
- $\forall X \in \mathcal{X},\ \forall Y \in \mathcal{Y} : A \Rightarrow [F]^d_{X,Y} \geq [F^*]^d_{X,Y}(1 - \varepsilon_2)$, *and*

- *Pr*$[A] \geq 1 - \varepsilon_3$,

then for any class of plaintext attacks ATK,

$$AdvF^{\mathrm{ATK}(d)}(F) \leq \varepsilon_1 + \varepsilon_2 + \varepsilon_3.$$

Proof:

$$\begin{aligned}[F]^d_{X,Y} &= \Pr[F(X) = Y] \geq \Pr[F(X) = Y \wedge A] = \Pr[F(X) = Y|A] \cdot \Pr[A] \\ &\geq [F^*]^d_{X,Y}(1 - \varepsilon_2)(1 - \varepsilon_3) \geq [F^*]^d_{X,Y}(1 - \varepsilon_2 - \varepsilon_3)\end{aligned}$$

Now we may apply Theorem 3.1.1. ■

Similar theorems may be proved for permutations:

Theorem 3.1.3 *Let C be a random permutation on a set $\mathcal{M}$. Let $\mathcal{X}$ be the subset of $\mathcal{M}^d$ of all $(x_1, \ldots, x_d)$ with pairwise different entries. Let C^* be a perfect random permutation on $\mathcal{M}$, $p_0 = 1/|\mathcal{M}|^{\underline{d}}$, and d an integer. If there is a subset $\mathcal{Y} \subseteq \mathcal{M}^d$, and two positive real values ε_1 and ε_2 such that:*

- $|\mathcal{Y}|p_0 \geq 1 - \varepsilon_1$, *and*
- $\forall X \in \mathcal{X},\ \forall Y \in \mathcal{Y} : [C]^d_{X,Y} \geq [C^*]^d_{X,Y}(1 - \varepsilon_2)$.

then for any class of plaintext attacks ATK,

$$AdvC^{\mathrm{ATK}(d)}(C) \leq \varepsilon_1 + \varepsilon_2.$$

Proof: The proof is similar to the one of Theorem 3.1.1 and is omitted. ■

Corollary 3.1.4 *Let C be a random permutation on a set $\mathcal{M}$. Let $\mathcal{X}$ be the subset of $\mathcal{M}^d$ of all $(x_1, \ldots, x_d)$ with pairwise different entries. Let C^* be a perfect random permutation on $\mathcal{M}$, $p_0 = 1/|\mathcal{M}|^{\underline{d}}$, and d an integer. If there is a subset $\mathcal{Y} \subseteq \mathcal{M}^d$, a condition A, and three positive real values ε_1, ε_2, and ε_3 such that:*

- $|\mathcal{Y}|p_0 \geq 1 - \varepsilon_1$, *and*
- $\forall X \in \mathcal{X},\ \forall Y \in \mathcal{Y} : A \Rightarrow [C]^d_{X,Y} \geq [C^*]^d_{X,Y}(1 - \varepsilon_2)$, *and*
- *Pr*$[A] \geq 1 - \varepsilon_3$,

then for any class of plaintext attacks ATK,

$$AdvC^{\mathrm{ATK}(d)}(C) \leq \varepsilon_1 + \varepsilon_2 + \varepsilon_3.$$

Proof: The proof is similar to the one of the Corollary 3.1.2 and is omitted. ■

When dealing with ciphertext and plaintext-ciphertext attacks, we are not able to make restrictions on the set of outputs. The following theorem is actually a generalization of the previous one for $\varepsilon_1 = 0$.

Theorem 3.1.5 *Let C be a random permutation on a set $\mathcal{M}$. Let $\mathcal{X}$ be the subset of $\mathcal{M}^d$ of all $(x_1, \ldots, x_d)$ with pairwise different entries. Let C^* be a perfect random permutation on $\mathcal{M}$, and d an integer. If there is a positive real value ε such that $\forall X, Y \in \mathcal{X} : [C]^d_{X,Y} \geq [C^*]^d_{X,Y}(1 - \varepsilon)$ then for any class of attacks* ATK,

$$AdvC^{\mathrm{ATK}(d)}(C) \leq \varepsilon.$$

Proof: [This is an extension of Lemma 5 in [22].] Let $p_0 = 1/\mathcal{M}^{\underline{d}}$, D be the best d-limited distinguisher between C and C^*, and $\mathcal{A}$ the set of all pairs (X, Y) such that the distinguisher outputs "accept". As in the previous proofs, we may assume that all queries X occurred in $\mathcal{A}$ have pairwise different entries, i.e. $X_\tau, Y_\tau \in \mathcal{X}$.

$$\begin{aligned}AdvC^{\mathrm{ATK}(d)}_D(C) = p^*_D - p_D &= \sum_{\tau=(Q,R)} 1_{(X_\tau,Y_\tau)\in\mathcal{A}} \Pr[Q] \cdot \left([C^*]^d_{X_\tau,Y_\tau} - [C]^d_{X_\tau,Y_\tau}\right) \\ &\leq \sum_{\tau=(Q,R)} 1_{(X_\tau,Y_\tau)\in\mathcal{A}} \Pr[Q] \cdot (p_0 - p_0(1 - \varepsilon)) \\ &\leq \varepsilon p_0|\mathcal{Y}| \leq \varepsilon\end{aligned}$$

■

Corollary 3.1.6 *Let C be a random permutation on a set $\mathcal{M}$. Let $\mathcal{X}$ be the subset of $\mathcal{M}^d$ of all $(x_1, \ldots, x_d)$ with pairwise different entries. Let C^* be a perfect random permutation on $\mathcal{M}$, and d an integer. If there is a condition A, and two positive real values ε_1 and ε_2 such that:*

- $\forall X, Y \in \mathcal{X} : A \Rightarrow [C]^d_{X,Y} \geq [C^*]^d_{X,Y}(1 - \varepsilon_1)$*, and*
- *Pr*$[A] \geq 1 - \varepsilon_2$,

then for any class of attacks ATK,

$$AdvC^{\mathrm{ATK}(d)}(C) \leq \varepsilon_1 + \varepsilon_2.$$

Proof: The proof is similar to the one of the Corollary 3.1.2 and is omitted. ■

Consider now the norm N_∞ (see Appendix C). The following theorem shows how this norm bounds the advantage of a distinguisher.

Theorem 3.1.7 *Let F be a random function from $\mathcal{M}_1$ to $\mathcal{M}_2$, and d an integer. Then for any class of attacks* ATK,

$$AdvF^{\mathrm{ATK}(d)}(F) \leq DecF^d_{N_\infty}(F).$$

Proof: [A similar theorem can be found in [21] for CPA.]

Let $\varepsilon = DecF^d_{N_\infty}(F) = N_\infty([F]^d - [F^*]^d) = \max_{X,Y} \frac{|[F]^d_{X,Y} - [F^*]^d_{X,Y}|}{[F^*]^d_{X,Y}}$. It means that for all pairs $(X, Y) \in \mathcal{M}_1 \times \mathcal{M}_2$

$$\varepsilon \geq \left| \frac{[F]^d_{X,Y}}{[F^*]^d_{X,Y}} - 1 \right|.$$

Therefore, $[F]^d_{X,Y} \leq (1 + \varepsilon)\,[F^*]^d_{X,Y}$, and for any distinguisher D

$$\begin{aligned} p_D &= \sum_{\tau=(Q,R)} 1_{(X_\tau,Y_\tau)\in\mathcal{A}} \Pr[Q] \cdot [F]^d_{X_\tau,Y_\tau} \\ &\leq \sum_{\tau=(Q,R)} 1_{(X_\tau,Y_\tau)\in\mathcal{A}} \Pr[Q]\,(1+\varepsilon)\,[F^*]^d_{X_\tau,Y_\tau} = (1+\varepsilon)\,p^*_D \leq p^*_D + \varepsilon \end{aligned}$$

Hence, $p_D - p^*_D \leq \varepsilon$. If $p_D - p^*_D \geq 0$, then $AdvF^{\mathrm{ATK}(d)}_D(F) = p_D - p^*_D \leq \varepsilon$. Otherwise, we can construct another distinguisher D' which returns inverse answers as D, i.e. accepts whenever D rejects, and vice versa. For this distinguisher $p_{D'} = 1 - p_D$. Similarly, $p^*_{D'} = 1 - p^*_D$. Therefore, $AdvF^{\mathrm{ATK}(d)}_D(F) = p^*_D - p_D = p_{D'} - p^*_{D'}$, and since the above inequality holds for any distinguisher, it is also less or equal to ε. Consequently,

$$AdvF^{\mathrm{ATK}(d)}(F) = \max_D \left\{ AdvF^{\mathrm{ATK}(d)}_D(F) \right\} \leq \varepsilon.$$

■

A similar theorem holds also for permutations.

Theorem 3.1.8 *Let C be a cipher on $\mathcal{M}$, and d an integer. Then for any class of attacks* ATK,

$$AdvC^{\mathrm{ATK}(d)}(C) \leq DecC^d_{N_\infty}(C).$$

Proof: The proof is similar to the one of the previous theorem and is omitted. ■

Although this result is general for any class of attacks, N_∞ is not a matrix norm, thus Theorem 2.4.4 cannot be applied. In the following chapters we derive norms for individual attacks. They give exacter bounds on the advantage, and all of them are matrix norms. Furthermore, we study the attacks more generally, considering distinguishers between any two fixed random functions F_1 and F_2 with the same domain and range (we do not limit ourselves to comparison with a perfect random function).

3.2 Known Plaintext Attack

The **known plain attack** is the simplest and least powerful type of attack — the attacker may access some independent randomly chosen plaintext-ciphertext pairs. A d-limited distinguisher D for a known plaintext attack between two independent random functions F_1 and F_2 defined from a set $\mathcal{M}_1$ to a set $\mathcal{M}_2$ obtains d pairs of inputs and outputs, and has to decide which of the functions was implemented. It works as follows:

DISTINGUISHER 3.1 (KPA): d-limited known-plaintext-attack distinguisher

1. Get $X = (x_1, \ldots, x_d)$, and $Y = (F_i(x_1), \ldots, F_i(x_d))$, where $i \in \{1, 2\}$, and $x_1, \ldots, x_d$ are pairwise different and uniformly distributed on $\mathcal{M}_1$.
2. Depending on X and Y, output "accept" if you "think" the oracle implements F_1 or "reject" otherwise.

Let $\mathcal{A}$ be the set of all pairs (X, Y) such that the distinguisher outputs "accept". The probability that it outputs "accept" when the oracle implements the function F_i ($i = 1, 2$) is

$$p_i = \sum_X \text{Pr}[X] \sum_Y 1_{(X,Y)\in\mathcal{A}} [F_i]^d_{X,Y}$$

Since the distinguisher gets a random set of d pairwise different x_i's,

$$p_i = \sum_X \frac{1}{|\mathcal{M}_1|^d} \sum_Y 1_{(X,Y)\in\mathcal{A}} [F_i]^d_{X,Y} = \frac{1}{|\mathcal{M}_1|^d} \sum_{X,Y} 1_{(X,Y)\in\mathcal{A}} [F_i]^d_{X,Y}$$

Hence, the advantage of the distinguisher is

$$\begin{aligned} Adv_D^{\text{KPA}(d)}(F_1, F_2) = |p_1 - p_2| &= \frac{1}{|\mathcal{M}_1|^d} \left| \sum_{X,Y} 1_{(X,Y)\in\mathcal{A}} \left([F_1]^d_{X,Y} - [F_2]^d_{X,Y} \right) \right| \\ &\leq \frac{1}{|\mathcal{M}_1|^d} \sum_{X,Y} 1_{(X,Y)\in\mathcal{A}} \left| [F_1]^d_{X,Y} - [F_2]^d_{X,Y} \right| \end{aligned}$$

The advantage is maximal if $\mathcal{A}$ contains *all* pairs (X, Y) such that $[F_1]^d_{X,Y} - [F_2]^d_{X,Y}$ have the same sign. In that case,

$$Adv^{\text{KPA}(d)}(F_1, F_2) = \frac{1}{|\mathcal{M}_1|^d} \sum_{X,Y} 1_{(X,Y)\in\mathcal{A}} \left| [F_1]^d_{X,Y} - [F_2]^d_{X,Y} \right|$$

Theorem 3.2.1 *Let F_1 and F_2 be two independent random functions from a set $\mathcal{M}_1$ to a set $\mathcal{M}_2$, and d be an integer. Then*

$$Adv^{\text{KPA}(d)}(F_1, F_2) = \frac{1}{2\,|\mathcal{M}_1|^d} \left\| [F_1]^d - [F_2]^d \right\|_1.$$

Proof: Since $\sum_Y \text{Pr}[F_i(X) = Y] = 1$ for any X, then $\sum_{X,Y}([F_1]^d_{X,Y} - [F_2]^d_{X,Y}) = 0$, and from the definition of $\mathcal{A}$ (the terms have the same sign), it follows that

$$\sum_{X,Y} 1_{(X,Y)\notin\mathcal{A}} |[F_1]^d_{X,Y} - [F_2]^d_{X,Y}| = \sum_{X,Y} 1_{(X,Y)\in\mathcal{A}} |[F_1]^d_{X,Y} - [F_2]^d_{X,Y}|$$

Therefore,

$$\begin{aligned} \left\| [F_1]^d - [F_2]^d \right\|_1 &= \sum_{X,Y} \left| [F_1]^d_{X,Y} - [F_2]^d_{X,Y} \right| \\ &= \sum_{X,Y} 1_{(X,Y)\in\mathcal{A}} \left| [F_1]^d_{X,Y} - [F_2]^d_{X,Y} \right| + \sum_{X,Y} 1_{(X,Y)\notin\mathcal{A}} \left| [F_1]^d_{X,Y} - [F_2]^d_{X,Y} \right| \\ &= 2 \sum_{X,Y} 1_{(X,Y)\in\mathcal{A}} \left| [F_1]^d_{X,Y} - [F_2]^d_{X,Y} \right| \\ &= 2\,|\mathcal{M}_1|^d \, Adv^{\text{KPA}(d)}(F_1, F_2) \end{aligned}$$

■

Corollary 3.2.2 *Let F be a random function from $\mathcal{M}_1$ to $\mathcal{M}_2$, and d be an integer. Then*

$$AdvF^{\mathrm{KPA}(d)}(F) = \frac{1}{2\,|\mathcal{M}_1|^d}\, DecF^d_{\|\cdot\|_1}(F).$$

Similar theorems for the advantage of ciphers can be proved.

Theorem 3.2.3 *Let C_1 and C_2 be two independent ciphers on a set $\mathcal{M}$, and d be an integer. Then*

$$Adv^{\mathrm{KPA}(d)}(C_1, C_2) = \frac{1}{2\,|\mathcal{M}_1|^d}\, \left\| [C_1]^d - [C_2]^d \right\|_1$$

Proof: The proof is similar to the one of the previous theorem and is omitted. ■

Corollary 3.2.4 *Let C be a cipher on a set $\mathcal{M}$, and d be an integer. Then*

$$AdvC^{\mathrm{KPA}(d)}(C) = \frac{1}{2\,|\mathcal{M}_1|^d}\, DecC^d_{\|\cdot\|_1}(C).$$

Known Plaintext Attack Against Unbalanced Feistel Networks

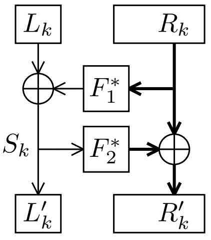
In this subsection we examine security of the two-round unbalanced Feistel network (UFN) against the known plaintext attack.

Theorem 3.2.5 *Let F_1^*, F_2^* be two independent perfect random functions, F_1^* from $\mathcal{M}_2$ to $\mathcal{M}_1$ and F_2^* from $\mathcal{M}_1$ to $\mathcal{M}_2$, $\Psi[F_1^*, F_2^*]$ a two-round UFN on $\mathcal{M} = \mathcal{M}_1 \times \mathcal{M}_2$, and d an integer. Then*

$$AdvC^{\mathrm{KPA}(d)}(\Psi[F_1^*, F_2^*]) \leq \frac{d^2}{\min\{|\mathcal{M}_1|, |\mathcal{M}_2|\}}$$

Proof: The proof is based on the technique introduced in [16].

Any d-limited known-plaintext-attack distinguisher has access to d plaintext/ciphertext pairs $(x_1, y_1), \ldots, (x_d, y_d)$. When the oracle implements the unbalanced Feistel network, the ciphertexts are calculated as depicted on the following figure.



$$x_k = [L_k, R_k]$$
$$y_k = [L'_k, R'_k]$$

$$L'_k = S_k = L_k \oplus F_1^*(R_k)$$
$$R'_k = R_k \oplus F_2^*(S_k)$$

If all R_k's are pairwise distinct, then the sequence of all S_k's (and thus also of all L'_k's) is perfectly random, because the function F_1^* is perfectly random. If all S_k's are pairwise distinct, then the sequence of R'_k's is also perfectly random, because the function F_2^* is perfectly random. If both sequences (of L'_k's and of R'_k's) are perfectly random, then sequence of $Y_k = [L'_k, R'_k]$ is also perfectly random. In this case, output of the cipher Ψ looks like a perfect cipher.

Without loss of generality, we may assume that all x_k's are pairwise distinct. Since x_k's are uniformly distributed, then for all $k \neq l$, Pr$[R_k = R_l] = \frac{1}{|\mathcal{M}_2|}$.

If $R_k = R_l$ then $L_k \neq L_l$ (since $x_k \neq x_l$), and thus $S_k = L_k + F_1^*(R_k) \neq L_l + F_1^*(R_l) = S_l$, what means that Pr$[S_k = S_l] = 0$. If $R_k \neq R_l$ then the probability that $S_k = S_l$ is $\frac{1}{|\mathcal{M}_1|}$, since $F_1^*(R_k)$ and $F_1^*(R_l)$ are independent and perfectly random. Therefore, Pr$[S_k = S_l] \leq \frac{1}{|\mathcal{M}_1|}$.

Whenever the distinguisher gets inputs $x_1, \ldots, x_d$ such that the sequence of Y_i's is perfectly random, it cannot distinguish them, and returns the same value whether the oracle implements $\Psi[F_1^*, F_2^*]$ or C^*. Hence, from Theorem 2.2.3:

$$\begin{aligned} AdvC^{\mathrm{KPA}(d)}(\Psi[F_1^*, F_2^*]) &\leq 1 - \Pr[\forall\, k \neq l : R_k \neq R_l \wedge S_k \neq S_l] \\ &= \Pr[\exists\, k \neq l : R_k = R_l \vee S_k = S_l] \\ &\leq \Pr[\exists\, k \neq l : R_k = R_l] + \Pr[\exists\, k \neq l : S_k = S_l] \\ &\leq \binom{d}{2} \frac{1}{|\mathcal{M}_2|} + \binom{d}{2} \frac{1}{|\mathcal{M}_1|} \\ &\leq 2 \binom{d}{2} \frac{1}{\min\{|\mathcal{M}_1|, |\mathcal{M}_2|\}} = \frac{d^2}{\min\{|\mathcal{M}_1|, |\mathcal{M}_2|\}} \end{aligned}$$

■

Corollary 3.2.6 (Corollary 3.2.4) *Let F_1^*, and F_2^* be two independent perfect random functions, F_1^* from a set $\mathcal{M}_1$ to a set $\mathcal{M}_2$ and F_2^* from $\mathcal{M}_2$ to $\mathcal{M}_1$, and d be an integer. Then*

$$DecC^d_{\|\cdot\|_1}(\Psi[F_1^*, F_2^*]) \leq 2\, \frac{d^2}{\min\{|\mathcal{M}_1|, |\mathcal{M}_2|\}}.$$

3.3 Chosen Plaintext Attack

The **chosen plaintext attack** is more advantageous for an attacker than the known plaintext attack, because he can choose messages which are encrypted for him. A d-limited chosen-plaintext-attack distinguisher D between two independent random functions F_1 and F_2 from a set $\mathcal{M}_1$ to a set $\mathcal{M}_2$ works as follows:

DISTINGUISHER 3.2 (CPA): d-limited chosen-plaintext-attack distinguisher [21]

1. Choose plaintext messages $X = (x_1, \ldots, x_d)$.
2. Query the oracle with X, and get $Y = (F_i(x_1), \ldots, F_i(x_d))$, where $i \in \{1, 2\}$.
3. Depending on X and Y, output “accept” if you “think” the oracle implements F_1 or “reject” otherwise.

Let $\mathcal{A}$ be again the set of all pairs (X, Y) such that the distinguisher outputs “accept”. The probability that it outputs “accept” when the oracle implements the function F_i $(i = 1, 2)$ is

$$p_i = \sum_X \Pr[X] \sum_Y 1_{(X,Y)\in\mathcal{A}} [F_i]^d_{X,Y}$$

and the advantage of the distinguisher is

$$\begin{aligned} Adv_D^{\mathrm{CPA}(d)}(F_1, F_2) = |p_1 - p_2| &= \left| \sum_X \Pr[X] \sum_Y 1_{(X,Y)\in\mathcal{A}} \left([F_1]^d_{X,Y} - [F_2]^d_{X,Y}\right) \right| \\ &\leq \sum_X \Pr[X] \sum_Y 1_{(X,Y)\in\mathcal{A}} \left|[F_1]^d_{X,Y} - [F_2]^d_{X,Y}\right| \end{aligned}$$

The advantage is maximal when $\mathcal{A}$ contains *all* pairs (X, Y) such that $[F_1]^d_{X,Y} - [F_2]^d_{X,Y}$ have the same sign and when $\Pr[X] = 1$ for the best choice. In that case,

$$Adv^{\mathrm{CPA}(d)}(F_1, F_2) = \max_X \sum_Y 1_{(X,Y)\in\mathcal{A}} \left|[F_1]^d_{X,Y} - [F_2]^d_{X,Y}\right|.$$

Lemma 3.3.1 *Let D be the best d-limited chosen-plaintext-attack distinguisher. The distinguisher can always choose $X = (x_1, \ldots x_d)$ with pairwise distinct x_i's in order to obtain the best advantage.*

Proof: Without loss of generality we may assume that $d \leq |\mathcal{M}_1|$, so that there are enough elements to choose from. Let $X = (x_1, \ldots x_d)$ is such that $\sum_Y 1_{(X,Y)\in\mathcal{A}} \left|[F_1]^d_{X,Y} - [F_2]^d_{X,Y}\right|$ is maximal. Let there be $c < d$ different x_j's in X and σ be a monotone function from $\{0, \ldots, c\}$ to $\{0, \ldots, d\}$ such that all $x_{\sigma(j)}$'s are pairwise different. Further, if there is a pair (k, l) such that $x_k = x_l$ and $y_k \neq y_l$

then $[F_i]^d_{X,Y} = 0$. Otherwise, we can choose $d-c$ new inputs $x'_{c+1}, \ldots, x'_d$ distinct from $x_1, \ldots, x_c$, and since $\sum_y \Pr[F_i(x) = y] = 1$ for any fixed x,

$$\begin{aligned}
[F_i]^d_{X,Y} &= \Pr\left[\bigwedge_{j=1}^{d} F_i(x_j) = y_j\right] = \Pr\left[\bigwedge_{j=1}^{c} F_i(x_{\sigma(j)}) = y_{\sigma(j)}\right]\\
&= \sum_{y'_{c+1},\ldots,y'_d} \Pr\left[\bigwedge_{j=1}^{c} F_i(x_{\sigma(j)}) = y_{\sigma(j)} \wedge \bigwedge_{j=c+1}^{d} F_i(x'_j) = y'_j\right]\\
&= \sum_{y'_{c+1},\ldots,y'_d} \Pr\left[\bigwedge_{j=1}^{d} F_i(x'_j) = y'_j\right] = \sum_{y'_{c+1},\ldots,y'_d} [F_i]^d_{X',Y'}
\end{aligned}$$

with $x'_j = x_{\sigma(j)}$ and $y'_j = y_{\sigma(j)}$ for $j = 1, \ldots, c$. The advantage for the new choice is

$$\begin{aligned}
Adv_D^{\mathrm{CPA}(d)}(F_1, F_2) &= \sum_Y \left|[F_1]^d_{X,Y} - [F_2]^d_{X,Y}\right| = \sum_{y'_1,\ldots,y'_c} \left| \sum_{y'_{c+1},\ldots,y'_d} [F_1]^d_{X',Y'} - [F_2]^d_{X',Y'} \right|\\
&\le \sum_{Y'} \left|[F_1]^d_{X',Y'} - [F_2]^d_{X',Y'}\right|
\end{aligned}$$

Hence, we have a new choice of X with the advantage at least the same as the previous one. ■

Theorem 3.3.2 *Let F_1 and F_2 be two independent random functions from a set $\mathcal{M}_1$ to a set $\mathcal{M}_2$, and d be an integer. Then*

$$Adv^{\mathrm{CPA}(d)}(F_1, F_2) = \frac{1}{2} \left|\left|\left|[F_1]^d - [F_2]^d.\right|\right|\right|_\infty$$

Proof: Since $\sum_Y [F_i]^d_{X,Y} = 1$ for any fixed X, then $\sum_Y [F_1]^d_{X,Y} - [F_2]^d_{X,Y} = 0$, and thus from the definition of $\mathcal{A}$ (the terms have the same sign), it follows that for all X:

$$\sum_Y 1_{(X,Y)\notin\mathcal{A}} \left|[F_1]^d_{X,Y} - [F_2]^d_{X,Y}\right| = \sum_Y 1_{(X,Y)\in\mathcal{A}} \left|[F_1]^d_{X,Y} - [F_2]^d_{X,Y}\right|$$

Therefore,

$$\begin{aligned}
\left|\left|\left|[F_1]^d - [F_2]^d\right|\right|\right|_\infty &= \max_X \sum_Y \left|[F_1]^d_{X,Y} - [F_2]^d_{X,Y}\right|\\
&= \max_X \left\{ \sum_Y 1_{(X,Y)\in\mathcal{A}} \left|[F_1]^d_{X,Y} - [F_2]^d_{X,Y}\right| \right.\\
&\qquad \left. + \sum_Y 1_{(X,Y)\notin\mathcal{A}} \left|[F_1]^d_{X,Y} - [F_2]^d_{X,Y}\right| \right\}\\
&= \max_X\, 2 \sum_Y 1_{(X,Y)\in\mathcal{A}} \left|[F_1]^d_{X,Y} - [F_2]^d_{X,Y}\right|\\
&= 2 \max_X \sum_Y 1_{(X,Y)\in\mathcal{A}} \left|[F_1]^d_{X,Y} - [F_2]^d_{X,Y}\right|\\
&= 2\, Adv^{\mathrm{CPA}(d)}(F_1, F_2)
\end{aligned}$$

■

Corollary 3.3.3 *Let F be a random function, and d be an integer. Then*

$$AdvF^{\mathrm{CPA}(d)}(F) = \frac{1}{2}\, DecF^d_{|||\cdot|||_\infty}(F).$$

Similar theorems hold also for the advantage of ciphers.

Theorem 3.3.4 *Let C_1 and C_2 be two independent ciphers on a set $\mathcal{M}$, and d be an integer. Then*

$$Adv^{\mathrm{CPA}(d)}(C_1, C_2) = \frac{1}{2} \left|\left|\left|[C_1]^d - [C_2]^d.\right|\right|\right|_\infty$$

Proof: The proof is similar to the one of the previous theorem and is omitted. ■

Corollary 3.3.5 ([21]) *Let C be a cipher, and d be an integer. Then*

$$AdvC^{\mathrm{CPA}(d)}(C) = \frac{1}{2}\, DecC^{d}_{|||\cdot|||_\infty}(C)$$

The following theorem evaluates the decorrelation distance between a perfect random permutation and a perfect random function.

Theorem 3.3.6 *Let C^* be a perfect random permutation and F^* be a perfect random function on $\mathcal{M}$, and $d < \sqrt{|\mathcal{M}|}$ be an integer. Then*

$$\left|\left|\left|[C^*]^d - [F^*]^d\right|\right|\right|_\infty = 2\left(1 - \frac{|\mathcal{M}|^{\underline{d}}}{|\mathcal{M}|^d}\right) \leq \frac{d^{\underline{2}}}{|\mathcal{M}|}$$

Proof: Let X be a fixed d-tuple $(x_1, \ldots, x_d)$. Let c be the number of distinct values among $x_1, \ldots, x_d$. Then for every $Y = (y_1, \ldots, y_d)$:

$$\left|[C^*]^d_{X,Y} - [F^*]^d_{X,Y}\right| = \begin{cases} 0 & \text{if } \exists i,j : x_i = x_j \wedge y_i \neq y_j \\ \frac{1}{|\mathcal{M}|^c} & \text{if } \forall i,j : x_i = x_j \Rightarrow y_i \neq y_j, \\ & \text{but } \exists i,j : x_i \neq x_j \wedge y_i = y_j \\ \frac{1}{|\mathcal{M}|^{\underline{c}}} - \frac{1}{|\mathcal{M}|^c} & \text{if } \forall i,j : x_i = x_j \Leftrightarrow y_i = y_j \end{cases}$$

Hence,
$\sum_Y \left|[C^*]^d_{X,Y} - [F^*]^d_{X,Y}\right| = |\mathcal{M}|^{\underline{c}}\left(\frac{1}{|\mathcal{M}|^{\underline{c}}} - \frac{1}{|\mathcal{M}|^c}\right) + (|\mathcal{M}|^c - |\mathcal{M}|^{\underline{c}})\frac{1}{|\mathcal{M}|^c} = 2\left(1 - \frac{|\mathcal{M}|^{\underline{c}}}{|\mathcal{M}|^c}\right).$
Since for any $c < d$

$$\frac{\frac{|\mathcal{M}|^{\underline{c}}}{|\mathcal{M}|^c}}{\frac{|\mathcal{M}|^{\underline{d}}}{|\mathcal{M}|^d}} = \frac{|\mathcal{M}|^{d-c}}{(|\mathcal{M}| - c)\ldots(|\mathcal{M}| - d + 1)} > 1$$

(i.e. $\frac{|\mathcal{M}|^{\underline{c}}}{|\mathcal{M}|^c} > \frac{|\mathcal{M}|^{\underline{d}}}{|\mathcal{M}|^d}$), then the greatest value one can obtain is when all x_i's are different.

In that case, $2 \cdot \left(1 - \frac{|\mathcal{M}|^{\underline{d}}}{|\mathcal{M}|^d}\right) \leq 2 \cdot \frac{d^{\underline{2}}}{2|\mathcal{M}|} = \frac{d^{\underline{2}}}{|\mathcal{M}|}$. ■

Corollary 3.3.7 *Let C^* be a perfect random permutation and F^* be a perfect random function on $\mathcal{M}$, and $d < \sqrt{|\mathcal{M}|}$ be an integer. Then*

$$AdvC^{\mathrm{CPA}(d)}(F^*) = AdvF^{\mathrm{CPA}(d)}(C^*) \leq \frac{d^{\underline{2}}}{2|\mathcal{M}|}.$$

Proof: Follows from Theorem 3.3.6, and Corollary 3.3.3 or Corollary 3.3.5. ■

Chosen Plaintext Attack Against Unbalanced Feistel Networks

In the previous section we proved that in the random oracle model two-round UFNs are secure against known plaintext attacks with a reasonable number of queries. Here we show that UFNs are not secure against chosen plaintext attack.

Theorem 3.3.8 ([14]) *Let F_1 and F_2 be any two independent random functions — F_1 from a set $\mathcal{M}_2$ to a set $\mathcal{M}_1$ and F_2 from $\mathcal{M}_1$ to $\mathcal{M}_2$. Then $\Psi[F_1, F_2]$ is not secure against the chosen plaintext attack.*

Proof: There is a 2-limited chosen-plaintext-attack distinguisher between $\Psi[F_1, F_2]$ and a perfect cipher:

DISTINGUISHER 3.3 (D): 2-limited CPA distinguisher for $\Psi[F_1, F_2]$

1. Choose two plaintexts such that $x_1 = [L_1, R]$, and $x_2 = [L_2, R]$.
2. Get $y_1 = [L'_1, R'_1]$, and $y_2 = [L'_2, R'_2]$.

   If the oracle implements Ψ, then

   $$y_i = \Psi[F_1, F_2](x_i) = [L_i \oplus F_1(R), R \oplus F_2(L_i \oplus F_1(R))] = [L'_i, R'_i].$$

   XOR of the left parts $L'_1 \oplus L'_2$ of the ciphertexts is thus $L_1 \oplus L_2$.
3. If $L'_1 \oplus L'_2 = L_1 \oplus L_2$ then output “accept”, otherwise output “reject”.

If the oracle implements Ψ, then the distinguisher always "accepts", i.e. $p = 1$. If the oracle implements a perfect cipher, then the probability the distinguisher "accepts" is the same as the probability that two random ciphertexts have the same left parts, i.e. $p^* = \frac{1}{|\mathcal{M}_1|}$. The advantage of the distinguisher is thus $AdvC_D^{\mathrm{CPA}(2)}(C) = 1 - \frac{1}{|\mathcal{M}_1|}$.

By adding further plaintext, and following the same construction, a d-limited distinguisher with advantage $AdvC_D^{\mathrm{CPA}(d)}(C) = 1 - \frac{1}{|\mathcal{M}_1|^{d-1}}$ for any $d \geq 2$ can be created. ■

3.4 Adaptive Chosen Plaintext Attack

Adaptive attacks are more powerful than non-adaptive ones, since the attacker may modify his choice according to plaintext and ciphertext messages he obtains from the oracle. Luby and Rackoff defined in their paper [14] that a cipher C is called **pseudorandom** if there is no d-limited adaptive-chosen-plaintext-attack distinguisher between C and a perfect cipher for any d polynomial in $\lg(|\mathcal{M}_1|)$.

We will first consider a distinguisher between two general random functions. A d-limited adaptive-chosen-plaintext-attack distinguisher between two independent random functions F_1 and F_2 from a set $\mathcal{M}_1$ to a set $\mathcal{M}_2$ works as follows:

DISTINGUISHER 3.4 (ACPA): d-limited adaptive-chosen-plaintext-attack distinguisher [21]

```
1. For $k = 1$ to $d$ do
    1.1 Choose a plaintext message $x_k$, depending on the previous plaintexts and ciphertexts.
    1.2 Query the oracle with $x_k$, and get $y_k = F_i(x_k)$, where $i \in \{1, 2\}$.
2. Depending on $X = (x_1, \ldots, x_d)$ and $Y = (y_1, \ldots, y_d)$, output "accept" if you "think" the oracle implements $F_1$ or "reject" otherwise.
```

Let $\mathcal{A}$ be again the set of all pairs (X, Y) such that the distinguisher outputs "accept". The probability that it outputs "accept" when the oracle implements the function F_i ($i = 1, 2$) is

$$p_i = \sum_{X,Y} 1_{(X,Y)\in\mathcal{A}} \Pr[x_1] \Pr[x_2|x_1, y_1] \ldots \Pr[x_d|x_1, y_1, \ldots x_{d-1}, y_{d-1}] \, [F_i]^d_{X,Y}$$

and the advantage of the distinguisher is

$$\begin{aligned} Adv_D^{\mathrm{ACPA}(d)}(F_1, F_2) &= |p_1 - p_2| \\ &= \left| \sum_{X,Y} 1_{(X,Y)\in\mathcal{A}} \Pr[x_1] \ldots \Pr[x_d|x_1, y_1, \ldots x_{d-1}, y_{d-1}] \left([F_1]^d_{X,Y} - [F_2]^d_{X,Y} \right) \right| \\ &\leq \sum_{X,Y} 1_{(X,Y)\in\mathcal{A}} \Pr[x_1] \ldots \Pr[x_d|x_1, y_1, \ldots x_{d-1}, y_{d-1}] \left| [F_1]^d_{X,Y} - [F_2]^d_{X,Y} \right| \end{aligned}$$

The advantage is maximal when $\mathcal{A}$ contains *all* pairs (X, Y) such that $[F_1]^d_{X,Y} - [F_2]^d_{X,Y}$ have the same sign and when probabilities $\Pr[x_i|x_1, y_1, \ldots x_{i-1}, y_{i-1}] = 1$ for the best choice. In that case,

$$Adv^{\mathrm{ACPA}(d)}(F_1, F_2) = \max_{x_1} \sum_{y_1} \max_{x_2} \sum_{y_2} \ldots \max_{x_d} \sum_{y_d} 1_{(X,Y)\in\mathcal{A}} \left| [F_1]^d_{X,Y} - [F_2]^d_{X,Y} \right|$$

Since the queries depend on the previous responses of the oracle, we can define a function $f : \mathcal{M}_2^d \to \mathcal{M}_1^d$ such that

$$Adv^{\mathrm{ACPA}(d)}(F_1, F_2) = \sum_{Y} 1_{(X,Y)\in\mathcal{A}} |[F_1]^d_{f(Y),Y} - [F_2]^d_{f(Y),Y}|$$

i.e. $X = f(Y) = [x_1, \ldots, x_d]$ is the best choice of $x_1, \ldots, x_d$ for the fixed responses of the oracle Y. Since the value of x_i is fixed for given x_1, y_1, ..., x_{i-1}, y_{i-1},

$$\forall\, \tilde{y}_i, \ldots \tilde{y}_d : \quad x_i = [f(Y)]_i = [f(y_1, \ldots, y_{i-1}, \tilde{y}_i, \ldots \tilde{y}_d)]_i$$

and we will write $x_i = f(y_1, \ldots, y_{i-1}, *)$.

Lemma 3.4.1 *Let D be the best d-limited adaptive-chosen-plaintext-attack distinguisher. The distinguisher may always choose $X = (x_1, \ldots x_d)$ with pairwise distinct x_i's in order to obtain the best advantage.*

Proof: Let Y be such that $X = f(Y) = (x_1, \ldots, x_d)$, that not all x_i's are distinct. Let k be the smallest index such that there is $l < k$, and $Y = (y_1, \ldots, y_d)$, for which $x_k = x_l$. (Thus, for all Y and all $i, j < k$: $[f(Y)]_i \neq [f(Y)]_j$.) Then for all $\tilde{Y} = (y_1, \ldots, y_{k-1}, \tilde{y}_k, \ldots, \tilde{y}_d)$, and for all $i \leq k$:

$$x_i = [f(y_1, \ldots, y_{k-1}, \tilde{y}_k, \ldots, \tilde{y}_d)]_i$$

Furthermore, if $\tilde{y}_k \neq y_l$ then $[F_i]^d_{f(\tilde{Y}), \tilde{Y}} = 0$. Otherwise for any x'_d distinct from all $[f(Y)]_j$'s

$$[F_i]^d_{f(\tilde{Y}),\tilde{Y}} = \Pr\left[\bigwedge_{j=1}^{d} F_i([f(\tilde{Y})]_j) = \tilde{y}_j\right] = \sum_{y'_d} \Pr\left[F_i(x'_d) = y'_d \wedge \bigwedge_{\substack{j \in \{1,\ldots,d\} \\ i \neq k}} F_i([f(\tilde{Y})]_j) = \tilde{y}_j\right]$$

$$= \sum_{y'_d} \Pr\left[\bigwedge_{j=1}^{d} F_i([f'(Y')]_j) = y'_j\right] = \sum_{y'_d} [F_i]^d_{f'(Y'),Y'}$$

where $Y' = (y_1, \ldots, y_{k-1}, \tilde{y}_{k+1}, \ldots, \tilde{y}_d, y'_d)$, and f' is a new function such that for any $\tilde{y}_k, \ldots, \tilde{y}_d$

1. $[f'(y_1, \ldots, y_{j-1}, *)]_j = x_j$ for all $j \leq k$
2. $[f'(y_1, \ldots, y_{k-1}, \tilde{y}_k, \ldots, \tilde{y}_{j-1}, *)]_j = [f(y_1, \ldots, y_{k-1}, y_l, \tilde{y}_k, \ldots, \tilde{y}_{j-1}, *)]_{j+1}$, for all $k < j < d$
3. $[f'(y_1, \ldots, y_{k-1}, \tilde{y}_k, \ldots, \tilde{y}_{d-1}, *)]_d = x'_d(Y)$, where $x'_d(Y)$ is a new value, different from all $x_i = f'(Y)$, for all $Y = (y_1, \ldots, y_{k-1}, \tilde{y}_k, \ldots, \tilde{y}_{d-1}, *)$
4. $[f'(\tilde{y}_1, \ldots, \tilde{y}_{k-1}, \tilde{y}_k, \ldots, \tilde{y}_d)]_j = [f(\tilde{y}_1, \ldots, \tilde{y}_{k-1}, \tilde{y}_k, \ldots, \tilde{y}_d)]_j$, for all other values

Therefore, for a fixed values of $y_1, \ldots, y_{k-1}$

$$\sum_{\substack{\tilde{y}_k, \ldots, \tilde{y}_d \\ \tilde{Y} = (y_1, \ldots, y_{k-1}, \tilde{y}_k, \ldots, \tilde{y}_d)}} 1_{(f(\tilde{Y}),\tilde{Y}) \in \mathcal{A}} \left|[F_1]^d_{f(\tilde{Y}),\tilde{Y}} - [F_2]^d_{f(\tilde{Y}),\tilde{Y}}\right|$$

$$= \sum_{\substack{\tilde{y}_{k+1}, \ldots, \tilde{y}_d \\ \tilde{Y} = (y_1, \ldots, y_{k-1}, y_l, \tilde{y}_{k+1}, \ldots, \tilde{y}_d)}} 1_{(f(\tilde{Y}),\tilde{Y}) \in \mathcal{A}} \left|[F_1]^d_{f(\tilde{Y}),\tilde{Y}} - [F_2]^d_{f(\tilde{Y}),\tilde{Y}}\right|$$

$$\leq \sum_{\tilde{y}_{k+1}, \ldots, \tilde{y}_d} \left| \sum_{\substack{y'_d \\ Y' = (y_1, \ldots, y_{k-1}, \tilde{y}_k, \ldots, \tilde{y}_d, y'_d)}} 1_{(f(Y'),Y') \in \mathcal{A}} \left([F_1]^d_{f(Y'),Y'} - [F_2]^d_{f(Y'),Y'}\right) \right|$$

$$\leq \sum_{\substack{\tilde{y}_k, \ldots, \tilde{y}_d \\ Y' = (y_1, \ldots, y_{k-1}, \tilde{y}_k, \ldots, \tilde{y}_d)}} 1_{(f'(Y'),Y') \in \mathcal{A}} \left|[F_1]^d_{f'(Y'),Y'} - [F_2]^d_{f'(Y'),Y'}\right|$$

and

$$Adv_D^{\text{ACPA}(d)}(F_1, F_2) = \sum_{Y} 1_{(f(Y),Y) \in \mathcal{A}} \left|[F_1]^d_{f(Y),Y} - [F_2]^d_{f(Y),Y}\right|$$

$$\leq \sum_{Y} 1_{(f'(Y),Y) \in \mathcal{A}} \left|[F_1]^d_{f'(Y),Y} - [F_2]^d_{f'(Y),Y}\right|$$

Repeating this construction we get a new choice with all x_i's different and with advantage greater or equal to the previous one. Consequently, we may assume that the best distinguisher always uses different queries ■

Theorem 3.4.2 ([25]) *Let F_1 and F_2 be two independent random functions from a set $\mathcal{M}_1$ to a set $\mathcal{M}_2$, and d be an integer. Then*

$$Adv^{\text{ACPA}(d)}(F_1, F_2) = \frac{1}{2} \left\|[F_1]^d - [F_2]^d\right\|_a$$

Proof: Since for a fixed $x_1, \ldots, x_k$ and $y_1 \ldots, y_{k-1}$, it holds $\sum_{y_k} \Pr\left[\bigwedge_{i=1}^{k} F_i(x_i) = y_i\right] = \Pr\left[\bigwedge_{i=1}^{k-1} F_i(x_i) = y_i\right]$, then

$$\begin{aligned}
\sum_Y [F_i]^d_{f(Y),Y} &= \sum_Y \Pr\left[\bigwedge_{j=1}^{d} F_i([f(Y)]_j) = y_j\right] \\
&= \sum_{y_1,\ldots,y_{d-1}} \sum_{y_d} \Pr\left[\bigwedge_{j=1}^{d} F_i([f(y_1,\ldots,y_{j-1},*)]_j) = y_j\right] \\
&= \sum_{y_1,\ldots,y_{d-1}} \Pr\left[\bigwedge_{j=1}^{d-1} F_i([f(y_1,\ldots,y_{j-1},*)]_j) = y_j\right] \\
&\ldots \\
&= \sum_{y_1} \Pr\left[F_i([f(*)]_1) = y_1\right] = 1
\end{aligned}$$

Therefore, $\sum_Y [F_1]^d_{f(X),Y} - [F_2]^d_{f(X),Y} = 0$, and

$$\sum_Y 1_{(f(Y),Y)\in\mathcal{A}} \left|[F_1]^d_{f(X),Y} - [F_2]^d_{f(X),Y}\right| = \sum_Y 1_{(f(Y),Y)\notin\mathcal{A}} \left|[F_1]^d_{f(X),Y} - [F_2]^d_{f(X),Y}\right|$$

Hence,

$$\begin{aligned}
\left\|[F_1]^d - [F_2]^d\right\|_a &= \max_{x_1}\sum_{y_1}\max_{x_2}\sum_{y_2}\ldots\max_{x_d}\sum_{y_d}\left|[F_1]^d_{X,Y} - [F_2]^d_{X,Y}\right| \\
&= \sum_Y \left|[F_1]^d_{f(Y),Y} - [F_2]^d_{f(Y),Y}\right| \\
&= \sum_Y 1_{(f(Y),Y)\in\mathcal{A}} \left|[F_1]^d_{f(Y),Y} - [F_2]^d_{f(Y),Y}\right| \\
&\quad + \sum_Y 1_{(f(Y),Y)\notin\mathcal{A}} \left|[F_1]^d_{f(Y),Y} - [F_2]^d_{f(Y),Y}\right| \\
&= 2\sum_Y 1_{(f(Y),Y)\in\mathcal{A}} \left|[F_1]^d_{f(Y),Y} - [F_2]^d_{f(Y),Y}\right| \\
&= 2\,Adv^{\mathrm{ACPA}(d)}(F_1,F_2)
\end{aligned}$$

■

Corollary 3.4.3 *Let F be a random function, and d be an integer. Then*

$$AdvF^{\mathrm{ACPA}(d)}(F) = \frac{1}{2}\,DecF^d_{\|\cdot\|_a}(F).$$

Similar theorems hold also for permutations.

Theorem 3.4.4 *Let C_1 and C_2 be two independent random permutations on a set $\mathcal{M}$, and d be an integer. Then*

$$Adv^{\mathrm{ACPA}(d)}(C_1,C_2) = \frac{1}{2}\left\|[C_1]^d - [C_2]^d\right\|_a$$

Corollary 3.4.5 *Let C be a cipher, and d be an integer. Then*

$$AdvC^{\mathrm{ACPA}(d)}(C) = \frac{1}{2}\,DecC^d_{\|\cdot\|_a}(C)$$

Like for the chosen plaintext attack, we evaluate here the distance between a perfect random function and a perfect random permutation in the adaptive chosen plaintext attack.

Theorem 3.4.6 *Let C^* be a perfect cipher and F^* be a perfect random function on $\mathcal{M}$, and $d < \sqrt{|\mathcal{M}|}$ be an integer. Then*

$$\left\|[C^*]^d - [F^*]^d\right\|_a = 2\left(1 - \frac{|\mathcal{M}^{\underline{d}}|}{|\mathcal{M}^d|}\right) \le \frac{d^{\underline{2}}}{|\mathcal{M}|}.$$

Proof: Let $A = [C^*]^d - [F^*]^d$. For any fixed $X = (x_1, \ldots, x_d)$ with c distinct values among $x_1, \ldots, x_d$, and for any $Y = (y_1, \ldots, y_d)$:

$$|A_{X,Y}| = \begin{cases} 0 & \text{if } \exists i,j : x_i = x_j \wedge y_i \neq y_j \\ \frac{1}{|\mathcal{M}|^c} & \text{if } \forall i,j : x_i = x_j \Rightarrow y_i \neq y_j, \\ & \text{but } \exists i,j : x_i \neq x_j \wedge y_i = y_j \\ \frac{1}{|\mathcal{M}|^{\underline{c}}} - \frac{1}{|\mathcal{M}|^c} & \text{if } \forall i,j : x_i = x_j \Leftrightarrow y_i = y_j \end{cases}$$

Thus, there are only two distinct non-zero values in the matrix A. In the following, we first proof which of the two values is greater, then we introduce a function which, as we show later, simulates the calculation of the norm $\|A\|_a$, then we show how the x_i's have to be chosen in order to get the maximal values in the norm, and at last we evaluate the norm.

A. $\frac{1}{|\mathcal{M}|^c} > \frac{1}{|\mathcal{M}|^{\underline{c}}} - \frac{1}{|\mathcal{M}|^c}$:
$\frac{1}{|\mathcal{M}|^c} - \left(\frac{1}{|\mathcal{M}|^{\underline{c}}} - \frac{1}{|\mathcal{M}|^c}\right) = \frac{2}{|\mathcal{M}|^c} - \frac{1}{|\mathcal{M}|^{\underline{c}}} = \frac{1}{|\mathcal{M}|^{\underline{c}}}\left(2\frac{|\mathcal{M}|^{\underline{c}}}{|\mathcal{M}|^c} - 1\right) \geq$
$\frac{1}{|\mathcal{M}|^{\underline{c}}}\left[2\left(1 - \frac{c^{\underline{2}}}{2\,|\mathcal{M}|}\right) - 1\right] = \frac{1}{|\mathcal{M}|^{\underline{c}}}\left[1 - \frac{c^{\underline{2}}}{|\mathcal{M}|}\right] > 0$

B. Let S(k, c) be defined as follows:
1. $S(0,c) = \frac{1}{|\mathcal{M}|^{\underline{c}}} - \frac{1}{|\mathcal{M}|^c}$
2. $S(k,c) = c \cdot \frac{1}{|\mathcal{M}|^{c+1}} + (|\mathcal{M}| - c) \cdot S(k-1, c+1)$

Then the closed formula for S is

$$S(k,c) = \sum_{i=0}^{k-1}(c+i) \cdot (|\mathcal{M}| - c)^{\underline{i}} \cdot \frac{1}{|\mathcal{M}|^{c+i+1}} + (|\mathcal{M}| - c)^{\underline{k}} \cdot \left(\frac{1}{|\mathcal{M}|^{\underline{c+k}}} - \frac{1}{|\mathcal{M}|^{c+k}}\right) \tag{3.1}$$

and $S(k,c) < S(k+1,c)$ for any constant c.
Proof (by induction on k):

1.
   - $S(0,c) = \sum_{i=0}^{-1}(c+i) \cdot (|\mathcal{M}| - c)^{\underline{i}} \cdot \frac{1}{|\mathcal{M}|^{c+i+1}} + (|\mathcal{M}| - c)^{\underline{0}} \cdot \left(\frac{1}{|\mathcal{M}|^{\underline{c}}} - \frac{1}{|\mathcal{M}|^c}\right) = \frac{1}{|\mathcal{M}|^{\underline{c}}} - \frac{1}{|\mathcal{M}|^c}$
   - $S(1,c) = \sum_{i=0}^{0}(c+i) \cdot (|\mathcal{M}| - c)^{\underline{i}} \cdot \frac{1}{|\mathcal{M}|^{c+i+1}} + (|\mathcal{M}| - c)^{\underline{1}} \cdot \left(\frac{1}{|\mathcal{M}|^{\underline{c+1}}} - \frac{1}{|\mathcal{M}|^{c+1}}\right) = c \cdot \frac{1}{|\mathcal{M}|^{c+1}} + (|\mathcal{M}| - c) \cdot \left(\frac{1}{|\mathcal{M}|^{\underline{c+1}}} - \frac{1}{|\mathcal{M}|^{c+1}}\right) = c \cdot \frac{1}{|\mathcal{M}|^{c+1}} + (|\mathcal{M}| - c) \cdot S(0, c+1)$
   - $S(1,c) = c \cdot \frac{1}{|\mathcal{M}|^{c+1}} + (|\mathcal{M}| - c) \cdot \left(\frac{1}{|\mathcal{M}|^{\underline{c+1}}} - \frac{1}{|\mathcal{M}|^{c+1}}\right) =$
   $c \cdot \left(\frac{1}{|\mathcal{M}|^{c+1}} - \left(\frac{1}{|\mathcal{M}|^{\underline{c+1}}} - \frac{1}{|\mathcal{M}|^{c+1}}\right)\right) + \frac{|\mathcal{M}|}{|\mathcal{M}|^{\underline{c+1}}} - \frac{|\mathcal{M}|}{|\mathcal{M}|^{c+1}} \overset{A}{>} \frac{|\mathcal{M}|}{|\mathcal{M}|^{\underline{c+1}}} - \frac{|\mathcal{M}|}{|\mathcal{M}|^{c+1}} >$
   $\frac{|\mathcal{M}|-c}{|\mathcal{M}|^{\underline{c+1}}} - \frac{1}{|\mathcal{M}|^c} = \frac{1}{|\mathcal{M}|^{\underline{c}}} - \frac{1}{|\mathcal{M}|^c} = S(0,c)$
2. Assume that Equation (3.1) is correct for all $k < k_0$ for a constant k_0, and that for any c, $S(k-1,c) < S(k,c)$.
3.
   - $S(k+1,c) = c \cdot \frac{1}{|\mathcal{M}|^{c+1}} + (|\mathcal{M}| - c) \cdot S(k, c+1) =$
   $c \cdot \frac{1}{|\mathcal{M}|^{c+1}} + (|\mathcal{M}| - c) \cdot \left[\sum_{i=0}^{k-1}(c+i+1) \cdot (|\mathcal{M}| - c - 1)^{\underline{i}} \cdot \frac{1}{|\mathcal{M}|^{c+i+2}} + (|\mathcal{M}| - c - 1)^{\underline{k}} \cdot \left(\frac{1}{|\mathcal{M}|^{\underline{c+k+1}}} - \frac{1}{|\mathcal{M}|^{c+k+1}}\right)\right] =$
   $c \cdot \frac{1}{|\mathcal{M}|^{c+1}} + (|\mathcal{M}| - c) \cdot \left[\sum_{i=1}^{k}(c+i) \cdot (|\mathcal{M}| - c - 1)^{\underline{i-1}} \cdot \frac{1}{|\mathcal{M}|^{c+i+1}} + (|\mathcal{M}| - c - 1)^{\underline{k}} \cdot \left(\frac{1}{|\mathcal{M}|^{\underline{c+k+1}}} - \frac{1}{|\mathcal{M}|^{c+k+1}}\right)\right] = c \cdot (|\mathcal{M}| - c)^{\underline{0}} \frac{1}{|\mathcal{M}|^{c+1}} + \sum_{i=1}^{k}(c+i) \cdot (|\mathcal{M}| - c)^{\underline{i}} \cdot \frac{1}{|\mathcal{M}|^{c+i+1}} + (|\mathcal{M}| - c)^{\underline{k+1}} \cdot \left(\frac{1}{|\mathcal{M}|^{\underline{c+k+1}}} - \frac{1}{|\mathcal{M}|^{c+k+1}}\right) =$
   $\sum_{i=0}^{k}(c+i) \cdot (|\mathcal{M}| - c)^{\underline{i}} \cdot \frac{1}{|\mathcal{M}|^{c+i+1}} + (|\mathcal{M}| - c)^{\underline{k+1}} \cdot \left(\frac{1}{|\mathcal{M}|^{\underline{c+k+1}}} - \frac{1}{|\mathcal{M}|^{c+k+1}}\right)$
   - Since

$$S(k,c) = c \cdot \frac{1}{|\mathcal{M}|^{c+1}} + (|\mathcal{M}| - c) \cdot S(k-1, c+1)$$
$$S(k+1,c) = c \cdot \frac{1}{|\mathcal{M}|^{c+1}} + (|\mathcal{M}| - c) \cdot S(k, c+1)$$

and from the induction assumption $S(k-1,c+1) < S(k,c+1)$ then $S(k,c) < S(k+1,c)$ as well.

Recall that

$$\|A\|_a = \begin{cases} \max\limits_{x_1} \sum\limits_{y_1} \|\pi_{x_1,y_1}(A)\|_a, & d > 1, \\ \max\limits_{x_1} \sum\limits_{y_1} A_{x_1,y_1}, & d = 1, \end{cases}$$

In the following, we show how to choose the individual x_i's. We start with the smallest size (1), i.e. when all $x_1, \ldots, x_{d-1}$ are already fixed, and then continue by induction.

1. Let $x_1, \ldots, x_{d-1}$ and $y_1, \ldots, y_{d-1}$ be fixed. Let c be the number of distinct values among $x_1, \ldots, x_{d-1}$. And let $B = \pi_{x_1,y_1}\pi_{x_2,y_2}\ldots\pi_{x_{d-1},y_{d-1}}(A)$. Then

   Case I(d – 1, c): If $\exists i,j \le d-1$ such that $x_i = x_j$ and $y_i \ne y_j$ (i.e. there is no such permutation or function) then B is a zero matrix, and thus for any x_d, $\sum_{y_d} B_{x_d,y_d} = 0$. It means that choosing any value for x_d the result will not change.

   Case II(d – 1, c): If $\forall i,j \le d-1$, $x_i = x_j \Rightarrow y_i = y_j$, but $\exists i,j \le d-1$ such that $x_i \ne x_j$ and $y_i = y_j$ (i.e. there is a function, but no permutation)

   a) If we choose x_d so that $\exists j : x_d = x_j$ then
      - if $y_d \ne y_j$ then $B_{x_d,y_d} = 0$
      - if $y_d = y_j$ then $B_{x_d,y_d} = \frac{1}{|\mathcal{M}|^c}$

      Hence, $\sum_{y_d} B_{x_d,y_d} = \frac{1}{|\mathcal{M}|^c}$

   b) If $x_d \notin \{x_1, \ldots, x_{d-1}\}$, it adds a new distinct value, and thus all elements in the row are $\frac{1}{|\mathcal{M}|^{c+1}}$. Hence, $\sum_{y_d} B_{x_d,y_d} = |\mathcal{M}| \cdot \frac{1}{|\mathcal{M}|^{c+1}} = \frac{1}{|\mathcal{M}|^c}$

   It means that in this case one can choose any value for x_d as well.

   Case III(d – 1, c): If $\forall i,j \le d-1$, $x_i = x_j \Leftrightarrow y_i = y_j$ (i.e. there is a function as well as a permutation)

   a) If we choose x_d so that $\exists j : x_d = x_j$ then
      - if $y_d \ne y_j$ then $B_{x_d,y_d} = 0$
      - if $y_d = y_j$ then $B_{x_d,y_d} = \frac{1}{|\mathcal{M}|^{\underline{c}}} - \frac{1}{|\mathcal{M}|^c}$

      Hence, $\sum_{y_d} B_{x_d,y_d} = \frac{1}{|\mathcal{M}|^{\underline{c}}} - \frac{1}{|\mathcal{M}|^c} \overset{(B)}{=} S(0,c)$

   b) If $x_d \notin \{x_1, \ldots, x_{d-1}\}$, it adds a new distinct value
      - if $y_d = y_j$ then there is no such permutation, and thus $B_{x_d,y_d} = \frac{1}{|\mathcal{M}|^{c+1}}$
      - if $y_d \ne y_j$ then $B_{x_d,y_d} = \frac{1}{|\mathcal{M}|^{\underline{c+1}}} - \frac{1}{|\mathcal{M}|^{c+1}}$

      Hence, $\sum_{y_d} B_{x_d,y_d} = c \cdot \frac{1}{|\mathcal{M}|^{c+1}} + (|\mathcal{M}| - c) \cdot \left(\frac{1}{|\mathcal{M}|^{\underline{c+1}}} - \frac{1}{|\mathcal{M}|^{c+1}}\right) \overset{(B)}{=} S(1,c)$

   Since $S(0,c) < S(1,c)$, in this case one has to choose a distinct x_d in order to get the better result.

2. Let $x_1, \ldots, x_k$ and $y_1, \ldots, y_k$ be fixed. Let c be number of distinct values among $x_1, \ldots, x_k$. And let $B = \pi_{x_1,y_1}\pi_{x_2,y_2}\ldots\pi_{x_k,y_k}(A)$. Assume that

   Case I(k, c): If $\exists i,j \le k$ such that $x_i = x_j$ and $y_i \ne y_j$ (i.e. there is no such permutation or function) then for any x_{k+1}, $\sum_{y_{k+1}} \left\|\pi_{x_{k+1},y_{k+1}}(B)\right\|_a = 0$. It means that in this case one can choose any value for x_{k+1}.

   Case II(k, c): If $\forall i,j \le k$, $x_i = x_j \Rightarrow y_i = y_j$, but $\exists i,j \le k$ such that $x_i \ne x_j$ and $y_i = y_j$ (i.e. there is a function, but no permutation)

   a) If $\exists j : x_{k+1} = x_j$ then $\sum_{y_{k+1}} \left\|\pi_{x_{k+1},y_{k+1}}(B)\right\|_a = \frac{1}{|\mathcal{M}|^c}$

   b) If $x_{k+1} \notin \{x_1, \ldots, x_k\}$, $\sum_{y_{k+1}} \left\|\pi_{x_{k+1},y_{k+1}}(B)\right\|_a = \frac{1}{|\mathcal{M}|^c}$

   It means that in this case one can choose any value for x_{k+1} as well.

   Case III(k, c): $\forall i,j \le k$, $x_i = x_j \Leftrightarrow y_i = y_j$ (i.e. there is a function as well as a permutation).

   a) If $\exists j : x_{k+1} = x_j$ then $\sum_{y_{k+1}} \left\|\pi_{x_{k+1},y_{k+1}}(B)\right\|_a = S(d-k-1,c)$

   b) If $x_{k+1} \notin \{x_1, \ldots, x_k\}$ then $\sum_{y_{k+1}} \left\|\pi_{x_{k+1},y_{k+1}}(B)\right\|_a = S(d-k,c)$

Since $S(d-k-1,c) < S(d-k,c)$, one has to choose a different x_{k+1} in order to get the better result.

3. Let $x_1,\ldots,x_{k-1}$ and $y_1,\ldots,y_{k-1}$ be fixed. Let c be number of distinct values among $x_1,\ldots,x_{k-1}$. And let $B = \pi_{x_1,y_1}\pi_{x_2,y_2}\ldots\pi_{x_{k-1},y_{k-1}}(A)$. Then

Case I(k – 1, c): If $\exists i,j \le k-1$ such that $x_i = x_j$ and $y_i \ne y_j$ (i.e. there is no such permutation or function) then B is a zero matrix, and thus for any x_k, $\sum_{y_k} \left\|\pi_{x_k,y_k}(B)\right\|_a = 0$. It means that in this case one can choose any value for x_k.

Case II(k – 1, c): If $\forall i,j \le k-1$, $x_i = x_j \Rightarrow y_i = y_j$, but $\exists i,j \le k-1$ such that $x_i \ne x_j$ and $y_i = y_j$ (i.e. there is a function, but no permutation)

a) If we choose x_d so that $\exists j : x_k = x_j$ then

- if $y_k \ne y_j$ then $\pi_{x_k,y_k}B_{x_k,y_k}$ is a zero matrix (i.e. $\rightsquigarrow$ I(k, c)), and $\left\|\pi_{x_k,y_k}(B)\right\|_a = 0$
- if $y_k = y_j$ then ($\rightsquigarrow$ II(k, c)) $\left\|\pi_{x_k,y_k}(B)\right\|_a = \frac{1}{|\mathcal{M}|^c}$

Hence, $\sum_{y_k} \left\|\pi_{x_k,y_k}(B)\right\|_a = \frac{1}{|\mathcal{M}|^c}$

b) If $x_k \notin \{x_1,\ldots,x_{k-1}\}$, it adds a new distinct value, and for any choice of y_d ($\rightsquigarrow$ II(k, c + 1)) $\pi_{x_k,y_k}B_{x_k,y_k} = \frac{1}{|\mathcal{M}|^{c+1}}$ Hence, $\sum_{y_k} \left\|\pi_{x_k,y_k}(B)\right\|_a = |\mathcal{M}| \cdot \frac{1}{|\mathcal{M}|^{c+1}} = \frac{1}{|\mathcal{M}|^c}$

It means that in this case one can choose any value for x_k as well.

Case III(k – 1, c): If $\forall i,j \le k-1$, $x_i = x_j \Leftrightarrow y_i = y_j$ (i.e. there is a function as well as a permutation)

a) If we choose x_d so that $\exists j : x_k = x_j$ then

- if $y_k \ne y_j$ then ($\rightsquigarrow$ I(k, c)) $\left\|\pi_{x_k,y_k}(B)\right\|_a = 0$
- if $y_k = y_j$ then ($\rightsquigarrow$ III(k, c)) $\left\|\pi_{x_k,y_k}(B)\right\|_a = S(d-k,c)$

Hence, $\sum_{y_k} \left\|\pi_{x_k,y_k}(B)\right\|_a = S(d-k,c)$

b) If $x_k \notin \{x_1,\ldots,x_{k-1}\}$, it adds a new distinct value

- if $y_k = y_j$ then ($\rightsquigarrow$ II(k, c + 1)) $\left\|\pi_{x_k,y_k}(B)\right\|_a = \frac{1}{|\mathcal{M}|^{c+1}}$
- if $y_k \ne y_j$ then ($\rightsquigarrow$ III(k, c + 1)) $\left\|\pi_{x_k,y_k}(B)\right\|_a = S(d-k,c+1)$

Hence,

$\sum_{y_k} \left\|\pi_{x_k,y_k}(B)\right\|_a = c \cdot \frac{1}{|\mathcal{M}|^{c+1}} + (|\mathcal{M}| - c) \cdot S(d-k,c+1) \overset{(B)}{=} S(d-k+1,c)$

Since $S(d-k,c) < S(d-k+1,c)$, one has to choose a different x_{k+1} in order to get the better result.

At the beginning we have to choose x_1, y_1, and in order to get the best result, we have to start the process in III(1, 1). Therefore, for any y_1, we get $S(d-1,1)$, and thus

$$\left\|A\right\|_a = \sum_{y_1} \pi_{x_1,y_1}A_{x_1,y_1} = |\mathcal{M}| \cdot S(d-1,1) =$$
$$|\mathcal{M}| \cdot \left[\sum_{i=0}^{d-2}(i+1)\cdot(|\mathcal{M}|-1)^{\underline{i}} \cdot \frac{1}{|\mathcal{M}|^{i+2}} + (|\mathcal{M}|-1)^{\underline{d-1}} \cdot \left(\frac{1}{|\mathcal{M}|^{\underline{d}}} - \frac{1}{|\mathcal{M}|^d}\right)\right] =$$
$$|\mathcal{M}| \cdot \left[\sum_{i=1}^{d-1} i\cdot(|\mathcal{M}|-1)^{\underline{i-1}} \cdot \frac{1}{|\mathcal{M}|^{i+1}} + (|\mathcal{M}|-1)^{\underline{d-1}} \cdot \left(\frac{1}{|\mathcal{M}|^{\underline{d}}} - \frac{1}{|\mathcal{M}|^d}\right)\right] =$$
$$\sum_{i=1}^{d-1} i \cdot \frac{|\mathcal{M}|^{\underline{i}}}{|\mathcal{M}|^{i+1}} + |\mathcal{M}|^{\underline{d}} \cdot \left(\frac{1}{|\mathcal{M}|^{\underline{d}}} - \frac{1}{|\mathcal{M}|^d}\right) = \frac{1}{|\mathcal{M}|} \cdot \sum_{i=0}^{d-1} i \cdot \frac{|\mathcal{M}|^{\underline{i}}}{|\mathcal{M}|^{i}} + \left(1 - \frac{|\mathcal{M}|^{\underline{d}}}{|\mathcal{M}|^d}\right)$$

Now, we show (by induction) that $\frac{1}{|\mathcal{M}|} \cdot \sum_{i=1}^{d-1} i \cdot \frac{|\mathcal{M}|^{\underline{i}}}{|\mathcal{M}|^{i}} = 1 - \frac{|\mathcal{M}|^{\underline{d}}}{|\mathcal{M}|^d}$

1. $d = 1:$
   - $\frac{1}{|\mathcal{M}|} \cdot \sum_{i=0}^{0} i \cdot \frac{|\mathcal{M}|^{\underline{i}}}{|\mathcal{M}|^{i}} = 0$
   - $1 - \frac{|\mathcal{M}|^{\underline{1}}}{|\mathcal{M}|^1} = 1 - \frac{|\mathcal{M}|}{|\mathcal{M}|} = 0$
2. Assume that $\frac{1}{|\mathcal{M}|} \cdot \sum_{i=1}^{k-1} i \cdot \frac{|\mathcal{M}|^{\underline{i}}}{|\mathcal{M}|^{i}} = 1 - \frac{|\mathcal{M}|^{\underline{k}}}{|\mathcal{M}|^k}$
3. For $k+1$: $\frac{1}{|\mathcal{M}|} \cdot \sum_{i=1}^{k} i \cdot \frac{|\mathcal{M}|^{\underline{i}}}{|\mathcal{M}|^{i}} = \frac{1}{|\mathcal{M}|} \cdot \sum_{i=1}^{k-1} i \cdot \frac{|\mathcal{M}|^{\underline{i}}}{|\mathcal{M}|^{i}} + \frac{1}{|\mathcal{M}|} \cdot k \cdot \frac{|\mathcal{M}|^{\underline{k}}}{|\mathcal{M}|^{k}} =$
   $1 - \frac{|\mathcal{M}|^{\underline{k}}}{|\mathcal{M}|^k} + \frac{k\cdot|\mathcal{M}|^{\underline{k}}}{|\mathcal{M}|^{k+1}} = 1 - \frac{|\mathcal{M}|^{\underline{k}}(|\mathcal{M}|-k)}{|\mathcal{M}|^{k+1}} = 1 - \frac{|\mathcal{M}|^{\underline{k+1}}}{|\mathcal{M}|^{k+1}}$

Hence, the greatest value, one can obtain, is:

$$\frac{1}{|\mathcal{M}|} \cdot \sum_{i=0}^{d-1} i \cdot \frac{|\mathcal{M}|^{\underline{i}}}{|\mathcal{M}|^{i}} + \left(1 - \frac{|\mathcal{M}|^{\underline{d}}}{|\mathcal{M}|^d}\right) = 2\left(1 - \frac{|\mathcal{M}|^{\underline{d}}}{|\mathcal{M}|^d}\right) \le 2\,\frac{d^{\underline{2}}}{2\,|\mathcal{M}|} = \frac{d^{\underline{2}}}{|\mathcal{M}|} \qquad \blacksquare$$

Corollary 3.4.7 *Let C^* be a perfect cipher and F^* be a perfect random function on $\mathcal{M}$, and $d < \sqrt{|\mathcal{M}|}$ be an integer. Then*

$$AdvC^{\mathrm{ACPA}(d)}(F^*) = AdvF^{\mathrm{ACPA}(d)}(C^*) \leq \frac{d^2}{2|\mathcal{M}|}$$

Proof: Follows from Theorem 3.4.6, and Corollary 3.4.3, or Corollary 3.4.5. ■

Adaptive Chosen Plaintext Attack Against Unbalanced Feistel Networks

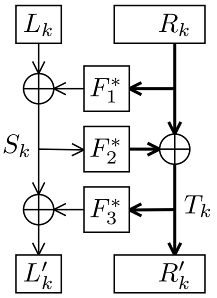
In this subsection we examine security of three-round UFNs in the random oracle model against the adaptive chosen plaintext attack.

Theorem 3.4.8 ([25]) *Let F_1^*, F_2^*, F_3^* be three independent perfect random functions, F_1^* and F_3^* from $\mathcal{M}_2$ to $\mathcal{M}_1$ and F_2^* from $\mathcal{M}_1$ to $\mathcal{M}_2$, $\Psi[F_1^*, F_2^*, F_3^*]$ a 3-round UFN, and d an integer. Then,*

$$AdvC^{\mathrm{ACPA}(d)}(\Psi[F_1^*, F_2^*, F_3^*]) \leq \frac{d^2}{\min\{|\mathcal{M}_1|, |\mathcal{M}_2|\}}$$

Proof: The proof is based on the technique introduced in [16].

Any d-limited known-plaintext-attack distinguisher has access to d plaintext/ciphertext pairs $(x_1, y_1), \ldots, (x_d, y_d)$. When the oracle implements the unbalanced Feistel network, the ciphertexts are calculated as depicted on the following figure.



$$x_k = [L_k, R_k]$$
$$y_k = [L'_k, R'_k]$$

$$S_k = L_k \oplus F_1^*(R_k)$$
$$R'_k = T_k = R_k \oplus F_2^*(S_k)$$
$$L'_k = S_k \oplus F_3^*(T_k)$$

If all S_k's are pairwise distinct, then the T_k's are perfectly random, since the function F_2^* is perfectly random. If all T_k's (and thus also R'_k's) are pairwise distinct, then L'_k's are also perfectly random, since the function F_3^* is perfectly random. Consequently, if all S_k's and T_k's are pairwise different, then $y_k = [L'_k, R'_k]$ are perfectly random. In that case, the cipher Ψ looks like a perfect cipher.

Without loss of generality, we may assume that all x_k's chosen by the distinguisher are distinct. In that case, for any $k \neq l$:

- If $R_k = R_l$ then $L_k \neq L_l$ (since $x_k \neq x_l$), and thus $S_k = L_k + F_1^*(R_k) \neq L_l + F_1^*(R_l) = S_l$, and Pr$[S_k = S_l] = 0$.
- If $R_k \neq R_l$ then the probability that $S_k = S_l$ is $\frac{1}{|\mathcal{M}_1|}$, since $F_1^*(R_k)$ and $F_1^*(R_l)$ are independent and random.

Therefore, Pr$[S_k = S_l] \leq \frac{1}{|\mathcal{M}_1|}$.

After the first round, the input into the second round is $[S_k, R_k]$ and $[S_l, R_l]$. As shown, either $R_k \neq R_l$, or when $R_k = R_l$ then $S_k \neq S_l$. Thus, we may continue with the same technique as for the first round, and get Pr$[T_k = T_l] \leq \frac{1}{|\mathcal{M}_2|}$. Hence, from Theorem 2.2.3

$$\begin{aligned}
AdvC^{\mathrm{ACPA}(d)}(\Psi[F_1^*, F_2^*, F_3^*]) &\leq 1 - \Pr[\forall\, k \neq l : S_k \neq S_l \wedge T_k \neq T_l] \\
&= \Pr[\exists\, k \neq l : S_k = S_l \vee T_k = T_l] \\
&\leq \Pr[\exists\, k \neq l : S_k = S_l] + \Pr[\exists\, k \neq l : T_k = T_l] \\
&\leq \sum_{1 \leq k < l \leq d} \Pr[S_k = S_l] + \sum_{1 \leq k < l \leq d} \Pr[T_k = T_l] \\
&= \binom{d}{2} \frac{1}{|\mathcal{M}_1|} + \binom{d}{2} \frac{1}{|\mathcal{M}_2|} \\
&\leq 2\binom{d}{2} \frac{1}{\min\{|\mathcal{M}_1|, |\mathcal{M}_2|\}} = \frac{d^2}{\min\{|\mathcal{M}_1|, |\mathcal{M}_2|\}}
\end{aligned}$$

Note that adding a new round with any round function (even a weak one) before the first round cannot increase the advantage, because we only need the property of pairwise difference of S_k's, which is preserved. ■

Corollary 3.4.9 *Let F_1^*, F_2^*, F_3^* be three independent perfect random functions, F_1^* and F_3^* from a set $\mathcal{M}_2$ to a set $\mathcal{M}_1$ and F_2^* from $\mathcal{M}_1$ to $\mathcal{M}_2$, $\Psi[F_1^*, F_2^*, F_3^*]$ a 3-round UFN, and d an integer. Then*

$$DecC^d_{\|\cdot\|_a}(\Psi[F_1^*, F_2^*, F_3^*]) \leq 2\,\frac{d^2}{\min\{|\mathcal{M}_1|, |\mathcal{M}_2|\}}$$

Corollary 3.4.10 *Let $F_1, \ldots, F_r$ be $r \geq 3$ independent random functions, F_i for all odd i from a set $\mathcal{M}_2$ to a set $\mathcal{M}_1$, and F_i for all even i from $\mathcal{M}_1$ to $\mathcal{M}_2$ such that $AdvF^{\mathrm{ACPA}(d)}(F_i) \leq \varepsilon$, and d be an integer. Then*

$$AdvC^{\mathrm{ACPA}(d)}(\Psi[F_1, \ldots, F_r]) \leq \frac{1}{2}\left[2\left(3\varepsilon + \frac{d^2}{\min\{|\mathcal{M}_1|, |\mathcal{M}_2|\}}\right)\right]^{\lfloor\frac{r}{3}\rfloor}$$

Proof: Follows from Theorem 2.4.4 and 2.4.2, and from the note at the end of the proof of Theorem 3.4.8. ■

3.5 Chosen Ciphertext Attack

The **chosen ciphertext attack** is similar to the chosen plaintext attack with the difference that the attacker may choose several ciphertexts, rather than plaintexts. A d-limited chosen-ciphertext-attack distinguisher D between two independent random permutations C_1 and C_2 on a set $\mathcal{M}$ works as follows:

DISTINGUISHER 3.5 (CCA): d-limited chosen-plaintext-attack distinguisher

1. Choose ciphertext messages $Y = (y_1, \ldots, y_d)$.
2. Query the oracle with Y, and get $X = (C_i^{-1}(y_1), \ldots, C_i^{-1}(y_d))$, where $i \in \{1, 2\}$.
3. Depending on X and Y, output "accept" if you "think" the oracle implements C_1 or "reject" otherwise.

It is easy to see that each chosen ciphertext attack is actually a chosen plaintext attack distinguishing C_1^{-1} and C_2^{-1}. Formally: Let $\mathcal{A}$ be the set of all pairs (X, Y), such that the distinguisher outputs "accept". Then the probability that it outputs "accept" when the oracle implements the function C_i ($i = 1, 2$) is

$$\begin{aligned} p_i &= \sum_Y \Pr[Y] \sum_X 1_{(X,Y)\in\mathcal{A}}\, [C_i]^d_{X,Y} = \sum_X \Pr[X] \sum_Y 1_{(Y,X)\in\mathcal{A}}\, [C_i]^d_{Y,X} \\ &= \sum_X \Pr[X] \sum_Y 1_{(X,Y)\in\mathcal{A}'}\, \left[[C_i]^d\right]^T_{X,Y} \overset{\text{Lemma 2.3.3}}{=} \sum_X \Pr[X] \sum_Y 1_{(X,Y)\in\mathcal{A}'}\, [C_i^{-1}]^d_{X,Y} \end{aligned}$$

Thus, for each chosen-ciphertext-attack distinguisher D with the acceptance set $\mathcal{A}$, there is a chosen-plaintext-attack distinguisher D' with the acceptance set $\mathcal{A}' = \{(X, Y)|(Y, X) \in \mathcal{A}\}$, and with the same advantage. Therefore,

$$Adv^{\mathrm{CCA}(d)}(C_1, C_2) = Adv^{\mathrm{CPA}(d)}_{D'}(C_1^{-1}, C_2^{-1}) \leq Adv^{\mathrm{CPA}(d)}(C_1^{-1}, C_2^{-1})$$

In the same way, any chosen-plaintext distinguishing attack for C_1^{-1} and C_2^{-1} can be transformed into a chosen-ciphertext attack. Hence,

$$Adv^{\mathrm{CPA}(d)}(C_1^{-1}, C_2^{-1}) \leq Adv^{\mathrm{CCA}(d)}(C_1, C_2)$$

and consequently,

$$Adv^{\mathrm{CCA}(d)}(C_1, C_2) = Adv^{\mathrm{CPA}(d)}(C_1^{-1}, C_2^{-1}).$$

Theorem 3.5.1 *Let C_1 and C_2 be two independent ciphers on a set $\mathcal{M}$, and d be an integer. Then*

$$Adv^{\mathrm{CCA}(d)}(C_1, C_2) = \frac{1}{2}\,|||C_1^{-1}, C_2^{-1}|||_\infty$$

Corollary 3.5.2 *Let C be a cipher, and d be an integer. Then*

$$AdvC^{\mathrm{CCA}(d)}(C) = \frac{1}{2}\,DecC^d_{|||\cdot|||_\infty}(C^{-1})$$

3.6 Adaptive Chosen Ciphertext Attack

The **adaptive chosen ciphertext attack** differs from the chosen ciphertext attack so that the attacker may choose the ciphertexts adaptively depending on the previous responses of the oracle. A d-limited adaptive-chosen-plaintext-attack distinguisher between two independent random permutations C_1 and C_2 on a set $\mathcal{M}$ works as follows:

DISTINGUISHER 3.6 (ACCA): d-limited adaptive-chosen-plaintext-attack distinguisher

```
1. For $k = 1$ to $d$ do
    1.1 Choose a ciphertext message $y_k$, depending on the previous plaintexts and ciphertexts.
    1.2 Get $x_k = C_i^{-1}(y_k)$, where $i \in \{1, 2\}$.
2. Depending on $X = (x_1, \ldots, x_d)$ and $Y = (y_1, \ldots, y_d)$, output "accept" if you "think" the oracle
   implements $C_1$ or "reject" otherwise.
```

Similarly as for the chosen plaintext attack, the adaptive chosen plaintext attack is actually an adaptive chosen plaintext attack distinguishing C_1^{-1}, and C_1^{-1}. Formally: Let $\mathcal{A}$ be the set of all pairs (X, Y) such that the distinguisher outputs "accept". Then the probability that it outputs "accept" when the oracle implements the function C_i ($i = 1, 2$) is

$$
\begin{aligned}
p_i &= \sum_{Y,X} 1_{(X,Y)\in\mathcal{A}} \Pr[y_1] \Pr[y_2|y_1, x_1] \ldots \Pr[y_d|y_1, x_1, \ldots, y_{d-1}, x_{d-1}] \, [C_i]^d_{X,Y} \\
&= \sum_{X,Y} 1_{(Y,X)\in\mathcal{A}} \Pr[x_1] \Pr[x_2|x_1, y_1] \ldots \Pr[x_d|x_1, y_1, \ldots, x_{d-1}, y_{d-1}] \, [C_i]^d_{Y,X} \\
&= \sum_{X,Y} 1_{(X,Y)\in\mathcal{A}'} \Pr[x_1] \Pr[x_2|x_1, y_1] \ldots \Pr[x_d|x_1, y_1, \ldots, x_{d-1}, y_{d-1}] \, \left[[C_i]^d\right]^T_{X,Y} \\
&= \sum_{X,Y} 1_{(X,Y)\in\mathcal{A}'} \Pr[x_1] \Pr[x_2|x_1, y_1] \ldots \Pr[x_d|x_1, y_1, \ldots, x_{d-1}, y_{d-1}] \, [C_i^{-1}]^d_{X,Y}
\end{aligned}
$$

where $\mathcal{A}' = \{(X, Y) | (Y, X) \in \mathcal{A}\}$. Thus, similarly as in the previous section,

$$
Adv^{\mathrm{ACCA}(d)}(C_1, C_2) = Adv^{\mathrm{ACPA}(d)}(C_1^{-1}, C_2^{-1}).
$$

Theorem 3.6.1 *Let C_1 and C_2 be two independent random permutations on a set $\mathcal{M}$, and d an integer. Then*

$$
Adv^{\mathrm{ACCA}(d)}(C_1, C_2) = \frac{1}{2} \left\| C_1^{-1}, C_2^{-1} . \right\|_a
$$

Corollary 3.6.2 *Let C be a cipher, and d an integer. Then*

$$
AdvC^{\mathrm{ACCA}(d)}(C) = \frac{1}{2} \, DecC^d_{\|\cdot\|_a}(C^{-1}).
$$

(Adaptive) Chosen Ciphertext Attack Against Unbalanced Feistel Networks

The unbalanced Feistel network is a self-inverse structure, i.e. the encryption and decryption schemes are identical, only the order of the round functions is reversed. Therefore, the unbalanced Feistel network resists the (adaptive) chosen ciphertext attack if and only if it resists the (adaptive) chosen plaintext attack. Thus, from Theorem 3.3.8 and Theorem 3.4.8 we have that the two round unbalanced Feistel networks do not withstand the chosen ciphertext attack, and the three round unbalanced Feistel networks withstand the adaptive chosen ciphertext attack.

Note that in the general case the resistance to the plaintext attacks does not automatically imply resistance to the ciphertext attacks. For example see Section 7.2.3.

3.7 Chosen Plaintext-Ciphertext Attack

The **chosen plaintext-ciphertext attack** is the most powerful non-adaptive attack. A d-limited adaptive-chosen-plaintext-ciphertext-attack distinguisher between two functions C_1 and C_2 on a set $\mathcal{M}$ works as follows:

DISTINGUISHER 3.7 (CPCA): d-limited chosen-plaintext-ciphertext-attack distinguisher

1. Choose queries $Q = (q_1, \ldots, q_d)$ so that q_k is either $(0, x_k)$, or $(1, y_k)$.
2. Query the oracle with Q, and get $R = (r_1, \ldots, r_d)$, where $r_k = y_k = C_i(x_k)$ if $q_k = (0, x_k)$, or $r_k = x_k = C_i^{-1}(y_k)$ if $q_k = (1, y_k)$, for $i \in \{1, 2\}$.
3. Depending on $X = (x_1, \ldots, x_d)$ and $Y = (y_1, \ldots, y_d)$, output "accept" if you "think" the oracle implements C_1, or "reject" otherwise.

Let $\mathcal{A}$ be the set of all (Q, R) such that the distinguisher outputs "accept". We can define a set $\mathcal{M}' = \{0, 1\} \times \mathcal{M}$, and for both ciphers C_i new functions $F_i : \mathcal{M}' \to \mathcal{M}$ so that $F_i(0, x) = C_i(x)$, and $F_i(1, x) = C_i^{-1}(x)$. The probability that the distinguisher outputs "accept" when the oracle implements the function C_i ($i = 1, 2$) is then

$$p_i = \sum_Q \Pr[Q] \sum_R 1_{(Q,R)\in\mathcal{A}} \, [C_i]^d_{X,Y} = \sum_Q \Pr[Q] \sum_R 1_{(Q,R)\in\mathcal{A}} \, [F_i]^d_{Q,R}$$

Thus, we have transformed the chosen plaintext-ciphertext attack distinguishing between ciphers C_1 and C_2 into a chosen plaintext attack distinguishing between functions F_1 and F_2. In the same way, we can transform any chosen plaintext attack distinguishing between functions F_1 and F_2 into a chosen plaintext-ciphertext attack distinguishing between ciphers C_1 and C_2. Therefore,

$$Adv^{\mathrm{CPCA}(d)}(C_1, C_2) = Adv^{\mathrm{CPA}(d)}(F_1, F_2)$$

Theorem 3.7.1 *Let C_1 and C_2 be two independent ciphers on a set $\mathcal{M}$, and d an integer. Let F_1 and F_2 be constructed from C_1 and C_2 as described above. Then*

$$Adv^{\mathrm{CPCA}(d)}(C_1, C_2) = \frac{1}{2} \left|\left|\left| [F_1]^d - [F_2]^d \right|\right|\right|_\infty$$

Corollary 3.7.2 *Let C be a cipher, and d an integer. Let F and F' be created from C and C^* as described above. Then*

$$AdvC^{\mathrm{CPCA}(d)}(C) = \frac{1}{2} \left|\left|\left| [F]^d - [F']^d \right|\right|\right|_\infty$$

3.8 Adaptive Chosen Plaintext-Ciphertext Attack

The **adaptive chosen plaintext-ciphertext attack** is the most powerful attack. A cipher C is called **super-pseudorandom** [14] if there is no d-limited adaptive-chosen-plaintext-ciphertext-attack distinguisher between C and a perfect cipher for any d polynomial in $\log_2 |\mathcal{M}|$. A d-limited adaptive-chosen-plaintext-ciphertext-attack distinguisher between two functions C_1 and C_2 on a set $\mathcal{M}$ works as follows:

DISTINGUISHER 3.8 (ACPCA): d-limited adaptive-chosen-plaintext-ciphertext-attack distinguisher [21]

1. For $k = 1$ to d do
   - 1.1 Choose a query $q_k \in \{(0, x_k), (1, y_k)\}$ depending on the previous plaintexts and ciphertexts.
   - 1.2 Query the oracle with q_k and get the request $r_k = y_k = C_i(x_k)$ or $r_k = x_k = C_i^{-1}(y_k)$, where $i \in \{1, 2\}$.
2. Depending on $X = (x_1, \ldots, x_d)$ and $Y = (y_1, \ldots, y_d)$, output "accept" if you "think" the oracle implements C_1 or "reject" otherwise.

Let $\tau = ((q_1, r_1), \ldots, (q_d, r_d))$ be the sequence of queries of the distinguisher and responses of the oracle. Let $\mathcal{M}' = \{0, 1\} \times \mathcal{M}$, and F_1 and F_2 be functions from $\mathcal{M}'$ to $\mathcal{M}$ defined as $F_i(0, x) = C_i(x)$ and $F_i(1, x) = C_i^{-1}(x)$.

Let $\mathcal{A}$ be the set of all traces τ such that the distinguisher outputs "accept". Then the probability that it outputs "accept" when the oracle implements the function C_i ($i = 1, 2$) is

$$\begin{aligned} p_i &= \sum_{\tau=(Q,R)} 1_{\tau\in\mathcal{A}} \Pr[q_1] \Pr[q_2|q_1, r_1] \ldots \Pr[q_d|q_1, r_1, \ldots, q_{d-1}, r_{d-1}]\ [C_i]^d_{X_\tau, Y_\tau} \\ &= \sum_{\tau=(Q,R)} 1_{\tau\in\mathcal{A}} \Pr[q_1] \Pr[q_2|q_1, r_1] \ldots \Pr[q_d|q_1, r_1, \ldots, q_{d-1}, r_{d-1}]\ [F_i]^d_{Q,R} \end{aligned}$$

In this way, we have transformed the adaptive chosen plaintext-ciphertext attack distinguishing between ciphers C_1 and C_2 into a adaptive chosen plaintext attack distinguishing between functions F_1 and F_2. In the same way, we can transform any adaptive chosen plaintext attack distinguishing between functions F_1 and F_2 into a adaptive chosen plaintext-ciphertext attack distinguishing between ciphers C_1 and C_2. Therefore,

$$Adv^{\text{ACPCA}(d)}(C_1, C_2) = Adv^{\text{ACPA}(d)}(F_1, F_2)$$

Theorem 3.8.1 ([25]) *Let C_1 and C_2 be two independent ciphers on a set $\mathcal{M}$, and d an integer. Then*

$$Adv^{\text{ACPCA}(d)}(C_1, C_2) = \frac{1}{2} \left\| [C_1]^d - [C_2]^d \right\|_s$$

Proof: Let F_1 and F_2 be defined as above. Then

$$\begin{aligned} DecC^d_{\|\cdot\|_s}(C_1, C_2) = DecF^d_{\|\cdot\|_a}(F_1, F_2) &= 2\, Adv^{\text{ACPA}(d)}(F_1, F_2) \\ &= 2\, Adv^{\text{ACPCA}(d)}(C_1, C_2) \end{aligned}$$

■

Corollary 3.8.2 ([25]) *Let C be a cipher, and d an integer.*

$$AdvC^{\text{ACPCA}(d)}(C) = \frac{1}{2}\, DecC^d_{\|\cdot\|_s}(C)$$

Adaptive Chosen Plaintext-Ciphertext Attack Against Unbalanced Feistel Networks

In Section 3.4 we proved that in the random oracle model the three-round UFNs are secure against adaptive chosen plaintext attacks with a reasonable number of queries. Here we show that the three-round UFNs are not secure against adaptive chosen plaintext-ciphertext attacks.

Theorem 3.8.3 ([14]) *Let F_1, F_2, F_3 be any three independent random functions — F_1 and F_3 from a set $\mathcal{M}_2$ to a set $\mathcal{M}_1$ and F_2 from $\mathcal{M}_1$ to $\mathcal{M}_2$. Then $\Psi[F_1, F_2, F_3]$ is not secure against the adaptive chosen plaintext-ciphertext attack.*

Proof: There is a 3-limited adaptive-chosen-plaintext-ciphertext-attack distinguisher between the function $\Psi[F_1, F_2, F_3]$ and a perfect cipher:

DISTINGUISHER 3.9 (D): 3-limited distinguisher for $\Psi[F_1, F_2, F_3]$

1. Create a plaintext $x_1 = [L_1, R]$ for any L_1 and R, and get a ciphertext $y_1 = [L'_1, R'_1]$.
2. Create a plaintext $x_2 = [L_2, R]$ for any L_2, and get a ciphertext $y_2 = [L'_2, R'_2]$.
3. Create the ciphertext $y_3 = [L'_2 \oplus L_1 \oplus L_2, R'_2]$, and get a plaintext $x_3 = [L_3, R_3]$.
4. If the oracle implements Ψ, then for all $k = 1, 2, 3$:

$$\begin{aligned} R'_k &= R_k \oplus F_2(L_k \oplus F_1(R_k)) \\ L'_k &= L_k \oplus F_1(R_k) \oplus F_3(R'_k) \end{aligned}$$

   with $R_1 = R_2 = R$. Therefore,

$$\begin{aligned} L'_3 &= L_3 \oplus F_1(R_3) \oplus F_3(R'_3) = L_3 \oplus F_1(R_3) \oplus F_3(R'_2) \\ &= L_3 \oplus F_1(R_3) \oplus L_2 \oplus F_1(R) \oplus L'_2 \overset{(def.)}{=} L'_2 \oplus L_1 \oplus L_2 \end{aligned}$$

   Hence, $L_3 \oplus F_1(R_3) = L_1 \oplus F_1(R)$. Furthermore,

$$\begin{aligned} R'_3 &= R_3 \oplus F_2(L_3 \oplus F_1(R_3)) = R_3 \oplus F_2(L_1 \oplus F_1(R)) \\ &= R_3 \oplus R'_1 \oplus R \overset{(def.)}{=} R'_2 \end{aligned}$$

   Hence, $R_3 = R \oplus R'_1 \oplus R'_2$.
5. If $R_3 = R \oplus R'_1 \oplus R'_2$, output "accept", otherwise output "reject".

If the oracle implements Ψ, then the distinguisher always "accepts", i.e. $p = 1$. If the oracle implements a perfect cipher, then the probability the distinguisher "accepts" is the same as the probability that for two random ciphertexts $y_1 = [L'_1, R'_1]$ and $y_2 = [L'_2, R'_2]$, and a random plaintext $x_3 = [L_3, R_3]$, $R_3 = R + R'_1 + R'_2$, i.e. $p^* = \frac{1}{|\mathcal{M}_2|}$. The advantage of the distinguisher is thus $AdvC_D^{\text{ACPCA}(d)}(C) = 1 - \frac{1}{|\mathcal{M}_2|}$.

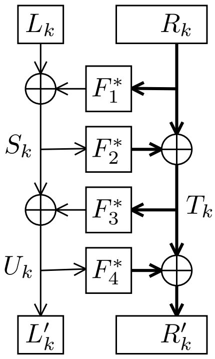
By adding further plaintexts and ciphertexts, a $3d$-limited distinguisher with advantage $AdvC_D^{\text{ACPCA}(3d)}(C) = 1 - \frac{1}{|\mathcal{M}_2|^d}$ for any $d \geq 1$ can be created. ■

Theorem 3.8.4 ([25]) *Let $F_1^*, F_2^*, F_3^*, F_4^*$ be four independent perfect random functions, F_1^* and F_3^* from $\mathcal{M}_2$ to $\mathcal{M}_1$ and F_2^* and F_3^* from $\mathcal{M}_1$ to $\mathcal{M}_2$, $\Psi[F_1^*, F_2^*, F_3^*, F_4^*]$ a 4-round UFN on $\mathcal{M} = \mathcal{M}_1 \times \mathcal{M}_2$, and d an integer. Then*

$$AdvC^{\text{ACPCA}(d)}(\Psi[F_1^*, F_2^*, F_3^*, F_4^*]) \leq \frac{d^2}{\min\{|\mathcal{M}_1|, |\mathcal{M}_2|\}}$$

Proof: The proof uses the technique introduced in [16].

Any d-limited known-plaintext-attack distinguisher has access to d plaintext/ciphertext pairs $(x_1, y_1), \ldots, (x_d, y_d)$. When the oracle implements the unbalanced Feistel network, the ciphertexts are calculated as depicted on the following figure.



$$\begin{aligned} x_k &= [L_k, R_k] \\ y_k &= [U_k, V_k] \end{aligned}$$

$$\begin{aligned} S_k &= L_k \oplus F_1^*(R_k) \\ T_k &= R_k \oplus F_2^*(S_k) \\ L'_k &= U_k = S_k \oplus F_3^*(T_k) \\ R'_k &= T_k \oplus F_3^*(T_k) \end{aligned}$$

In the case that the distinguisher queries the oracle with a plaintext: If all R_k's are pairwise distinct, then S_k's are perfectly random, since the function F_1^* is perfectly random. If all S_k's are pairwise distinct, then the T_k's are perfectly random, since the function F_2^* is perfectly random. If all S_k's and T_k's are perfectly random, then y_k's are also perfectly random.

In the case that the distinguisher queries the oracle with a ciphertext: If all U_k's are pairwise distinct, then the T_k's are perfectly random, since the function F_4^* is perfectly random. If all T_k's are pairwise distinct, then the S_k's are perfectly random, since the function F_3^* is perfectly random. If all S_k's and T_k's are perfectly random, then the x_k's are also perfectly random.

Without loss of generality, we may assume that queries are always distinct from all previous queries as well as from obtained answers of the oracle. Since Ψ is a permutation, all x_k's as well as all y_k's are pairwise different. In similar way as in proof of Theorem 3.4.8 we can obtain the following probabilities:

$$\Pr[S_k = S_l] = \begin{cases} 0 & R_k = R_l \\ \dfrac{1}{|\mathcal{M}_1|} & R_k \neq R_j \end{cases} \qquad \Pr[T_k = T_l] = \begin{cases} 0 & U_k = U_l \\ \dfrac{1}{|\mathcal{M}_2|} & U_k \neq U_j \end{cases}$$

Therefore, $\Pr[S_k = S_l] \leq \frac{1}{|\mathcal{M}_1|}$, $\Pr[T_k = T_l] \leq \frac{1}{|\mathcal{M}_2|}$, and from Theorem 2.2.3

$$\begin{aligned} AdvC^{\mathrm{ACPCA}(d)}(\Psi[F_1^*, F_2^*, F_3^*, F_4^*]) &\leq 1 - \Pr[\forall\, k \neq l : S_k \neq S_l \wedge T_k \neq T_l] \\ &= \Pr[\exists\, k \neq l : S_k = S_l \vee T_k = T_j] \\ &\leq \Pr[\exists\, k \neq l : S_k = S_j] + \Pr[\exists\, k \neq l : T_k = T_l] \\ &\leq \binom{d}{2}\frac{1}{|\mathcal{M}_1|} + \binom{d}{2}\frac{1}{|\mathcal{M}_2|} = \frac{d^2}{\min\{|\mathcal{M}_1|, |\mathcal{M}_2|\}} \end{aligned}$$

Note that adding a new round (even a weak one) between the second and third round cannot increase the advantage, because the proof does not depend on what happens between these two rounds. ■

Corollary 3.8.5 *Let $F_1^*, F_2^*, F_3^*, F_4^*$ be four independent perfect random functions, F_1^* and F_3^* from a set $\mathcal{M}_2$ to a set $\mathcal{M}_1$ and F_2^* and F_4^* from $\mathcal{M}_1$ to $\mathcal{M}_2$, $\Psi[F_1^*, F_2^*, F_3^*, F_4^*]$ a 4-round UFN on $\mathcal{M} = \mathcal{M}_1 \times \mathcal{M}_2$, and d an integer. Then*

$$DecC^d_{\|\cdot\|_s}(\Psi[F_1^*, F_2^*, F_3^*, F_4^*]) \leq 2\,\frac{d^2}{\min\{|\mathcal{M}_1|, |\mathcal{M}_2|\}}$$

Corollary 3.8.6 *Let $F_1, \ldots, F_r$ be $r \geq 4$ independent random functions such that $AdvF^{\mathrm{ACPCA}(d)}(F_i) \leq \varepsilon$, and d be an integer. Then*

$$AdvC^{\mathrm{ACPCA}(d)}(\Psi[F_1, \ldots, F_r]) \leq \frac{1}{2}\left[2\left(4\varepsilon + \frac{d^2}{\min\{|\mathcal{M}_1|, |\mathcal{M}_2|\}}\right)\right]^{\lfloor \frac{r}{4} \rfloor}$$

Proof: Follows from Theorem 2.4.4 and 2.4.2, and from the note at the end of the proof of Theorem 3.8.4. ■

3.9 Summary

In this chapter, we studied general types of attacks. For each one we found an associated matrix norm $\|\cdot\|$, and showed that the advantage between two functions (permutations) F_1, F_2 is

$$Adv^{\mathrm{ATK}(d)}(F_1, F_2) = \frac{1}{2}\,\|[F_1]^d - [F_2]^d\|,$$

The norm $\|\cdot\|$ is:

- $\|\cdot\|_1$ the for the known plaintext attack;
- $|||\cdot|||_\infty$ for the chosen plaintext attack and chosen ciphertext attack;
- $\|\cdot\|_a$ for the adaptive chosen plaintext attack and adaptive chosen ciphertext attack;
- $\|\cdot\|_s$ for the adaptive chosen plaintext-ciphertext attack.

From the definition of the norms, it follows that

$$Adv^{\mathrm{KPA}(d)}(F_1,F_2) \leq \left\{ \begin{array}{ll} Adv^{\mathrm{CPA}(d)}(F_1,F_2) & \leq Adv^{\mathrm{ACPA}(d}(F_1,F_2) \\ & \leq Adv^{\mathrm{CPCA}(d)}(F_1,F_2) \\ Adv^{\mathrm{CCA}(d)}(F_1,F_2) & \leq Adv^{\mathrm{ACCA}(d)}(F_1,F_2) \end{array} \right\} \leq Adv^{\mathrm{ACPCA}(d)}(F_1,F_2)$$

The relationship is depicted on the following picture so that the arrows direct from the weaker attacks to the stronger ones, i.e. from attacks with the lower upper-bound on the advantage to the ones with the higher upper-bound.

$$\begin{tikzcd}
& \mathrm{CPA} \arrow[r] \arrow[dr] & \mathrm{ACPA} \arrow[dr] & \\
\mathrm{KPA} \arrow[ur] \arrow[rr] \arrow[dr] & & \mathrm{CPCA} \arrow[r] & \mathrm{ACPCA} \\
& \mathrm{CCA} \arrow[r] \arrow[ur] & \mathrm{ACCA} \arrow[ur] &
\end{tikzcd}$$

We further examined unbalanced Feistel networks $\Psi[F_1,\ldots F_r] : \mathcal{M}_1 \times \mathcal{M}_2 \rightarrow \mathcal{M}_1 \times \mathcal{M}_2$ against the attacks, and showed that

- The 2-round UFNs are secure against known plaintext attack in the random oracle model;
- There is no 2-round UFN secure against chosen plaintext attacks;
- The 3-round UFNs are secure against adaptive chosen plaintext attack in the random oracle model;
- There is no 3-round UFN secure against adaptive chosen plaintext-ciphertext attacks;
- The 4-round UFNs are secure against adaptive chosen plaintext-ciphertext attacks in the random oracle model.

More formally: Let $\mathcal{D}_d = \frac{d^2}{\min\{|\mathcal{M}_1|,|\mathcal{M}_2|\}}$. Then for the d-limited known plaintext, adaptive chosen plaintext, and adaptive chosen plaintext-ciphertext attacks,

- $AdvC^{\mathrm{KPA}(d)}(\Psi[F_1^*,F_2^*]) \leq \mathcal{D}_d$
- $AdvC^{\mathrm{ACPA}(d)}(\Psi[F_1^*,F_2^*,F_3^*]) \leq \mathcal{D}_d$
- $AdvC^{\mathrm{ACPCA}(d)}(\Psi[F_1^*,F_2^*,F_3^*,F^*4]) \leq \mathcal{D}_d$

Note that the bounds are minimal for the balanced Feistel network. Further, the attacker must dispose $d \ll \sqrt{\min\{|\mathcal{M}_1|,|\mathcal{M}_2|\}}$ plaintext/ciphertext pairs in order to ensure security against the particular attack, and then the minimal number of rounds of a UFN is 3 for pseudorandomness, and 4 for super-pseudorandomness. However, for design of a UFN cipher, we usually need to quantify how many rounds it has to have in order to achieve a defined minimal security, or to find out what size of attack the cipher is able to withstand.

Consider now a UFN defined on $\{0,1\}^n = \{0,1\}^m \times \{0,1\}^{n-m}$, with $m \leq \frac{n}{2}$, and thus with $\min\{2^m, 2^{n-m}\} = 2^m$. Corollary 3.4.10 implies that in the ideal case, when the primitive functions are perfectly random,

$$AdvC^{\mathrm{ACPA}(d)}(\Psi[F_1^*,\ldots,F_{3k}^*]) \leq \frac{1}{2}\left(2 \cdot \frac{d^2}{2^m}\right)^k$$

Hence, in order to achieve the pseudorandomness with advantage less than 2^{-l}, we need

$$k \geq \frac{l-1}{m - 2\lg d - 1},$$

and thus at least $3\lceil \frac{l-1}{m-2\lg d-1} \rceil$ rounds, or we have to bound the size of the attack to

$$d \leq 2^{\frac{\lfloor \frac{r}{3} \rfloor (m-1)-l+1}{2k}},$$

for an r-round UFN.

Similarly, for the super-pseudorandomness, if we want to achieve advantage smaller than 2^{-l}, we need at least $4\lceil \frac{l-1}{m-2\lg d-1} \rceil$ rounds, or to bound the size of the attack to $d \leq 2^{\frac{k(m-1)-l+1}{2k}}$ for a $4k$-round UFN.

Evaluating the minimal number of rounds involves two parameters: the maximal size of the attack (d), and the upper-bound of the advantage.

The size of the attack may be surely limited by 2^n because that is the number of plaintext/ciphertext pairs, and thus with $d = 2^n$ the attacker already has all plaintext/ciphertext pairs and does not need to attack the cipher at all. However, in the calculation of the advantage, we require $d < 2^{\frac{m-1}{2}}$, but even this upper-bound may be considered to be unrealistic, since an attacker is usually able to get only a few plaintext/ciphertext pairs. Note that we require plaintext/ciphertext pairs encrypted with the right key, not any pairs which can be easily obtained whenever the encryption algorithm is known, which is the accepted rule known as Kerckhoffs's principle. To obtain many plaintext/ciphertext pairs encrypted with the right key would probably mean access to the encryption device together with the key for quite a long time, which is a security problem beyond the framework of this work. Therefore, we can consider d to be small.

On the other hand, advantage expresses the probability that an attacker can distinguish a cipher from a perfectly random output. It can be seen as amount of calculation necessary to distinguish between them. Thus, whenever n is considered to be an appropriate block length, i.e. when the exhaustive search of size 2^n is deemed to be infeasible, 2^{-n} can be considered as an appropriate value of advantage for the cipher of size n bits [17]. Applying this for UFNs, we get

$$k \geq \frac{n-1}{m-1-2\lg d}. \tag{3.2}$$

Chapter 4

Composed Attacks

In the previous chapter we showed under which conditions a cipher may withstand different types of attacks. It is natural to ask whether some combination of these attacks can lead to a more efficient attack against the cipher. This chapter gives some answers to this question. First, we examine the case when an attacker has a simple chosen plaintext attack and repeats it many times in order to get a better advantage. This kind of attack is called iterated attack. The differential and linear cryptanalysis ([4], [15]) happen to be included in this class of attacks, and because of their popularity, we focus on them separately. Then we examine combined attacks which make use of several distinct attacks with the same goal to build a stronger attack, and give upper bounds of their advantage. Here we consider the average advantage of the attacks. Although the small average advantage does not exclude the possibility of weak keys in a particular cipher, it shows that the attack does not work on average, which implies that the fraction of weak keys is negligible against the average case.

4.1 Basic Differential Cryptanalysis

Although **differential cryptanalysis** was invented by Eli Biham and Adi Shamir [4] in order to recover the encryption key of a cipher, we study here its distinguishing variant which exploits the underlying idea of the differential cryptanalysis. In our notion, the basic differential cryptanalysis is the $2d$-limited distinguisher with a characteristic $(a, b) \in \mathcal{M}^+ \times \mathcal{M}$ between a cipher C and a perfect cipher C^* both defined on $\mathcal{M} = \{0,1\}^m$, working as follows:

DISTINGUISHER 4.1 (DCA): $2d$-limited basic-differential-cryptanalysis distinguisher [21]

```
1. For $k = 1$ to $d$ do
    1.1 Choose a random plaintext message $x_k$.
    1.2 Query the oracle with $x_k$ and $x'_k = x_k + a$, and get $y_k = \tilde{C}(x_k)$ and $y'_k = \tilde{C}(x'_k)$, where $\tilde{C}$ is
        either $C$ or $C^*$.
    1.3 If $\tilde{C}(x_k + a) = \tilde{C}(x_k) + b$, stop and output “accept”.
2. Output “reject”.
```

Differential cryptanalysis depends on the following probability [21]:

$$DP^C[a,b](x) = \text{Pr}[C(x+a) = C(x) + b\,]$$

where x has the uniform distribution over all plaintext messages. The probability of success in one round is

$$\begin{aligned} DP^C[a,b] = E(DP^C[a,b](x)) &= \sum_x \text{Pr}[x] \cdot \text{Pr}[C(x+a) = C(x) + b\,] \\ &= \frac{1}{|\mathcal{M}|} \sum_x \text{Pr}[C(x+a) = C(x) + b\,] \\ &= \frac{1}{|\mathcal{M}|} \sum_x \sum_y \text{Pr}[C(x) = y \wedge C(x+a) = y + b\,] \\ &= \frac{1}{|\mathcal{M}|} \sum_x \sum_y [C]^2_{(x,x+a)(y,y+b)}. \end{aligned}$$

Since $a \neq 0$, for a perfect cipher:

$$DP^{C^*}[a,b] = \frac{1}{|\mathcal{M}|}\sum_x \sum_y \frac{1}{|\mathcal{M}|^{\underline{2}}} = \frac{1}{|\mathcal{M}|-1}.$$

The probability that the distinguisher accepts when C is implemented is

$$\begin{aligned}
p &= \sum_{X=(x_1,\ldots,x_d)\in\mathcal{M}^d} \Pr[X]\left[1-\prod_{i=1}^{d}(1-DP^C[a,b](x_i))\right] \\
&= \frac{1}{|\mathcal{M}|^d}\sum_{X=(x_1,\ldots,x_d)\in\mathcal{M}^d}\left[1-\prod_{i=1}^{d}(1-DP^C[a,b](x_i))\right] \\
&= 1-\frac{1}{|\mathcal{M}|^d}\sum_{X=(x_1,\ldots,x_d)\in\mathcal{M}^d}\prod_{i=1}^{d}(1-DP^C[a,b](x_i)) \\
&= 1-\frac{1}{|\mathcal{M}|^d}\left[\sum_{x\in\mathcal{M}}(1-DP^C[a,b](x))\right]^d \\
&= 1-\left[\frac{1}{|\mathcal{M}|}\sum_{x\in\mathcal{M}}(1-DP^C[a,b](x))\right]^d \leq d\cdot DP^C[a,b]
\end{aligned}$$

For a perfect cipher it gives,

$$p^* \leq d\cdot DP^{C^*}[a,b] = \frac{d}{|\mathcal{M}|-1}$$

Hence,

$$AdvC_D^{\mathrm{DCA}(2d)}(C) = |p-p^*| \leq \max\{p,p^*\} \tag{4.1}$$

$$= d\cdot\max\left\{DP^C[a,b], \frac{1}{|\mathcal{M}|-1}\right\} \tag{4.2}$$

Theorem 4.1.1 ([21]) *Let C be a cipher on $\mathcal{M}$, and d an integer. Then*

$$AdvC^{\mathrm{DCA}(2d)}(C) \leq \frac{d}{|\mathcal{M}|-1} + d\cdot DecC^2_{|||\cdot|||_\infty}(C).$$

Proof:

$$\begin{aligned}
\left|DP^C[a,b]-DP^{C^*}[a,b]\right| &= \left|\frac{1}{|\mathcal{M}|}\sum_x\sum_y [C]^2_{(x,x+a)(y,y+b)} - \frac{1}{|\mathcal{M}|}\sum_x\sum_y [C^*]^2_{(x,x+a)(y,y+b)}\right| \\
&\leq \frac{1}{|\mathcal{M}|}\sum_x\sum_y\left|[C]^2_{(x,x+a)(y,y+b)} - [C^*]^2_{(x,x+a)(y,y+b)}\right| \\
&\leq \frac{1}{|\mathcal{M}|}\sum_x\sum_y\sum_{y'}\left|[C]^2_{(x,x+a)(y,y')} - [C^*]^2_{(x,x+a)(y,y')}\right| \\
&\leq \frac{1}{|\mathcal{M}|}\sum_x\max_{x'}\sum_y\sum_{y'}\left|[C]^2_{(x,x')(y,y')} - [C^*]^2_{(x,x')(y,y')}\right| \\
&\leq \max_x\max_{x'}\sum_y\sum_{y'}\left|[C]^2_{(x,x')(y,y')} - [C^*]^2_{(x,x')(y,y')}\right| \\
&= DecC^2_{|||\cdot|||_\infty}(C)
\end{aligned}$$

Therefore,

$$DP^C[a,b] \leq \frac{1}{|\mathcal{M}|-1} + DecC^2_{|||\cdot|||_\infty}(C),$$

and from 4.2

$$AdvC^{\mathrm{DCA}(2d)}(C) \leq d\cdot\left(\frac{1}{|\mathcal{M}|-1} + DecC^2_{|||\cdot|||_\infty}(C)\right).$$

■

4.2 General Iterated Attack

In case of basic differential cryptanalysis, a simple 2-limited non-adaptive plaintext attack was repeated several times. This approach can be generalized to any non-adaptive plaintext attack: An **iterated attack** iterates n times an elementary d-limited plaintext-attack distinguisher D between a cipher C and a perfect cipher on $\mathcal{M}$. It is defined by

1. the underlying d-limited attack D;
2. the number of iterations n (complexity) of the underlying attack (the number of iterations n of the attack is assumed to be large, the order of the underlying attack d small);
3. a plaintext distribution $\mathcal{D}$ on $\mathcal{M}^d$, which describes the distribution on queries of the underlying d-limited distinguisher (if D is the uniform distribution, we will refer to **known-plaintext iterated attack**);
4. a test function $\mathcal{T}\colon \mathcal{M}^d \times \mathcal{M}^d \to \{0,1\}$ that corresponds to the acceptance set of the underlying d-limited distinguisher D — it returns 1 for all pairs (X,Y) for which D accepts; and
5. an acceptance set $\mathcal{A} \subseteq \{0,1\}^n$, that contains all combinations of the test function outputs for which the iterated distinguisher accepts;

and works as follows:

DISTINGUISHER 4.2 (IA): nd-limited iterated-attack distinguisher

```
1. For $k = 1$ to $n$ do
    1.1 Choose a random vector of plaintext messages $X_k = (x_{k1}, \dots, x_{kd})$ with distribution $\mathcal{D}$.
    1.2 Get $Y_k = (C_i(x_{k1}), \dots, C_i(x_{kd}))$, where $i \in \{1,2\}$, $C_1 = C$, and $C_2 = C^*$.
    1.3 Get the answer of the underlying $d$-limited distinguisher: $T_k = \mathcal{T}(X_k, Y_k)$.
2. If $(T_1, \dots T_n) \in \mathcal{A}$, output "accept", otherwise output "reject".
```

At the first glance, it is tempting to believe that a cipher resists this type of attack once it has a small d-wise decorrelation bias. The following example shows that this is not true.

Example 4.2.1 ([21]) *Let C be a cipher with a perfect d-wise decorrelation. Assume that each instance (induced by a key) is totally defined by any d distinct plaintext-ciphertext pairs. [For example, the cipher defined in Example 2.3.5 — $C(x) = ax + b$ — has perfect 2-wise decorrelation, and each key (a,b) can be reconstructed from any two distinct plaintext-ciphertext pairs.] Thus, there are $|\mathcal{M}|^{\underline{d}}$ different instances, and they can be indexed by numbers from 1 to $|\mathcal{M}|^{\underline{d}}$: $c_1, \dots, c_{|\mathcal{M}|^{\underline{d}}}$. For each d-tuple of plaintexts $X = (x_1, \dots, x_d)$ and ciphertexts $Y = (y_1, \dots, y_d)$ we can define an index function $I : \mathcal{M}^d \to \{1, \dots, |\mathcal{M}|^d\}$, so that $I(X,Y)$ is the unique index k such that $c_k(x_i) = y_i$ for all $i = 1, \dots, d$. Let now D be an $2d$-limited iterated-attack-distinguisher with the following properties:*

1. *$\mathcal{D}$ is uniform distribution;*
2. $\mathcal{T}(X,Y) = \begin{cases} 1 & \text{\textit{if} } I(X,Y) \equiv 0 \pmod{\mu} \\ 0 & \text{\textit{otherwise}} \end{cases}$
   with a given modulus $\mu = \frac{n}{a}$ for a constant $a < n$;
3. $\mathcal{A} = \{0,1\}^n \setminus \{(0, \dots, 0)\}$.

If the oracle implements C then for each pair (X,Y), there is exactly one value k such that $I(X,Y) = k$, and

$$\mathcal{T}(X,Y) = \begin{cases} 1 & \text{\textit{if} } k \equiv 0 \pmod{\mu} \\ 0 & \text{\textit{otherwise}} \end{cases}$$

Therefore,

$$p = \mathit{Pr}[k \equiv 0 \bmod \mu] = \frac{1}{\mu}$$

If the oracle implements C^* then the output is random, and thus any of k is equally possible. Therefore,

$$\mathit{Pr}[T_i = 1] = \frac{1}{\mu} \qquad \mathit{Pr}[T_i = 0] = 1 - \frac{1}{\mu}$$

and

$$\begin{aligned} p^* &= Pr[\exists i : T_i = 1] = 1 - Pr[\forall i : T_i = 0] \\ &= 1 - \prod_{i=1}^{n} Pr[T_i = 0] = 1 - \left(1 - \frac{1}{\mu}\right)^n \end{aligned}$$

Hence, if $n \geq 2$

$$\begin{aligned} AdvC^{\mathrm{IA}(nd)}(C) &= 1 - \left(1 - \frac{1}{\mu}\right)^n - \frac{1}{\mu} \geq 1 - \left(1 - \frac{1}{\mu}\right)^2 - \frac{1}{\mu} \\ &= \frac{1}{\mu}\left(1 - \frac{1}{\mu}\right) \end{aligned}$$

which can be quite large although C is perfectly pairwise decorrelated. The idea behind this attack is that the test $\mathcal{T}$ provides the same expected result for C and C^*, but a different standard deviation.

The previous example shows that decorrelation of order d is not sufficient to prove the security of a cipher against iterated attacks of order d. The following theorem proves that the decorrelation of order $2d$ is sufficient.

Theorem 4.2.2 *Let C be a cipher on $\mathcal{M}$ such that $AdvC^{\mathrm{CPA}(2d)}(C) \leq \varepsilon$ for some integer $d < \sqrt{|\mathcal{M}|}$. Let n be an integer. Let $\mathcal{D}$ be a plaintext distribution, and δ the probability that for two independent random plaintext vectors $X = (x_1, \ldots, x_d)$ and $X' = (x'_1, \ldots, x'_d)$ with distribution $\mathcal{D}$ there is i and j such that $x_i = x'_j$. Then for iterated attacks of order d and complexity n with plaintext distribution $\mathcal{D}$*

$$AdvC^{\mathrm{IA}(nd)}(C) \leq 3\left[\left(2\delta + \frac{2d^2}{|\mathcal{M}|} + 3\varepsilon\right) n^2\right]^{\frac{1}{3}} + n\varepsilon.$$

Proof: [The proof is based on the technique introduced in [24].]

The proof is presented in several steps.

Let $C_1 := C$, and $C_2 := C^*$.

1. Let Z_i be the probability (over the distribution of X) such that the test $\mathcal{T}$ accepts $(X, C_i(X))$, i.e.
$$Z_i = \sum_X \Pr[X]\mathcal{T}(X, C_i(X)) = E_X(\mathcal{T}(X, C_i(X))).$$
Note that Z_i depends on C_i.
2. Let $\tilde{X}$ denote a vector $(X_1, \ldots, X_n)$, and $\tilde{Y}$ denote $(Y_1, \ldots, Y_n)$, where all X_k's and Y_k's are vectors from $\mathcal{M}^d$. The probability that the attack accepts when the oracle implements C_i is:
$$\begin{aligned} p_i &= \sum_{\tilde{X}} \Pr[\tilde{X}] \sum_{\tilde{Y}} \Pr[C_i(\tilde{X}) = \tilde{Y}] \sum_{T_1,\ldots,T_n} 1_{(T_1,\ldots,T_n)\in\mathcal{A}} \Pr[\forall k : \mathcal{T}(X_k, Y_k) = T_k] \\ &= \sum_{\tilde{X}} \Pr[\tilde{X}] \sum_{\tilde{Y}} \Pr[C_i(\tilde{X}) = \tilde{Y}] \sum_{T_1,\ldots,T_n} 1_{(T_1,\ldots,T_n)\in\mathcal{A}} \prod_{k=1}^{n} Z_i^{T_k}(1 - Z_i)^{1-T_k} \\ &= \sum_{\tilde{X}} \Pr[\tilde{X}] \sum_{\tilde{Y}} \Pr[C_i(\tilde{X}) = \tilde{Y}] \sum_{T_1,\ldots,T_n} 1_{(T_1,\ldots,T_n)\in\mathcal{A}} Z_i^{T_1+\ldots+T_n}(1 - Z_i)^{n-T_1-\ldots-T_n} \\ &= E_{C_i}(f(Z_i)) \end{aligned}$$
where
$$f(z) = \sum_{T_1,\ldots,T_n} 1_{(T_1,\ldots,T_n)\in\mathcal{A}} z^{T_1+\ldots+T_n}(1 - z)^{n-T_1-\ldots-T_n} = \sum_{k=0}^{n} a_k z^k (1 - z)^{n-k}$$
for some constants a_k such that $a_k \leq \binom{n}{k}$.[1] Since $f(z)$ is a polynomial of degree at most n,
$$\begin{aligned} f'(z) &= \sum_{k=0}^{n} \left(k a_k z^{k-1}(1 - z)^{n-k} - a_k z^k (n - k)(1 - z)^{n-k-1}\right) \\ &= \sum_{k=1}^{n} k a_k z^{k-1}(1 - z)^{n-k} - \sum_{k=0}^{n-1} a_k z^k (n - k)(1 - z)^{n-k-1}. \end{aligned}$$

[1] $a(k) = \binom{n}{k}$ if and only if $\mathcal{A}$ contains all vectors with exactly k ones.

For the first sum,

$$\begin{aligned}\sum_{k=1}^{n} k a_k z^{k-1}(1-z)^{n-k} \leq \sum_{k=1}^{n} k\binom{n}{k} z^{k-1}(1-z)^{n-k} &= n\sum_{k=1}^{n}\binom{n-1}{k-1} z^{k-1}(1-z)^{n-k}\\ &= n\sum_{k=0}^{n-1}\binom{n-1}{k} z^{k}(1-z)^{n-1-k} = n(z+1-z)^{n-1} = n\end{aligned}$$

and for the second sum

$$\begin{aligned}\sum_{k=0}^{n-1} a_k z^{k}(n-k)(1-z)^{n-k-1} &\leq \sum_{k=0}^{n-1}\binom{n}{k} z^{k}(n-k)(1-z)^{n-k-1}\\ &= n\sum_{k=0}^{n-1}\binom{n-1}{k} z^{k}(1-z)^{n-1-k} = n(z+1-z)^{n-1} = n\end{aligned}$$

holds. Consequently,

$$|f'(z)| \leq \max\left\{\sum_{k=1}^{n} k a_k z^{k-1}(1-z)^{n-k}, \sum_{k=0}^{n-1} a_k z^{k}(n-k)(1-z)^{n-k-1}\right\} \leq n$$

Therefore,

$$|f(Z_1) - f(Z_2)| \leq n|Z_1 - Z_2|.$$

3. The probability that one round is accepted, i.e. that $\mathcal{T}(X_k, Y_k) = 1$, is

$$\begin{aligned}p_i^{\text{round}_k} &= \sum_X \text{Pr}[X] \sum_Y \text{Pr}[C_i(X) = Y] \cdot \text{Pr}[\mathcal{T}(X,Y) = 1]\\ &= \sum_X \text{Pr}[X] \sum_Y \text{Pr}[C_i(X) = Y] \cdot Z_i = E_{C_i}(Z_i)\end{aligned}$$

Since one round of the iterated attack is actually a chosen-plaintext attack,

$$|E_{C_1}(Z_1) - E_{C_2}(Z_2)| = \left|p_1^{\text{round}_k} - p_2^{\text{round}_k}\right| \leq AdvC^{\text{CPA}(d)}(C) \leq AdvC^{\text{CPA}(2d)}(C) \leq \varepsilon$$

4. Since

$$Z_i^2 = \left[\sum_X \text{Pr}[X]\mathcal{T}(X, C_i(X))\right]^2 = \sum_{X_1,X_2} \text{Pr}[X_1]\text{Pr}[X_2]\mathcal{T}(X_1, C_i(X_1))\mathcal{T}(X_2, C_i(X_2))$$

and since X_k's are chosen independently from each other,

$$Z_i^2 = \sum_{X'=(x_1,\ldots,x_{2d})} \text{Pr}[X']\mathcal{T}'(X', C_i(X'))]$$

corresponds to another test $\mathcal{T}'$ with $2d$ entries such that

$$\mathcal{T}'((X_1,X_2),(C_i(X_1),C_i(X_2)) = \mathcal{T}(X_1,C_i(X_1)) \cdot \mathcal{T}(X_2,C_i(X_2)),$$

i.e. $\mathcal{T}'((X_1,X_2),(C_i(X_1),C_i(X_2))$ accepts if and only if both $\mathcal{T}(X_1,C_i(X_1))$ and $\mathcal{T}(X_2,C_i(X_2))$ accept. Therefore from 3:

$$\left|E_{C_1}(Z_1^2) - E_{C_2}(Z_2^2)\right| \leq AdvC^{\text{CPA}(2d)}(C) \leq \varepsilon$$

and

$$\begin{aligned}|V_{C_1}(Z_1) - V_{C_2}(Z_2)| &= \left|E_{C_1}(Z_1^2) - E^2(Z_1) - E_{C_2}(Z_2^2) + E^2(Z_2)\right|\\ &\leq \left|E_{C_1}(Z_1^2) - E_{C_2}(Z_2^2)\right|\\ &\quad + |(E_{C_1}(Z_1) - E_{C_2}(Z_2)) \cdot (E_{C_1}(Z_1) + E_{C_2}(Z_2))|\\ &\leq \varepsilon + \varepsilon \cdot 2 = 3\varepsilon\end{aligned}$$

Hence, $V_{C_1}(Z_1) \leq V_{C_2}(Z_2) + 3\varepsilon$.

5. From the Chebyshev inequality, one finds:

$$\Pr[|Z_i - E_{C_i}(Z_i)| > \lambda] \le \frac{V_{C_i}(Z_i)}{\lambda^2}$$

Hence using 3,

$$\begin{aligned}
\Pr[|Z_1 - Z_2| > 2\lambda + \varepsilon] &\le \Pr[|Z_1 - E_{C_1}(Z_1)| + |E_{C_1}(Z_1) - E_{C_2}(Z_2)| \\
&\quad + |E_{C_2}(Z_2) - Z_2| > 2\lambda + \varepsilon] \\
&\le \Pr[|Z_1 - E_{C_1}(Z_1)| > \lambda] + \Pr[|Z_2 - E_{C_2}(Z_2)| > \lambda] \\
&\quad + \Pr[|E_{C_1}(Z_1) - E_{C_2}(Z_2)| > \varepsilon] \\
&\le \frac{V_{C_1}(Z_1)}{\lambda^2} + \frac{V_{C_2}(Z_2)}{\lambda^2} \le \frac{2V_{C_2}(Z_2) + 3\varepsilon}{\lambda^2}
\end{aligned}$$

6. Using 2 and 5:

$$\begin{aligned}
AdvC^{\mathrm{IA}(nd)}(C) &= |p_1 - p_2| \\
&= \left| \sum_{\tilde{X}} \Pr[\tilde{X}] \sum_{\tilde{Y}} \Big(\Pr[C_1(\tilde{X}) = \tilde{Y}] \cdot f(Z_1) - \Pr[C_2(\tilde{X}) = \tilde{Y}] \cdot f(Z_2) \Big) \right| \\
&= \left| \sum_{\tilde{X}} \Pr[\tilde{X}] \sum_{\tilde{Y}} \sum_{\tilde{Y}'} \Big(\Pr[C_1(\tilde{X}) = \tilde{Y}] \cdot \Pr[C_2(\tilde{X}) = \tilde{Y}'] f(Z_1) \right. \\
&\qquad \left. - \Pr[C_1(\tilde{X}) = \tilde{Y}] \cdot \Pr[C_2(\tilde{X}) = \tilde{Y}'] f(Z_2) \Big) \right| \\
&\le \sum_{\tilde{X}} \Pr[\tilde{X}] \sum_{\tilde{Y}} \sum_{\tilde{Y}'} \Pr[C_1(\tilde{X}) = \tilde{Y}] \Pr[C_2(\tilde{X}) = \tilde{Y}'] \, |f(Z_1) - f(Z_2)| \\
&= E(|f(Z_1) - f(Z_2)|) \le E(n|Z_1 - Z_2|) \\
&\le n\,(\varepsilon + 2\lambda) + \frac{2V_{C_2}(Z_2) + 3\varepsilon}{\lambda^2}
\end{aligned}$$

7. Since

$$\begin{aligned}
E_{C_2}(Z_2^2) &= \sum_{X,X'} \sum_{Y,Y'} \Pr[X]\Pr[X']\Pr[X \stackrel{C_2}{\to} Y, X' \stackrel{C_2}{\to} Y'] \mathcal{T}(X,Y)\mathcal{T}(X',Y') \\
E_{C_2}^2(Z_2) &= \sum_{X,X'} \sum_{Y,Y'} \Pr[X]\Pr[X']\Pr[X \stackrel{C_2}{\to} Y]\Pr[X' \stackrel{C_2}{\to} Y'] \mathcal{T}(X,Y)\mathcal{T}(X',Y')
\end{aligned}$$

we have

$$\begin{aligned}
V_{C_2}(Z_2) &= E_{C_2}(Z_2^2) - E_{C_2}^2(Z_2) \\
&= \sum_{\substack{X,X' \\ Y,Y'}} \Pr[X]\Pr[X']\mathcal{T}(X,Y)\mathcal{T}(X',Y') \left(\Pr\left[\substack{X \stackrel{C_2}{\to} Y \\ X' \stackrel{C_2}{\to} Y'}\right] - \Pr[X \stackrel{C_2}{\to} Y]\Pr[X' \stackrel{C_2}{\to} Y'] \right)
\end{aligned}$$

This sum is maximal when $\mathcal{T}(X,Y)$ and $\mathcal{T}(X',Y')$ are 1 for all terms with the same sign. Since

$$\sum_{\substack{X,X' \\ Y,Y'}} \Pr[X]\Pr[X'] \left(\Pr[X \stackrel{C_2}{\to} Y, X' \stackrel{C_2}{\to} Y'] - \Pr[X \stackrel{C_2}{\to} Y]\Pr[X' \stackrel{C_2}{\to} Y'] \right) = 0$$

we get

$$\max V_{C_2}(Z_2) = \frac{1}{2} \sum_{\substack{X,X' \\ Y,Y'}} \Pr[X]\Pr[X'] \left| \Pr\left[\substack{X \stackrel{C_2}{\to} Y \\ X' \stackrel{C_2}{\to} Y'}\right] - \Pr[X \stackrel{C_2}{\to} Y]\Pr[X' \stackrel{C_2}{\to} Y'] \right|$$

a) Sum of all terms for X and X' with colliding entries (recall that there are collision only between X and X', but no collisions inside X, and inside X', because we may assume that the underlying distinguisher always chooses different queries):

$$
\begin{aligned}
&\frac{1}{2}\sum_{\mathrm{coll}(X,X')}\sum_{Y,Y'}\Pr[X]\Pr[X']\left|\Pr\left[\begin{smallmatrix} X\xrightarrow{C_2}Y \\ X'\xrightarrow{C_2}Y'\end{smallmatrix}\right]-\Pr[X\xrightarrow{C_2}Y]\Pr[X'\xrightarrow{C_2}Y']\right| \\
&\leq \frac{1}{2}\sum_{\mathrm{coll}(X,X')}\Pr[X]\Pr[X']\sum_{Y,Y'}\Pr\left[\begin{smallmatrix} X\xrightarrow{C_2}Y \\ X'\xrightarrow{C_2}Y'\end{smallmatrix}\right] \\
&\qquad+\frac{1}{2}\sum_{\mathrm{coll}(X,X')}\Pr[X]\Pr[X']\sum_{Y,Y'}\Pr[X\xrightarrow{C_2}Y]\Pr[X'\xrightarrow{C_2}Y'] \\
&=\frac{1}{2}\sum_{\mathrm{coll}(X,X')}\Pr[X]\Pr[X']+\frac{1}{2}\sum_{\mathrm{coll}(X,X')}\Pr[X]\Pr[X'] \\
&=\delta
\end{aligned}
$$

b) Sum of all terms for X and X' without colliding entries, and Y and Y' with colliding entries. Since C_2 is a permutation, if a collision inside Y occurs, then $\Pr[C(X)=Y]=0$. Similarly, if there is a collision in Y' then $\Pr[C(X)=Y]=0$, and if there is a collision in $Y\cup Y'$ then $\Pr[C(X)=Y,C(X')=Y']=0$. Furthermore, there are $d^2|\mathcal{M}|\left[(|\mathcal{M}|-1)^{\underline{d-1}}\right]^2$ possibilities to choose collisions between Y and Y'. Thus,

$$
\begin{aligned}
&\frac{1}{2}\sum_{\text{non-coll}(X,X')}\sum_{\mathrm{coll}(Y,Y')}\Pr[X]\Pr[X']\left|\Pr\left[\begin{smallmatrix} X\xrightarrow{C_2}Y \\ X'\xrightarrow{C_2}Y'\end{smallmatrix}\right]-\Pr[X\xrightarrow{C_2}Y]\Pr[X'\xrightarrow{C_2}Y']\right| \\
&=\frac{1}{2}\sum_{\text{non-coll}(X,X')}\Pr[X]\Pr[X']\sum_{\substack{\mathrm{coll}(Y,Y')\\ \text{non-coll}(Y)\\ \text{non-coll}(Y')}}\Pr[X\xrightarrow{C_2}Y]\Pr[X'\xrightarrow{C_2}Y'] \\
&\leq\frac{1}{2}\sum_{\text{non-coll}(X,X')}\Pr[X]\Pr[X']\,d^2|\mathcal{M}|\left[(|\mathcal{M}|-1)^{\underline{d-1}}\right]^2\frac{1}{|\mathcal{M}|^{\underline{d}}}\frac{1}{|\mathcal{M}|^{\underline{d}}} \\
&=\frac{d^2}{2\,|\mathcal{M}|}\sum_{\text{non-coll}(X,X')}\Pr[X]\Pr[X']\leq\frac{d^2}{2|\mathcal{M}|}\sum_{X,X'}\Pr[X]\Pr[X'] \\
&=\frac{d^2}{2|\mathcal{M}|}
\end{aligned}
$$

c) Sum of all terms for X and X' without colliding entries, and Y and Y' without colliding entries. There are $|\mathcal{M}|^{\underline{2d}}$ possibilities to choose non-colliding Y and Y'. Thus,

$$
\begin{aligned}
&\frac{1}{2}\sum_{\text{non-coll}(X,X')}\sum_{\text{non-coll}(Y,Y')}\Pr[X]\Pr[X']\left|\Pr\left[\begin{smallmatrix} X\xrightarrow{C_2}Y \\ X'\xrightarrow{C_2}Y'\end{smallmatrix}\right]-\Pr[X\xrightarrow{C_2}Y]\Pr[X'\xrightarrow{C_2}Y']\right| \\
&=\frac{1}{2}\sum_{\text{non-coll}(X,X')}\Pr[X]\Pr[X']\cdot|\mathcal{M}|^{\underline{2d}}\left(\frac{1}{|\mathcal{M}|^{\underline{2d}}}-\frac{1}{\left(|\mathcal{M}|^{\underline{d}}\right)^2}\right) \\
&=\frac{1}{2}\left(1-\frac{|\mathcal{M}|^{\underline{2d}}}{\left(|\mathcal{M}|^{\underline{d}}\right)^2}\right)\sum_{\text{non-coll}(X,X')}\Pr[X]\Pr[X'] \\
&\leq\frac{1}{2}\left(1-\frac{|\mathcal{M}|^{\underline{2d}}}{\left(|\mathcal{M}|^{\underline{d}}\right)^2}\right)\leq\frac{d^2}{2|\mathcal{M}|}
\end{aligned}
$$

Summing all three values gives

$$
V_{C_2}(Z_2)\leq\delta+\frac{d^2}{|\mathcal{M}|}.
$$

8. Let $\lambda = \left(\frac{2V_{C_2}(Z_2)+3\varepsilon}{n}\right)^{\frac{1}{3}}$. Then

$$\begin{aligned}
AdvC^{\mathrm{IA}(nd)}(C) &\overset{6.}{\leq} n\left(\varepsilon + 2\lambda\right) + \frac{2V_{C_2}(Z_2) + 3\varepsilon}{\lambda^2} \\
&= n\varepsilon + 2n\left(\frac{2V_{C_2}(Z_2) + 3\varepsilon}{n}\right)^{\frac{1}{3}} + \left(2V_{C_2}(Z_2) + 3\varepsilon\right)\left(\frac{n}{2V_{C_2}(Z_2) + 3\varepsilon}\right)^{\frac{2}{3}} \\
&= n\varepsilon + 3n\left(\frac{2V_{C_2}(Z_2) + 3\varepsilon}{n}\right)^{\frac{1}{3}} \\
&\overset{7.}{\leq} n\varepsilon + 3\left[n^2\left(2\delta + \frac{2d^2}{|\mathcal{M}|} + 3\varepsilon\right)\right]^{\frac{1}{3}}
\end{aligned}$$

■

Corollary 4.2.3 ([24]) *Let C be a cipher on $\mathcal{M}$ such that $AdvC^{\mathrm{CPA}(2d)}(C) \leq \varepsilon$ for some $d < \sqrt{|\mathcal{M}|}$. Let n be an integer. Then for an iterated attacks of order d and complexity n with uniform plaintext distribution*

$$AdvC^{\mathrm{IA}(nd)}(C) \leq 3\left[\left(\frac{4d^2}{|\mathcal{M}|} + 3\varepsilon\right)n^2\right]^{\frac{1}{3}} + n\varepsilon$$

holds.

Proof: Since the plaintext distribution is uniform, the probability that for two independent random $X = (x_1, \ldots, x_d)$ and $X' = (x'_1, \ldots, x'_d)$ there is i and j such that $x_i = x'_j$ is:

$$\begin{aligned}
\delta = Pr[\exists i, j : x_i = x'_j] &\leq \sum_{i,j} Pr[x_i = x'_j] = \sum_{i,j}\sum_{x} Pr[x_i = x \wedge x'_j = x] \\
&= d^2|\mathcal{M}|\frac{\left[(|\mathcal{M}| - 1)^{\underline{d-1}}\right]^2}{\left[|\mathcal{M}|^{\underline{d}}\right]^2} = \frac{d^2}{|\mathcal{M}|}
\end{aligned}$$

The inequality follows now directly from Theorem 4.2.2. ■

The result of the corollary says that with a small advantage ε against the $2d$-limited chosen-plaintext-attack distinguishers, one needs

$$n = \Omega\left(\min\left\{1/\sqrt{\varepsilon}, \sqrt{|\mathcal{M}|}\right\}\right)$$

in order to get a significant advantage, unless the distribution $\mathcal{D}$ has some special property. Further, note that the previous results are not tight because of the use of the Chebyshev inequality.

Note On The Basic Differential Cryptanalysis

From the definition of the basic differential cryptanalysis, it is easy to see, that it is an iterated attack of order 2 with the following parameters:

1. Distribution $\mathcal{D}$ is distribution of $(x, x + a)$ with uniformly distributed X;

2. $\mathcal{T}((x, x'), (y, y')) = 1$ if and only if $y + b = y'$.

3. $\mathcal{A} = \{0, 1\}^n \setminus \{(0, \ldots, 0)\}$.

Corollary 4.2.4 *Let C be a cipher on $\mathcal{M}$ such that $AdvC^{\mathrm{CPA4}}(C) = \frac{1}{2}DecC^4_{|||\cdot|||_\infty}(C) \leq \varepsilon$, and n be an integer. Then,*

$$AdvC^{\mathrm{DCA}(2n)}(C) \leq 3\left[n^2\left(\frac{12}{|\mathcal{M}|} + 3\varepsilon\right)\right]^{\frac{1}{3}} + n\varepsilon,$$

Proof: The probability that $X = (x, x + a)$ and $X' = (x', x' + a)$ have a colliding entry is

$$\begin{aligned}
\delta &= \Pr[x = x' \vee x + a = x' + a \vee x + a = x' \vee x = x' + a] \\
&\leq \Pr[x = x'] + \Pr[x + a = x'] + \Pr[x = x' + a] = \frac{3}{|\mathcal{M}|}.
\end{aligned}$$

Now, we can use Theorem 4.2.2 to get the result. ■

Since Corollary 4.2.3 requires decorrelation of order 4, this result is weaker than the result of Theorem 4.1.1.

4.3 Linear Cryptanalysis

The **linear cryptanalysis** is another well-known iterated attack introduced by Mitsuru Matsui in [15]. A d-limited linear-cryptanalysis distinguisher between a cipher C and a perfect cipher on $\mathcal{M} = \{0,1\}^m$ with a characteristic $(a,b) \in \mathcal{M}^+ \times \mathcal{M}^+$ and acceptance set $\mathcal{B} \subseteq \{0,1,\ldots,n\}$ works as follows:

DISTINGUISHER 4.3 (LCA): n-limited linear-cryptanalysis distinguisher [21]

```
1. Initialize the counter $u = 0$.
2. For $k = 1$ to $n$ do
    2.1 Choose a random plaintext message $x_k$.
    2.2 Get $y_k = \tilde{C}(x_k)$, where $\tilde{C}$ is either $C$ or $C^*$.
    2.3 If $x_k \cdot a = \tilde{C}(x_k) \cdot b$ (the inner dot product is the parity of the bitwise AND of the operands),
        then increment the counter $u$.
3. If $u \in \mathcal{B}$, output "accept", otherwise output "reject"
```

The linear cryptanalysis is an iterated attack of order 1 with the following parameters:

1. The distribution $\mathcal{D}$ is uniform over $\mathcal{M}$;
2. $d = 1$;
3. $\mathcal{T}(X,Y) = 1$ if and only if $X \cdot a = Y \cdot b$;
4. $\mathcal{A}(T_1,\ldots,T_n) = \{v \in \{0,1\}^n | \omega(v) \in \mathcal{B}\}$, where $\omega(v)$ is number of ones in v.

Corollary 4.3.1 ([24]) *Let C be a cipher on $\mathcal{M}$ such that $AdvC^{\mathrm{CPA}(2)}(C) = \frac{1}{2}DecC^2_{|||\cdot|||_\infty}(C) \leq \varepsilon$, and n be an integer. Then*

$$AdvC^{\mathrm{LCA}(n)}(C) \leq 3\left[n^2\left(\frac{4}{|\mathcal{M}|} + 3\varepsilon\right)\right]^{\frac{1}{3}} + n\varepsilon,$$

Proof: Directly follows from the Corollary 4.2.3 for $d = 1$. ■

Note that in [21], another result for the linear cryptanalysis is given:

$$\lim_{n \to +\infty} \frac{AdvC^{\mathrm{LCA}(n)}(C)}{n^{\frac{1}{3}}} \leq 9.3\left(\frac{1}{|\mathcal{M}|-1} + 2DecC^2_{|||\cdot|||_\infty}(C)\right).$$

It is better than that of the previous Corollary, however, it also depends on $DecC^2_{|||\cdot|||_\infty}$ and it is only an asymptotic result.

4.4 Combined Attacks

The iterated attacks take a chosen plaintext attack and repeat it many times in order to get a better advantage. The **combined attacks** are similar, but they use different attacks in each round. In this section we examine whether a combination of less efficient attacks can lead to a more efficient attack against the cipher. A combined attack against a cipher C on $\mathcal{M}$ is defined by

1. a set of n distinguishers $D_1,\ldots,D_k$ realizing different attacks, each of order d_k — unlike the iterated attacks a combined attack assumes a small number of iterations n (complexity) and large order d_k of the underlying attacks;
2. a set of test functions $\mathcal{T}_k : \mathcal{M}^{d_k} \times \mathcal{M}^{d_k} \to \{0,1\}$ which correspond to the acceptance sets of the underlying distinguishers D_k, i.e. it returns 1 for all pairs (X,Y) for which D_k accepts; and
3. an acceptance set $\mathcal{A} \subseteq \{0,1\}^n$ which contains all combinations of the outputs of the test functions for which the combined distinguisher accepts;

and works as follows:

DISTINGUISHER 4.4 (CA): Combined-attack distinguisher of order d and complexity n

```
1. For $k = 1$ to $n$ do
    1.1 Choose a random plaintext vector $X_k = (x_{k1}, \dots, x_{kd})$.
    1.2 Get $Y_k = (C_i(x_{k1}), \dots, C_i(x_{kd}))$, where $i \in \{1, 2\}$, $C_1 = C$, and $C_2 = C^*$.
    1.3 Get the answer of the distinguisher $D_k$: $T_k = \mathcal{T}_k(X_k, Y_k)$.
2. If $(T_1, \dots T_n) \in \mathcal{A}$, output "accept", otherwise output "reject".
```

Theorem 4.4.1 *Let C be a cipher on a set $\mathcal{M}$. Let $D_1, \dots, D_n$ be n distinguishers realizing some attacks on C with advantages $Adv_{D_1}, \dots, Adv_{D_n}$. Let d_k denote the number of queries of D_k and let D_k^2 denote the following distinguisher:*

1. *Run the distinguisher D_k and set a to the result.*
2. *Run the distinguisher D_k and set b to the result.*
3. *If $a = b = 1$, output "accept", otherwise output "reject".*

Let $Adv_{D_k^2}$ denote its advantage, and δ_k denote the probability that two independent runs of the distinguisher D_k have queries with an input in common. Then

$$AdvC^{\mathrm{CA}(d_1,\dots,d_n)}(C) \leq \sum_{k=1}^{n} \left[Adv_{D_k} + 3 \left(2\delta_k + \frac{2d_k^2}{|\mathcal{M}|} + 2Adv_{D_k} + Adv_{D_k^2} \right)^{\frac{1}{3}} \right]$$

Proof: [The proof is based on the technique introduced in [24].]

The proof follows the same steps as the proof of Theorem 4.2.2

Let $C_1 := C$, and $C_2 := C^*$.

1. Let $Z_{i^{[k]}}$ be probability that the test $\mathcal{T}_k$ accepts $(X, C_i(X))$, i.e. $\Pr[\mathcal{T}_k(X, C_i(X)) = 1]$. Then

$$Z_{i^{[k]}} = \sum_{X} \Pr[X] \mathcal{T}_k(X, C_i(X)) = E_X(\mathcal{T}_k(X, C_i(X))).$$

2. Let $\tilde{X}$ denote a vector $(X_1, \dots, X_n)$, and $\tilde{Y}$ denote $(Y_1, \dots, Y_n)$, where all X_k's and Y_k's are vectors from $\mathcal{M}^{d_k}$. Then the probability that the combined attack accepts when the oracle implements C_i is:

$$\begin{aligned} p_i &= \sum_{\tilde{X}} \Pr[\tilde{X}] \sum_{\tilde{Y}} \Pr[C_i(\tilde{X}) = \tilde{Y}] \sum_{T_1,\dots,T_n} 1_{(T_1,\dots,T_n)\in\mathcal{A}} \Pr[\forall k : \mathcal{T}_k(X_k, Y_k) = T_k] \\ &= \sum_{\tilde{X}} \Pr[\tilde{X}] \sum_{\tilde{Y}} \Pr[C_i(\tilde{X}) = \tilde{Y}] \sum_{T_1,\dots,T_n} 1_{(T_1,\dots,T_n)\in\mathcal{A}} \prod_{k=1}^{n} (Z_{i^{[k]}})^{T_k} (1 - Z_{i^{[k]}})^{1-T_k} \\ &= E_{C_i} \left(f(Z_{i^{[1]}}, Z_{i^{[2]}}, \dots, Z_{i^{[n]}}) \right) \end{aligned}$$

where

$$\begin{aligned} f(z_1, \dots, z_n) &= \sum_{T_1,\dots,T_n} 1_{(T_1,\dots,T_n)\in\mathcal{A}}\, z_1^{T_1}(1 - z_1^{1-T_1}) \cdots z_n^{T_n}(z_n^{1-T_n}) \\ &= \sum_{(k_1,\dots,k_n)\in\{0,1\}^n} a_{k_1 k_2 \dots k_n} z_1^{k_1}(1 - z_1^{1-k_1}) \cdots z_n^{k_n}(1 - z_n^{1-k_n}) \end{aligned}$$

for some constants $a_k \in \{0, 1\}$. Since $f(z)$ is a polynomial of partial degrees at most 1,

$$\frac{\partial f}{\partial z_1} = \sum_{k_2,\dots,k_n} (a_{1k_2\dots k_n} - a_{0k_2\dots k_n}) z_2^{k_2}(1 - z_2^{1-k_2}) \cdots z_n^{k_n}(1 - z_n^{1-k_n})$$

Since

$$\begin{aligned}
S_1 &:= \sum_{k_2,\ldots,k_n} a_{1k_2\ldots k_n} z_2^{k_2}(1-z_2^{1-k_2})\cdots z_n^{k_n}(1-z_n^{1-k_n})\\
&\leq \sum_{k_2,\ldots,k_n} z_2^{k_2}(1-z_2^{1-k_2})\cdots z_n^{k_n}(1-z_n^{1-k_n})\\
&= \sum_{k_2} z_2^{k_2}(1-z_2^{1-k_2}) \sum_{k_3,\ldots,k_n} z_3^{k_3}(1-z_3^{1-k_3})\cdots z_n^{k_n}(1-z_n^{1-k_n})\\
&= (z_2+1-z_2) \sum_{k_3,\ldots,k_n} z_3^{k_3}(1-z_3^{1-k_3})\cdots z_n^{k_n}(1-z_n^{1-k_n})\\
&= \sum_{k_3,\ldots,k_n} z_3^{k_3}(1-z_3^{1-k_3})\cdots z_n^{k_n}(1-z_n^{1-k_n}) = \ldots = 1
\end{aligned}$$

and similarly,

$$S_2 := \sum_{k_2,\ldots,k_n} a_{0k_2\ldots k_n} z_2^{k_2}(1-z_2^{1-k_2})\cdots z_n^{k_n}(1-z_n^{1-k_n}) \leq 1$$

Thus,

$$\frac{\partial f}{\partial z_1} \leq \max\{S_1, S_2\} \leq 1$$

In a similar way, we get for all partial derivatives: $\frac{\partial f}{\partial z_k} \leq 1$. Therefore,

$$|f(Z_{1^{[1]}}, Z_{1^{[2]}}, \ldots, Z_{1^{[n]}}) - f(Z_{2^{[1]}}, Z_{2^{[2]}}, \ldots, Z_{2^{[n]}})| \leq \sum_{k=1}^{n} |Z_{1^{[k]}} - Z_{2^{[k]}}|.$$

3. The probability that one round is accepted (i.e. $\mathcal{T}_k(\tilde{X}_k, \tilde{Y}_k) = 1$) is

$$p_i^{\text{round}_k} = E_{C_i}(Z_{i^{[k]}})$$

Therefore,

$$|E_{C_1}(Z_{1^{[k]}}) - E_{C_2}(Z_{2^{[k]}})| = \left|p_1^{\text{round}_k} - p_2^{\text{round}_k}\right| \leq Adv_{D_k}$$

4. Since $Z^2_{i^{[k]}}$ corresponds to the attack D_k^2,

$$\left|E_{C_1}(Z^2_{1^{[k]}}) - E_{C_2}(Z^2_{2^{[k]}})\right| \leq Adv_{D_k^2}$$

and

$$\begin{aligned}
|V_{C_1}(Z_{1^{[k]}}) - V_{C_2}(Z_{2^{[k]}})| &= \left|E_{C_1}(Z^2_{1^{[k]}}) - E^2(Z_{1^{[k]}}) - E_{C_2}(Z^2_{2^{[k]}}) + E^2(Z_{2^{[k]}})\right|\\
&\leq Adv_{D_k^2} + 2Adv_{D_k}
\end{aligned}$$

Hence, $V_{C_1}(Z_{1^{[k]}}) \leq V_{C_2}(Z_{2^{[k]}}) + Adv_{D_k^2} + 2Adv_{D_k}$.

5. From the Chebyshev inequality, one can derive:

$$\Pr[|Z_{i^{[k]}} - E_{C_i}(Z_{i^{[k]}})| > \lambda_k] \leq \frac{V_{C_i}(Z_{i^{[k]}})}{\lambda_k^2}$$

Hence,

$$\Pr[|Z_{1^{[k]}} - Z_{2^{[k]}}| > 2\lambda_k + Adv_{D_k}] \leq \frac{2V_{C_2}(Z_{2^{[k]}}) + Adv_{D_k^2} + 2Adv_{D_k}}{\lambda_k^2}$$

6. From 2. and 5.:

$$\begin{aligned}
AdvC^{\mathrm{CA}(d_1,\ldots,d_n)}(C) &= |p_1 - p_2|\\
&\leq E(|f(Z_{1^{[1]}}, \ldots, Z_{1^{[n]}}) - f(Z_{2^{[1]}}, \ldots, Z_{2^{[n]}})|)\\
&\leq E\left(\sum_{k=0}^{n} |Z_{1^{[k]}} - Z_{2^{[k]}}|\right)\\
&\leq \sum_{k=1}^{n} \left(2\lambda_k + Adv_{D_k} + \frac{2V_{C_2}(Z_{2^{[k]}}) + Adv_{D_k^2} + 2Adv_{D_k}}{\lambda_k^2}\right)
\end{aligned}$$

7. As proved in Theorem 4.2.2 (Step 7)

$$V_{C_2}(Z_{2^{[k]}}) \leq \frac{1}{2} \sum_{\substack{X,X' \\ Y,Y'}} \Pr[X]\Pr[X'] \left| \Pr\left[{X \overset{C_2}{\to} Y \atop X' \overset{C_2}{\to} Y'} \right] - \Pr[X \overset{C_2}{\to} Y]\Pr[X' \overset{C_2}{\to} Y'] \right|$$
$$\leq \delta_k + \frac{d_k^2}{|\mathcal{M}|}.$$

8. Let $\lambda_k = \left(2V_{C_2}(Z_{2^{[k]}}) + Adv_{D_k^2} + 2Adv_{D_k}\right)^{\frac{1}{3}}$. Then

$$AdvC^{\mathrm{CA}(d_1,\dots,d_n)}(C) \leq \sum_{k=1}^{n} \left(3 \left[2V_{C_2}(Z_{2^{[k]}}) + Adv_{D_k^2} + 2Adv_{D_k} \right]^{\frac{1}{3}} + Adv_{D_k} \right)$$
$$\leq \sum_{k=1}^{n} \left(3 \left[2\delta_k + \frac{2d_k^2}{|\mathcal{M}|} + Adv_{D_k^2} + 2Adv_{D_k} \right]^{\frac{1}{3}} + Adv_{D_k} \right)$$

■

Similarly as for the generalized iterated attacks, the result of the previous theorem is not tight because of the use of the Chebyshev inequality.

Corollary 4.4.2 ([24]) *Let C be a cipher on a set $\mathcal{M}$. Let $D_1, \dots, D_n$ be n attacks on C with advantages $Adv_{D_1}, \dots, Adv_{D_n}$. Let d_k, denote the number of queries of D_k. Let D_k^2 and $Adv_{D_k^2}$ have the same meaning as in the previous theorem. If the underlying attacks have a uniform distribution on the plaintexts, then*

$$AdvC^{\mathrm{CA}(d_1,\dots,d_n)}(C) \leq \sum_{k=1}^{n} \left[Adv_{D_k} + 3\left(\frac{4d_k^2}{|\mathcal{M}|} + 2Adv_{D_k} + Adv_{D_k^2} \right)^{\frac{1}{3}} \right]$$

Proof: If the attacks have the uniform distribution of the plaintexts, $\delta_k \leq \frac{d_k^2}{|\mathcal{M}|}$. The result follows now from Theorem 4.4.1. ■

4.5 Conclusions

In this chapter, we analyzed two attacks which use less efficient attacks against a cipher as building blocks for a more efficient one: An iterated attack uses one chosen-plaintext attack with small size and repeat it many times. In opposite, a combined attack takes a few distinct attacks with large size and each one is executed once. In both types, to ensure security against the composed attacks using an attack of order d, security against the underlying attack of order d is not sufficient; we need to ensure security against the underlying attacks of order at least $2d$.

In the previous chapter the advantage of different types of attack against unbalanced Feistel networks was evaluated. We can use these results to determine the advantage of a composed attack against UFNs. Consider for example a differential cryptanalysis against an r-round unbalanced Feistel network with perfectly random round functions $\Psi[F_1^*, \dots, F_r^*]$ on $\{0,1\}^n = \{0,1\}^m \times \{0,1\}^{n-m}$. It depends on the chosen plaintext attack with the following advantage (Corollary 3.4.10):

$$AdvC^{\mathrm{CPA}(d)}(\Psi[F_1^*, \dots, F_r^*]) \leq AdvC^{\mathrm{ACPA}(d)}(\Psi[F_1^*, \dots, F_r^*])$$
$$\leq \frac{1}{2}\left(2\frac{d^2}{\min\{|\mathcal{M}_1|, |\mathcal{M}_2|\}} \right)^k$$

where $k = \lfloor \frac{r}{3} \rfloor$. Let $\min\{2^m, 2^{n-m}\} = 2^m$. Assume that we want to upper-bound the advantage by some value a. Then from Theorem 4.1.1

$$AdvC^{\mathrm{DCA}(2d)}(C) \leq d\left(\frac{1}{2^n - 1} + \frac{1}{2}\left(\frac{d^2}{2^m} \right)^k \right) \leq a$$
$$\left(\frac{d^2}{2^m} \right)^k \leq 2\left(\frac{a}{d} - \frac{1}{2^n - 1} \right) =: b$$
$$k \geq \frac{\lg b}{2\lg d - m}$$

and thus we need at least $\frac{3\ \lg b}{2\lg d - m}$ rounds to achieve resistance to the basic differential cryptanalysis. Similar formulas can be derived for other attacks and their combinations.

Chapter 5

Modes of Operation

The block size of the block ciphers is usually rather small (modern ciphers use 128 bit blocks), much smaller than the usual length of messages. Thus, it is necessary to extend the encryption of one block to encryption of the whole message with as little overhead as possible. Modes of operation handle this problem. They were first specified by the FIPS 81 standard [18] for DES, however, they are independent of the actual block cipher being used. In this chapter we describe individual modes, discuss their properties, namely security, fault-tolerance, and efficiency, and evaluate their advantage in the random oracle model, i.e. with the assumption that the underlying block cipher is perfect.

> **Notation:** Most of the modes use an *initialization vector* (IV) to randomize the plaintext message which is prepended to the ciphertext as the zeroth block. It makes it possible to encrypt two equal messages into distinct ciphertexts. To handle this feature, we will consider a modified notion of the perfect cipher C^* for these modes: It will be a function from $\mathcal{M}^*$ to $\mathcal{M}^*$ that for any input message $x_1x_2\ldots x_n$ consisting of n elements of $\mathcal{M}$ returns perfectly random output $y_0y_1\ldots y_n$ of $n+1$ elements of $\mathcal{M}$. Further, the advantage of the best distinguisher between a cipher used in a particular mode Mode and a perfect cipher querying an oracle with up to q messages containing together up to d blocks will be denoted by $AdvC^{\mathrm{ATK}(d|q)}(\mathrm{Mode}[C])$.

5.1 ECB Mode

The **electronic codebook (ECB) mode** is the simplest way of using a block cipher: the message is split into blocks which are independently encrypted using the same key. Formally, let $X = x_1x_2\ldots$ be a plaintext message. The ciphertext message $Y = y_1y_2\ldots$ is calculated by $y_i = C(x_i)$, and the ciphertext is decrypted by $x_i = C^{-1}(y_i)$ (see Figure 5.1).

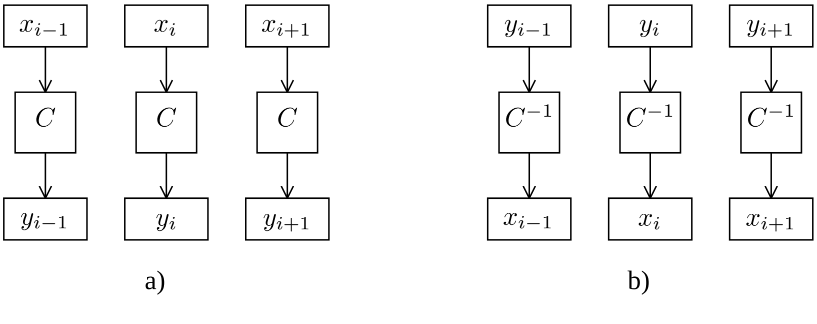


Figure 5.1: Electronic codebook mode: a) encryption, b) decryption

Security: The main problem of the ECB mode is that the same plaintext blocks are always encrypted into the same ciphertext blocks. Thus an attacker can start to compile the codebook without knowing the key (and cipher) - if he finds out the plaintext block corresponding to a ciphertext block, he is able to decrypt this ciphertext block whenever the block occurs again. It is also possible to mount statistical attacks on the underlying plaintext, irrespective of the strength of the block cipher. Furthermore, an attacker can modify ciphertext messages without knowing the key - he can insert, remove, interchange, or replay blocks (replay attack) without being detected, thus some authentication should be included into the plaintext.

Fault-tolerance: Alternation of a plaintext/ciphertext block only causes alternation of the corresponding ciphertext/plaintext block. If a part of the ciphertext whose size is not a multiple of the block size is inserted or lost, then all subsequent blocks are decrypted incorrectly.

Efficiency: Encryption as well as decryption may be done in parallel — any block can be encrypted or decrypted regardless of the other blocks.

Theorem 5.1.1 *Let C be a cipher on $\mathcal{M}$, ATK a class of attacks, and d and q integers ($q \le d$). Then,*

$$AdvC^{\mathrm{ATK}(d|q)}(\mathrm{ECB}[C]) = AdvC^{\mathrm{ATK}(d)}(C).$$

Proof: From any attack on a cipher in the ECB mode, it is possible to create an attack on the underlying cipher by ignoring the division to the individual messages, and similarly from an attack on the cipher, an attack on $\mathrm{ECB}[C]$ can be created by grouping the blocks into messages. ■

5.2 CBC Mode

In the **cipher block chaining (CBC) mode**, each plaintext block is XOR-ed with the previous ciphertext block before it is encrypted, and thus each ciphertext block dependents on all the previous plaintext blocks. Since there is no previous ciphertext block at the beginning of the encryption, an initialization vector is chosen at random. Formally, for a plaintext $X = x_1x_2\ldots$, the corresponding ciphertext $Y = y_0y_1y_2$ is computed as follows: $y_0 = \mathrm{IV}$, and $y_i = C(x_i \oplus y_{i-1})$ for all $i \ge 1$; the ciphertext is decrypted by $x_i = y_{i-1} \oplus C^{-1}(y_i)$ (see Figure 5.2).

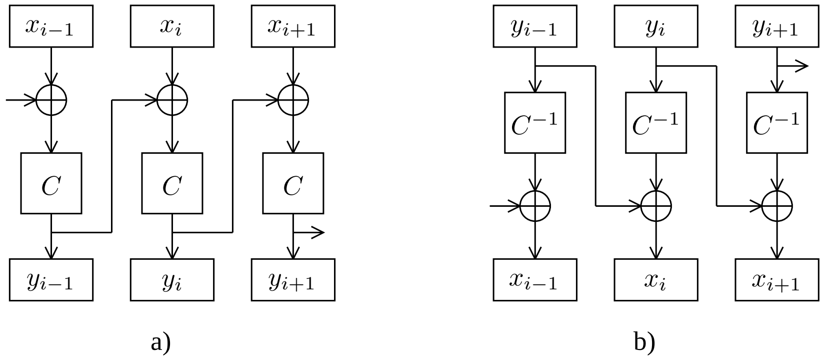


Figure 5.2: Cipher block chaining mode: a) encryption, b) decryption

Security: Some blocks can be inserted or removed from the beginning or end of an encrypted message without being detected, thus the message should have some indicator of the beginning and end. Plaintext patterns are concealed. The initialization vector should be unique; otherwise ciphertexts of two messages are equal until the first difference in their plaintexts.

Fault-tolerance: Alternation of a plaintext block x_i causes alternation of the corresponding ciphertext block y_i and of all subsequent blocks. However, it is not significant for decryption, because decryption reverses this effect, and the decrypted plaintext has the same error as the original one. On the other hand this feature may be used for creating of a message authentication code. Similarly, alternation of a ciphertext block y_i fully damages the corresponding plaintext block and causes alternation of the following plaintext block in the same bits; all following blocks are decrypted correctly. If a part of the ciphertext whose size is not a multiple of the block size is inserted or lost, then all subsequent blocks are decrypted incorrectly. Synchronization errors of full block sizes are recoverable with one garbled block.

Efficiency: Encryption cannot be done in parallel. Decryption may be done in parallel and enables random access to plaintext data. No preprocessing is possible.

In order to be able to decrypt the message, C has to be a cipher. However, for simplicity of the proof, we first consider the case where the underlying primitive is a perfect random function. Since attacks querying the oracle with ciphertexts need to calculate the inverse of the underlying primitive, with this substitution we are limited to plaintext attacks.

Lemma 5.2.1 *Let F^* be a perfect random function on $\mathcal{M}$, $\mathrm{ATK} \in \{\mathrm{CPA}, \mathrm{ACPA}\}$ a class of attacks, and d and q be integers ($q \le d$). Then*

$$AdvC^{\mathrm{ATK}(d|q)}(\mathrm{CBC}[F^*]) \le \frac{d^2}{2\,|\mathcal{M}|}.$$

Proof: [The proof is based on the technique introduced in [2].]

Let $C_1 := \text{CBC}[F^*]$, and $C_2 = C^*$. Assume that the attacker obtains the following plaintexts: $X_j = x_{j1}x_{j2},\ldots,x_{jn_j}$, and ciphertexts: $Y_j = y_{j0}y_{j1}y_{j2},\ldots,y_{jn_j}$, where $1 \leq j \leq q$, and n_j are the number of blocks in the individual messages. Thus, $n_1 + n_2 + \cdots + n_q = d$.

Let all y_{jl}, for $1 \leq j \leq q$, and $0 \leq l < n_k$, be ordered in one sequence, so that y_k denotes the k-th element of the sequence ($0 \leq k < d$), i.e.

$$y_k = y_{ab},$$

where a is the greatest integer such that $\sum_{j=1}^{a-1} n_j \leq k$, and $b = k - \sum_{j=1}^{a} n_j$. Let x_k denote the plaintext block encrypted in y_k (i.e. x_{ab}) if $b \neq 0$, or x_{an_a} if $b = 0$.

Let D_k denote the following event:

$$\forall u, v \leq k, u \neq v : y_u \oplus x_{u+1} \neq y_v \oplus x_{v+1},$$

(i.e. there is no collision in inputs to the function F^*). Let $D_{-1} = 1$, and $D = D_{d-1}$.

If D occurs, then the sequence of all ciphertext blocks y_{ab} is perfectly random because the elements are either initialization vectors generated at random, or outputs of F^*, but in that case the function was evaluated only for non-colliding (pairwise different) inputs, and thus they are perfectly random. Consequently, if D occurs, the distinguisher cannot distinguish between $\text{CBC}[F^*]$ and C^*.

Let $\Pr[\neg D_k | D_{k-1}]$ be the probability that a collision occurs the first time in the k-th element provided that the oracle implements the CBC mode with perfect cipher. If y_k is y_{ab} for some a, and $b > 0$ then since y_{k-1} did not have a collision, the cipher is evaluated for a new value, and

$$\begin{aligned}\Pr[\neg D_k | D_{k-1}] &= \Pr[\exists u < k : y_k = y_u \oplus x_{u+1} \oplus x_{k+1} | D_{k-1}] \\ &= \Pr[\exists u < k : F^*(y_{k-1} \oplus x_k) = y_u \oplus x_{u+1} \oplus x_{k+1} | D_{k-1}] = \frac{k}{|\mathcal{M}|}.\end{aligned}$$

If y_k is y_{a0} for some a, then it is a random IV, and

$$\Pr[\neg D_k | D_{k-1}] = \Pr[\exists u < k : y_{a0} = y_u \oplus x_{u+1} \oplus x_{k+1} | D_{k-1}] = \frac{k}{|\mathcal{M}|}.$$

Now, using Theorem 2.2.3

$$AdvC^{\text{ATK}(d|q)}(\text{CBC}[F^*]) = \Pr[\neg D] = \Pr[\neg D_{d-1}] \leq \sum_{k=0}^{d-1} \Pr[\neg D_k | D_{k-1}] = \sum_{k=0}^{d-1} \frac{k}{|\mathcal{M}|} = \frac{d^2}{2\,|\mathcal{M}|}$$

■

Theorem 5.2.2 *Let C^* be a perfect cipher on $\mathcal{M}$, $\text{ATK} \in \{\text{CPA}, \text{ACPA}\}$ a class of attacks, and d and q integers ($q \leq d$). Then*

$$AdvC^{\text{ATK}(d|q)}(\text{CBC}[C^*]) \leq \frac{d^2}{|\mathcal{M}|}.$$

Proof: Follows from Theorems 2.4.2 ($\text{ATK}^+ = \text{ATK}$) and 5.2.2, and Corollaries 3.3.7 or Corollary 3.4.7. ■

Corollary 5.2.3 *Let C be a cipher on $\mathcal{M}$, $\text{ATK} \in \{\text{CPA}, \text{ACPA}\}$ a class of attacks, and d and q integers ($q \leq d$). Then*

$$AdvC^{\text{ATK}(d|q)}(\text{CBC}[C]) \leq AdvC^{\text{ATK}(d)}(C) + \frac{d^2}{|\mathcal{M}|}.$$

Proof: Follows from Theorem 2.4.2 ($\text{ATK}^+ = \text{ATK}$), and Theorem 5.2.2. ■

We showed that the CBC mode is secure against the (adaptive) chosen plaintext attack, i.e. it is pseudorandom. In the rest of this section we discuss the lower bound on the advantage of the CBC mode.

Theorem 5.2.4 *Let F^* be a perfect random function on $\mathcal{M}$, and $d < \sqrt{2\mathcal{M}}$ and q be integers ($q \leq d$). Then,*

$$AdvC^{\mathrm{CPA}(d|q)}(\mathrm{CBC}[F^*]) \geq \left(1 - \frac{1}{e} - \frac{1}{|\mathcal{M}|}\right) \cdot \frac{d^2}{2\,|\mathcal{M}|}.$$

Proof: [The proof is based on the technique introduced in [2].]

Consider the following distinguisher:

DISTINGUISHER 5.1 (CBC): d-limited distinguisher for CBC

```
1. Create $q$ messages $X_k = x_{k1} \ldots x_{kn_q}$ ($1 \leq k \leq q$, $\sum_{k=1}^{q} n_k = d$), having the same
   value in all blocks, i.e. $\exists v \in \mathcal{M} : \forall k, a : x_{ka} = v$.
2. For $k = 1$ to $q$ do
    2.1 Get $Y_k = y_{k0} y_{k1} \ldots y_{kn_q} = C(X_k)$, where $C$ is either $\mathrm{CBC}[F^*]$ or $C^*$.
3. If $\exists u, v \leq q, a < n_u, b < n_v, (u,a) \neq (v,b) : y_{ua} = y_{vb}$
    3.1 Then if $y_{u(a+1)} = y_{v(b+1)}$ then output "accept".
4. Output "reject".
```

Note that unlike the attacks against the underlying ciphers, when a mode of operation using the initialization vector to randomize input is attacked, it makes sense to query the oracle with equal content.

Let D_k and D have the same meaning as in the proof of Theorem 5.2.2. When the oracle implements the CBC mode, the distinguisher accepts whenever a collision occurs. When the oracle implements a perfect cipher, the distinguisher accepts if there are two equal subblocks of length 2 in the ciphertext. Thus,

$$\begin{aligned} p_0 &= \Pr[\neg D] \\ p_1 &\leq \binom{d}{2} \frac{1}{|\mathcal{M}|^2} \\ |p_0 - p_1| &\geq \Pr[\neg D] - \frac{d^2}{2\,|\mathcal{M}|^2}. \end{aligned}$$

Since

$$\begin{aligned} \Pr[D] = \Pr[D_{d-1}] &= \prod_{k=0}^{d-1} \Pr[D_k | D_{k-1}] = \prod_{k=0}^{d-1} (1 - \Pr[\neg D_k | D_{k-1}]) \\ &= \prod_{k=0}^{d-1} \left(1 - \frac{k}{|\mathcal{M}|}\right) \leq \prod_{k=0}^{d-1} e^{-\frac{k}{|\mathcal{M}|}} = e^{-\sum_{k=0}^{d-1} -\frac{k}{|\mathcal{M}|}} = e^{-\frac{d^2}{2\,|\mathcal{M}|}} \end{aligned}$$

we get

$$\Pr[\neg D] = 1 - \Pr[D_{d-1}] \geq 1 - e^{-\frac{d^2}{2\,|\mathcal{M}|}} \geq \left(1 - \frac{1}{e}\right) \frac{d^2}{2\,|\mathcal{M}|}.$$

Therefore,

$$AdvC^{\mathrm{CPA}(d|q)}(\mathrm{CBC}[F^*]) \geq \left(1 - \frac{1}{e} - \frac{1}{|\mathcal{M}|}\right) \cdot \frac{d^2}{2\,|\mathcal{M}|}.$$

■

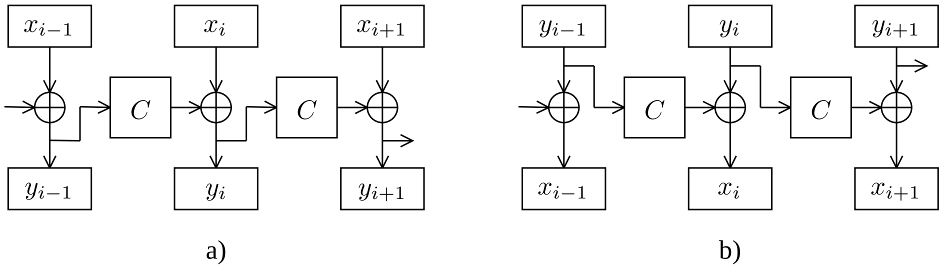
The distinguisher from the previous proof has a chance to distinguish $\mathrm{CBC}[F^*]$ from C^* only if a collision occurs. The probability that a collision occurs is higher when the number of chosen plaintexts d is higher. However, the higher the probability of collision, the higher the probability that it occurs also when C^* is implemented ($\frac{d^2}{2\,|\mathcal{M}|} \rightsquigarrow 1$). Consequently, the distinguisher has the greatest lower bound on the advantage when $\left(\frac{d^{\underline{2}}}{2|\mathcal{M}|}\right)^{\underline{2}}$ is maximal, thus when $d^{\underline{2}} = |\mathcal{M}|$, i.e. $d \approx \sqrt{|\mathcal{M}|}$. In that case, $AdvC^{\mathrm{CPA}(d|q)}(\mathrm{CBC}[F^*]) \geq \frac{1}{4}\left(1 - \frac{1}{e}\right)$.

The lower bound indicates that even if the CBC mode uses a perfect random function for the underlying primitive, it always leaks some information. However, the upper bound from Theorem 5.2.2 implies that the attack described in the proof of Theorem 5.2.4 is essentially the best one, up to a constant factor.

5.3 CFB Mode

The **cipher feedback (CFB) mode** is actually a *self-synchronizing stream cipher.* As defined in [18], the feedback structure is more complex than the one we present here. We discuss the original form at the end of this section.

The key stream in the CFB mode is produced by encrypting the previous ciphertext block y_{i-1}, which is XOR-ed with the plaintext block. As in the ECB mode, an initialization vector (IV) has to be chosen before the encryption, i.e. $y_0 = \mathrm{IV}$, and $y_i = x_i \oplus K_i$ where $K_i = C(y_{i-1})$, or shortly $y_i = x_i \oplus C(y_{i-1})$, for all $i \geq 1$; the decryption works in the same way $x_i = y_i \oplus C(y_{i-1})$ (see Figure 5.3).



a) b)

Figure 5.3: Cipher feedback mode: a) encryption, b) decryption

Security: Some blocks can be inserted or removed from the beginning or the end of an encrypted message without being detected, thus the message should have some indicator of the beginning and end of the message. Plaintext patterns are concealed. The initialization vector should be unique; otherwise ciphertexts of two messages are equal until the first difference in their plaintexts.

Fault-tolerance: Alternation of a plaintext block x_i causes alternation of the corresponding ciphertext block y_i and of all subsequent blocks. However, it is not significant for the decryption, because the decryption reverses this effect, and the recovered plaintext has the same error. On the other hand this feature may be used for creating a message authentication code. Similarly, alternation of a ciphertext block y_i causes alternation of the corresponding plaintext block in the same bits and fully damages the following plaintext block. Loss or addition of a full block is recoverable with one garbled block; synchronization errors of other sizes are unrecoverable.

Efficiency: Encryption cannot be done in parallel. Decryption can be done in parallel and enables random access to plaintext data. No preprocessing is possible.

Since the underlying cipher is never used for decryption, it does not need to be a cipher (does not need to be invertible). We will further consider a perfect random function as the underlying primitive. The advantage of the CFB mode with a cipher we discuss in the summary section of this chapter.

Theorem 5.3.1 *Let F^* be a perfect random function on $\mathcal{M}$,* ATK *a class of attacks, and d and q integers ($q \leq d$). Then*

$$AdvC^{\mathrm{ATK}(d|q)}(\mathrm{CFB}[F^*]) \leq \frac{d^{\underline{2}}}{2|\mathcal{M}|}.$$

Proof: Assume that the attacker obtains the following plaintexts: $X_j = x_{j1}x_{j2},\ldots,x_{jn_j}$, and ciphertexts: $Y_j = y_{j0}y_{j1}y_{j2},\ldots,y_{jn_j}$, where $1 \leq j \leq q$, and n_j are the number of blocks in the individual messages. Thus, $n_1 + n_2 + \cdots + n_q = d$.

Let all y_{jl}, for $1 \leq j \leq q$, and $0 \leq l < n_k$, be ordered in one sequence, and let the y_k denote the k-th element of the sequence ($0 \leq k < d$), i.e.

$$y_k = y_{ab},$$

where a is the greatest integer such that $\sum_{j=1}^{a-1} n_j \leq k$, and $b = k - \sum_{j=1}^{a} n_j$. Let x_k denote the plaintext block encrypted in y_k (i.e $y_k = x_k \oplus F^*(y_{k-1})$ if $b > 0$).

Let D_k denote the following event:

$$\forall u, v \leq k, u \neq v : y_u \neq y_v.$$

Let $D_{-1} = 1$, and $D = D_{d-1}$.

If D occurs, then the sequence of all ciphertext blocks y_{ab} is perfectly random because the elements are either initialization vectors generated at random or $y_k = F^*(y_{k-1}) \oplus x_k$, but in that case the function was evaluated only for non-colliding inputs, and thus they are perfectly random. Consequently, if C occurs, the distinguisher cannot distinguish between $\mathrm{CBC}[F^*]$ and C^*.

Let $\Pr[\neg D_k | D_{k-1}]$ be the probability, that a collision occurs in the k-th element provided that the oracle implements the CBC mode with perfect random cipher. If y_k is y_{ab} for some a, and $b > 0$ then since y_{k-1} did not have a collision, $F^*$is evaluated for a new value, and

$$\begin{aligned} \Pr[\neg D_k | D_{k-1}] &= \Pr[\exists u < k : y_k = y_u | D_{k-1}] \\ &= \Pr[\exists u < k : F^*(y_{k-1}) = y_u \oplus x_k | D_{k-1}] \leq \frac{k}{|\mathcal{M}|}. \end{aligned}$$

If y_k is equal to y_{a0} for some a then it is a random IV, and

$$\Pr[\neg D_k | D_{k-1}] = \Pr[\exists u < k : y_{a0} = y_u | D_{k-1}] \leq \frac{k}{|\mathcal{M}|}.$$

Now, as in the proof of Theorem 5.2.2,

$$AdvC^{\mathrm{ATK}(d|q)}(\mathrm{CBC}[F^*]) = \Pr[\neg D] \leq \frac{d^2}{2\,|\mathcal{M}|}$$

■

Corollary 5.3.2 *Let F be a random function on $\mathcal{M}$,* ATK *a class of general attacks from Chapter 3, and d and q integers ($q \leq d$). Then*

$$AdvC^{\mathrm{ATK}(d|q)}(\mathrm{CFB}[F]) \leq AdvF^{\mathrm{ATK}^+(d)}(F) + \frac{d^2}{2\,|\mathcal{M}|}.$$

Proof: Follows from Theorems 2.4.2 and 5.3.1. ■

Theorem 5.3.3 *Let F^* be a perfect random function on $\mathcal{M}$, and d and q be integers ($q \leq d$). Then*

$$AdvC^{\mathrm{CPA}(d|q)}(\mathrm{CFB}[F^*]) \geq \left(1 - \frac{1}{e} - \frac{1}{|\mathcal{M}|}\right) \cdot \frac{d^2}{2\,|\mathcal{M}|}.$$

Proof: The proof is similar to the one of Theorem 5.2.4 and is omitted. ■

The feedback structure defined in [18] allows one to divide the plaintext message into smaller units than the block length of the underlying cipher. In that case, the output of the cipher must be reduced into a smaller block which is then XORed with the plaintext unit. The input to the cipher consist of several previous ciphertext blocks, and not only of the last one as in the simplified version, and is stored in a shift register, which shifts after each encryption and adds the last key-stream block (see Figure 5.4).

Assume that the plaintext message is divided into blocks from a set $\mathcal{M}$. From the construction of the input into the cipher, it follows that the cipher has to be defined on $\mathcal{M}^L$, where L is the size of the feedback (the number of the last ciphertext blocks stored in the shift register). Output of the cipher is after encryption reduced from an element of $\mathcal{M}^L$ to an element of $\mathcal{M}$. Actually, we do not need to concern the structure of the function reducing the output of the cipher to the appropriate size; we can consider the composition of the cipher and the reduction function to be a random function from $\mathcal{M}^L$ to $\mathcal{M}$.

As in the simplified OFB mode, the shift register has to be initialized by a random initial vector at the beginning of encryption. This is, however, much longer comparing with the plaintext blocks than it was in the simplified version. The plaintext $X = x_1 x_2 \ldots$ is encrypted into $Y = y_{-L+1} \ldots y_0 y_1 y_2 \ldots$, where $y_{-L+1} \ldots y_0$ is the initial vector.

The advantage of the original CFB mode may be evaluated in the same way as of the simplified version.

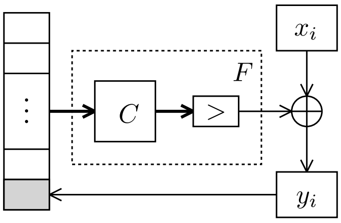


Figure 5.4: Generalized ciphertext feedback mode

Theorem 5.3.4 *Let F^* be a perfect random function from $\mathcal{M}^L$ to $\mathcal{M}$,* ATK *a class of attacks, and d and q integers ($q \leq d$). Then*

$$AdvC^{\mathrm{ATK}(d|q)}(\mathrm{CFB}[F^*]) \leq \frac{d^2}{2\,|\mathcal{M}|^L}.$$

Proof: The proof is similar to the one of Theorem 5.3.1, but the sequence $Y_0, Y_1, \ldots, Y_d$ is created from the big inputs to the function F^*, i.e.

$$Y_k = y_{a(b-L)} y_{ab-L+1} \cdots y_{ab}$$

where a is the greatest integer such that $\sum_{j=1}^{a-1} n_j \leq k$, and $b = k - \sum_{j=1}^{a} n_j$. Let Y_k^i denote $y_{a(b-L+i)}$ for all $0 \leq i < L$. Let x_k be the plaintext block encrypted in Y_k^{L-1}.

Let D_k again denote the event:

$$\forall u, v \leq k, u \neq v : Y_u \neq Y_v.$$

Let $D_{-1} = 1$, and $D = D_{d-1}$.

If D occurs then the sequence of all ciphertext blocks y_{ab} is perfectly random.

$$\begin{aligned}\Pr[\neg D_k | D_{k-1}] &= \Pr[\exists u < k : Y_k = Y_u | D_{k-1}] \\ &= \Pr[\exists u < k\, \forall i : Y_k^i = Y_u^i | D_{k-1}].\end{aligned}$$

For a fixed u, and i, If $\Pr[Y_k^i = Y_u^i | D_{k-1}] = \frac{1}{|\mathcal{M}|}$, since Y_k^i is either a random IV block, or $Y_k^i = x_k \oplus F^*(Y_{k-L+1+i})$ which are all evaluated for distinct inputs. Therefore,

$$\Pr[\neg D_k | D_{k-1}] \leq \frac{k}{|M|^L}.$$

Now, similarly as in the proof of Theorem 5.2.2,

$$AdvC^{\mathrm{ATK}(d|q)}(\mathrm{CBC}[F^*]) = \Pr[\neg D] \leq \frac{d^2}{2\,|\mathcal{M}|^L}$$

■

Thus, if we consider both the simplified and the original CFB mode with inputs to the underlying function from $\mathcal{M}^L$, they have the same advantage. The only disadvantage of using the original CFB mode is the bigger size of the initialization vector.

5.4 OFB Mode

The **output feedback (OFB) mode** is very similar to the CFB mode, but the input to the underlying cipher is the output of the previous encryption, rather than the ciphertext unit. It is actually a *synchronous stream cipher.* Also the OFB mode was defined in [18] in more complex structure. Similarly as in the previous section, we first simplify it, and at the end of the section we discuss the original scheme.

The key stream in the OFB mode is produced by repeated encryption of an initialization vector (IV), and is XOR-ed with the plaintext blocks, e.g. $K_0 = \mathrm{IV}$, $K_i = C(K_{i-1})$ for $i \geq 1$. The ciphertext consists of the initial vector, and all encrypted blocks, i.e. $y_0 = K_0$, and $y_i = x_i \oplus K_i$; and $x_i = y_i \oplus K_i$, (see

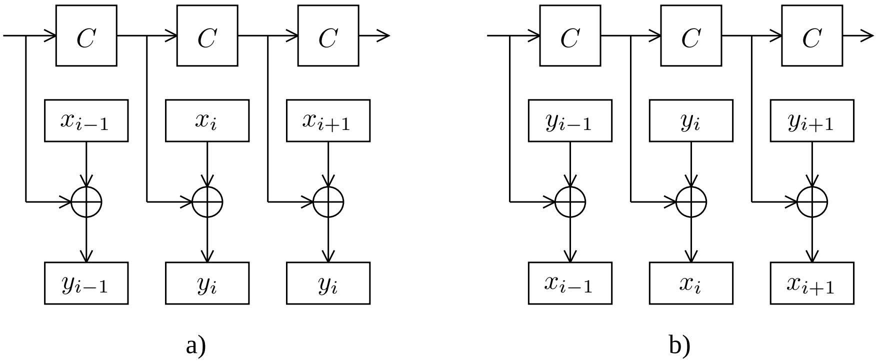


Figure 5.5: Output feedback mode: a) encryption, b) decryption

Figure 5.5). This mode is sometimes called internal feedback mode, because the feedback mechanism is independent of both the plaintext and ciphertext.

Security: Plaintexts are very easy to manipulate — any alternation of the ciphertext directly affects the same bits of the plaintext. Plaintext patterns are concealed. The initialization vector should be unique; otherwise the same key streams are generated.

Fault-tolerance: A plaintext (ciphertext) alternation affects only the corresponding block of the ciphertext (plaintext). Addition or loss of a part of the ciphertext is unrecoverable.

Efficiency: OFB processing cannot be done in parallel, unless the key sequence K_i is precomputed.

Similarly as for the CFB mode we may use any random function, not only a cipher, for the underlying primitive.

Theorem 5.4.1 *Let F^* be a perfect random function on $\mathcal{M}$,* ATK *a class of attacks, and d and q integers ($q \leq d$). Then*

$$AdvC^{\mathrm{ATK}(d|q)}(\mathrm{OFB}[F^*]) \leq \frac{d^2}{2\,|\mathcal{M}|}.$$

Proof: Assume that the attacker obtains the following plaintexts: $X_j = x_{j1}x_{j2},\dots,x_{jn_j}$, and ciphertexts: $Y_j = y_{j0}y_{j1}y_{j2},\dots,y_{jn_j}$, where $1 \leq j \leq q$, and n_j are number of blocks in the individual messages. Thus, $n_1 + n_2 + \cdots + n_q = d$.

Let all y_{jl}, for $1 \leq j \leq q$, and $1 \leq l \leq n_k$, be ordered in one sequence, and let the y_k denote the k-th element of the sequence ($0 \leq k < d$), i.e.

$$y_k = y_{ab},$$

where a is the biggest integer such that $\sum_{j=1}^{a-1} n_j \leq k+1$, and $b = k + 1 - \sum_{j=1}^{a} n_j$. Let x_k denotes the plaintext block encrypted in y_k.

Let D_k denote the following event:

$$\forall u, v \leq k, u \neq v : y_u \oplus x_u \neq y_v \oplus x_v.$$

Let $D_{-1} = 1$, and $D = D_{d-1}$.

If D occurs then the sequence of all ciphertext blocks y_{ab} is perfectly random because the elements are either initialization vectors generated at random or $F^*(K_k) \oplus x_k$ evaluated for non-colliding inputs. Consequently, if D occurs the distinguisher cannot distinguish between $\mathrm{CBC}[F^*]$ and C^*.

Let $\Pr[\neg D_k|D_{k-1}]$ be the probability, that a collision occurs in the k-th element provided that the oracle implements the CBC mode with a perfect cipher. Then

$$\begin{aligned}\Pr_1[\neg D_k|D_{k-1}] &= \Pr[\exists u < k : y_k = y_u \oplus x_u \oplus x_k|D_{k-1}]\\ &= \Pr[\exists u < k : F^*(K_{k-1}) = y_u \oplus x_u|D_{k-1}]\\ &= \Pr[\exists u < k : F^*(y_{k-1} \oplus x_{k-1}) = y_u \oplus x_u|D_{k-1}] \leq \frac{k}{|\mathcal{M}|}.\end{aligned}$$

Similarly as in the proof of Theorem 5.2.2,

$$AdvC^{\mathrm{ATK}(d|q)}(\mathrm{OFB}[F^*]) = \Pr[\neg D] \leq \frac{d^2}{2\,|\mathcal{M}|}$$

■

Corollary 5.4.2 *Let F be a random function on $\mathcal{M}$,* ATK *a class of general attacks from Chapter 3, and d and q integers ($q \leq d$). Then*

$$AdvC^{\mathrm{ATK}(d|q)}(\mathrm{OFB}[F]) \leq AdvF^{\mathrm{ATK}^+(d)}(F) + \frac{d^2}{2\,|\mathcal{M}|}.$$

Proof: Follows from Theorems 2.4.2 and 5.4.1. ■

Theorem 5.4.3 *Let F^* be a perfect random function on $\mathcal{M}$, and d and q integers ($q \leq d$). Then*

$$AdvC^{\mathrm{CPA}(d|q)}(\mathrm{OFB}[F^*]) \geq \left(1 - \frac{1}{e} - \frac{1}{|\mathcal{M}|}\right) \cdot \frac{d^2}{2\,|\mathcal{M}|}.$$

Proof: The proof is similar to the one of Theorem 5.2.4 and is omitted. ■

The feedback structure defined in [18] is very similar to the generalized structure of the CFB mode, with the difference that the shift register is filled with the key-stream block, rather than ciphertext block (see Figure 5.6). The plaintext message is again divided into blocks from a set $\mathcal{M}$, and the cipher is defined on $\mathcal{M}^L$, where L is the size of feedback. As in the previous section, we can consider the composition of the cipher and the reduction function as a random function from $\mathcal{M}^L$ to $\mathcal{M}$. The length of the initialization vector is L blocks as well.

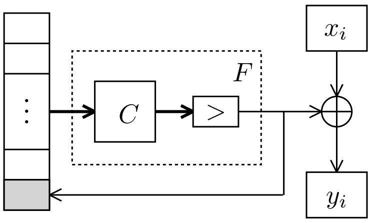


Figure 5.6: Generalized output feedback mode

Theorem 5.4.4 *Let F^* be a perfect random function from $\mathcal{M}^L$ to $\mathcal{M}$,* ATK *a class of attacks, and d and q integers ($q \leq d$). Then*

$$AdvC^{\mathrm{ATK}(d|q)}(\mathrm{CFB}[F^*]) \leq \frac{d^2}{2\,|\mathcal{M}|}.$$

Proof: The proof is similar of that of Theorem 5.3.4 and is omitted. ■

5.5 Counter Mode

An alternative mode to the OFB mode was suggested by Diffie [5]. It is called **counter mode**, and differs from the OFB mode only in the way the input for the next encryption is determined; instead of the previous ciphertext, the content of a counter is taken, i.e. $K_i = C(\mathrm{counter}(i))$. The counter is initialized by a random value, and the plaintext $X = x_1 x_2 \ldots$ is encrypted into a ciphertext $Y = y_0 y_1 y_2 \ldots$ as $y_0 = K_0$ and $y_i = x_i \oplus K_i$, and decrypted as $x_i = y_i \oplus K_i$ (see Figure 5.7).

Since the counter mode is a modified version of the OFB mode, they have the same properties (for details see the previous section).

The counter on the set $\mathcal{M}$ is a function which orders the elements of the set. In other words, it is a bijection $c : \{0, \ldots |\mathcal{M}|\} \rightarrow \mathcal{M}$. For example, if $\mathcal{M} = \{0,1\}^m$, the counter is a bijection from $\{0, \ldots 2^m\}$ to $\mathcal{M}$, and can be implemented for example as the binary string representation of the input integer.

> **Notation:** The i-th successor of an element $y \in \mathcal{M}$ with respect to the counter c, i.e. $c(c^{-1}(y) + i)$, will be denoted by $y + i$.

Theorem 5.5.1 *Let F^* be a perfect random function on $\mathcal{M}$,* ATK *a class of attacks, and d and q integers ($q \leq d$). Then*

$$AdvC^{\mathrm{ATK}(d|q)}(\mathrm{CRT}[F^*]) \leq \frac{qd}{|\mathcal{M}|}.$$

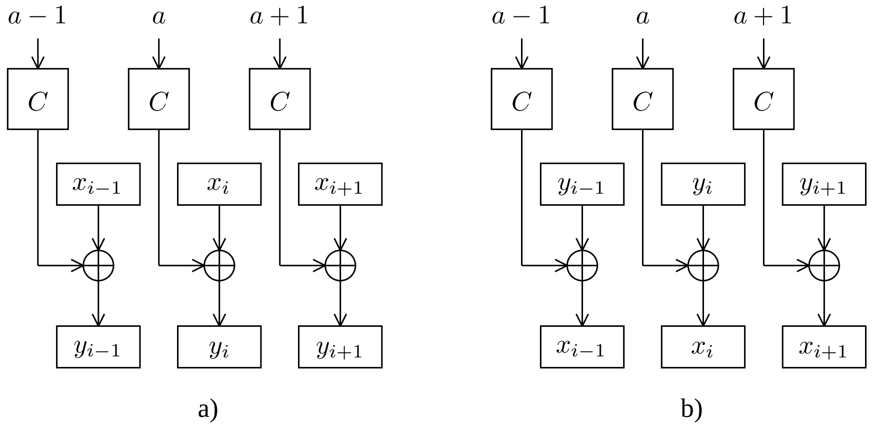


Figure 5.7: Counter mode: a) encryption, b) decryption

Proof: Let $C_1 := \mathrm{CRT}[F^*]$, and $C_2 = C^*$ be ciphers on $\mathcal{M}$. Assume that the attacker obtains the following plaintexts: $X_j = x_{j1}x_{j2}, \ldots, x_{jn_j}$, and ciphertexts: $Y_j = y_{j0}y_{j1}y_{j2}, \ldots, y_{jn_j}$, where $1 \le j \le q$, and n_j are number of blocks in the individual messages. Thus, $n_1 + n_2 + \cdots + n_q = d$

Let D_k denote the following event:

$$\forall (u,a) \neq (v,b), 1 \le u, v \le k, 1 \le a \le n_u, 1 \le b \le v : y_{u,0} + a \neq y_{v,0} + b,$$

(i.e. there is no collision in the outputs of the counter). Let $D_0 = 1$, and $D = D_q$.

If D occurs and the oracle implements $\mathrm{CRT}[F^*]$, all evaluation of F^* are for distinct inputs. Consequently, if D occurs, all $y_{ka} = F^*(y_{k0} + a) \oplus x_{ka}$ are random, and the distinguisher cannot distinguish between $\mathrm{CRT}[F^*]$ and C^*.

The plaintexts Y_k and Y_l cause a collision for some $l < k$ if and only if there is at least one pair (a, b) such that $y_{k0} + a = y_{l0} + b$. Thus, a collision with Y_l is possible if and only if $y_{k0} > y_{l0} - n_k$, and $y_{k0} < y_{l0} + n_l$, i.e. there are $n_l + n_k - 1$ possibilities for y_{k0} to get a collision with Y_l. Therefore, the probability that an element of Y_k has a collision with any of the previous blocks is:

$$\Pr[\neg D_k | D_{k-1}] \le \frac{\sum_{l=1}^{k-1}(n_k + n_l - 1)}{|\mathcal{M}|} = \frac{(k-1)(n_k - 1) + \sum_{l=1}^{k-1} n_l}{|\mathcal{M}|}$$

Now, using Theorem 2.2.3

$$\begin{aligned} AdvC^{\mathrm{ATK}(d|q)}(\mathrm{CRT}[F^*]) = \Pr[\neg D] = \Pr[\neg D_q] &\le \sum_{k=1}^{q} \Pr[\neg D_k | D_{k-1}] \\ &\le \sum_{k=1}^{q} \frac{(k-1)(n_k-1) + \sum_{l=1}^{k-1} n_l}{|\mathcal{M}|} \le \sum_{k=1}^{q} \frac{kn_k}{|\mathcal{M}|} + \sum_{k=1}^{q}\sum_{l=1}^{k-1} \frac{n_l}{|\mathcal{M}|} \\ &= \sum_{k=1}^{q} \frac{kn_k}{|\mathcal{M}|} + \sum_{l=1}^{q}\sum_{k=1}^{q-l} \frac{n_l}{|\mathcal{M}|} = \sum_{k=1}^{q} \frac{kn_k}{|\mathcal{M}|} + \sum_{k=1}^{q} \frac{(q-k)n_k}{|\mathcal{M}|} \\ &= \sum_{k=1}^{q} \frac{qn_k}{|\mathcal{M}|} = \frac{(q-1)}{|\mathcal{M}|} \sum_{k=1}^{q} n_k = \frac{qd}{|\mathcal{M}|} \end{aligned}$$

■

Corollary 5.5.2 *Let F be a random function from $\mathcal{M}_1$ to $\mathcal{M}_2$,* ATK *a class of attacks from Chapter 3, and d and q integers ($q \le d$). Then*

$$AdvC^{\mathrm{ATK}(d|q)}(\mathrm{CRT}[F]) \le AdvF^{\mathrm{ATK}^+(d)}(F) + \frac{dq}{|\mathcal{M}|}.$$

Proof: Follows from Theorems 2.4.2 and 5.5.1. ■

Theorem 5.5.3 *Let F^* be a perfect random function on $\mathcal{M}$,* ATK *a class of attacks,* ATK *a class of attacks, and d and q integers ($q \leq d$). Then*

$$AdvC^{\mathrm{ATK}(d|q)}(\mathrm{CRT}[F^*]) \geq \left(1 - \frac{1}{e}\right)\left(1 - \frac{1}{|\mathcal{M}|}\right)\frac{qd}{|\mathcal{M}|}.$$

Proof: The proof is similar to the one of Theorem 5.2.4.

DISTINGUISHER 5.2 (CRT): d-limited distinguisher for CRT

```
1. Create $q$ messages $X_k = x_{k1} \ldots x_{kn_q}$ ($1 \leq k \leq q$, $\sum_{k=1}^{q} n_k = d$), having the same value in all blocks, i.e. $\exists v \in \mathcal{M} : \forall k, a : x_{ka} = v$.
2. For $k = 1$ to $q$ do
   2.1 Get $Y_k = y_{k0}y_{k1} \ldots y_{kn_q} = C(X_k)$, where $C$ is either $\mathrm{CRT}[F^*]$ or $C^*$.
3. If $\exists u, v \leq q, a < n_u, b < n_v, (u, a) \neq (v, b) : y_{u0} + a = y_{v0} + b$
   3.1 Then if $y_{ua} = y_{vb}$ then output “accept”.
4. Output “reject”.
```

Let D_k and D have the same meaning as in the proof of Theorem 5.2.2. When the oracle implements the CRT mode, the distinguisher accepts whenever a collision occurs. If the oracle implements a perfect cipher, the distinguisher accepts if the collision occurs and the outputs for the blocks with the equal counter are equal. Since in both cases the initialization vector for the counter is chosen randomly, the probability of the collision is for both the same. Therefore,

$$\begin{aligned} p_0 &= \mathrm{Pr}[\neg D] \\ p_1 &= \mathrm{Pr}[\neg D]\frac{1}{|\mathcal{M}|} \\ |p_0 - p_1| &= \mathrm{Pr}[\neg D]\left(1 - \frac{1}{|\mathcal{M}|}\right) \end{aligned}$$

If D_{k-1} holds, then D_k can hold only if the initialization vector of Y_k does not overlap with any of the previous messages, i.e.

$$Pr[D_k|D_{k-1}] \leq \frac{|\mathcal{M}| - \sum_{l=1}^{k-1} n_l}{|\mathcal{M}|} = 1 - \frac{\sum_{l=1}^{k-1} n_l}{|\mathcal{M}|}$$

Hence,

$$\begin{aligned} \mathrm{Pr}[D] = \mathrm{Pr}[D_q] = \prod_{k=1}^{q} \mathrm{Pr}[D_k|D_{k-1}] &= \prod_{k=1}^{q}\left(1 - \frac{\sum_{l=1}^{k-1} n_l}{|\mathcal{M}|}\right) \leq \prod_{k=1}^{q} e^{-\frac{\sum_{l=1}^{k-1} n_l}{|\mathcal{M}|}} \\ &= e^{-\sum_{k=1}^{q}\frac{\sum_{l=1}^{k-1} n_l}{|\mathcal{M}|}} = e^{-\frac{\sum_{l=1}^{q}\sum_{k=1}^{q-l} n_l}{|\mathcal{M}|}} = e^{-\frac{\sum_{l=1}^{q}(q-l)n_l}{|\mathcal{M}|}} \\ &\leq e^{-\frac{\sum_{l=1}^{q} qn_l}{|\mathcal{M}|}} = e^{-\frac{qd}{|\mathcal{M}|}} \end{aligned}$$

and

$$\mathrm{Pr}[\neg D] = 1 - \mathrm{Pr}[D] \geq 1 - e^{-\frac{qd}{|\mathcal{M}|}} \geq \left(1 - \frac{1}{e}\right)\frac{qd}{|\mathcal{M}|}$$

Therefore,

$$AdvC^{\mathrm{CPA}(d|q)}(\mathrm{CBC}[C^*]) \geq \left(1 - \frac{1}{e}\right)\left(1 - \frac{1}{|\mathcal{M}|}\right)\frac{qd}{|\mathcal{M}|}$$

■

When the counter mode is modified (denote it CRT') so that the counter for encryption of the next message is set to the value the previous one finished with, there is no overlap between encrypted messages. Therefore:

Theorem 5.5.4 *Let F^* be a perfect random function on $\mathcal{M}$, ATK a class of attacks, and $d < |\mathcal{M}|$ and q integers ($q \leq d$). Then*

$$AdvC^{\mathrm{ATK}(d|q)}(\mathrm{CRT}'[F^*]) = 0.$$

Proof: The proof is similar to the one of Theorem 5.5.1, with the difference that $Pr[\neg D_k] = 0$. ■

Corollary 5.5.5 *Let F be a random function on $\mathcal{M}$, ATK a class of attacks, and d and q integers ($q \leq d$).*

$$AdvC^{\mathrm{ATK}(d|q)}(\mathrm{CRT}'[F]) \leq AdvF^{\mathrm{ATK}^+(d)}(F).$$

Proof: Follows from Theorems 2.4.2 and 5.5.3. ■

5.6 Summary

Since messages are usually longer than the block size, it was necessary to extend encryption to long messages of unspecified length. The simplest method is to split the message into units of the block length and encrypt them independently. This does not increase advantage comparing to advantage of the cipher. However, this method has other disadvantages discussed in Section 5.1. For other methods,

- $AdvC^{\mathrm{ATK}(d|q)}(\mathrm{CBC}[C]) \leq AdvC^{\mathrm{ATK}(d)}(C) + \frac{d^2}{|\mathcal{M}|}$,
- $AdvC^{\mathrm{ATK}(d|q)}(\mathrm{CFB}[F]) \leq AdvF^{\mathrm{ATK}^+(d)}(F) + \frac{d^2}{2|\mathcal{M}|}$,
- $AdvC^{\mathrm{ATK}(d|q)}(\mathrm{OFB}[F]) \leq AdvF^{\mathrm{ATK}^+(d)}(F) + \frac{d^2}{2|\mathcal{M}|}$,
- $AdvC^{\mathrm{ATK}(d|q)}(\mathrm{CRT}[F]) \leq AdvF^{\mathrm{ATK}^+(d)}(F) + \frac{dq}{|\mathcal{M}|}$,
- $AdvC^{\mathrm{ATK}(d|q)}(\mathrm{CRT}'[F]) \leq AdvF^{\mathrm{ATK}^+(d)}(F)$,

where $\mathrm{ATK} \in \{\mathrm{CPA}, \mathrm{ACPA}\}$ for CBC, or any of the attacks defined in Chapter 3 for the other modes. All the modes except of CBC do not use inversion of the underlying primitive in both encryption and decryption, thus the primitive may be any random function (not only a cipher as in the CBC mode).

When a cipher C is used in the modes other than CBC,

$$AdvF^{\mathrm{ATK}^+(d)}(C) = \frac{1}{2}\|C - F^*\| \leq \frac{1}{2}\left(\|C - C^*\| + \|C^* - F^*\|\right) \leq AdvC^{\mathrm{ATK}^+(d)}(C) + \frac{d^2}{2|\mathcal{M}|},$$

for the norm $\|\cdot\|$ associated with the particular attack. Therefore, using a cipher in the CFB or OFB mode gives the same upper-bound for advantage as for the CBC mode.

Advantage of the counter mode depends on the number of messages, and gives a lower upper-bound than the previous modes for any $q \leq \frac{d-1}{2}$. (However, for $q > \frac{d-1}{2}$ there are some plaintext messages of length 1, which cause inefficiency in encryption due to the expansion of the ciphertext — if the plaintext contains only one block, the length is doubled. Thus, it may be expected that the plaintext messages are larger than one block.) Furthermore, using the modified counter mode, initializing its counter with the value the previous encryption finished with, does not increase the resulting advantage at all.

Part II

Provable Secure Scalable Block Ciphers

Chapter 6

Scalability of Block Ciphers

Security properties of a block cipher depend on two of its parameters — the block and key size. While in theory bigger block and key sizes contribute to security against the brute force attack on the cipher, in practice increasing these parameters also means greater computational complexity of their implementation. Thus, they must be carefully selected so that the cipher ensures both the adequate level of security with respect to current as well as predicted advances in technologies, and also an acceptable performance.

In the first modern ciphers, like DES or IDEA, the key and block sizes were fixed by design so that they fitted requirements of the time they were designed. They had some lifetime expectation after which the cipher needed to be reviewed. In case of DES, the American encryption standard introduced in 1977 by the National Institute of Standards and Technology (NIST) for securing unconfidential data, it was stipulated to be reviewed every five years, the first time in 1982. Indeed, NIST renewed the DES standard three times, in 1983, 1988, and in 1993; the last time with hesitation and controversy. In 1999, DES was, after all, admitted to be no more appropriate and recommended to be used only in legacy systems. Despite approval of the DES standard, in the beginning of 90's the process of replacement of the DES with its variant — Triple-DES (which was standardized by NIST in 1999) — started. Only in September 1997 the selection process for the new encryption standard was initiated. It finished in October 2000 by selecting Rijndael as the new AES.

The history of the American encryption standard and of its updates shows how difficult a design of a generally accepted standard and the migration process from one algorithm to another (first from the DES to Triple-DES [9], nowadays to AES) is. The natural scalability of an encryption algorithm extends its life span, because demands on higher security may be solved simply by changing (increasing) parameters of the algorithm. The AES candidates were already required to support key/block size combinations of 128/128, 192/128, and 256/128 bits. Thus, all the AES candidates were required to be partially scalable, although the required scalability is not very flexible. Submitted algorithms were allowed to support other key/block size combinations. For example, the winning candidate Rijndael supports key and block sizes of 128, 192, and 256 bit in any combination, with possibility to extend any of these parameters to any multiple of 32 bits. In general, the scalability property of symmetric ciphers may be classified as follows:

1. **unscalable** — the key and block size are fixed by design to one value;
2. **partially scalable** — only a small finite set of key/block size combinations is provided;
3. **strongly scalable** — the key and block sizes are restricted but unlimited (e.g. the block size must be multiple of 32);
4. **fully scalable** — the key and block sizes are unrestricted (generally, it may not be secure to use large blocks with short keys, but for this classification the possibility of any combination is essential, not its security).

With reference to this classification, the traditional symmetric ciphers like DES or IDEA are unscalable, the AES candidates were required to be scalable, and the winning candidate Rijndael is strongly scalable. There is to date no commonly used fully scalable cipher.

6.1 Key Size Scalability

Block ciphers are usually schemes built on other key-dependent cryptographic primitives. In order to ensure higher level of security of the scheme, the primitives use different keys, so that the total length of the sub-keys may be very large. This problem is solved by introducing a main key to the scheme, from which the sub-keys are generated by a **key expansion algorithm**. This algorithm must satisfy all requirements on a

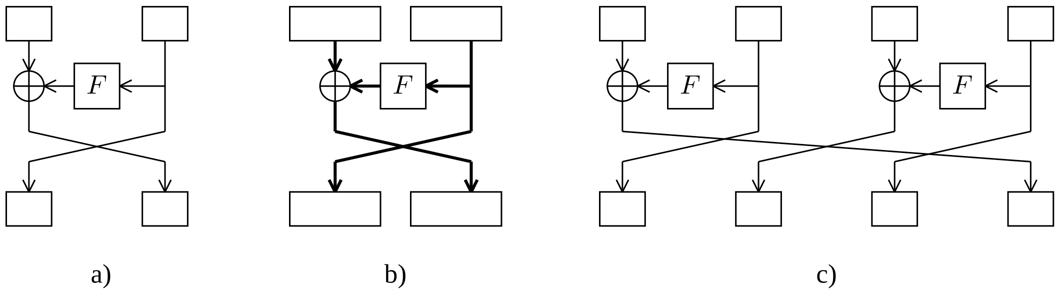


Figure 6.1: Scaling approaches: a) original scheme, b) scaling through primitives, c) scaling through structure

cryptographically strong random sequence generator. However, it may be separated from the encryption scheme and provided as an independent plug-in module. Since security of the random sequence generators is an extensive topic, we will not address it in the thesis, and assume that the sub-keys for the primitives are perfectly random. Further, since there are random sequence generators expanding a short seed of a variable length into a sequence of any length, we will consider the full scalability with respect to the key size.

6.2 Block Size Scalability

In general, there are two approaches how the block size of a cipher may be increased: either by scaling the cryptographic primitives of the cipher, or by scaling its structure. In the first case, the structure of the scheme is preserved, and the cryptographic primitives of the original cipher are substituted by new ones with larger inputs and outputs (see Figure 6.1 b). The disadvantage of this approach is that the requirement of scalability is transferred to the primitives of the scheme, and there are currently only few primitives (hash functions and random sequence generators) which may be used in this design.

When the scheme is scaled through the structure, the original scheme is modified in order to be able to deal with larger blocks, usually by dividing the input blocks into sub-blocks of the original block size, and the original cryptographic primitives are used (see Figure 6.1 c). In any case, security of the original scheme does not imply security of the enlarged one. As indicated by Theorem 2.4.1, in the first case it is necessary to prove security of the new primitives, in the other case security of the modified scheme.

A Feistel-like scheme which is scalable through primitives was investigated in [8], and led to a design of a practical scalable cipher TST. Up to now security of the TST algorithm has been assessed more from an empirical view (on the base of statistical properties) [7], and more detailed analytical investigation is still lacking. In the next chapter we examine TST using the analytical, rather than empirical approach.

The Chapter 8 examines security of the IDEA scheme and its scalability through the structure. It also shortly discusses its scalability through primitives.

Chapter 7

TST

TST is an iterative block cipher based on a scheme similar to an unbalanced Feistel network. It was designed by Valér Čanda and Tran van Trung, and analyzed in [7] using statistical methods. In this chapter we analyze it in the random oracle model.

The underlying scheme of the TST cipher introduces some changes to the unbalanced Feistel network; however, we show that we can use the security proofs provided in Chapter 3 for the unbalanced Feistel network just with minor changes. Therefore, we will focus more on the analysis of the primitives of the scheme, with the special attention to one of them: We show that a hash function used in the TST scheme is weak and even addition of another primitive to the unbalanced Feistel scheme does not provide sufficient compensation for the weakness. Then we investigate other hash functions and their use in the scheme, and show how one can select the best one with respect to security of the overall scheme. Finally, we calculate number of rounds which ensures pseudorandomness and super-pseudorandomness of the scheme.

7.1 Unbalanced Feistel Networks and TST

A Feistel Network is a general iterative method of transforming functions into block ciphers. It was invented by Horst Feistel in the late 1960s during his work at IBM Thomas J Watson Research Labs on the Lucifer cipher, and later named after him. There are many ciphers, including the former US encryption standard DES, built on this method. It is based on repeated execution of a **round transformation** consisting of the following steps:

1. the input message X is divided into two halves $X = [L, R]$;
2. the right half R is transformed by a function, called **round function** or **F-function** [19], and XOR-ed to the left part L (i.e. $L \oplus F(R)$);

The round function usually depends on a key, called **round key**. For a fixed function F and a round key, the round transformation (denoted by r_F) is a permutation, and has the very practical property of being self-invertible, i.e. for all input messages X, $r_F(r_F(X)) = X$. The **Feistel network** (also called **Feistel cipher**) Ψ consists of several round transformations followed by a block exchange operation (see Figure 7.1):

$$\Psi[F_1, F_2, \dots F_n] = r_{F_n} \circ \sigma \circ \dots \circ \sigma \circ r_{F_2} \circ \sigma \circ r_{F_1},$$

where $\sigma([L, R]) = [R, L]$. Note that the last block exchange is omitted. Computation of a Feistel network on a plaintext message $X = [L, R]$ may be described by the following sequence of round transformations:

- $[L_0, R_0] = [L, R]$ is the initial pair;
- $[L_i, R_i] = r_{F_i}([L_{i-1}, R_{i-1}]) \circ \sigma = [R_{i-1}, L_{i-1} \oplus F_i(R_{i-1})]$, for $i = 1, \dots n-1$;
- $[L_n, R_n] = r_{F_n}([L_{n-1}, R_{n-1}]) = [L_{n-1} \oplus F_n(R_{n-1}), R_{n-1}]$.

The round functions $F_1, \dots, F_n$ may be the same function with different round keys, but also possibly different functions.

Since each round of the Feistel network is self-invertible, the whole scheme may be easily inverted using the same round transformations in the reverse order. Hence, $\Psi^{-1}[F_1, F_2, \dots F_n] = r_{F_1} \circ \sigma \circ r_{F_2} \circ \sigma \circ \dots \circ r_{F_n}$.

In general, inputs of round transformations may be divided into two parts of different lengths. Such schemes are called **unbalanced Feistel networks (UFN)** [19]. There are two types of UFN, depending on the input division: If the input of the F-function is larger than the output, the UFN is called **source**

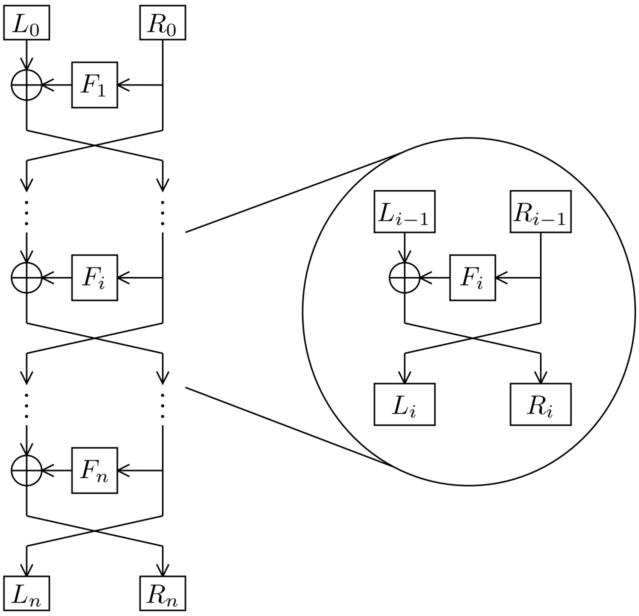


Figure 7.1: Feistel Network

heavy (see Figure 7.2 a), in the other case, it is called **target heavy** (see Figure 7.2 b) [19]. It is possible to combine both types into a **mixed UFN** (see Figure 7.2 c). We will call the round consisting of both unbalanced structures a **double-round**. Note that a mixed UFN does not necessarily have to consist of even number of transformations, it may finish with a single unbalanced round (in the middle of a double-round).

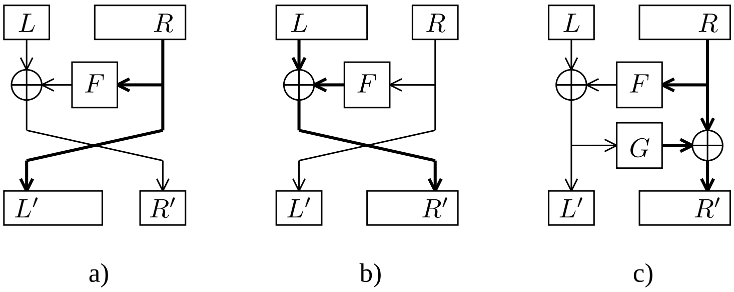


Figure 7.2: Unbalanced Feistel networks

The TST cipher as proposed in [8] is based on the structure of the unbalanced Feistel network, but brings some modifications to it (see Figure 7.3 a):

- It changes the XOR operation to any invertible operations on binary strings ($\odot$, $\oslash$, and $\otimes$);
- uses un-keyed round functions (H, S) with addition of a round key to the right part before each (double-)round;
- adds a new permutation (P) to the left part; and
- applies string rotation after each (double-)round.

In general, the function H is a hash function, and S a random sequence generator.

The scheme of TST (Figure 7.3 a) involves three primitive cryptographic functions. Assuming the block size of $(n+1)m$ bit, they are:

1. a hash function $H : \{0,1\}^{nm} \to \{0,1\}^m$;
2. a substitution box $S : \{0,1\}^m \to \{0,1\}^{nm}$; and
3. a permutation substitution box $P : \{0,1\}^m \to \{0,1\}^m$.

Both S and P are represented by a table of respectively $2^m \times nm$ and $2^m \times m$ bits generated by a random bit or number generator (see Algorithm 7.1, and Algorithm 7.2). For the hash function H, the authors of [8] suggest two basic structures depicted in Figures 7.5, and 7.8 using a simple key-independent function from $\{0,1\}^m$ to $\{0,1\}^m$ as the underlying function (see Section 7.6).

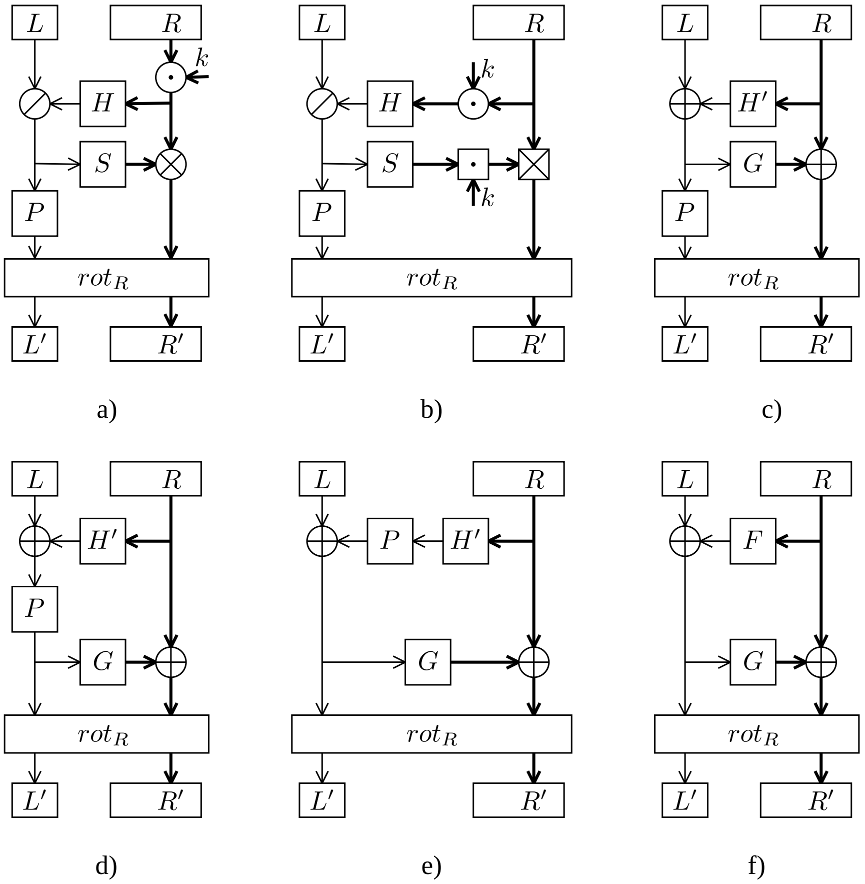


Figure 7.3: TST (a) and its simplification (b–f)

Since the function H does not depend on the key (it is not a real primitive of the scheme), we are not able to use the ROM directly. For this reason we make some changes to the scheme which, however, do not weaken its security. First, the addition of the key k ($\odot$) may be attached to both functions H, and S: $H'(X) = H(X \odot k)$, $G(X) = k \boxdot S(X)$, so that both sub-rounds in a double-round have the same key, and the $\otimes$ is substituted by another operation $\boxtimes$, where $\boxdot$ and $\boxtimes$ are such that $(a \odot b) \otimes c = a \boxtimes (c \boxdot b)$ for all entries (see Figure 7.3 b). Even if such a pair of functions does not exist, one may use $\boxdot = \odot$ and $\boxtimes = \otimes$. This scheme is not equivalent to the original scheme (in the sense it may give different outputs), but we buy flexibility in choice of the functions H' and G which actually introduce randomness into the scheme.

For compatibility with the UFN, we will first consider only the XOR function for $\oslash$ and $\boxtimes$. However, in general, $\oslash$ and $\boxtimes$ could be any randomness-preserving operation (in the sense that if one of the operands is random, the result of the operation remains random) because, as mentioned before, the functions H' and G introduce the randomness, and the operations $\oslash$ and $\boxtimes$ just transfer the randomness to the data blocks, thus we may do this substitution without loss of generality.

Using these substitutions, one gets a modified form of TST — further called **basic** TST — which actually is a mixed UFN with two additional operations — the permutation P, and the final rotation rot_R (see Figure 7.3 c). Note that the simplification we made is only in its form. In fact, functions H' and G may be any key-dependent functions with the appropriate input and output sizes, thus the scheme is actually more general than the original one, and is easier to optimize.

The basic TST scheme may be further simplified. The permutation P was introduced to the scheme in order to improve its confusion property [8]. For the same reason, it is advantageous to put it before the second sub-round (see Figure 7.3 d), bringing more randomness to the input of the function G. Further, we move the permutation P before the $\oplus$ operation (see Figure 7.3 e), and put together with H' into one stronger substitution $F = H' \oplus P$ (see Figure 7.3 f). As we show in the next section, this step weakens the security of the scheme against the chosen plaintext attack, but has an implementation advantage discussed in Section 7.7. The scheme depicted in Figure 7.3 f) will be further called **simplified** TST.

7.2 Security of the TST Scheme

In this section we examine the security of both the basic and the simplified TST scheme. The similarity of both of them with the unbalanced Feistel network will allow us to use the proof for the UFNs from Chapter 3 with slight modifications.

> **Notation:** Further we will consider the TST schemes with block size of $(n+1)m$ bits divided into an m-bit and an nm-bit parts, i.e. using the notation used in chapter 3, $\mathcal{M}_1 = \{0,1\}^m$, and $\mathcal{M}_2 = \{0,1\}^{nm}$. The basic TST scheme will be denoted by Φ, the simplified TST by Θ.

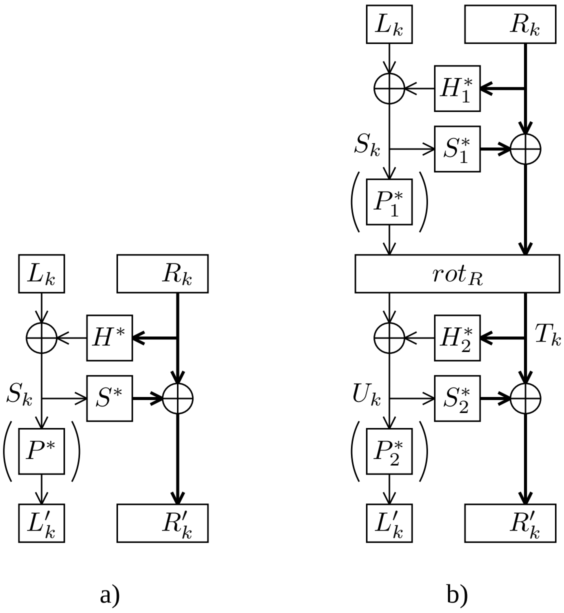


Figure 7.4: One- and two-round TST

7.2.1 Known Plaintext Attack

First, consider the one-round TST (see Figure 7.4a without the permutation P). Since the rotation function is omitted after the last round, the one-round simplified TST is actually a UFN with the m-bit left part and nm-bit right part, and thus, Theorem 3.2.5 may be applied.

Theorem 7.2.1 *Let $H^* : \mathcal{M}_2 \to \mathcal{M}_1$ and $S^* : \mathcal{M}_1 \to \mathcal{M}_2$ two independent perfect random function, and d be an integer. Then*

$$AdvC^{\mathrm{KPA}(d)}(\Theta[H^*, S^*]) \leq \frac{d^2}{|\mathcal{M}_1|}.$$

Proof: See proof of Theorem 3.2.5. ■

In the basic TST the permutation P is added. However, if the output of the one-round simplified TST is undistinguishable from a perfectly random output, the application of a perfect random function will not affect this property.

Theorem 7.2.2 *Let $H^* : \mathcal{M}_2 \to \mathcal{M}_1$, $S^* : \mathcal{M}_1 \to \mathcal{M}_2$, and $P^* : \mathcal{M}_1 \to \mathcal{M}_1$ be three independent perfect random functions, and d an integer. Then*

$$AdvC^{\mathrm{KPA}(d)}(\Phi[H^*, S^*, P^*]) \leq \frac{d^2}{|\mathcal{M}_1|}.$$

Proof: The proof of security of the 2-round UFN (see Theorem 3.2.5, page 24) depends on the fact that:

> If all R_k's are pairwise distinct, then the sequence of all S_k's (and thus also of all L'_k's) is perfectly random, because the function F_1^* [here H^*] is perfectly random. If all S_k's are pairwise distinct, then the sequence of R'_k's is also perfectly random, because the function F_2^* [here S^*] is perfectly random. If both sequences (of L'_k's and of R'_k's) are perfectly random, then the sequence of $Y_k = [L'_k, R'_k]$ is also perfectly random. In this case, output of the cipher Ψ looks like a perfect cipher.

Since the output of a perfect random function on a set of perfectly random inputs (L'_k of the UFN) is also perfectly random, security of the one-round basic TST may be proved in the same way as Theorem 3.2.5. ■

Hence, both the basic and the simplified TST can withstand the known plaintext attack.

7.2.2 Adaptive Chosen Plaintext Attack

Since the one-round simplified TST is equivalent to a two-round UFN, it is not secure against the chosen plaintext attack — Distinguisher 3.3 may be applied. However, two-round simplified TST also resists the chosen plaintext attack:

Theorem 7.2.3 *Let $H^* : \mathcal{M}_2 \to \mathcal{M}_1$, $S^* : \mathcal{M}_1 \to \mathcal{M}_2$, and $P^* : \mathcal{M}_1 \to \mathcal{M}_1$ be three independent perfect random functions, and d an integer. Then*

$$AdvC^{\mathrm{ACPA}(d)}(\Phi[H^*, S^*, P^*]) \leq \frac{d^2}{|\mathcal{M}_1|}.$$

Proof: We may assume that all queries to the oracle are pairwise different. Let $\mathcal{Y} = \{Y = (y_1, y_2, \ldots, y_d) | \forall k \neq l : L'_k \neq L'_l\}$. Consider any fixed value of $Y \in \mathcal{Y}$. The output of the cipher is Y if and only if $L'_k = P^*(S_k)$ and $R'_k = R_k \oplus S^*(S_k)$. Let E_k be the following event:

$$E_k = [P^*(S_k) = L'_k \wedge S^*(S_k) = R_k \oplus R'_k]$$

If all values S_k are pairwise different, then (since all L'_k's are pairwise different):

$$\Pr[\forall k : P^*(S_k) = L'_k] = \frac{1}{|\mathcal{M}_1|^{\underline{d}}} \geq \frac{1}{|\mathcal{M}_1|^d}$$
$$\Pr[\forall k : S^*(S_k) = R_k \oplus R'_k] = \frac{1}{|\mathcal{M}_2|^d}$$

Thus, in that case

$$\begin{aligned}[\Phi[H^*, S^*, P^*]]^d_{X,Y} &\geq \frac{1}{|\mathcal{M}_1|^d |\mathcal{M}_2|^d} = \frac{|\mathcal{M}_1 \times \mathcal{M}_2|^{\underline{d}}}{|\mathcal{M}_1 \times \mathcal{M}_2|^d} \cdot [C^*]^d_{X,Y} \\ &\geq \left(1 - \frac{d^{\underline{2}}}{2\,|\mathcal{M}_1 \times \mathcal{M}_2|}\right) [C^*]^d_{X,Y}\end{aligned}$$

Now, we can use Corollary 3.1.4 with the following parameters:

1. $\varepsilon_1 = \frac{d^{\underline{2}}}{2|\mathcal{M}_1|}$ (since $\Pr[\exists\, k \neq l : y^L_k = y^L_l] \leq \frac{d^{\underline{2}}}{2|\mathcal{M}_1|}$),
2. $\varepsilon_2 = \frac{d^{\underline{2}}}{2\,|\mathcal{M}_1 \times \mathcal{M}_2|}$, and
3. $\varepsilon_3 = \frac{d^{\underline{2}}}{2\,|\mathcal{M}_1|}$ (since $\Pr[\exists k \neq l : S_k = S_l] \leq \frac{d^{\underline{2}}}{2\,\mathcal{M}_1}$),

and we get

$$\begin{aligned}AdvC^{\mathrm{ACPA}(d)}(\Phi[H^*, S^*, P^*]) &\leq \frac{d^{\underline{2}}}{2|\mathcal{M}_1|} + \frac{d^{\underline{2}}}{2\,|\mathcal{M}_1 \times \mathcal{M}_2|} + \frac{d^{\underline{2}}}{2\,|\mathcal{M}_1|} \leq \frac{d^2 - d + 1 + d^2 - d}{2\,|\mathcal{M}_1|} \\ &\leq \frac{d^2}{|\mathcal{M}_1|}\end{aligned}$$

■

7.2.3 Adaptive Chosen Plaintext-Ciphertext attack

The one-round basic TST withstands the adaptive chosen plaintext attack, but it is distinguishable using a simple 2-limited chosen ciphertext attack.

Theorem 7.2.4 *Let $H : \mathcal{M}_2 \to \mathcal{M}_1$, $S : \mathcal{M}_1 \to \mathcal{M}_2$, and $P : \mathcal{M}_1 \to \mathcal{M}_1$ are three random functions, and d an integer. Then*

$$AdvC^{\mathrm{CCA}(2)}(\Phi[H, S, P]) \geq 1 - \frac{1}{|\mathcal{M}_2|}.$$

Proof: Consider the following distinguisher:

Distinguisher 7.1 (D_1): 2-limited CCA distinguisher for the basic TST

1. Create $Y_1 = [L', R'_1]$ and $Y_2 = [L', R'_2]$. at random.
2. Query the oracle with Y_1 and Y_2, and get $X_1 = [L_1, R_1]$ and $X_2 = [L_2, R_2]$.

   If the oracle implements the basic TST, then

   $$R_i = R'_i \oplus S(P^{-1}(L'))$$

   and thus $R_1 \oplus R_2 = R'_1 \oplus R'_2$.
3. If $R_1 \oplus R_2 = R'_1 \oplus R'_2$ then output "accept".
4. Output "reject".

When the oracle implements the basic TST, the Distinguisher D_1 always answers correctly, i.e. $p_0 = 1$. When the oracle implements a perfect random function, the probability that the distinguisher answers incorrectly is $p_1 = \frac{1}{|\mathcal{M}_2|}$. Therefore, advantage of this distinguisher is

$$AdvC^{\mathrm{CCA}(2)}(\Phi[H, S, P]) = |p_0 - p_1| = 1 - \frac{1}{|\mathcal{M}_2|}.$$

■

Adding one round, we get both the basic as well as the simplified TST schemes resistant to the adaptive chosen plaintext-ciphertext attack.

Theorem 7.2.5 *Let $H_1^*, H_2^* : \mathcal{M}_2 \to \mathcal{M}_1$ and $S_1^*, S_2^* : \mathcal{M}_1 \to \mathcal{M}_2$ be four independent perfect random functions, and d an integer. Then*

$$AdvC^{\mathrm{ACPCA}(d)}(\Theta[H_1^*, S_1^*, H_2^*, S_2^*]) \leq \frac{d^2}{|\mathcal{M}_1|}.$$

Proof: The only difference between the 2-round simplified TST (see Figure 7.4 b) and the 4-round UFN is the rotation function in the middle. Since the proof of Theorem 3.8.4 does not depend on what happens between calculation of S_k, and T_k (where the rotation is located), the proof of security of the 2-round simplified TST is the same as the proof of Theorem 3.8.4. ■

Theorem 7.2.6 *Let $H_1^*, H_2^* : \mathcal{M}_2 \to \mathcal{M}_1$, $S_1^*, S_2^* : \mathcal{M}_1 \to \mathcal{M}_2$, and $P_1^*, P_2^* : \mathcal{M}_1 \to \mathcal{M}_1$ be six independent perfect random functions, and d an integer. Then*

$$AdvC^{\mathrm{ACPCA}(d)}(\Phi[H_1^*, S_1^*, P_1^*, H_2^*, S_2^*, P_2^*]) \leq \frac{d^2}{|\mathcal{M}_1|}.$$

Proof: The 2-round basic TST is again enhanced by the permutation P — this time twice. The first time between S_k and T_k, where also the rotation is located, and thus they (the first permutation and rotation) are, for the same reason as the rotation in the proof of the previous theorem, uninteresting. The second permutation is placed just before the output, and with the same argument as in the proof of Theorem 7.2.2, the proof of Theorem 3.8.4 may be applied. ■

7.3 Conclusions about the Security of the TST Schemes

Summarizing the results of the previous sections, we have for any $d \ll 2^{\frac{m}{2}}$:

- The 1-round TST schemes are secure against the known plaintext attacks ($AdvC^{\mathrm{KPA}(d)}(\Theta[H^*, S^*]) \leq \frac{d^2}{|\mathcal{M}_1|}$, and $AdvC^{\mathrm{KPA}(d)}(\Phi[H^*, S^*, P^*]) \leq \frac{d^2}{|\mathcal{M}_1|}$).
- The 1-round simplified TST schemes are not secure against the chosen plaintext attack, but the 1-round basic TST schemes are secure against the chosen plaintext attack ($AdvC^{\mathrm{CPA}(d)}(\Phi[H^*, S^*, P^*]) \leq \frac{d^2}{|\mathcal{M}_1|}$).
- The 1-round TST schemes are not secure against the chosen plaintext-ciphertext attack;
- The 2-round TST schemes are secure against the adaptive chosen plaintext-ciphertext attack ($AdvC^{\mathrm{ACPCA}(d)}(\Theta[H_1^*, S_1^*, H_2^*, S_2^*]) \leq \frac{d^2}{|M_1|}$, and $AdvC^{\mathrm{ACPCA}(d)}(\Phi[H_1^*, S_1^*, P_1^*, H_2^*, S_2^*, P_2^*])$ $\leq \frac{d^2}{|M_1|}$).

Similarly as for the unbalanced Feistel networks, Theorem 2.4.4 implies that for security against an attacker with access to more plaintext/ciphertext pairs, the number of rounds has to be increased to at least $a\frac{l-1}{m-1-2\lg d}$, where 2^{-l} is the requested upper-bound for advantage, and $a = 1$ for the known plaintext attack against both schemes and the adaptive chosen plaintext attack against the basic TST, or $a = 2$ for the adaptive chosen ciphertext against the simplified TST and the adaptive chosen plaintext-ciphertext attack against both of them. (For more about the value l see section 3.9.)

Thus we showed that the basic TST is stronger against the adaptive chosen plaintext attack than the simplified TST. However, if we want to achieve security against the adaptive chosen plaintext-ciphertext attack, they must have both at least 2 rounds, and thus for implementation using a sufficient number of rounds, the simplified TST is more advantageous without loss of security.

To complete the analysis of the TST cipher, we have to examine its primitives. The rest of this chapter deals with this task.

7.4 S-Box S

The S-box S is a random expansion function which takes a short input of length m and returns a long output of length nm. It is generated at random from a key using a pseudorandom bit generator as follows:

ALGORITHM 7.1 Generation of the S-box S [7]

```
INPUT:    key $K$
OUTPUT:   random S-box $S$

1. For $i = 0$ to $2^m - 1$ do
    1.1 For $j = 0$ to $mn - 1$ do
        1.1.1 $S_{ij} = \text{Random}(\{0,1\}, K)$
2. Return $S$
```

The following theorem shows that the S-box is a perfect random function provided that the underlying generator is a perfect random bit generator (i.e. it generates independent perfectly random bits).

Theorem 7.4.1 *Let S^* be a function from $\{0,1\}^m$ to $\{0,1\}^{mn}$ represented by a table generated by a perfect random bit generator, which generates independent random bits, according to the Algorithm 7.1. Then for any integer d and any class of attack* ATK*,*

$$AdvF^{\mathrm{ATK}(d)}(S^*) = 0.$$

Proof: Since the bits generated by the generator are independent and random, then for any $X = (x_1, \ldots, x_d)$, and $Y = (y_1, \ldots, y_d)$:

- If $\exists i, j$ such that $x_i = x_j$, and $y_i \neq y_j$, then $\Pr[S^*(X) = Y] = 0 = \Pr[F^*(X) = Y]$

- Assume that $\forall i,j : x_i = x_j \Rightarrow y_i = y_j$, and there are c different values among x_i's (we may without loss of generality assume that they are $x_1,\ldots,x_c$). Let $s[j]$ denote the j-th bit of a string s, and S_{ij} are the bits generated according to Algorithm 7.1. Then

$$\begin{aligned}\text{Pr}[S^*(X) = Y] &= \prod_{i=1}^{c} \text{Pr}[S^*(x_i) = y_i] = \prod_{i=1}^{c} \text{Pr}\left[\bigwedge_{j=1}^{mn} S^*(x_i)[j] = y_i[j]\right] \\ &= \prod_{i=1}^{c} \text{Pr}\left[\bigwedge_{j=1}^{k} S_{ij} = y_i[j]\right] = \prod_{i=1}^{c}\prod_{j=1}^{mn} \text{Pr}\left[S_{ij} = y_i[j]\right] \\ &= \prod_{i=1}^{c}\prod_{j=1}^{mn} \frac{1}{2} = \left(\frac{1}{2^{mn}}\right)^{c} = \text{Pr}[F^*(X) = Y]\end{aligned}$$

■

7.5 S-Box P

As defined in [7], the permutation S-box P is generated from a key using a pseudorandom number generator. It is generated with the following algorithm:

Algorithm 7.2 Generation of the S-box P [7]

```
INPUT:   key $K$
OUTPUT:  random S-box $P$

1. For $i = 0$ to $2^m - 1$ do
    1.1 $P_i = i$
2. For $i = 0$ to $2^m - 1$ do
    2.1 $j = \text{Random}(\{i,\ldots,2^m-1\}, K)$
    2.2 $P_i \leftrightarrow P_j$
3. Return $P$
```

We can again prove that having a perfect random number generator, which generates independent perfectly random numbers, the permutation generated by Algorithm 7.2 is perfectly random.

Theorem 7.5.1 *Let P^* be a permutation on $\{0,1\}^m$ represented by a table generated by a perfect random generator according to the algorithm 7.2. Then for any integer d and any class of attacks* ATK*,*

$$AdvC^{\text{ATK}(d)}(P^*) = 0.$$

Proof: For each permutation p on $\{0,1\}^m$, there is a sequence of exchanges (numbers) ξ_p generated by the Algorithm 7.2 which cause its occurrence. There are $2^m!$ possible permutations generated by the algorithm, which is also the number of all possible permutations. Thus each permutation is generated by exactly one sequence of exchanges. If the random number generator is perfect, each of the exchange sequences (and thus also each of the permutation) occurs with the same probability.

Let p_ξ be a permutation induced by an exchange sequence ξ. Consider an input $X = (x_1,\ldots,x_d)$, and output $Y = (y_1,\ldots,y_d)$:

- If $\exists i,j$ such that $x_i = x_j$, and $y_i \neq y_j$, or $x_i \neq x_j$, and $y_i = y_j$ then there is no such permutation, and $\text{Pr}[P^*(X) = Y] = 0 = \text{Pr}[C^*(X) = Y]$.
- If $\forall i,j : x_i = x_j \Rightarrow y_i = y_j$, and there are c different values among x_i's (we may without loss of generality assume that they are $x_1,\ldots,x_c$), then $\text{Pr}[P^*(X) = Y] = \text{Pr}[\bigwedge_{i=1}^{c} P^*(x_i) = y_i] = \text{Pr}[\exists\xi : p_\xi(x_i) = y_i] = \frac{(2^m-d)!}{2^m!} = \frac{1}{(2^m)^{\underline{d}}} = \text{Pr}[C^*(X) = Y]$.

■

7.6 Hash Function H

In this section we examine different implementation of the hash function. In [7] two basic structures depicted in Figures 7.5, and 7.8 are proposed. For the underlying function T, the following functions are proposed:

- $f(x) = \text{rot}_c(x)$ where c is relatively prime to m.
- $f(x) = c \cdot x \bmod 2^m$ where c is an m-bit prime.
- $f(x) = x(2x+1) \bmod 2^m$
- or another simple non-linear function.

We will discuss security of the proposed schemes, and suggest some improvements.

7.6.1 Weak Structure

First, consider the hierarchical structure H_1 depicted in Figure 7.5.

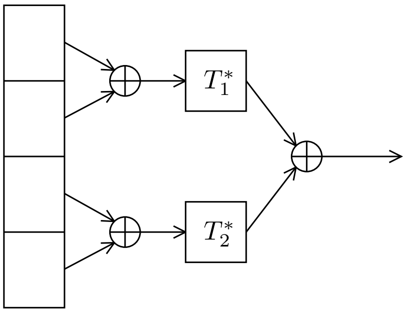


Figure 7.5: Weak Hierarchical Structure

Since the function T_i is executed only after the first XOR function, any two inputs with two blocks leading to the same execution of T_i with equal XOR cause equal outputs of the hash function. This feature is exploited in the following theorem.

Theorem 7.6.1 *Let H_1^* be a function from $\mathcal{M}_1^n$ to $\mathcal{M}_2$ defined for any $X = (x_1, \ldots, x_n) \in \mathcal{M}_1^n$ as*

$$H_1^*(x_i) = x_i,$$
$$H_1^*(x_1, \ldots, x_n) = T^* \left(H_1^*(x_1, \ldots, x_{\lceil \frac{n}{2} \rceil}) \oplus H_1^*(x_{\lceil \frac{n}{2} \rceil + 1}, \ldots, x_n) \right),$$

where T^* is a perfect random function from $\mathcal{M}_1$ to $\mathcal{M}_2$. Then

$$AdvF^{\text{CPA}(2)}(H_1^*) \geq 1 - \frac{1}{|\mathcal{M}_2|}.$$

Proof: Consider the following distinguisher between H_1^* and a perfect cipher:

DISTINGUISHER 7.2 (D_1): 2-limited CPA distinguisher for H_1

1. Create $X_1 = (x_1, x_2, x_3, \ldots, x_n)$ at random.
2. Create $X_2 = (x_1 \oplus c, x_2 \oplus c, x_3, \ldots, x_n)$ for a non-zero constant c.
3. Query the oracle with X_1 and X_2, and get Y_1 and Y_2, where Y_i is either $H_1(X_i)$ or $F^*(X_i)$.

   If $Y_i = H_1(X_i)$ then XORing the first two blocks in both of the plaintext messages eliminates the constant c (i.e. $x_1 \oplus x_2 = x_1 \oplus c \oplus x_2 \oplus c$) and all T_i's are executed with the same arguments for both X_1 and X_2. Therefore $Y_1 = Y_2$.
4. If $Y_1 = Y_2$ then output "accept".
5. Output "reject".

When the oracle implements H_1, the distinguisher D_1 always answers correctly, i.e. $p_0 = 1$. When the oracle implements a perfect random function, the probability that two random strings of length m are equal is $p_1 = \frac{1}{|\mathcal{M}_2|}$. Therefore the advantage of this 2-limited distinguisher is

$$AdvF_{D_1}^{\mathrm{CPA}(2)}(H_1^*) = |p_0 - p_1| = 1 - \frac{1}{|\mathcal{M}_2|}.$$

■

The advantage may be increased by adding further chosen plaintexts in a similar way as X_2, so that $AdvF^{\mathrm{CPA}(d)}(H_1^*) \geq 1 - \frac{1}{|\mathcal{M}_2|^{d-1}}$. Thus, the hierarchical structure H_1 is not secure against chosen plaintext attacks. In the following we show in more detail how this attack can be extended to the whole TST scheme and that the additional S-box P cannot stop the attack.

Consider the difference between two inputs of a TST round (consisting of one m-bit block for the left part and one nm-bit blocks for the right part) X_1 and X_2. If the attacker creates two messages $X_1 = (x_0, x_1, x_2, x_3, \ldots, x_n)$, and $X_2 = (x_0, x_1 \oplus c, x_2 \oplus c, x_3, \ldots, x_n)$, their differential characteristic is $(0, c, c, 0, \ldots, 0)$. The right parts (without x_0) are fed into the function H_1^*. Let h_i be the output of the function H_1^* for X_i. As shown in the previous distinguisher, $h_1 = h_2$, and thus also $x_0 \oplus h_i$, $P^*(x_0 \oplus h_i)$, as well as $S^*(x_0 \oplus h_i)$ give the same values for both inputs. Therefore, the outputs of the TST round have the same difference as the inputs, i.e. $(0, c, c, 0, \ldots 0)$ — see also Figure 7.6. If the rotation were not in the scheme, the characteristic would propagate through the whole cipher.

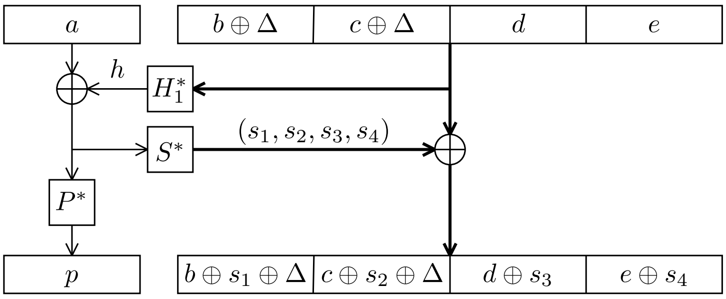


Figure 7.6: Characteristic Propagation

In the design of TST, a round key is added to the right part of the plaintext, and only then it is hashed (Figure 7.3 a). If the key is XORed with the plaintext, the changes in the second message are eliminated as described in the previous proof. The author of [7] suggests to use a modular addition operation to combine round keys with messages in order to make an attack more difficult. Consider the following distinguisher between H_1^* and a perfect cipher used on a plaintext with the added key:

DISTINGUISHER 7.3 (D_2): 2-limited CPA distinguisher for H_1

1. Create $X_1 = ((00\ldots0)_2, (00\ldots0)_2, x_3, \ldots, x_n)$ at random.
2. Create $X_2 = ((10\ldots0)_2, (10\ldots0)_2, x_3, \ldots, x_n)$.
3. Query the oracle with X_1 and X_2 and get Y_1 and Y_2, where Y_i is either $H_1(X_i)$ or $F^*(X_i)$.

   If $Y_i = H_1(X_i)$, and the added key is $k = (k_1, k_2, \ldots, k_n)$, then for the first plaintext the first XOR is $(0 + k_1) \oplus (0 + k_2)$. For the second plaintext, we have $(2^{m-1} + k_1) \oplus (2^{m-1} + k_2)$. The only bits changed in $2^{m-1} + k_i$ ($i \in \{1, 2\}$) are the most significant ones, and since one-bit addition is actually XOR function, they are for both plaintexts equal. Therefore, $k_1 \oplus k_2 = (2^{m-1} + k_1) \oplus (2^{m-1} + k_2)$, and $Y_1 = Y_2$.
4. If $Y_1 = Y_2$ then output "accept".
5. Output "reject".

When the oracle implements H_1, the distinguisher D_2 always answers correctly, i.e. $p_0 = 1$. When the oracle implements a perfect random function, the probability that the distinguisher accepts is $p_1 = \frac{1}{2^m}$. Therefore,

$$Adv^{\mathrm{CPA}(2)}(H_1) \geq Adv_{D_2}^{\mathrm{CPA}(2)}(H_1) = |p_0 - p_1| = 1 - \frac{1}{2^m}.$$

It means that the introduction of the addition operation, as suggested in [8], does not improve the scheme in the ROM.

7.6.2 Weak Underlying Primitive

Now we take a look at the second structure of the hash function suggested in [7]. It is the scheme depicted in Figure 7.8, using a weak function f (a simple key-independent function) for T. This is actually the hash function which was implemented and tested by the author of [7]. Here we can apply the following attack:

Theorem 7.6.2 *Let H_2 be a function from $\mathcal{M}_1^n$ to $\mathcal{M}_2$ defined for any $X = (x_1, \ldots, x_n) \in \mathcal{M}_1^n$ as*

$$H_2(x_1, \ldots, x_n) = f(f(\ldots f(x_1) \oplus x_2 \ldots) \oplus x_n),$$

where f is a simple key-independent function from $\mathcal{M}_1$ to $\mathcal{M}_2$ Then

$$AdvF^{\mathrm{CPA}(2)}(H_2) \geq 1 - \frac{1}{|\mathcal{M}_2|}.$$

Proof: Consider the following distinguisher between H_1^* and a perfect cipher:

DISTINGUISHER 7.4 (D_3): 2-limited CPA distinguisher for H_2

1. Create $X_1 = (x_1, x_2, x_3, \ldots, x_n)$ at random.
2. Create $X_2 = (x_1 \oplus a, x_2 \oplus c, x_3, \ldots, x_n)$ for any two constants a and c such that $f(x_1 \oplus a) = f(x_1) \oplus c$, and at least one of them is nonzero.
3. Query the oracle with X_1 and X_2 and get Y_1 and Y_2, where Y_i is either $H_2(X_i)$ or $F^*(X_i)$.

   If $Y_i = H_2(X_i)$ then the calculation for the message X_1 follows this calculation sequence:

   $$\begin{aligned} r_1 &= f(x_1) \\ r_i &= f(r_{i-1} \oplus x_i) \qquad \text{for all } i = 2, \ldots, n \end{aligned}$$

   The second calculation sequence starts with:

   $$\begin{aligned} r_1' &= f(x_1 \oplus a) = f(x_1) \oplus c = r_1 \oplus c \\ r_2' &= f(r_1 \oplus c \oplus x_2 \oplus c) = f(r_1 \oplus x_2) = r_2 \end{aligned}$$

   and thus in all further steps they both have the same values. Therefore, $Y_1 = Y_2$.
4. If $Y_1 = Y_2$ then output "accept".
5. Output "reject".

When the oracle implements H_2, the distinguisher D_3 always answers correctly, i.e. $p_0 = 1$, and therefore the advantage of this 2-limited distinguisher is again

$$AdvF^{\mathrm{CPA}(2)}(H_2) \geq AdvF_D^{\mathrm{CPA}(2)}(H_2) = |p_0 - p_1| = 1 - \frac{1}{|\mathcal{M}_2|}.$$

■

The advantage may be increased by adding further chosen plaintexts in a similar way as X_2, so that $AdvF^{\mathrm{CPA}(d)}(H_2) \geq 1 - \frac{1}{|\mathcal{M}_2|^{d-1}}$.

In order to increase the security of the hash function, the author of [7] suggested a doubled scheme consisting of two executions of the hash function H_2 — once as described above, and once with a shifted message — and finally XORing their outputs, i.e. $H = H_2(x_1, \ldots, x_n) \oplus H_2(x_n, x_1, \ldots, x_{n-1})$ (see Figure 7.7). However, when the messages are constructed as in Distinguisher 7.4, the difference is eliminated when the modified blocks are in any place in the message provided they follow each other as described. Thus, the double structure does not bring any improvement against this distinguishing attack.

Since we are able to create two distinct messages with the same output of H_2, the remark in the previous section about spreading the characteristic through the round holds also for this hash scheme.

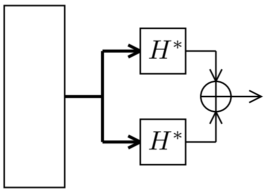


Figure 7.7: Double Hash Function

7.6.3 Strong Structure And Strong Primitive

The results described above do not imply that there is no appropriate hash scheme for TST. In what follows we show how careful analysis of various candidates and subsequent modifications allows us to find schemes with a small advantage and choose the best one.

Serial Hash Function

Consider the same structure as H_2, but with perfect random function T^* as the primitive (let's denote it by H_2^*).

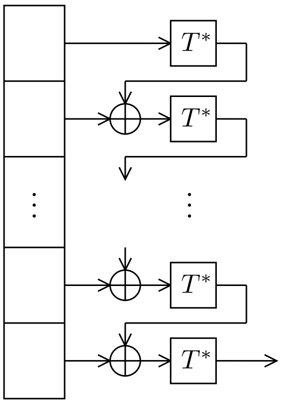


Figure 7.8: Serial Structure

Theorem 7.6.3 ([22]) *Let H_2^* be a function from $\mathcal{M}_1^n$ to $\mathcal{M}_2$ defined for any $X = (x_1, \ldots, x_n) \in \mathcal{M}_1^n$ as*

$$H_2^*(x_1, \ldots, x_n) = T^*(T^*(\ldots T^*(x_1) \oplus x_2 \ldots) \oplus x_n),$$

where T^* is a perfect random function from $\mathcal{M}_1$ to $\mathcal{M}_2$. Then for any integer d,

$$AdvF^{\mathrm{ACPA}(d)}(H_2^*) \leq \frac{(dn)^2}{2\,|\mathcal{M}_2|}.$$

Proof: Assume that the attacker disposes of d plaintext/ciphertext pairs (X_i, Y_i) such that $X_i = (x_{i1}, \ldots, x_{in})$, $Y_i = (y_{i1}, \ldots, y_{in})$, and $1 \leq i \leq d$. Without loss of generality we may assume that all X_i's are pairwise different. For each input X_i, the computation of the function H_2^* follows these steps:

1. $y_{i1} = T^*(x_{i1})$,
2. $y_{ia} = T^*(y_{i(a-1)} \oplus x_{ia})$, for all $1 < a \leq n$,

and $Y_i = y_{in}$

Inputs to the individual executions of T^* can be collected in a $d \times n$ matrix I, so that $I_{ia} = y_{ia-1} \oplus x_{ia}$, and $y_{i0} = 0$, for all i. Thus, the rows contain the computational sequences, and the columns inputs to the same instance of the function T^*.

If inputs into the last instance of T^*, i.e. I_{in} are all pairwise distinct then all outputs Y_i are perfectly random, since T^* is perfectly random, and the attacker cannot distinguish them from perfectly random outputs.

In the following we look at the probability that I_{in} are all distinct.

Let ColFree be an event that for all pairs of inputs I_{ia} and I_{jb} such that $1 \leq i, j \leq d$, and $1 \leq a, b < n$, if $I_{ia} \neq I_{jb}$ then $y_{ia} \neq y_{jb}$, i.e. that T^* returns different outputs for different inputs. The probability that there is a collision is

$$\Pr[\overline{\mathsf{ColFree}}] \leq \binom{d(n-1)}{2} \frac{1}{|\mathcal{M}_2|}$$

When ColFree occurs,

$$\begin{aligned}
\Pr[I_{in} = I_{jn}] &= \Pr[y_{i(n-1)} \oplus x_{in} = y_{j(n-1)} \oplus x_{jn}] \\
&= \begin{cases} \Pr[y_{i(n-1)} = y_{j(n-1)}] \overset{\mathsf{ColFree}}{=} \Pr[I_{i(n-1)} = I_{j(n-1)}] & \text{if } x_{in} = x_{jn} \\ \Pr[y_{i(n-1)} = y_{j(n-1)} \oplus x_{in} \oplus x_{jn}] & \text{if } x_{in} \neq x_{jn} \end{cases}
\end{aligned}$$

Let r be the smallest integer such that $x_{in-r} \neq x_{jn-r}$. Note that $r \geq 1$, because $X_i \neq X_j$. Then

$$\begin{aligned}
\Pr[I_{in} = I_{jn}] &= \Pr[I_{i(n-1)} = I_{j(n-1)}] = \ldots = \Pr[I_{i(n-r)} = I_{j(n-r)}] \\
&= \Pr[y_{i(n-r-1)} = y_{j(n-r-1)} \oplus x_{i(n-r)} \oplus x_{j(n-r)}] \\
&= \begin{cases} 0 & \text{if } I_{i(n-r-1)} = I_{j(n-r-1)} \\ \dfrac{1}{|\mathcal{M}_2|} & \text{if } I_{i(n-r-1)} \neq I_{j(n-r-1)} \end{cases} \\
&\leq \frac{1}{|\mathcal{M}_2|}
\end{aligned}$$

Note that the ColFree condition is that strong even if we require the non-collision condition only within one column — if there were a collision between two different columns, the same pair of inputs, causing this collision, could occur also in some column, which would force the collision.

From Theorem 2.2.3

$$\begin{aligned}
AdvF^{\mathrm{ACPA}(d)}(H_2^*) &= 1 - \Pr[\forall i, j : I_{in} \neq I_{jn}] = \Pr[\exists i, j : I_{in} = I_{jn}] \\
&= \Pr\left[\exists i, j : (I_{in} = I_{jn} \wedge \mathsf{ColFree}) \vee (I_{in} = I_{jn} \wedge \overline{\mathsf{ColFree}})\right] \\
&\leq \Pr\left[\mathsf{ColFree} \wedge \exists i, j : I_{in} = I_{jn}\right] + \Pr[\overline{\mathsf{ColFree}} \wedge \exists i, j : I_{in} = I_{jn}] \\
&\leq \Pr\left[\exists i, j : I_{in} = I_{jn} \wedge \mathsf{ColFree}\right] + \Pr[\overline{\mathsf{ColFree}}] \\
&\leq \binom{d}{2} \frac{1}{|\mathcal{M}_2|} + \binom{d(n-1)}{2} \frac{1}{|\mathcal{M}_2|} \\
&= \frac{d(d-1) + d(n-1)(d(n-1)-1)}{2\,|\mathcal{M}_2|} \\
&= \frac{(dn)^2 - dn - 2d^2 n - 2d + 2d^2}{2\,|\mathcal{M}_2|} \\
&\leq \frac{dn(dn-1)}{2\,|\mathcal{M}_2|}
\end{aligned}$$

■

The non-collision condition in the previous proof has to be rather strong, and causes a quite high advantage. This can be improved by using different random functions $T_1^*, \ldots, T_n^*$ for each input block (let's denote the scheme by H_3^*). In this case, a collision between two different columns cannot cause a collision inside a column, and, as the following theorem shows, the upper-bound of the advantage decreases.

Theorem 7.6.4 *Let H_3^* be a function from $\mathcal{M}_1^n$ to $\mathcal{M}_2$ defined for any $X = (x_1, \ldots, x_n) \in \mathcal{M}_1^n$ as*

$$H_3^*(x_1, \ldots, x_n) = T_n^*(T_{n-1}^*(\ldots T_1^*(x_1) \oplus x_2 \ldots) \oplus x_n),$$

where T_i^*'s are independent perfect random functions from $\mathcal{M}_1$ to $\mathcal{M}_2$. Then for any integer d,

$$AdvF^{\mathrm{ACPA}(d)}(H_3^*) \leq \frac{nd^2}{2\,|\mathcal{M}_2|}.$$

Proof: The proof is similar to the previous one with the difference that $\mathsf{ColFree}$ is defined as the event that for all pairs of inputs in the same column I_{ia} and I_{ja} such that $1 \leq i, j \leq d$, and $1 \leq a < n$, if $I_{ia} \neq I_{ja}$, then $y_{ia} \neq y_{ja}$. (Since T_i^* are independent, there is no intercolumnar dependency). Thus,

$$\Pr[\overline{\mathsf{ColFree}}] \leq (n-1)\binom{d}{2}\frac{1}{|\mathcal{M}_2|}$$

and

$$\begin{aligned} AdvF^{\mathrm{ACPA}(d)}(H_3^*) &\leq \Pr\left[\exists i, j : I_{in} = I_{jn} \wedge \mathsf{ColFree}\right] + \Pr[\overline{\mathsf{ColFree}}] \\ &\leq \binom{d}{2}\frac{1}{|\mathcal{M}_2|} + \binom{d}{2}\frac{n-1}{|\mathcal{M}_2|} \\ &\leq \frac{nd^2}{2\,|\mathcal{M}_2|} \end{aligned}$$

■

Parallel Hash Function

In section 7.6.1 we have shown that the message blocks in the hierarchical structure H_1 should **not** be XORed before the execution of functions T_i^*. The simplest way to avoid this is to apply the functions on the message blocks first, and then simply XOR the results together (see Figure 7.9).

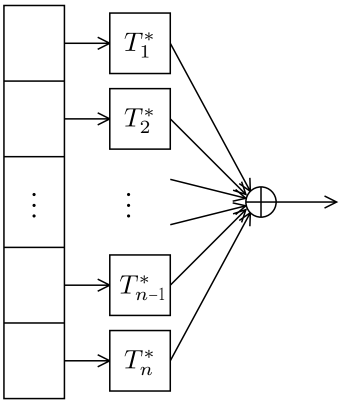


Figure 7.9: Parallel Structure (1)

The following theorem shows that this scheme is perfect as long as an attacker cannot obtain more than three plaintext/ciphertext pairs.

Theorem 7.6.5 *Let H_4^* be a function from $\mathcal{M}_1^n$ to $\mathcal{M}_2$ defined for any $X = (x_1, \ldots, x_n) \in \mathcal{M}_1^n$ as*

$$H_4^*(x_1, \ldots, x_n) = \bigoplus_{a=1}^{n} T_a^*(x_a),$$

where T_i^* are independent perfect random functions from $\mathcal{M}_1$ to $\mathcal{M}_2$. Then for all $d \leq 3$ and any class of attacks ATK,

$$AdvF^{\mathrm{ATK}(d)}(H_4^*) = 0.$$

Proof: Without loss of generality assume that all queries to the oracle are pairwise different. Let X_i consist of n blocks, i.e. $X_i = x_{i1}, \ldots x_{in}$. Let $z_{ik} = T_k^*(x_{ik})$.

If $d = 1$:

$$\begin{aligned} \Pr\left[H_4^*(x_1) = y_1\right] &= \sum_{\substack{z_{11},\ldots,z_{1n} \\ \bigoplus_{a=1}^{n} z_{1a} = y_1}} \Pr\left[\bigwedge_{a=1}^{n} T_a^*\left(x_{1a}\right) = z_{1a}\right] \\ &= \sum_{\substack{z_{11},\ldots,z_{1n} \\ \bigoplus_{a=1}^{n} z_{1a} = y_1}} \prod_{a=1}^{n} \Pr\left[T_a^*\left(x_{1a}\right) = z_{1a}\right] \\ &= |\mathcal{M}_2|^{n-1} \cdot \frac{1}{|\mathcal{M}_2|^n} = \frac{1}{|\mathcal{M}_2|} = \Pr[F^*(x_1) = y_1] \end{aligned}$$

If $d = 2, 3$:

Assume that for the a-th block, there are $k_a \in \{0, \dots, d\}$ different values among $x_{1a}, x_{2a}, \dots, x_{da}$. Since all T_a's are independent, for any $z_{1a}, z_{2a}, \dots, z_{da}$

$$\Pr[\forall i \leq d : T_a^*(x_{ia}) = z_{ia}] = \begin{cases} 0 & \exists i, j : x_{ia} = x_{ja} \wedge z_{ia} \neq z_{ja} \\ \frac{1}{|\mathcal{M}_2|^{k_a}} & \text{otherwise} \end{cases}$$

Consider first the case that $d = 3$. We can choose a random z_{ia} without having a collision (thus having the non-zero probability) in the following way: There must be i and b such that for $x_{ib} \neq x_{jb}$ both $j \in \{1, 2, 3\} \setminus \{i\}$, otherwise all three inputs were equal. Without loss of generality we may assume that $i = 3$. For all $z_{11}, \dots, z_{1(d-1)}$ we may set any value from $\mathcal{M}$. The last one is calculated from the previous ones and $\bigoplus_{a=1}^{n} z_{1a} = y_1$. All equations $x_{1a} = x_{2a}$ induce $z_{2a} := z_{1a}$. However, there must be at least one the free x_{2a}, which can be calculated from $\bigoplus_{a=1}^{n} z_{2a} = y_2$ after setting other free positions at random. For $i = 3$, first all equations with $j = 1, 2$ are set, all free positions except of x_{3b} are chosen at random, and x_{3b} is calculated from $\bigoplus_{a=1}^{n} z_{3a} = y_3$. Since in each block, there are k_a different values among $x_{1a}, x_{2a}, \dots, x_{da}$, there are altogether $\sum_{a=1}^{n}(3 - k_a) + d$ fixed values. In a similar way we get that if $d = 2$, there are $\sum_{a=1}^{n}(2 - k_a) + d$ values. Therefore,

$$\begin{aligned} \Pr\left[\bigwedge_{i=1}^{d} H_4^*(x_i) = y_i\right] &= \sum_{\substack{i=2,3: z_{i1}, \dots, z_{in} \\ \bigoplus_{a=1}^{n} z_{ia} = y_i}} \Pr\left[\bigwedge_{i=1}^{d} \bigwedge_{a=1}^{n} T_a^*\left(x_{ia}\right) = z_{ia}\right] \\ &= \sum_{\substack{i=2,3: z_{i1}, \dots, z_{in} \\ \bigoplus_{a=1}^{n} z_{ia} = y_i}} \prod_{a=1}^{n} \Pr\left[\bigwedge_{i=1}^{d} T_a^*\left(x_{ia}\right) = z_{ia}\right] \\ &= \sum_{\substack{i=2,3: z_{i1}, \dots, z_{in} \\ i: \bigoplus_{a=1}^{n} z_{ia} = y_i \\ a(d-k_a \text{equations}): z_{ia} = z_{ja}}} \prod_{a=1}^{n} \frac{1}{|\mathcal{M}_2|^{k_a}} \\ &= |\mathcal{M}_2|^{(n-1)\cdot d - \sum_{a=1}^{n}(d-k_a)} \cdot |\mathcal{M}_2|^{-\sum_{a=1}^{n} k_a} = \frac{1}{|\mathcal{M}_2|^{d}} \end{aligned}$$

Therefore, H_4^* has perfect 3-wise decorrelation, and it is not possible to distinguish it from a perfect random function seeing up to three plaintext/ciphertext pairs. ■

However, when the attacker may choose four plaintext messages, it is possible to distinguish this scheme from a perfect random one.

Theorem 7.6.6 *Let H_4^* be a function from $\mathcal{M}_1^n$ to $\mathcal{M}_2$ defined as in the previous theorem. Then*

$$AdvF^{\mathrm{CPA}(4)}(H_4^*) \geq 1 - \frac{1}{|\mathcal{M}_2|}.$$

Proof: Consider the following distinguisher between H_4^* and a perfect cipher.

DISTINGUISHER 7.5 (D_4): 4-limited CPA distinguisher for H_5

1. Create $X_1 = (s, u, x_3, \ldots, x_n)$ at random.
2. Create $X_2 = (s, v, x_3, \ldots, x_n)$ so that $u \neq v$.
3. Create $X_3 = (t, u, x_3, \ldots, x_n)$ so that $s \neq t$.
4. Create $X_4 = (t, v, x_3, \ldots, x_n)$.
5. Query the oracle with X_1, X_2, X_3 and X_4, and get Y_1, Y_2, Y_3 and Y_4, where Y_i is either $H_5(X_i)$ or $F^*(X_i)$.

   If $Y_i = H_5(X_i)$ then

   $$\begin{aligned} T_1^*(s) \oplus T_2^*(u) \oplus T_3^*(x_3) \ldots \oplus T_n^*(x_n) &= Y_1 \\ T_1^*(s) \oplus T_2^*(v) \oplus T_3^*(x_3) \ldots \oplus T_n^*(x_n) &= Y_2 \\ T_1^*(t) \oplus T_2^*(u) \oplus T_3^*(x_3) \ldots \oplus T_n^*(x_n) &= Y_3 \\ T_1^*(t) \oplus T_2^*(v) \oplus T_3^*(x_3) \ldots \oplus T_n^*(x_n) &= Y_4 \end{aligned}$$

   Therefore, $0 = Y_1 \oplus Y_2 \oplus Y_3 \oplus Y_4$.
6. If $Y_1 \oplus Y_2 \oplus Y_3 \oplus Y_4 = 0$ then output “accept”.
7. Output “reject”.

When the oracle implements H_4^*, the distinguisher D_4 always answers correctly, i.e. $p_0 = 1$. When the oracle implements a perfect random function, probability that XOR of four random values from $\mathcal{M}$ gives 0 is $p_1 = \frac{1}{|\mathcal{M}_2|}$. Therefore the advantage of this 4-limited distinguisher is

$$AdvF^{\mathrm{CPA}(4)}(H_4^*) \geq AdvF_D^{\mathrm{CPA}(4)}(H_4^*) = |p_0 - p_1| = 1 - \frac{1}{|\mathcal{M}_2|}.$$

■

The advantage may be increased by adding further pairs of chosen plaintexts in a similar way as X_3 and X_4, so that $AdvF^{\mathrm{CPA}(2d)}(H_4^*) \geq 1 - \frac{1}{|\mathcal{M}_2|^{d-1}}$.

Thus this scheme is not secure when the attacker can obtain more than three plaintext/ciphertext pairs. However, by adding another random function to the output of the scheme (see Figure 7.10), we can achieve small advantage also for bigger d, while retaining the zero advantage for $d \leq 3$.

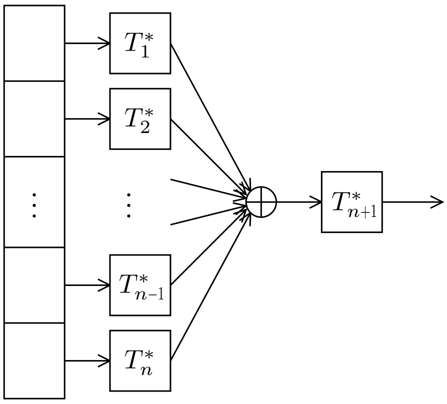


Figure 7.10: Parallel Structure (2)

Theorem 7.6.7 *Let H_5^* be a function from $\mathcal{M}_1^n$ to $\mathcal{M}_2$ defined for any $X = (x_1, \ldots, x_n) \in \mathcal{M}_1^n$ as*

$$H_5^*(x_1, \ldots, x_n) = T_{n+1}^* \left(\bigoplus_{a=1}^{n} T_a^*(x_a) \right),$$

where $T_1^*, \ldots, T_n^*$ are independent perfect random functions from $\mathcal{M}_1$ to a set $\mathcal{M}$, and T_{n+1}^* is a perfect random function from $\mathcal{M}$ to $\mathcal{M}_2$. Then for any integer d,

$$AdvF^{\mathrm{ACPA}(d)}(H_5^*) \leq \frac{d^2}{2\,|\mathcal{M}_2|}.$$

Proof: If all inputs to T_{n+1}^* are pairwise distinct, then the outputs Y_i's are perfectly random, since the function T_{n+1}^* is perfectly random. Let Z_i denote input into T_{n+1}^* for i-th message. Since $X_i \neq X_j$, there must be an r such that $x_{ir} \neq x_{jr}$. Therefore,

$$\begin{aligned}\Pr[Z_i = Z_j] &= \Pr\left[\bigoplus_{a=1}^{n} T_a^*(X_{ia}) = \bigoplus_{a=1}^{n} T_a^*(X_{ja})\right] \\ &= \Pr\left[T_r^*(x_{ir}) = T_r^*(x_{jr}) \oplus \bigoplus_{\substack{1\leq a<n \\ a\neq r}} T_a^*(x_{ia}) \oplus \bigoplus_{\substack{1\leq a<n \\ a\neq r}} T_a^*(x_{ja})\right] = \frac{1}{|\mathcal{M}_2|}\end{aligned}$$

Hence,

$$AdvF^{\mathrm{ACPA}(d)}(H_5^*) = 1 - \Pr[\forall i,j : Z_i \neq Z_j] = \Pr[\exists i,j : Z_i = Z_j] \leq \frac{d^2}{2\,|\mathcal{M}_2|}$$

■

This scheme has a small advantage for $d \ll \sqrt{2\,|\mathcal{M}|}$. Theorem 7.6.5 implies that for $d \leq 3$ the sequence of inputs to the function T_{n+1}^* is perfectly random. Since T_{n+1}^* is a perfect random function, the sequence of its outputs is in this case also perfectly random. It means that the function H_5^* is perfectly random for $d \leq 3$ as well.

This scheme has a disadvantage comparing to all previous hash schemes — it is less efficient due to the additional random function. The following scheme shows that one of the first-level functions may be omitted without increasing the upper bound of its advantage (see Figure 7.11).

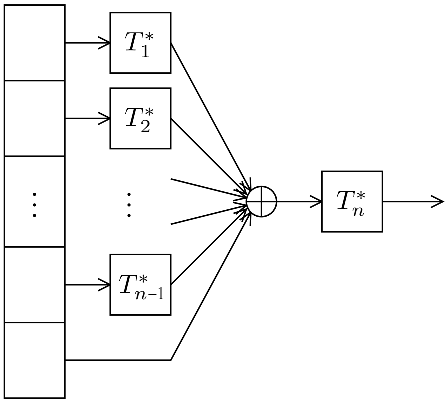


Figure 7.11: Parallel Structure (3)

Theorem 7.6.8 *Let H_6^* be a function from $\mathcal{M}_1^n$ to $\mathcal{M}_2$ defined for any $X = (x_1, \ldots, x_n) \in \mathcal{M}_1^n$ as*

$$H_6^*(x_1, \ldots, x_n) = T_n^*\left(x_n \oplus \bigoplus_{a=1}^{n-1} T_a^*(x_a)\right),$$

where $T_1^*, \ldots T_{n-1}^*$ are independent perfect random functions from $\mathcal{M}_1$ to $\mathcal{M}_1$, and T_n^* is a perfect random function from $\mathcal{M}_1$ to $\mathcal{M}_2$. Then for any integer d,

$$AdvF^{\mathrm{ACPA}(d)}(H_6^*) \leq \frac{d^2}{2\,|\mathcal{M}_2|}.$$

Proof: If all inputs to T_n^* are pairwise distinct, then the outputs are perfectly random, since the function T_n^* is perfectly random.

Assume that all inputs are pairwise distinct. Consider two cases: First, when in a pair of plaintexts X_i, X_j all blocks up to the last one are equal, i.e. when for all $1 \le a < n$, $x_{ia} = x_{ja}$. Then $x_{in} \neq x_{jn}$. Let $A := \bigoplus_{a=1}^{n-1} T_a^*(x_{ia}) = \bigoplus_{a=1}^{n-1} T_a^*(x_{ja})$.

$$\Pr[Z_i = Z_j] = \Pr\left[x_{in} \oplus A = x_{jn} \oplus A\right] = 0$$

If there is $r < n$ such that $x_{ir} \neq x_{jr}$ then

$$\begin{aligned}\Pr[Z_i = Z_j] &= \Pr\left[T_r^*(x_{ir}) = T_r^*(x_{jr}) \oplus x_{in} \oplus x_{jn} \oplus \bigoplus_{\substack{1 \le a < n \\ a \neq r}} T_a^*(x_{ia}) \oplus \bigoplus_{\substack{1 \le a < n \\ a \neq r}} T_a^*(x_{ja})\right] \\ &= \frac{1}{|\mathcal{M}_2|}\end{aligned}$$

Hence,

$$AdvF^{\mathrm{ATK}(d)}(H_6^*) = 1 - \Pr[\forall i, j : Z_i \neq Z_j] = \Pr[\exists i, j : Z_i = Z_j] \le \frac{d^2}{2\,|\mathcal{M}_2|}$$

■

Consider again the case that $d \le 3$: Skipping the last block of each message, the XOR of the outputs of $T_1^*, \ldots, T_{n-1}^*$ is perfectly random, adding the last block, the sequence still remains perfectly random, and thus applying the last function T_n^* we get again a perfectly random sequence. Therefore, the function H_6^* is perfectly random for $d \le 3$ as well.

7.7 Conclusions

Analysis of the basic and simplified TST showed that already the one-round basic TST is resistant to the chosen plaintext attack. The simplified TST needs two rounds to withstand it. However, they both need two rounds to resist the adaptive chosen plaintext-ciphertext attack. Thus, assuming that one implements the scheme in the securer way, the simplified TST is more advantageous, because it saves execution of one random permutation per round.

In the TST cipher as proposed in [8], both S-boxes S and P are perfectly random if they are generated by a perfect random bit, or number generator, respectively. If we consider a weak hash function H and a modified scheme without the rotation, the perfect pseudorandomness of the S-box P does not compensate for the weakness of the hash function. However, we showed that the rotation at the end of each round stops propagation of the characteristic of the chosen plaintexts, and thus it improves the security of the scheme in this case — if the hash function is strong, the rotation is not significant for the security of the scheme. (see Sections 7.6.1 and 7.6.2).

Both hash schemes H_1 and H_2 suggested in [7] are weak. Although we have not shown how its weakness can be exploited[1], by a sequence of carefully analyzed improvements we have found another hash scheme (H_6^*) which is provable secure — it is perfectly random when the attacker can obtain up to 3 plaintext/ciphertext pairs, and for attacks of larger sizes the probability of distinguishing it from a perfect random function is still small if the size of the attack $d \ll \sqrt{2\,|\mathcal{M}_1|}$ (for TST splitting plaintext messages into m and nm bits $d \ll 2^{\frac{m}{2}}$). Therefore, we suggest using hash function H_6^* instead of the proposed weak functions.

The primitive functions of the hash function H_6^*, $T_1^*, \ldots, T_n^*$, can be generated using a random bit generator in the same way as the S-box S^*, and stored in tables of size $2^m \times m$. Since the S-box S^* is represented by a $2^m \times nm$ table, if memory space is important, the table of S^* can be divided into n parts, so that each function T_i^* is defined by $(i-1)m$-th … $(im-1)$-th columns of the table, as depicted in the Figure 7.12.

For $d \le 3$, both the S-box S^* and hash function H_6^* are perfectly random, thus from Theorems 7.2.1–7.2.6 using the same method as in Section 3.9, we get that in order to make the basic scheme pseudorandom

[1] Even if we were able to find a distinguisher which distinguishes the output of TST from a perfect random one, it would not tell us how to construct an attack which reconstructs a key, or at least which encrypts or decrypts another message.

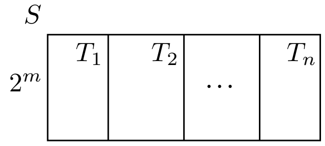


Figure 7.12: Table representation of the primitive functions of the hash functions H_6^*

one needs at least k rounds, and to make it super-pseudorandom at least $2k$ rounds with

$$k \geq \frac{(n+1)m-1}{m-1-2\lg d}.$$

For the simplified TST one needs $2k$ rounds for both pseudorandomness and pseudorandomness, with the same parameter k. For $d=2$ and for a nowadays common block length of 128 bits, it gives

$$k \geq \frac{127}{m-3}.$$

The closer the size of the left part is to one half of the block size (to the balanced scheme), the fewer rounds are necessary to obtain sufficient randomness. However, since the table of the S-box grows exponentially with the size of the left part (it is a $2^m \times nm$-bit table), it cannot be very large in order to get an efficient implementation. The following table shows the number of rounds for pseudorandomness and super-pseudorandomness for $m = 8$, 12, and 16.

m	k	pseudorandomness		super-pseudorandomness
		basic TST	simple TST	basic and simple TST
8	26	26	52	52
12	15	15	30	30
16	10	10	20	20

For comparison with another schemes discussed in this thesis, see Chapter 9. Evaluation of AES in the random oracle model was done in [17]. Using the parameters mentioned above, it requires 384 rounds for pseudorandomness and 640 rounds for super-pseudorandomness.

Note that the bound for k is calculated from Theorem 2.4.4 which does not provide tight bounds. This causes rather big difference between the calculated number of rounds and the number of rounds which is necessary to obtain pseudorandomness of the simplified TST scheme. The reason is following: A $2r$-round simplified TST is equivalent to an r-round unbalanced Feistel network. Using the parameters mentioned above we get, that the lower-bound for the number of rounds of the Feistel network is $3k$ for pseudorandomness (see Section 3.9). Thus, for the considered values of m we get that 78, 45, and 30 rounds respectively, ensure pseudorandomness of the Feistel network. It follows that for those values of m, already 39, 22.5, and 15 rounds respectively ensure pseudorandomness of the simplified TST. This is significantly less then what we obtained by the direct calculation using the theorem (52, 30, 10). This disproportion is caused by the fact, that we cannot stop the calculation in the middle of a round. From Theorem 7.2.3 we know that the threshold number of rounds for obtaining pseudorandomness is two for the simplified TST. This is equivalent to 4 rounds of an unbalanced Feistel network; however, the Feistel network needs only 3 rounds for pseudorandomness. Thus, the last half-round of the simplified TST is redundant. With the increasing number of rounds the redundant half-rounds cumulate. (Note that this disproportion does not occur in calculation of super-pseudorandomness, since the 4 rounds of an UFN are equivalent to 2 rounds of the simplified TST.)

If $d > 3$ the advantage of the TST schemes changes only slightly:

$$\begin{aligned} AdvC^{\mathrm{ACPCA}(d)}(\Phi[H_{61}^*, S_1^*, P_1^*, H_{62}^*, S_2^*, P_2^*) &= AdvC^{\mathrm{ACPCA}(d)}(\Phi[F_1^*, F_2^*, F_3^*, F_4^*, F_5^*, F_6^*]) \\ &\quad + 2\, AdvC^{\mathrm{ACPA}(d)}(H_6^*) + 2\, AdvC^{\mathrm{ACPA}(d)}(S^*) \\ &\quad + 2\, AdvC^{\mathrm{ACPA}(d)}(P^*) \\ &\leq \frac{d^2}{2^m} + 2\frac{d^2}{2 \cdot 2^m} + 2 \cdot 0 + 2 \cdot 0 = \frac{d^2}{2^{m-1}} \end{aligned}$$

for the basic TST, and

$$\begin{aligned}
AdvC^{\mathrm{ACPCA}(d)}(\Theta[H^*_{61}, S^*_1, H^*_{62}, S^*_2) &= AdvC^{\mathrm{ACPCA}(d)}(\Theta[F^*_1, F^*_2, F^*_3, F^*_4]) \\
&\quad + 2\,AdvC^{\mathrm{ACPA}(d)}(H^*_6) + 2\,AdvC^{\mathrm{ACPA}(d)}(S^*) \\
&\leq \frac{d^2}{2^m} + 2\frac{d^2}{2 \cdot 2^m} + 2 \cdot 0 = \frac{d^2}{2^{m-1}}
\end{aligned}$$

for the simplified one. Thus, in order to obtain super-pseudorandomness, we need at least $2\,\frac{(n+1)m-1}{m-2-2\lg d}$ rounds for both schemes.

Chapter 8

IDEA

IDEA (International Data Encryption Algorithm) is an iterated block cipher, first presented in 1990 as PES (Proposed Encryption Standard) [13] by Xuejia Lai and James L. Massey. It is based on a scheme similar to the Feistel network, and its round transformation works as follows [23] (see also Figure 8.1 a):

1. the input message X is divided into two halves $X = [L, R]$;
2. the difference of the two input parts $\Delta = L - R$ is calculated and provided as the input to the round function F;
3. the output of the round function is added to both parts forming the output of the round $[L', R'] = [L + F(\Delta), R + F(\Delta)]$.

The $+$ and $-$ operations are modular addition and subtraction — assuming a block size of $2m$ bits, the parts are m bits long and the operations are calculated modulo 2^m.

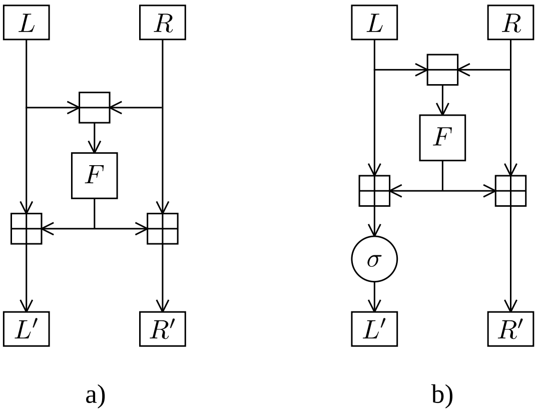


Figure 8.1: IDEA: a) original scheme, b) modified scheme

Considering the original scheme, it is easy to distinguish it from a perfect cipher using the following 1-limited KPA distinguisher:

DISTINGUISHER 8.1 (D_1): 1-limited KPA distinguisher for 1-round IDEA

1. Get $X = [L, R]$.
2. Get $Y = [L', R'] = \tilde{C}(X)$ where $\tilde{C}$ is either IDEA or a perfect cipher.
   If the oracle implements IDEA, $L' - R' = L + F(\Delta) - R - F(\Delta) = L - R$.
3. If $L - R = L' - R'$, then output "accept".
4. Output "reject".

This attack can be extended to any number of rounds, since the exploited characteristic passes through all the rounds of the cipher. To avoid this attack, we add a permutation to the left part of the round output (see Figure 8.1 b). In general, σ may be any α-almost orthomorphism for a small α [23]; however, we will further consider only the 1-bit left-rotation for σ. Since the bit rotation is a simple permutation independent from the key, it is easily invertible, and may be omitted in the last round.

Notation: The name IDEA will be further used not only for the original cipher, but also for the underlying scheme and its modifications. The scheme of IDEA enhanced by the permutation σ will be denoted by Λ^σ. The parts of a block X will be denoted by X^L for the left part and X^R for the right part, their difference $X^L - X^R$ by ΔX. For the bit-rotation σ, the individual parts are represented as m-bit integers, i.e. $X^L, X^R \in \mathcal{M} = \{0, 1, \ldots, 2^m - 1\}$, and $\Lambda^\sigma : \mathcal{M} \times \mathcal{M} \to \mathcal{M} \times \mathcal{M}$.

In the rest of this chapter, we first discuss properties of the bit rotation σ, and security of the modified IDEA as described above and depicted in Figure 8.1 b). In the next two chapters we address the scalability of the scheme of IDEA. We define two scalable schemes based on IDEA — one using the underlying scheme in the greatest possible extent, the other one simplified, using just one per round. We evaluate security of both the scalable schemes.

8.1 Properties of the Rotation Permutation

By introducing the bit rotation, the output difference of one-round changes to

$$L' - R' = \sigma(L + F(\Delta)) - R - F(\Delta) = \sigma'(L + F(\Delta)) + \Delta$$

where $\sigma'(x) = \sigma(x) - x$. Unfortunately, the function σ' is not a permutation: The 1-bit left-rotation is calculated as $\sigma(x) = 2x + \mathrm{msb}(x)$, and thus $\sigma'(x) = x + \mathrm{msb}(x)$. Let $y = \sigma'(x)$. If $\mathrm{msb}(y) = 0$, then either $x = y$, or $x = y - 1$ under the condition that $\mathrm{msb}(y-1) = 1$. The second case occurs only for $y = 0$, and thus ${\sigma'}^{-1}(0)$ has two solutions: 0, and $2^m - 1$ (00 . . . 0, and 11 . . . 1). Similarly, if $\mathrm{msb}(y) = 1$, then $x = y - 1$ under the condition that $\mathrm{msb}(y-1) = 1$. The condition is satisfied by all y with $\mathrm{msb}(y) = 1$ except of 2^{m-1} (10 . . . 0). Summarizing,

$$\left|{\sigma'}^{-1}(y)\right| = \begin{cases} 0 & \text{if } y = 2^{m-1} \\ 2 & \text{if } y = 0 \\ 1 & \text{otherwise} \end{cases}$$

In the following lemma we analyze some properties of the bit-rotation σ, that we need in the following proofs of security of the schemes based on IDEA.

Lemma 8.1.1 *Let σ be a 1-bit left-rotation on $\mathcal{M} = \{0, \ldots, 2^m - 1\}$ and $\sigma'(x) = \sigma(x) - x$ for any $x \in \mathcal{M}$. Then*

$$\forall \delta \in \mathcal{M} \setminus \{0\} : \; \mathit{Pr}[\exists a \in \mathcal{M} : \; \sigma'(a) = \sigma'(a + \delta)] \le \frac{1}{|\mathcal{M}|} \tag{8.1}$$

$$\forall \delta \in \mathcal{M} \setminus \{0\} : \; \mathit{Pr}[\exists a, b \in \mathcal{M} : \; \sigma'(a) - \sigma'(b) = \delta] \le \frac{1}{|\mathcal{M}|} \tag{8.2}$$

$$\forall \delta \in \mathcal{M} : \; \mathit{Pr}[\exists a \in \mathcal{M} : \; \delta \pm \sigma'(a) \notin \sigma'(\mathcal{M})] \le \frac{2}{|\mathcal{M}|} \tag{8.3}$$

Proof:

1. $\sigma'(a) = \sigma'(a + \delta)$ if and only if $a = 0$ and $\delta = 2^m - 1$, or $a = 2^m - 1$ and $\delta = 1$. Therefore,

$$\Pr[\exists a : \sigma'(a) = \sigma'(a + \delta)] = \begin{cases} \dfrac{1}{|\mathcal{M}|} & \delta = 1 \vee \delta = 2^m - 1 \\ 0 & \text{otherwise.} \end{cases}$$

2. $$\Pr[\exists a, b : \sigma'(a) = \sigma'(b) + \delta] = \frac{1}{|\mathcal{M}|^2} \sum_{b \in \mathcal{M}} \left|{\sigma'}^{-1}(\sigma'(b) + \delta)\right|$$

Since

$$\left|{\sigma'}^{-1}(\sigma'(b) + \delta)\right| = 2 \Leftrightarrow \sigma'(b) + \delta = 0 \Leftrightarrow \sigma'(b) = -\delta[\neq 0]$$

this may occur at most once (for $b = {\sigma'}^{-1}(-\delta)$). On the other hand,

$$\left|{\sigma'}^{-1}(\sigma'(b) + \delta)\right| = 0 \Leftrightarrow \sigma'(b) + \delta = 2^{m-1} \Leftrightarrow \sigma'(b) = 2^{m-1} - \delta[\neq 2^{m-1}]$$

i.e. it must occur at least once. Therefore,

$$\begin{aligned}\Pr[\exists a, b : \sigma'(a) = \sigma'(b) + \delta] &\leq \begin{cases} \frac{1}{|\mathcal{M}|^2}\left[1 \cdot 2 + (|\mathcal{M}| - 2) \cdot 1\right] & \text{if } \left|{\sigma'}^{-1}(-\delta)\right| > 0 \\ \frac{1}{|\mathcal{M}|^2}\left[(|\mathcal{M}| - 1) \cdot 1\right] & \text{otherwise} \end{cases} \\ &\leq \frac{1}{|\mathcal{M}|}\end{aligned}$$

3.

$$\begin{aligned}\Pr[\exists a \in \mathcal{M} : \delta \pm \sigma'(a) \notin \sigma'(\mathcal{M})] &= \frac{1}{|\mathcal{M}|}\left|\{a|\delta \pm \sigma'(a) \notin \sigma'(\mathcal{M})\}\right| \\ &= \frac{1}{|\mathcal{M}|}\left|\{a|\sigma'(a) = \pm(2^{m-1} - \delta)\}\right| \\ &= \begin{cases} 0 & \text{if } \delta = 0 \\ \frac{2}{|\mathcal{M}|} & \text{if } \delta = 2^{m-1} \\ \frac{1}{|\mathcal{M}|} & \text{otherwise} \end{cases}\end{aligned}$$

■

8.2 Security of the IDEA Scheme

In this section we study the security of the basic IDEA scheme with addition of the 1-bit left rotation. We discuss known plaintext, adaptive chosen plaintext, and adaptive chosen plaintext-ciphertext attack, and show how many rounds resist these attacks.

8.2.1 Known Plaintext Attack

Since the one-round IDEA does not contain the left-rotation, we may apply the same attack against it as described in Distinguisher 8.1.

Theorem 8.2.1 *Let F be a random function on a group $\mathcal{M}$. Then a one-round* IDEA $\Lambda^\sigma[F]$ *is not secure against the known plaintext attack.*

Proof: Consider Distinguisher 8.1. When the oracle implements IDEA, the distinguisher D_1 always answers correctly, i.e. $p_0 = 1$. When the oracle implements a perfect random function, the probability that the condition holds is $p_1 = \frac{1}{|\mathcal{M}|}$. Therefore the advantage of this distinguisher is

$$AdvC^{\mathrm{KPA}(1)}(\Lambda^\sigma[F]) \geq AdvC_{D_1}^{\mathrm{KPA}(1)}(\Lambda^\sigma[F]) = |p_0 - p_1| \geq 1 - \frac{1}{|\mathcal{M}|}.$$

■

The one-round IDEA is thus not secure against the known plaintext attack. However, adding one round makes IDEA resistent to this type of attack.

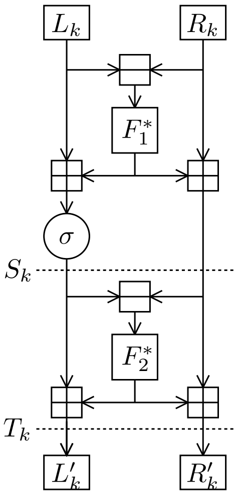
Theorem 8.2.2 *Let F_1^*, F_2^* be two independent perfect random functions on a group $\mathcal{M}$, and let d be an integer. Then*

$$AdvC^{\mathrm{KPA}(d)}(\Lambda^\sigma[F_1^*, F_2^*]) \leq \frac{d^2 + d + 1}{|\mathcal{M}|}.$$

Proof: A d-limited known-plaintext attack distinguisher has access to d plaintexts $x_1, x_2, \ldots, x_d$ and corresponding ciphertexts $y_1, y_2, \ldots, y_d$. When the oracle implements the IDEA scheme, the ciphertexts are calculated as depicted in Figure 8.2.

We may assume that all inputs in X to the oracle are pairwise different. Let $\mathcal{Y} = \{Y = (y_1, y_2, \ldots, y_d)|\forall k \neq l : \ \Delta y_k \neq \Delta y_l\}$. Consider any fixed value of $Y \in \mathcal{Y}$. Then $T_k = y_k$ if and only if $\Delta S_k = \Delta y_k$ and $T_k^L = L'_k$, i.e.

$$\begin{aligned}\Delta S_k &= \sigma'(L_k + F_1^*(\Delta x_k)) + \Delta x_k = \Delta y_k \\ T_k^L &= S_k^L + F_2^*(\Delta S_k) = S_k^L + F_2^*(\Delta y_k) = L'_k\end{aligned}$$



$$x_k = [L_k, R_k]$$
$$y_k = [L'_k, R'_k]$$

$$\Delta x_k = L_k - R_k$$
$$S_k = [\sigma(L_k + F_1^*(\Delta x_k)), R_k + F_1^*(\Delta x_k)]$$

$$\Delta S_k = S_k^L - S_k^R$$
$$T_k = y_k = [S_k^L + F_2^*(\Delta S_k), S_k^R + F_2^*(\Delta S_k)]$$

Figure 8.2: The 2-round IDEA scheme

Let E_k be the following event:

$$E_k = \left[F_1^*(\Delta x_k) \in {\sigma'}^{-1}(\Delta y_k - \Delta x_k) - L_k \wedge F_2^*(\Delta y_k) = L'_k - S_k^L \right].$$

Since all values Δy_k are pairwise different, we have

$$\Pr[\forall k : F_2^*(\Delta y_k) = L'_k - S_k^L] = \frac{1}{|\mathcal{M}|^d}.$$

Let C_1 and C_2 be the following conditions:

$$C_1 = [\forall k \neq l : \ \Delta x_k \neq \Delta x_l]$$
$$C_2 = [\forall k : \ \Delta y_k - \Delta x_k \in \sigma'(\mathcal{M})]$$

If both the conditions are satisfied,

$$\Pr[\forall k : F_1^*(\Delta x_k) \in {\sigma'}^{-1}(\Delta y_k - \Delta x_k) - L_k] = \prod_{k=1}^{d} \frac{\left|{\sigma'}^{-1}(\Delta y_k - \Delta x_k) - L_k\right|}{|\mathcal{M}|} \geq \frac{1}{|\mathcal{M}|^d}.$$

Therefore, in that case

$$[\Lambda^\sigma[F_1^*, F_2^*]]_{X,Y}^d \geq \frac{1}{|\mathcal{M}|^{2d}} = \frac{\left(|\mathcal{M}|^2\right)^d}{|\mathcal{M}|^{2d}} [C^*]_{X,Y}^d \geq \left(1 - \frac{d^2}{2\,|\mathcal{M}|^2}\right) [C^*]_{X,Y}^d.$$

Since the values of x_k are chosen randomly,
$\Pr[\neg C_1] = \Pr[\Delta x_k = \Delta x_l] = \Pr[L_k - R_k = L_l - R_l] = \frac{1}{|\mathcal{M}|}$. Further, $\Pr[\neg C_2] = \Pr[\Delta y_k - \Delta x_k \notin \sigma'(\mathcal{M})] \overset{(8.3)}{\leq} \frac{2}{|\mathcal{M}|}$. Therefore, the probability that the conditions are not satisfied is

$$\Pr\left[\exists\, k \neq l : \Delta x_k = \Delta x_l \vee \exists\, k : \Delta y_k - \Delta x_k \notin \sigma'(\mathcal{M})\right] \leq \frac{d^2}{2\,|\mathcal{M}|} + \frac{2d}{|\mathcal{M}|} = \frac{d^2 + 4d}{2\,|\mathcal{M}|}.$$

Now we can use Corollary 3.1.4 with the following parameters:

1. $\varepsilon_1 = \frac{d^2}{2|\mathcal{M}|}$ (since $\Pr[\exists\, k \neq l : \Delta y_k = \Delta y_l] \leq \frac{d^2}{2|\mathcal{M}|}$),
2. $\varepsilon_2 = \frac{d^2}{2\,|\mathcal{M}|^2}$, and
3. $\varepsilon_3 = \frac{d^2 + 4d}{2\,|\mathcal{M}|}$,

and we get

$$AdvC^{\mathrm{KPA}(d)}(\Lambda^{\sigma}[F_1^*, F_2^*]) \leq \frac{d^2}{2|\mathcal{M}|} + \frac{d^2}{2\,|\mathcal{M}|^2} + \frac{d^2 + 4d}{2\,|\mathcal{M}|} \leq \frac{d^2 - d + 1 + d^2 - d + 4d}{2\,|\mathcal{M}|}$$
$$\leq \frac{d^2 + d + 1}{|\mathcal{M}|}$$

■

8.2.2 Adaptive Chosen Plaintext Attack

Similarly as in the previous subsection, we first show that 2-round IDEA is not secure against the chosen plaintext attacks, and then we prove the resistance of the 3-round IDEA to the adaptive form of this type of attack, i.e. that it is pseudorandom.

Theorem 8.2.3 *Let F_1, and F_2 be two functions on a group $\mathcal{M}$. Then $\Lambda^{\sigma}[F_1, F_2]$ is not secure against the chosen plaintext attack.*

Proof: Consider the following distinguisher:

DISTINGUISHER 8.2 (D_2): 2-limited CPA distinguisher for the 2-round IDEA

1. Choose two plaintexts $x_1 = [L_1, R_1]$ and $x_2 = [L_2, R_2]$ so that $\Delta = L_1 - R_1 = L_2 - R_2$.
2. Query the oracle with x_1 and x_2, and get $y_1 = [L_1', R_1']$ and $y_2 = [L_2', R_2']$.

   If the oracle implements $\Lambda^{\sigma}[F_1, F_2]$, then

   $$y_k = \Lambda^{\sigma}[F_1, F_2](x_k) = [\sigma(L_k + F_1(\Delta)) + F_2(\Delta_k), R_k + F_1(\Delta) + F_2(\Delta_k)],$$

   where $\Delta_k = \sigma(L_k + F_1(\Delta)) - R_k - F_1(\Delta) = \sigma'(L_k + F_1(\Delta)) + \Delta$.

   Therefore,
   $$\Delta y_k = L_k' - R_k' = \Delta_k = \sigma'(L_k + F_1(\Delta)) + \Delta,$$
   and
   $$F_1(\Delta) = \sigma'^{-1}(\Delta y_k - \Delta) - L_k$$
   Hence,
   $$\sigma'^{-1}(\Delta y_1 - \Delta) - L_1 = \sigma'^{-1}(\Delta y_2 - \Delta) - L_2$$
3. If $\Delta y_1 - \Delta \notin \sigma(\mathcal{M})$, or $\Delta y_2 - \Delta \notin \sigma(\mathcal{M})$ then output "reject".
4. If $\sigma'^{-1}(\Delta y_1 - \Delta) - L_1 = \sigma'^{-1}(\Delta y_2 - \Delta) - L_2$ then output "accept".
5. Output "reject".

When the oracle implements $\Lambda^{\sigma}[F_1, F_2]$, then $\Delta y_k - \Delta \in \sigma(\mathcal{M})$ for both $k = 1, 2$, and the distinguisher always answers correctly, i.e. $p_0 = 1$.

When the oracle implements a perfect random function, there are two cases:

- $\Delta y_k - \Delta \notin \sigma(\mathcal{M})$ for at least one of the responses. In this case, the probability that the distinguisher answers incorrectly is 0.
- If both $\Delta y_k - \Delta$ ($k = 1, 2$) have a preimage in $\mathcal{M}$, the probability that the distinguisher answers incorrectly is

$$\Pr\left[\sigma'^{-1}(\Delta y_1 - \Delta) - L_1 = \sigma'^{-1}(\Delta y_2 - \Delta) - L_2\right] = \frac{1}{|\mathcal{M}|}.$$

Therefore, $p_1 \leq \frac{1}{|\mathcal{M}|}$, and the overall advantage of this distinguisher is

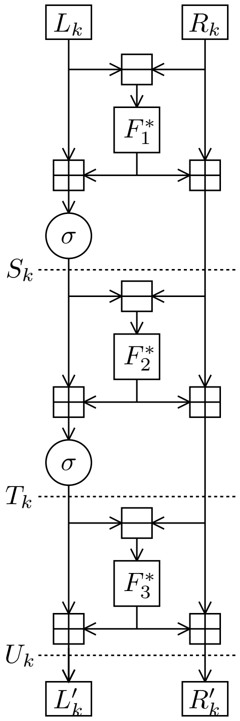
$$AdvC^{\mathrm{CPA}(2)}(\Lambda^{\sigma}[F_1, F_2]) \geq AdvC_{D_2}^{\mathrm{CPA}(2)}(\Lambda^{\sigma}[F_1, F_2]) = |p_0 - p_1| \geq 1 - \frac{1}{|\mathcal{M}|}.$$

■

Theorem 8.2.4 *Let F_1^*, F_2^*, F_3^* be three independent perfect random functions on a group $\mathcal{M}$, and d an integer. Then*

$$AdvC^{\mathrm{ACPA}(d)}(\Lambda^{\sigma}[F_1^*, F_2^*, F_3^*]) \leq \frac{d^2+d+1}{|\mathcal{M}|}.$$

Proof: The proof is similar to the one of Theorem 8.2.2. A d-limited known-plaintext attack distinguisher has access to d plaintexts $x_1, x_2, \ldots, x_d$ and corresponding ciphertexts $y_1, y_2, \ldots, y_d$. When the oracle implements the IDEA scheme, the ciphertexts are calculated as depicted in Figure 8.3.



$$\begin{aligned} x_k &= [L_k, R_k] \\ y_k &= [L'_k, R'_k] \end{aligned}$$

$$\begin{aligned} \Delta x_k &= L_k - R_k \\ S_k &= [\sigma(L_k + F_1^*(\Delta x_k)), R_k + F_1^*(\Delta x_k)] \end{aligned}$$

$$\begin{aligned} \Delta S_k &= S_k^L - S_k^R \\ T_k &= [\sigma(S_k^L + F_2^*(\Delta S_k)), S_k^R + F_2^*(\Delta S_k)] \end{aligned}$$

$$\begin{aligned} \Delta T_k &= T_k^L - T_k^R \\ U_k = y_k &= [T_k^L + F_3^*(\Delta T_k), T_k^R + F_3^*(\Delta T_k)] \end{aligned}$$

Figure 8.3: The 3-round IDEA scheme

We may assume that all inputs in X to the oracle are pairwise different. Let $\mathcal{Y} = \{Y = (y_1, y_2, \ldots, y_d) | \forall k \neq l : \; \Delta y_k \neq \Delta y_l\}$. Consider any fixed value of $Y \in \mathcal{Y}$. Then $U_k = y_k$ if and only if $\Delta T_k = \Delta y_k$ and $U_k^L = L'_k$, i.e.

$$\begin{aligned} \Delta T_k &= \sigma'(S_k^L + F_2^*(\Delta S_k)) + \Delta S_k = \Delta y_k \\ U_k^L &= T_k^L + F_3^*(\Delta T_k) = T_k^L + F_3^*(\Delta y_k) = L'_k \end{aligned}$$

Let E_k be the following event:

$$E_k = \left[F_2^*(\Delta S_k) \in \sigma'^{-1}(\Delta y_k - \Delta S_k) - S_k^L \wedge F_3^*(\Delta y_k) = L'_k - T_k^L\right].$$

Since all values Δy_k are pairwise different

$$\Pr[\forall k : F_3^*(\Delta y) = L'_k - T_k^L] = \frac{1}{|\mathcal{M}|^d}.$$

Let C_1 and C_2 be the following conditions:

$$\begin{aligned} C_1 &= [\forall k \neq l : \; \Delta S_k \neq S_l] \\ C_2 &= [\forall k : \; \Delta y_k - \Delta S_k \in \sigma'(\mathcal{M})] \end{aligned}$$

If both the conditions are satisfied,

$$\Pr[\forall k : F_2^*(\Delta S_k) \in \sigma'^{-1}(\Delta y_k - \Delta S_k) - L_k] = \prod_{k=1}^{d} \frac{\left|\sigma'^{-1}(\Delta y_k - \Delta S_k) - L_k\right|}{|\mathcal{M}|} \geq \frac{1}{|\mathcal{M}|^d}.$$

Therefore, in that case

$$[\Lambda^\sigma[F_1^*, F_2^*, F_3^*]]^d_{X,Y} \geq \frac{1}{|\mathcal{M}|^{2d}} = \frac{\left(|\mathcal{M}|^2\right)^{\underline{d}}}{|\mathcal{M}|^{2d}}[C^*]^d_{X,Y} \geq \left(1 - \frac{d^{\underline{2}}}{2\,|\mathcal{M}|^2}\right)[C^*]^d_{X,Y}$$

Now we evaluate the probability that the conditions are not satisfied. Since

$$\Delta S_k = \sigma'(L_k + F_1^*(\Delta x_k)) + \Delta x_k,$$

we get the following cases:

- If $\Delta x_k = \Delta x_l$:
  Since $x_k^L - x_k^R = \Delta x_k = \Delta x_l = x_l^L - x_l^R$ and $x_k \neq x_l$ then $x_k^L \neq x_l^L$. Thus $x_k^L + F_1^*(\Delta x_k) \neq x_l^L + F_1^*(\Delta x_l)$, and a collision occurs if $\sigma'(x_k^L + F_1^*(\Delta x_k)) = \sigma'(x_l^L + F_1^*(\Delta x_l))$. Therefore,

  $$\begin{aligned}\Pr[\Delta S_k = \Delta S_l] &= \Pr[\sigma'(x_k^L + F_1^*(\Delta x_k)) = \sigma'(x_l^L + F_1^*(\Delta x_l))]\\ &= \Pr[\exists a \in \mathcal{M} : \sigma'(a) = \sigma'(a+\delta)]\end{aligned}$$

  for a constant $\delta = x_k^L - x_l^L \in \mathcal{M} \setminus \{0\}$. From (8.1) we get that $\Pr[\Delta S_k = \Delta S_l] \leq \frac{1}{|\mathcal{M}|}$.
- If $\Delta x_k \neq \Delta x_l \wedge x_k^L + F_1^*(\Delta x_k) = x_l^L + F_1^*(\Delta x_l)$:
  Then $\sigma'(x_k^L + F_1^*(\Delta x_k)) + \Delta x_k \neq \sigma'(x_l^L + F_1^*(\Delta x_l)) + \Delta x_l$, and $\Pr[\Delta S_k = \Delta S_l] = 0$.
- If $\Delta x_k \neq \Delta x_l \wedge x_k^L + F_1^*(\Delta x_k) \neq x_l^L + F_1^*(\Delta x_l)$:
  Then $\Pr[\Delta S_k = \Delta S_l] = \Pr[\exists a, b \in \mathcal{M} : \sigma'(a) = \sigma'(b) + \delta]$, for a constant $\delta = \Delta x_l - \Delta x_k \in \mathcal{M} \setminus \{0\}$. From (8.2) we get that $\Pr[\Delta S_k = \Delta S_l] \leq \frac{1}{|\mathcal{M}|}$.

Summarizing all three cases, $\Pr[\Delta S_k = \Delta S_l] \leq \frac{1}{|\mathcal{M}|}$.
Further, $\Pr[\Delta y_k - \Delta S_k \notin \sigma'(\mathcal{M})] = \Pr[\Delta y_k - \sigma'(x_k^L + F_1^*(\Delta x_k)) - \Delta x_k \notin \sigma'(\mathcal{M})] = \Pr[\exists a \in \mathcal{M} : \delta - \sigma'(a) \notin \sigma'(\mathcal{M})] \overset{(8.3)}{\leq} \frac{2}{|\mathcal{M}|}$ for a constant $\delta = \Delta y_k - \Delta x_k$.
Therefore,

$$\Pr\left[\exists\, k \neq l : \Delta S_k = \Delta S_l \vee \exists\, k : \Delta y_k - \Delta S_k \notin \sigma'(\mathcal{M})\right] \leq \frac{d^{\underline{2}}}{2\,|\mathcal{M}|} + \frac{2d}{|\mathcal{M}|} = \frac{d^{\underline{2}} + 4d}{2\,|\mathcal{M}|},$$

and using Corollary 3.1.4 with the following parameters:

1. $\varepsilon_1 = \frac{d^{\underline{2}}}{2|\mathcal{M}|}$ (since $\Pr[\exists\, k \neq l : \Delta y_k = \Delta y_l] \leq \frac{d^{\underline{2}}}{2|\mathcal{M}|}$),
2. $\varepsilon_2 = \frac{d^{\underline{2}}}{2\,|\mathcal{M}|^2}$, and
3. $\varepsilon_3 = \frac{d^{\underline{2}}+4d}{2\,|\mathcal{M}|}$,

we get the upper-bound on the advantage:

$$AdvC^{\mathrm{ACPA}(d)}(\Lambda^\sigma[F_1^*, F_2^*, F_3^*]) \leq \frac{d^{\underline{2}}}{2|\mathcal{M}|} + \frac{d^{\underline{2}}}{2\,|\mathcal{M}|^2} + \frac{d^{\underline{2}} + 4d}{2\,|\mathcal{M}|} \leq \frac{d^2 + d + 1}{|\mathcal{M}|}$$

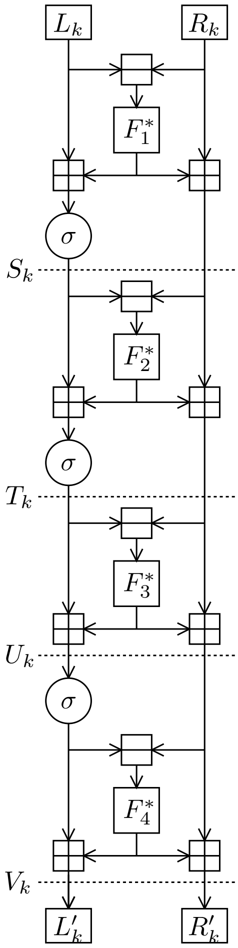
■

8.2.3 Adaptive Chosen Plaintext-Ciphertext Attack

Adaptive chosen plaintext-ciphertext attack is the strongest attack and resistance to it ensures super-pseudo-randomness of the scheme. We have not found any attack against the 3-round IDEA, however, we can prove that the 4-round IDEA resists the adaptive chosen-plaintext attack, and thus that it is super-pseudorandom. Super-pseudorandomness of 3-round IDEA is still an open problem.

Theorem 8.2.5 *Let $F_1^*, F_2^*, F_3^*, F_4^*$ be four independent perfect random functions on a group $\mathcal{M}$, and d an integer. Then*

$$AdvC^{\mathrm{ACPCA}(d)}(\Lambda^\sigma[F_1^*, F_2^*, F_3^*, F_4^*]) \leq \frac{d^2 + d + 1}{|\mathcal{M}|}.$$



$$\begin{aligned} x_k &= [L_k, R_k] \\ y_k &= [L'_k, R'_k] \end{aligned}$$

$$\begin{aligned} \Delta x_k &= L_k - R_k \\ S_k &= [\sigma(L_k + F_1^*(\Delta x_k)), R_k + F_1^*(\Delta x_k)] \end{aligned}$$

$$\begin{aligned} \Delta S_k &= S_k^L - S_k^R \\ T_k &= [\sigma(S_k^L + F_2^*(\Delta S_k)), S_k^R + F_2^*(\Delta S_k)] \end{aligned}$$

$$\begin{aligned} \Delta T_k &= \Delta U_k \\ U_k &= [T_k^L + F_3^*(\Delta T_k), T_k^R + F_3^*(\Delta T_k)] \end{aligned}$$

$$\begin{aligned} \Delta y_k &= L'_k - R'_k \\ U_k &= [\sigma^{-1}(L'_k - F_4^*(\Delta y_k)), R'_k - F_4^*(\Delta y_k)] \end{aligned}$$

Figure 8.4: The 4-round IDEA scheme

Proof: The proof is similar to the one of Theorem 8.2.2. Any d-limited known-plaintext attack distinguisher has access to d plaintexts $x_1, x_2, \ldots, x_d$ and corresponding ciphertexts $y_1, y_2, \ldots, y_d$. When the oracle implements the IDEA scheme, the ciphertexts are calculated as depicted in Figure 8.4. We may assume that all inputs in X to the oracle are pairwise different. Let $\lambda_k = (\Lambda^\sigma)^{-1}[F_4^*](y_k)$. Consider any fixed value of Y. Then $V_k = y_k$ if and only if

$$U_k = (\Lambda^\sigma)^{-1}[F_4^*](y_k) = [\sigma^{-1}(y_k^L - F_4^*(\Delta y_k)), y_k^R - F_4^*(\Delta y_k)],$$

i.e.

$$\begin{aligned} \Delta T_k &= \sigma'(S_k^L + F_2^*(\Delta S_k)) + \Delta S_k = \Delta U_k = \Delta\lambda_k \\ U_k^L &= T_k^L + F_3^*(\Delta T_k) = T_k^L + F_3^*(\Delta\lambda_k) = \lambda_k^L \end{aligned}$$

As in the proof of Theorem 8.2.4, we can define an event E_k:

$$E_k = \left[F_2^*(\Delta S_k) \in \sigma'^{-1}(\Delta\lambda_k - \Delta S_k) - S_k^L \wedge F_3^*(\Delta\lambda_k) = \lambda_k^L - T_k^L\right].$$

Let C_1, C_2, and C_3 be the following conditions:

- $C_1 = [\forall k \neq l : \Delta S_k \neq \Delta S_l]$
- $C_2 = [\forall k : \Delta\lambda_k - \Delta S_k \in \sigma'(\mathcal{M})]$
- $C_3 = [\forall k \neq l : \Delta\lambda_k \neq \Delta\lambda_l]$

Probabilities that conditions C_1 and C_2 are satisfied are evaluated in the proof of Theorem 8.2.4, and are as follows:

- $\Pr[\neg C_1] \leq \frac{d^2}{2\,|\mathcal{M}|}$

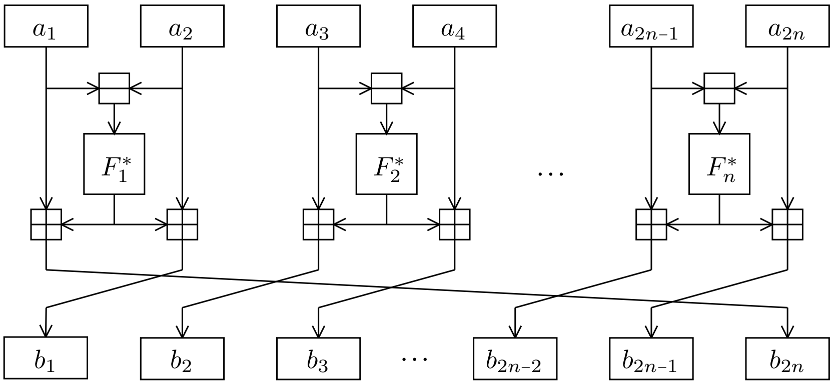


Figure 8.5: Scalable IDEA scheme

- Pr$[\neg C_2] \leq \frac{2d}{|\mathcal{M}|}$

Since $\Delta\lambda_k = \sigma^{-1}(y_k^L - F_4^*(\Delta y_k)) - y_k^L + F_4^*(\Delta y_k) + \Delta y_k =$ $\sigma^{-1}(y_k^L - F_4^*(\Delta y_k)) - \sigma(\sigma^{-1}(y_k^L - F_4^*(\Delta y_k)) + \Delta y_k = -\sigma'(\sigma^{-1}(y_k^L - F_4^*(\Delta y_k)) + \Delta y_k$, and since σ is a permutation, the probability Pr$[\Delta\lambda_k = \Delta\lambda_l]$ for a particular k and l can be analyzed in the same way as Pr$[\Delta S_k \neq \Delta S_l]$ with the same result

$$\text{Pr}[\neg C_3] \leq \frac{d^2}{2\,|\mathcal{M}|},$$

The parameters for Corollary 3.1.6 are now:

1. $\varepsilon_1 = \frac{d^2}{2\,|\mathcal{M}|^2}$, and
2. $\varepsilon_2 = \frac{2d^2+4d}{2\,|\mathcal{M}|} = \frac{d^2+2d}{|\mathcal{M}|}$.

Therefore,

$$\begin{aligned} AdvC^{\text{ACPCA}(d)}(\Lambda^{\sigma}[F_1^*, F_2^*, F_3^*, F_4^*]) &\leq \frac{d^2}{2\,|\mathcal{M}|^2} + \frac{d^2+2d}{|\mathcal{M}|} \leq \frac{1+d^2-d+2d}{|\mathcal{M}|} \\ &\leq \frac{d^2+d+1}{|\mathcal{M}|} \end{aligned}$$

■

8.3 Scalable Scheme Based on IDEA

The straightforward way to scale the original IDEA scheme is to divide the large input into sub-blocks of the length of the original IDEA block size, and apply the original scheme to each sub-block. In order to achieve dependency of the output sub-blocks on all the input sub-blocks, we need to employ the block rotation as depicted in Figure 8.5. Note that the dependency on all the input blocks is necessary. Otherwise we could apply the attack described in the proof of the following theorem.

Theorem 8.3.1 *Let C be a scalable encryption scheme, which divides the plaintext into n sub-blocks $x = (a_1, a_2, \ldots, a_n)$, and which calculates the ciphertext $y = (b_1, b_2, \ldots, b_n)$ so that $b_i = a_{j_i} \boxplus F_i(a_1, a_2, \ldots, a_n)$ (without any restriction on the functions F_i), where $j_1, j_2, \ldots, j_n$ is a permutation on $\{1, 2, \ldots, n\}$. If there are i and k such that the output of $F_i(a_1, a_2, \ldots, a_n)$ does not depend on a_k, then the scheme C is not secure against the chosen plaintext attack.*

Proof: Consider the following distinguisher:

DISTINGUISHER 8.3 (D_3): 2-limited CPA distinguisher for C

1. Choose two plaintexts $x_1 = (a_{11}, a_{12}, \dots, a_{1n})$, and $x_2 = (a_{21}, a_{22}, \dots, a_{2n})$ so that $a_{1k} \neq a_{2k}$, and for all $l \neq k$, $a_{1l} = a_{2l}$.
2. Query the oracle with x_1 and x_2, and get $y_1 = (b_{11}, b_{12}, \dots, b_{1n})$ and $y_2 = (b_{21}, b_{22}, \dots, b_{2n})$.

   If the oracle implements C then the function F_i does not depend on a_k, and thus $b_{1i} \boxminus b_{2i} = a_{1j_i} \boxplus F_i(a_{11}, a_{12}, \dots, a_{1n}) \boxminus a_{2j_i} \boxminus F_i(a_{21}, a_{22}, \dots, a_{2n}) = a_{1j_i} \boxminus a_{2j_i}$.
3. If $b_{1i} \boxminus b_{2i} = a_{1j_i} \boxminus a_{2j_i}$, output "accept".
4. Output "reject".

If the oracle implements C, the probability that the distinguisher D_3 accepts is $p_0 = 1$; if the oracle implements a perfect random permutation, $p_1 = \frac{1}{|\mathcal{M}|}$, where $\mathcal{M}$ is the set of possible values for b_i. Therefore,

$$AdvC^{\mathrm{CPA}(2)}(C) \geq AdvC^{\mathrm{CPA}(2)}_{D_3}(C) \geq 1 - \frac{1}{|\mathcal{M}|}.$$

■

The rotation of the sub-blocks ensures a good diffusion property of the cipher. However, it does not prevent an attack, similar to that one against the original scheme, exploiting the input-output differential characteristic.

Notation: The input block of scalable IDEA is divided into n sub-blocks, each consisting of two parts, i.e. altogether into $2n$ parts. The inputs to the round functions (the differences) are calculated for each sub-block separately. The difference between two subsequent parts a_i and a_{i+1} (i.e. $a_i - a_{i+1}$) of a block $x = (a_1, a_2, \dots, a_{2n})$ will be further denoted Δa_i.

Consider now the following 1-limited known plaintext attack distinguisher:

DISTINGUISHER 8.4 (D_4): 1-limited KPA distinguisher for IDEA

1. Get a plaintext $x = (a_1, a_2, \dots, a_{2n})$.
2. Get a ciphertext $y = (b_1, b_2, \dots, b_{2n})$.

   If the oracle implements a scalable IDEA with the round functions $F_1, F_2, \dots, F_n$, then for all $1 \leq i \leq n$:

$$\begin{aligned} b_{2i-1} &= a_{2i} + F_i(\Delta a_{2i-1}) \\ b_{2i} &= a_{2i+1} + F_{i+1}(\Delta a_{2i+1}). \end{aligned}$$

   Therefore,

$$\begin{aligned} \sum_{i=1}^{n} \Delta b_{2i-1} &= \sum_{i=1}^{n} a_{2i} + F_i(\Delta a_{2i-1}) - \sum_{i=1}^{n} a_{2i+1} + F_{i+1}(\Delta a_{2i+1}) \\ &= -\sum_{i=1}^{n} \Delta a_{2i-1}. \end{aligned}$$

3. If $\sum_{i=1}^{n} \Delta b_{2i-1} = -\sum_{i=1}^{n} \Delta a_{2i-1}$, then output "accept".
4. Output "reject".

When the oracle implements IDEA, the distinguisher D_4 always answers correctly, i.e. $p_0 = 1$. When the oracle implements a perfect random function, the probability that the condition in Step 3 holds is $p_1 = \frac{1}{|\mathcal{M}|}$. Therefore, the advantage of this distinguisher is $|p_0 - p_1| = 1 - \frac{1}{|\mathcal{M}|}$. The distinguisher may be extended to more rounds so that for an r-round IDEA it answer "accept" when $\sum_{i=1}^{n} \Delta b_{2i-1} = (-1)^r \sum_{i=1}^{n} \Delta a_{2i-1}$. The extended distinguisher has the same advantage as D_4.

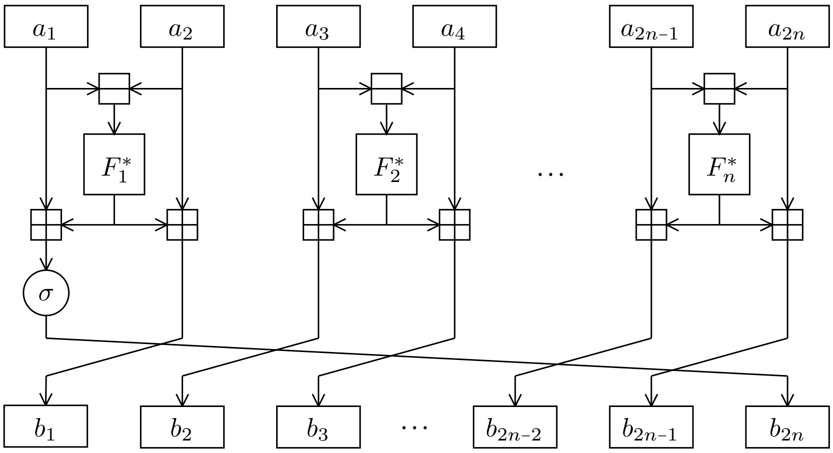


Figure 8.6: Modified scalable IDEA scheme

To avoid this attack we again need to add the permutation σ to the scheme. However, it is not necessary to use it in each sub-block, we may apply it just in one of them as depicted in Figure 8.6. After the last round, the permutation may be again omitted.

> **Notation:** The scalable scheme of IDEA enhanced by the permutation σ will be further denoted by Π_n^σ, i.e. $\Pi_n^\sigma : \mathcal{M}^{2n} \to \mathcal{M}^{2n}$ for a set $\mathcal{M} = \{0, 1, \ldots, 2^m - 1\}$.

In the following example we observe how the dependency of the output parts on the input parts spreads.

Example 8.3.2 *Consider a scalable scheme of* IDEA *with three sub-blocks (six parts). Let the input be* $(a_1, a_2, a_3, a_4, a_5, a_6)$ *and denote the* i*-th part of the output block after the first* r*-th rounds by* $a_{r_i} + F_{ri}(a_1, a_2, a_3, a_4, a_5, a_6)$*. The following table shows which part of the plaintext is on which position after the* r*-th round, and on which input parts the function* F_{ri} *depends.*

after	*1*		*2*		*3*		*4*		*5*		*6*	
the round r	i_1	F_{r1}	i_2	F_{r2}	i_3	F_{r3}	i_4	F_{r4}	i_5	F_{r5}	i_6	F_{r6}
1	*2*	*1, 2*	*3*	*3, 4*	*4*	*3, 4*	*5*	*5, 6*	*6*	*5, 6*	*1*	*1, 2*
2	*3*	*1–4*	*4*	*3–6*	*5*	*3–6*	*6*	*1, 2, 5, 6*	*1*	*1, 2, 5, 6*	*2*	*1–4*
3	*4*	*1–6*	*5*	*1–6*	*6*	*1–6*	*1*	*1–6*	*2*	*1–6*	*3*	*1–6*

Hence, the table indicates that we need at least 3 rounds to get all output parts dependent on all input parts.

This example may be generalized into the following theorem.

Theorem 8.3.3 *Let the scalable* IDEA *scheme divide the input into* $2n$ *parts. Then it requires at least* n *rounds to get all output parts dependent on all input parts.*

Proof: After the first round, each output part depends on two input parts. Then the parts are shifted to the left, and the input of each round function of the next round depends on all input parts contained in a particular sub-block. Considering the sub-block forming the input to the i-th round function $(1 \leq i \leq n)$:

1. The left part is the right part of the i-th output sub-block of the previous round. It depends on the same input parts as the input to the i-th round function of the previous round.
2. The right part is the left part of the $(i+1)$-th output sub-block of the previous round. It depends on the same input parts as the input to the $(i+1)$-th round function of the previous round. It contributes two input parts not contained in the left part.

Thus, after each round, each output part depends on two more input parts, and after the n-th round all $2n$ output parts depend on all $2n$ input parts. ■

8.3.1 Adaptive Chosen Plaintext Attack

Consider a chosen plaintext attack against a scalable scheme of IDEA where an attacker may obtain up to d plaintext/ciphertext pairs (x_k, y_k) with pairwise different plaintexts $x_k = (a_{k1}, \ldots, a_{k(2n)})$. Consider two fixed plaintexts x_k and x_l, for $k \neq l$. The attacker may choose the plaintexts so that all differences between subsequent parts are in both plaintexts equal, i.e. for all $1 \leq i \leq 2n-1$:

$$\Delta a_{ki} = \Delta a_{li}. \tag{8.4}$$

If $a_{k1} = a_{l1}$ then from (8.4) also all other $a_{ki} = a_{li}$, and thus $x_k = x_l$. Thus, assuming pairwise different plaintexts, this case does not occur. Let $a_{l1} = a_{k1} + \delta$, for a nonzero δ. Then from (8.4) we get that for all i, $a_{li} = a_{ki} + \delta$.

Let $y_k = (b_{k1}, \ldots, b_{k(2n)})$ be the output of the first round processing the plaintext x_k as depicted in Figure 8.6. Then for all $1 \leq i < n$

$$\begin{aligned}\Delta b_{l(2i-1)} &= a_{l(2i)} + F_i^*(\Delta a_{l(2i-1)}) - a_{l(2i+1)} - F_{i+1}^*(\Delta a_{l(2i+1)}) \\ &= a_{k(2i)} + \delta + F_i^*(\Delta a_{k(2i-1)}) - a_{k(2i+1)} - \delta - F_{i+1}^*(\Delta a_{k(2i+1)}) = \Delta b_{k(2i-1)}\end{aligned}$$

and $\Delta b_{k(2n-1)} = \Delta b_{l(2n-1)}$ if and only if

$$\begin{aligned}a_{k(2n)} + F_n^*(\Delta a_{k(2n-1)}) - \sigma\left(a_{k1} + F_1^*(\Delta a_{k1})\right) &= a_{l(2n)} + F_n^*(\Delta a_{l(2n-1)}) - \sigma\left(a_{l1} + F_1^*(\Delta a_{l1})\right) \\ &= a_{k(2n)} + \delta + F_n^*(\Delta a_{k(2n-1)}) \\ &\quad - \sigma\left(a_{k1} + \delta + F_1^*(\Delta a_{k1})\right)\end{aligned}$$

i.e. if

$$\begin{aligned}-\sigma\left(a_{k1} + F_1^*(\Delta a_{k1})\right) &= \delta - \sigma\left(a_{k1} + \delta + F_1^*(\Delta a_{k1})\right) \\ -\sigma'\left(a_{k1} + F_1^*(\Delta a_{k1})\right) - a_{k1} - F_1^*(\Delta a_{k1}) &= \delta - \sigma'\left(a_{k1} + \delta + F_1^*(\Delta a_{k1})\right) - a_{k1} - \delta - F_1^*(\Delta a_{k1}) \\ \sigma'\left(a_{k1} + F_1^*(\Delta a_{k1})\right) &= \sigma'\left(a_{k1} + \delta + F_1^*(\Delta a_{k1})\right)\end{aligned}$$

From (8.1),

$$\Pr[\Delta b_{k(2n-1)} = \Delta b_{l(2n-1)}] = \frac{1}{|\mathcal{M}|}. \tag{8.5}$$

> **Notation:** We will write that "there is an input to a round function distinct for two plaintexts $x_k \neq x_l$", if $x = (a_1, a_2, \ldots, a_{2n})$ and there is an i such that $\Delta a_{k(2i-1)} \neq \Delta a_{l(2i-1)}$.

The following lemma shows that we need at least n rounds to get the inputs to all the round functions distinct (for all pairs of plaintexts the attacker has access to) with a high probability.

Lemma 8.3.4 *Let $F_{11}^*, \ldots, F_{1n}^*, F_{21}^*, \ldots, F_{2n}^*, \ldots F_{r1}^*, \ldots, F_{rn}^*$ be independent perfect random functions on a group $\mathcal{M}$. Consider r rounds of the scalable* IDEA *(performing σ and block shift also after the last round). Let d be an integer. Let τ be any transcript, $X_\tau = (x_1, x_2, \ldots, x_d)$ with pairwise different entries, $Y_\tau = (y_1, y_2, \ldots, y_d)$, $x_k = (a_{k1}, \ldots, a_{k(2n)})$, and $y_k = (b_{k1}, \ldots, b_{k(2n)})$. Then*

1. *If $r < n$ and $d \geq 2$, it is possible to choose inputs X_τ such that there is at least one $1 \leq i \leq n$ and $1 \leq k, l \leq d$ with $\Delta b_{k(2i-1)} = \Delta b_{l(2i-1)}$;*

2. *If $r = n$:*

$$Pr[\exists k \neq l, i: \ \Delta b_{k(2i-1)} = \Delta b_{l(2i-1)}] \leq \frac{d^2 n^2}{2\,|\mathcal{M}|}.$$

Proof:

1. Let $r = n-1$ and $d = 2$. Let the attacker choose the plaintexts so that (8.4) holds. Then from (8.5) follows that there is at most one input into the next round-functions (to the last one) after the first round which differs for the two chosen plaintexts. After each following round the difference spreads to one more input to the left, so that after the r-th round at most the last r inputs are distinct, and thus after $n-1$ rounds the input to the first round function is still equal for both plaintexts.

2. Let S_k^r denote the output of the r-th round for the plaintext x_k. Then for all $1 \leq i < n$

$$\Delta S_{k(2i-1)}^{r+1} = \Delta S_{k(2i)}^r + F_{ri}^*(\Delta S_{k(2i-1)}^r) - F_{r(i+1)}^*(\Delta S_{k(2i+1)}^r)$$

and

$$\Delta S_{k(2n-1)}^{r+1} = S_{k(2n)}^r + F_{rn}^*(\Delta S_{k(2n-1)}^r) - \sigma(S_{k1}^r + F_{r1}^*(\Delta S_{k1}^r)).$$

Let $P^r[k,l,i]$ denote the probability that $\Delta S_{k(2i-1)}^r = \Delta S_{l(2i-1)}^r$. Then for all $1 \leq i < n$

$$P^{r+1}[k,l,i] \begin{cases} = 1 & \text{if } \Delta S_{k(2i)}^r = \Delta S_{l(2i)}^r \text{ and } \Delta S_{k(2i-1)}^r = \Delta S_{l(2i-1)}^r \text{ and} \\ & \Delta S_{k(2i+1)}^r = \Delta S_{l(2i+1)}^r \\ \leq \dfrac{1}{|\mathcal{M}|} & \text{otherwise} \end{cases} \tag{8.6}$$

$$P^{r+1}[k,l,n] \leq \begin{cases} 1 & \text{if } \Delta S_{k(2n-1)}^r = \Delta S_{l(2n-1)}^r \text{ and } \Delta S_{k1}^r = \Delta S_{l1}^r \\ \dfrac{1}{|\mathcal{M}|} & \text{otherwise} \end{cases} \tag{8.7}$$

The proof continues by induction for fixed k and l.

a) For the first round $S_k^0 = x_k$. If there is $1 \leq i < n$ such that $\Delta a_{ki} \neq \Delta a_{li}$, then from (8.6) and (8.7) follows that there is at least one input to the next round function distinct for both plaintexts with a high probability. If there is no such i, from (8.5) we know that with a high probability the inputs to the last round-function are distinct. In summary, there is at least one input to the next round functions i for which $P^1[k,l,i] \leq \frac{1}{|\mathcal{M}|}$.

b) Assume now that there are at least r values of i_j ($1 \leq j \leq r$) such that $P^r[k,l,i_j] = \Pr[\Delta S_{k(2i_j-1)} = \Delta S_{l(2i_j-1)}] \leq \frac{r}{|\mathcal{M}|}$ after the r-th round. In the worst case they are in sequence so that $i_{j+1} = i_j + 1$, which causes that the difference spreads only one part further, i.e.

$$\begin{aligned} P^{r+1}[k,l,i_{j_1}-1] = P^{r+1}[k,l,i_{j_1}] = P^{r+1}[k,l,i_{j_1}+1] = \ldots = P^{r+1}[k,l,i_{j_1}+r-1] \\ \leq \frac{1}{|\mathcal{M}|} \cdot (1-p) + 1 \cdot p, \end{aligned}$$

where p is the probability that the condition in (8.6), or in (8.7), is satisfied. In both cases this probability is smaller than $\Pr[\Delta S_{k(2i_j-1)} = \Delta S_{l(2i_j-1)}] = P^r[k,l,i_j] \leq \frac{r}{|\mathcal{M}|}$ for some of the i_j. Thus after the $(r+1)$-th round there are at least $r+1$ inputs to the next round functions such that

$$P^{r+1}[k,l,i_j] \leq \frac{1}{|\mathcal{M}|} \cdot 1 + 1 \cdot \frac{r}{|\mathcal{M}|} \leq \frac{r+1}{|\mathcal{M}|}.$$

Therefore, after the n-th round, for all inputs to the next round functions,

$$P^n[k,l,i] \leq \frac{n}{|\mathcal{M}|},$$

and the overall probability

$$\Pr[\exists k \neq l, i:\ \Delta b_{k(2i-1)} = \Delta b_{l(2i-1)}] \leq \binom{d}{2} \cdot n \cdot \frac{n}{|\mathcal{M}|} = \frac{d^2 n^2}{2\,|\mathcal{M}|}.$$

■

Using the previous lemma we can evaluate the number of rounds in order for the scheme to be secure against the chosen plaintext attack.

Theorem 8.3.5 *Let $F_{11}^*, \ldots, F_{1n}^*, F_{21}^*, \ldots, F_{2n}^*, \ldots, F_{(n+2)1}^*, \ldots, F_{(n+2)n}^*$ for $n > 1$ be independent perfect random functions on a group $\mathcal{M}$, and let d be an integer. Then*

$$AdvC^{\mathrm{ACPA}(d)}(\Pi_n^\sigma[F_{11}^*, \ldots, F_{(n+2)n}^*] \leq \frac{d^2(n+1)^2}{2\,|\mathcal{M}|}.$$

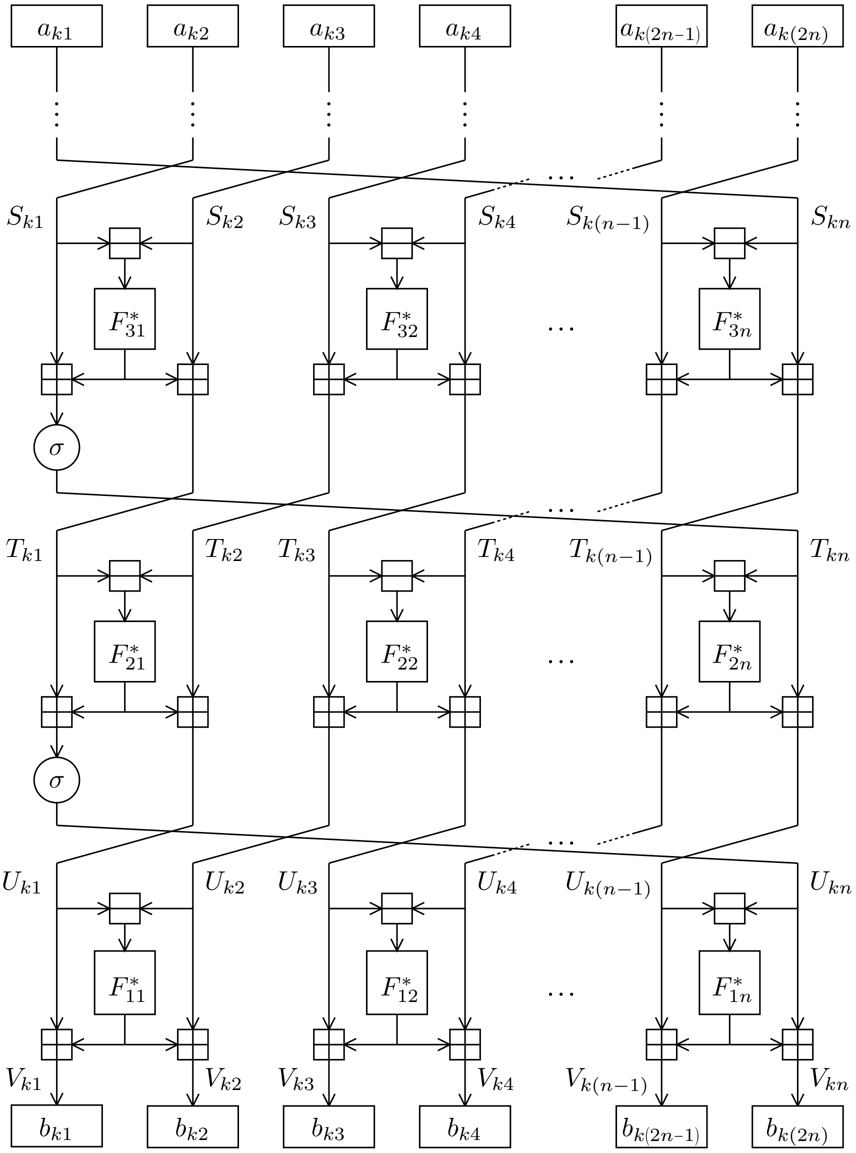


Figure 8.7: The last three rounds of an r-round scalable IDEA scheme

Proof: A d-limited known-plaintext attack distinguisher has access to d plaintexts $x_1, x_2, \dots, x_d$ and corresponding ciphertexts $y_1, y_2, \dots, y_d$. When the oracle implements the IDEA scheme, the ciphertexts are calculated as depicted in Figure 8.7. For convenience we will index the round functions in reverse order, i.e. F_{1i} are the round functions of the last round.

We may assume that all inputs in X to the oracle are pairwise different. Let $\mathcal{Y} = \{Y = (y_1, \dots, y_d) \in \mathcal{M} | \forall k : y_k = (b_{k1}, \dots, b_{k(2n)})$ and $\forall k \neq l \ \forall i : \Delta b_{k(2i-1)} \neq \Delta b_{l(2i-1)}\}$. Consider any fixed value of $Y \in \mathcal{Y}$. Then $V_k = y_k$ if and only if $V_{k(2i-1)} = b_{k(2i-1)}$ and $\Delta U_{k(2i-1)} = \Delta b_{k(2i-1)}$ for all $1 \leq i \leq n$.

$$V_{k(2i-1)} = U_{k(2i-1)} + F^*_{1i}(\Delta U_{k(2i-1)}) = U_{k(2i-1)} + F^*_{1i}(\Delta b_{k(2i-1)}) = b_{k(2i-1)}$$

and we can define an event E_k as follows:

$$E_k = \left[\forall 1 \leq i \leq n : \ F^*_{1i}(\Delta b_{k(2i-1)}) = b_{k(2i-1)} - U_{k(2i-1)}\right].$$

Since

$$\Delta U_{k(2i-1)} = T_{k(2i)} + F^*_{2i}(\Delta T_{k(2i-1)}) - T_{k(2i+1)} - F^*_{2(i+1)}(\Delta T_{k(2i+1)}) = \Delta b_{k(2i-1)},$$

for all $1 \leq i < n$, and

$$\Delta U_{k(2n-1)} = T_{k(2n)} + F^*_{2n}(\Delta T_{k(2n-1)}) - \sigma(T_{k1} + F^*_{21}(\Delta T_{k1})) = \Delta b_{k(2n-1)},$$

we get

$$\sum_{i=1}^{n} \Delta b_{k(2i-1)} = \sum_{i=1}^{n} \Delta U_{k(2i-1)} = -\sigma'(T_{k1} + F_{k1}^{*}(\Delta T_{k1})) - \sum_{i=1}^{n} \Delta T_{k(2i-1)}$$

and we can define another event F_k:

$$F_k = \Bigg[F_{21}^{*}(\Delta T_{k1}) \in {\sigma'}^{-1}\left(-\sum_{i=1}^{n} \Delta b_{k(2i-1)} - \sum_{i=1}^{n} \Delta T_{k(2i-1)} \right) - T_{k1}$$

$$\text{and}$$

$$F_{2i}^{*}(\Delta T_{k(2i-1)}) = -\Delta b_{k(2i-3)} + \Delta T_{k(2i-2)} + F_{2(i-1)}^{*}(\Delta T_{k(2i-3)})$$

$$\text{for all } 1 < i \leq n$$

$$\Bigg].$$

Since for all i all values $\Delta b_{k(2i-1)}$ are pairwise different

$$\Pr[\forall k : E_k] = \frac{1}{|\mathcal{M}|^{nd}},$$

and if all $\Delta T_{k(2i-1)}$ are pairwise different and $-\sum_{i=1}^{n} \Delta b_{k(2i-1)} - \sum_{i=1}^{n} \Delta T_{k(2i-1)} \in \sigma'(\mathcal{M})$ for all k, then

$$\Pr\left[\forall k : F_{21}^{*}(\Delta T_{k1}) \in {\sigma'}^{-1}\left(-\sum_{i=1}^{n} \Delta b_{k(2i-1)} - \sum_{i=1}^{n} \Delta T_{k(2i-1)} \right) - T_{k1} \right]$$
$$= \prod_{k=1}^{d} \frac{\left| {\sigma'}^{-1}\left(-\sum_{i=1}^{n} \Delta b_{k(2i-1)} - \sum_{i=1}^{n} \Delta T_{k(2i-1)} \right) \right|}{|\mathcal{M}|} \geq \frac{1}{|\mathcal{M}|^{d}}.$$

and

$$\Pr\left[\forall k, 1 < i \leq n : F_{2i}^{*}(\Delta T_{k(2i-1)}) = -\Delta b_{k(2i-3)} + \Delta T_{k(2i-2)} + F_{2(i-1)}^{*}(\Delta T_{k(2i-3)}) \right]$$
$$= \frac{1}{|\mathcal{M}|^{(n-1)d}}.$$

Thus,

$$\Pr[\forall k : F_k] \geq \frac{1}{|\mathcal{M}|^{nd}},$$

Let C_1, and C_2 be the following conditions:

$$C_1 = \left[\forall k : \; -\sum_{i=1}^{n} \Delta b_{k(2i-1)} - \sum_{i=1}^{n} \Delta T_{k(2i-1)} \in \sigma'(\mathcal{M}) \right]$$
$$C_2 = \left[\forall\, k \neq l \; \forall i : \; \Delta T_{k(2i-1)} \neq \Delta T_{l(2i-1)} \right]$$

If both C_1 and C_2 hold, then

$$\left[\Pi_n^{\sigma}[F_{11}^{*}, \ldots, F_{(n+2)n}^{*}] \right]_{X_\tau, Y_\tau}^{d} \geq \frac{1}{|\mathcal{M}|^{2nd}} = \frac{\left(|\mathcal{M}|^{2n} \right)^{\underline{d}}}{|\mathcal{M}|^{2nd}} [C^{*}]_{X,Y}^{d} \geq \left(1 - \frac{d^{\underline{2}}}{2\,|\mathcal{M}|^{2n}} \right) [C^{*}]_{X,Y}^{d}$$

Now we evaluate the probability that the conditions C_1 and C_2 are not satisfied. For a fixed k:

$$\Pr\left[-\sum_{i=1}^{n} \Delta b_{k(2i-1)} - \sum_{i=1}^{n} \Delta T_{k(2i-1)} \notin \sigma'(\mathcal{M}) \right]$$
$$= \Pr\left[-\sum_{i=1}^{n} \Delta b_{k(2i-1)} - \sum_{i=1}^{n} \Delta S_{k(2i-1)} + \sigma'(S_{k1} + F_{31}^{*}(\Delta S_{k1})) \notin \sigma'(\mathcal{M}) \right]$$
$$= \Pr\left[\exists a \in \mathcal{M} : \delta + \sigma'(a) \notin \sigma'(\mathcal{M}) \right]$$

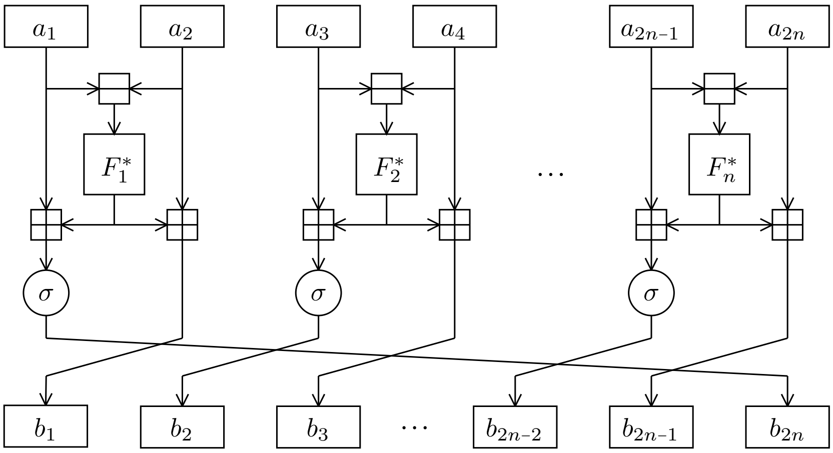


Figure 8.8: Scalable IDEA scheme with bit rotation in all sub-blocks

for a constant $\delta = -\sum_{i=1}^{n} \Delta b_{k(2i-1)} - \sum_{i=1}^{n} \Delta S_{k(2i-1)}$. From (8.3) we get that this probability is not greater than $\frac{2}{|\mathcal{M}|}$. Therefore,

$$\Pr[\neg C_1] \le \frac{2d}{|\mathcal{M}|}.$$

From Lemma 8.3.4 we know that

$$\Pr[\neg C_2] \le \frac{d^2 n^2}{2\,|\mathcal{M}|}.$$

Now we can use Corollary 3.1.4 with the following parameters:

1. $\varepsilon_1 = \frac{d^2 n}{2\,|\mathcal{M}|}$ (since $\Pr[\exists\, k \neq l\ \exists i : \Delta b_{k(2i-1)} \neq \Delta b_{l(2i-1)} \le \frac{d^2 n}{2\,|\mathcal{M}|}$,
2. $\varepsilon_2 = \frac{d^2}{2\,|\mathcal{M}|^{2n}}$, and
3. $\varepsilon_3 = \frac{d^2 n^2 + 4d}{2\,|\mathcal{M}|}$,

and we get

$$\begin{aligned} AdvC^{\mathrm{ACPA}(d)}(\Pi_n^{\sigma}[F_{11}^*, \ldots, F_{(n+2)n}^*]) &\le \frac{d^2 n}{2\,|\mathcal{M}|} + \frac{d^2}{2\,|\mathcal{M}|^{2n}} + \frac{d^2 n^2 + 4d}{2\,|\mathcal{M}|} \\ &\le \frac{d^2 n - dn + 1 + d^2 n^2 - dn^2 + 4d}{2\,|\mathcal{M}|} \\ &\le \frac{d^2 (n+1)^2}{2\,|\mathcal{M}|} \end{aligned}$$

■

In Lemma 8.3.4 we found the minimal number of rounds necessary to ensure a high probability that all inputs to the round functions are pairwise different. The worst case occurred when the attacker created two plaintexts with all differences of successive parts equal for both the plaintexts. For the scalable schemes with $n \ge 3$, we may avoid this attack by adding the bit rotation σ not only to the first sub-block but to all sub-blocks of the first round (see Figure 8.8). In this way we ensure that after the first round there are at least two different inputs to the next round functions. If the attacker chose all the differences between subsequent parts equal, all inputs to the next round functions would be distinct with a high probability for both plaintexts (from the same reason as in (8.5) for the last sub-block).

Thus, using σ in each sub-block, we get

$$\Delta S_{k(2i-1)}^{r+1} = S_{k(2i)}^{r} + F_{ri}^*(\Delta S_{k(2i-1)}^{r}) - \sigma(S_{k(2i+1)}^{r} + F_{r(i+1)}^*(\Delta S_{k(2i+1)}^{r})).$$

and we have the following cases:

1. If there is an i such that $\Delta a_{k(2i-1)} \neq \Delta a_{l(2i-1)}$ then there are at least two j ($j \in \{i, i-1\}$) such that $P[k,l,j] \leq \frac{1}{|\mathcal{M}|}$.
2. If there is no i such that $\Delta a_{k(2i-1)} \neq \Delta a_{l(2i-1)}$, but at least two i such that $\Delta a_{k(2i)} \neq \Delta a_{l(2i)}$, then for all these i, $P[k,l,i_j] \leq \frac{1}{|\mathcal{M}|}$.
3. If there is no i such that $\Delta a_{k(2i-1)} \neq \Delta a_{l(2i-1)}$ and only one i such that $\Delta a_{k(2i)} \neq \Delta a_{l(2i)}$, then the input to the i-th next round function is influenced directly, i.e. $\Pr[\Delta S^1_{k(2i-1)} = \Delta S^r_{l(2i-1)}] \leq \frac{1}{|\mathcal{M}|}$, and we show that there is at least one more next-round function with distinct inputs for the two fixed plaintexts.

   Let $a_{l(2i)} = a_{k(2i)} + \delta_1$ and $a_{l(2i+1)} = a_{k(2i+1)} + \delta_2$, then for all $j \leq 2i$, $a_{lj} = a_{kj} + \delta_1$; and for all $j > 2i$, $a_{lj} = a_{kj} + \delta_2$. Since $\Delta a_{k(2i)} \neq \Delta a_{l(2i)}$, $\delta_1 \neq \delta_2$.

   a) Let $\delta_1 = 0$ and $\delta_2 \neq 0$. Then $\Delta b_{k(2n-1)} = \Delta b_{l(2n-1)}$ if and only if

   $$\begin{aligned} &a_{k(2n)} + F_n^*(\Delta a_{k(2n-1)}) - \sigma\left(a_{k1} + F_1^*(\Delta a_{k1})\right) \\ &= a_{l(2n)} + F_n^*(\Delta a_{l(2n-1)}) - \sigma\left(a_{l1} + F_1^*(\Delta a_{l1})\right) \\ &= a_{k(2n)} + \delta_2 + F_n^*(\Delta a_{k(2n-1)}) - \sigma\left(a_{k1} + F_1^*(\Delta a_{k1})\right) \end{aligned}$$

   i.e. $\Pr[\Delta b_{k(2n-1)} = \Delta b_{l(2n-1)}] = 0$.

   b) Similarly, if $\delta_2 = 0$ and $\delta_1 \neq 0$, $\Delta b_{k(2n-1)} = \Delta b_{l(2n-1)}$ if and only if

   $$\begin{aligned} &a_{k(2n)} + F_n^*(\Delta a_{k(2n-1)}) - \sigma\left(a_{k1} + F_1^*(\Delta a_{k1})\right) \\ &= a_{l(2n)} + F_n^*(\Delta a_{l(2n-1)}) - \sigma\left(a_{l1} + F_1^*(\Delta a_{l1})\right) \\ &= a_{k(2n)} + F_n^*(\Delta a_{k(2n-1)}) - \sigma\left(a_{k1} + \delta_1 + F_1^*(\Delta a_{k1})\right) \end{aligned}$$

   and $\Pr[\Delta b_{k(2n-1)} = \Delta b_{l(2n-1)}] = 0$.

   c) Let both δ_1 and δ_2 be nonzero. Then there are at least four successive parts (two sub-blocks) with the same difference, i.e. there is an i' such that $a_{l(2i'+j)} = a_{k(2i'+j)} + \delta$ for $j \in \{-1,0,1,2\}$, and as $\Delta b_{k(2i'-1)} = \Delta b_{l(2i'-1)}$ if and only if

   $$\begin{aligned} &a_{k(2i')} + F_n^*(\Delta a_{k(2i'-1)}) - \sigma\left(a_{k(2i'+1)} + F_1^*(\Delta a_{k(2i'+1)})\right) \\ &= a_{l(2i')} + F_n^*(\Delta a_{l(2i'-1)}) - \sigma\left(a_{l(2i'+1)} + F_1^*(\Delta a_{l(2i'+1)})\right) \\ &= a_{k(2i')} + \delta + F_n^*(\Delta a_{k(2i'-1)}) - \sigma\left(a_{k(2i'+1)} + \delta + F_1^*(\Delta a_{k(2i'+1)})\right) \end{aligned}$$

   i.e. if $\sigma'\left(a_{k(2i'+1)} + F_1^*(\Delta a_{k(2i'+1)})\right) = \sigma'\left(a_{k1} + \delta + F_1^*(\Delta a_{k1})\right)$, and from (8.1), $\Pr[\Delta b_{k(2i-1)} = \Delta b_{l(2i-1)}] = \frac{1}{|\mathcal{M}|}$. Thus for example, if $n = 3$ and the distinct difference is between the second and the third part, we get the inputs to the first and second function of the next round distinct for the fixed plaintexts (see also Figure 8.9).

When $n = 2$, we can construct the plaintexts so that $a_{l1} = a_{k1} + \delta_1$, $a_{l2} = a_{k2} + \delta_1$, $a_{l3} = a_{k3} + \delta_2$, and $a_{l4} = a_{k4} + \delta_2$ with $\delta_1 \neq \delta_2$. Then $\Delta b_{k3} = \Delta b_{l3}$ if and only if

$$\begin{aligned} a_{k4} + F_n^*(\Delta a_{k3}) - \sigma\left(a_{k1} + F_1^*(\Delta a_{k1})\right) &= a_{l4} + F_n^*(\Delta a_{l3}) - \sigma\left(a_{l1} + F_1^*(\Delta a_{l1})\right) \\ &= a_{k4} + \delta_2 + F_n^*(\Delta a_{k3}) - \sigma\left(a_{k1} + \delta_1 + F_1^*(\Delta a_{k1})\right) \end{aligned}$$

i.e. if

$$\delta_2 = \sigma\left(a_{k1} + \delta_1 + F_1^*(\Delta a_{k1})\right) - \sigma\left(a_{k1} + F_1^*(\Delta a_{k1})\right)$$

Thus, $\Pr[\Delta b_{k(2i-1)} = \Delta b_{l(2i-1)}] = \Pr[\delta_2 = \sigma(\delta_1 + r) - \sigma(r)]$ for a random r. Since

$$\delta_2 = \sigma(\delta_1 + r) - \sigma(r) = 2(\delta_1 + r) + \mathrm{msb}(\delta_1 + r) - 2r - \mathrm{msb}(r) = 2\delta_1 + \mathrm{msb}(\delta_1 + r) - \mathrm{msb}(r)$$

the probability depends only on the most significant bits, and we can choose the values δ_1 and δ_2 so that the probability is high. For example, let $\delta_1 = 1$ and $\delta_2 = 2$. Then $2 = \sigma(r+1) - \sigma(r)$ if and only if

$$\mathrm{msb}(r+1) = \mathrm{msb}(r).$$

If $\mathrm{msb}(r) = 0$ then $r < 2^{m-1}$, and if $r \neq 2^{m-1} - 1$, then $\mathrm{msb}(r+1) = 0$ as well. Similarly, if $\mathrm{msb}(r) = 1$ and $r \neq 2^m - 1$ then $\mathrm{msb}(r+1) = 1$ as well. Therefore,

$$\Pr[\sigma(r+1) - \sigma(r) = 2] = \frac{|\mathcal{M}| - 2}{|\mathcal{M}|} = 1 - \frac{2}{|\mathcal{M}|},$$

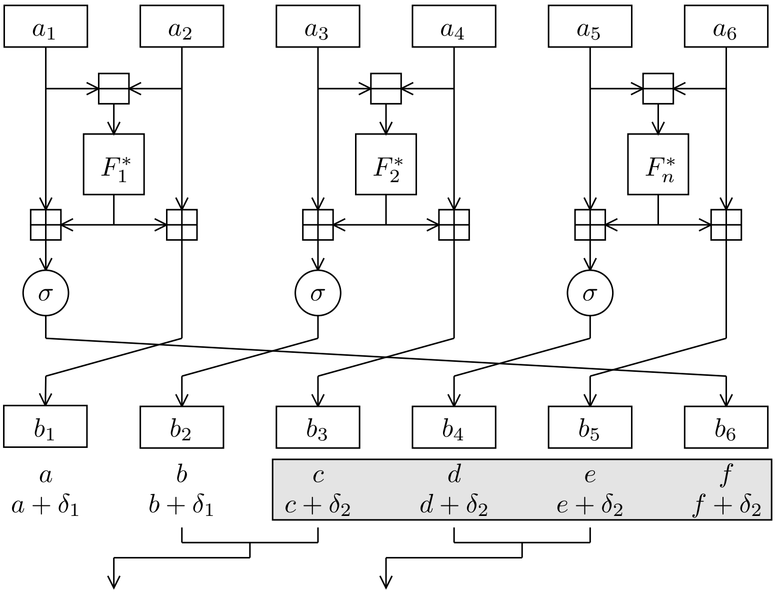


Figure 8.9: Case C: Differences after the first round

and we get with a very high probability only one input to the next round different for the two plaintexts.

Adding the bit rotation permutation σ to all sub-blocks of the first round, we get using the proof of Lemma 8.3.4 that for all $1 \le i \le n$, $P^{n-1}[k,l,i] \le \frac{n-1}{|\mathcal{M}|}$ and consequently,

$$AdvC^{\mathrm{ACPA}(d)}(\Pi_n^\sigma[F^*_{11},\dots,F^*_{(n+1)n}]) \le \frac{d^2 n}{2\,|\mathcal{M}|} + \frac{d^2}{2\,|\mathcal{M}|^{2n}} + \frac{d^2 n^2 + 4d}{2\,|\mathcal{M}|} \le \frac{d^2 n^2 + 1}{2\,|\mathcal{M}|}$$

Note that we cannot save more rounds, since choosing the plaintext so that they differ only in one part (i.e. there is an j such that $a_{kj} \ne a_{lj}$, and for all $i \ne j$, $a_{ki} = a_{li}$), we need $n-1$ rounds to get all inputs to the next round functions pairwise different for both the plaintexts.

8.3.2 Adaptive Chosen Plaintext-Ciphertext Attack

Similarly as for the basic scheme we can make use of the previous theorem in order to evaluate the number of rounds necessary for making the scalable scheme super-pseudorandom.

Theorem 8.3.6 *Let $F^*_{11},\dots,F^*_{1n},F^*_{21},\dots,F^*_{2n},\dots,F^*_{(2n+2)1},\dots,F^*_{(2n+2)n}$ for $n > 1$ be independent perfect random functions on a group $\mathcal{M}$, and let d be an integer. Then*

$$AdvC^{\mathrm{ACPCA}(d)}(\Pi_n^\sigma[F^*_{11},\dots,F^*_{(2n+2)n}] \le \frac{d^2 n^2}{|\mathcal{M}|}.$$

Proof: Similarly as for the basic scheme of IDEA, we may use the proof of Theorem 8.3.5, adding a new condition:

$$C_3 = \left[\forall\, k \ne l\; \forall i:\; \Delta\pi_{k(2i-1)} \ne \Delta\pi_{l(2i-1)}\right],$$

for $\pi_k = (\Pi_n^\sigma)^{-1}\,[F^*_{(n+3)1},\dots F^*_{(2n+2)n}](y_k)$. The probability that C_3 does not hold may be analyzed in the same way as in Lemma 8.3.4 with the same result, i.e.

$$\Pr[\neg C_3] \le \frac{d^2 n^2}{2\,|\mathcal{M}|}.$$

Now we can use Corollary 3.1.6 with the following parameters:

1. $\varepsilon_1 = \frac{d^2}{2\,|\mathcal{M}|^{2n}}$, and
2. $\varepsilon_2 = \frac{2d^2 n^2 + 4d}{2\,|\mathcal{M}|} = \frac{d^2 n^2 + 2d}{|\mathcal{M}|}$,

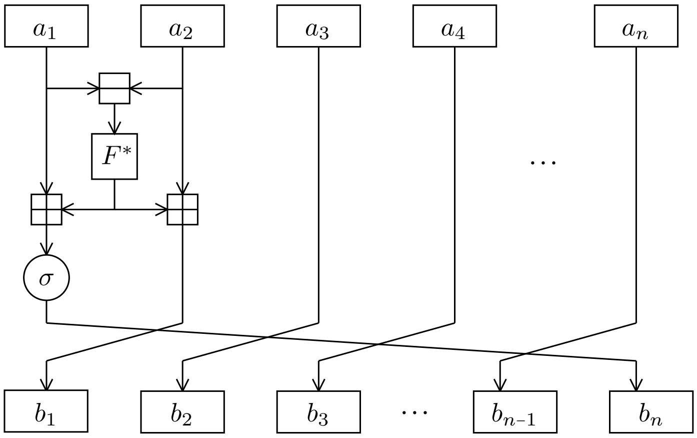


Figure 8.10: Simple scalable IDEA scheme

and we get

$$AdvC^{\mathrm{ACPCA}(d)}(\Pi_n^\sigma[F_{11}^*,\ldots,F_{(2n+2)n}^*]) \le \frac{d^2}{2\,|\mathcal{M}|^{2n}} + \frac{d^2n^2+2d}{|\mathcal{M}|} \le \frac{d^2n^2}{|\mathcal{M}|}.$$

■

Using the same argument as for the adaptive chosen plaintext attack, we can save two rounds in schemes with $n > 3$ by adding the bit rotation σ to all sub-blocks after the first and before the last round, so that we need altogether only $2n$ rounds to ensure super-pseudorandomness, and the advantage of this scheme is

$$AdvC^{\mathrm{ACPCA}(d)}(\Pi_n^\sigma[F_{11}^*,\ldots,F_{(2n)n}^*]) \le \frac{d^2}{2\,|\mathcal{M}|^{2n}} + \frac{d^2n^2+2d}{|\mathcal{M}|} \le \frac{d^2n^2}{|\mathcal{M}|}.$$

8.4 Simple Scalable Scheme Based on IDEA

The scalable scheme in the previous section enables parallelization of the computation so that the ciphertext of an r-round scalable scheme can be calculated in r steps computing all n round-functions in parallel in each step. When the parallel computation is not possible, the ciphertext can be calculated in rn-steps. Another solution is to design a scalable scheme which requires only one primitive function per round. An example of such a scheme is depicted in Figure 8.10. In this section, we analyze this type of scheme, and show how many rounds we need to ensure their pseudorandomness and super-pseudorandomness.

Similarly as in the previous scheme we enhanced the scheme by the permutation operation, since for the even number of rounds we get the same characteristic as described in Distinguisher 8.3. Note that this attack cannot be applied against this schemes with odd number of sub-blocks.

First, we again observe the input/output dependencies between individual parts.

Example 8.4.1 *Consider a simple scalable scheme of* IDEA *with four parts. Let the input to the scheme be* (a_1, a_2, a_3, a_4) *and denote the* i*-th part of the output block after the first r rounds by* $a_{r_i}+F_{ri}(a_1,a_2,a_3,a_4)$*. The following table shows which part of the plaintext is on which position after the* r*-th round, and on which input parts the function* F_{ri} *depends. Note that for* $1 < i < n$, $F_{ri} = F_{(r-1)(i+1)}$.

after	*1*		*2*		*3*		*4*	
the round r	i_1	F_{r1}	i_2	F_{r2}	i_3	F_{r3}	i_4	F_{r4}
1	*2*	*1, 2*	*3*	—	*4*	—	*1*	*1, 2*
2	*3*	*1–3*	*4*	—	*1*	*1, 2*	*2*	*1–3*
3	*4*	*1–4*	*1*	*1, 2*	*2*	*1–3*	*3*	*1–4*
4	*1*	*1–4*	*2*	*1–3*	*3*	*1–4*	*4*	*1–4*
5	*2*	*1–4*	*3*	*1–4*	*4*	*1–4*	*1*	*1–4*

Thus, the table shows that we need at least 5 rounds to get all output parts dependent on all input parts.

This example may be generalized to the following theorem.

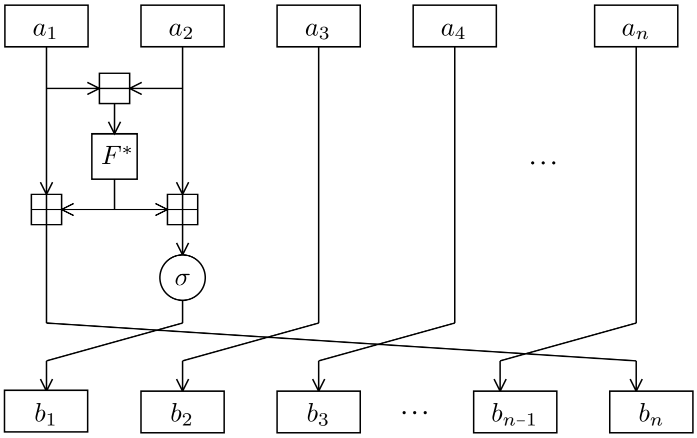


Figure 8.11: Modification of the simple scalable IDEA scheme

Theorem 8.4.2 *Let the scalable* IDEA *scheme divide the input into n parts. Then it requires at least $2n-3$ rounds to get all output parts dependent on all input parts.*

Proof: After the first round, the first and the last output parts depend on the first and the second input parts; all other output parts are in the plaintext form. After each following round, the first and the last output block depend on one more input block coming after the block shift from the third part to the second part. Thus, we need at least $n-1$ rounds to get the first and the last block dependent on all inputs. After that we need further $n-2$ rounds to spread the dependency to the other $n-2$ parts of the input. ■

Consider now that an attacker obtains d plaintext/ciphertext pairs (x_k, y_k) with pairwise different plaintexts such that $x_k = (a_{k1}, \dots, a_{kn})$. Consider two fixed plaintexts x_k and x_l, with $k \neq l$. If the plaintexts are such that all differences between subsequent parts are in both plaintexts equal, i.e. for all $1 \leq i \leq n-1$

$$\Delta a_{ki} = \Delta a_{li}, \tag{8.8}$$

there are two possible cases:

1. If $a_{k1} = a_{l1}$ then from (8.8) also all other $a_{ki} = a_{li}$, and thus $x_1 = x_2$.
2. If $a_{k1} \neq a_{l1}$, let $a_{l1} = a_{k1} + \delta$. Then from (8.8) we get that for all i, $a_{li} = a_{ki} + \delta$.

Since we assume all the plaintexts to be different, only the second case is relevant.

Let $(b_{k1}, \dots, b_{kn})$ be outputs of the first round as depicted in Figure 8.10. Then for all $1 < i < n-1$

$$\begin{aligned} &\Delta b_{li} = \Delta a_{li} = \Delta a_{ki} = \Delta b_{ki}, \text{ and} \\ &\Delta b_{l1} = a_{l2} + F^*(\Delta a_{l1}) - a_{l3} = a_{k2} + \delta + F^*(\Delta a_{k1}) - a_{k3} - \delta = \Delta b_{k1} \end{aligned}$$

Thus, the only difference may occur in $\Delta b_{k(n-1)} \neq \Delta b_{l(n-1)}$, and then we have to wait for the following $n-1$ rounds until we get it as input of the round function.

Consider now that the permutation σ is placed on the output of the second part rather than of the first one (see Figure 8.11) and the same plaintexts. Then for all $1 < i < n-1$

$$\begin{aligned} &\Delta b_{li} = \Delta a_{li} = \Delta a_{ki} = \Delta b_{ki}, \text{ and} \\ &\Delta b_{l(n-1)} = a_{ln} - a_{l1} - F^*(\Delta a_{l1}) = a_{kn} + \delta - a_{k1} - \delta - F^*(\Delta a_{k1}) = \Delta b_{k(n-1)} \end{aligned}$$

and $\Delta b_{k1} = \Delta b_{l1}$ if and only if

$$\sigma\left(a_{k2} + F^*(\Delta a_{k1})\right) - a_{k3} = \sigma\left(a_{l2} + F^*(\Delta a_{l1})\right) - a_{l3} = \sigma\left(a_{k2} + \delta + F^*(\Delta a_{k1})\right) - a_{k3} - \delta$$

i.e. if

$$\begin{aligned} \sigma'\left(a_{k2} + F^*(\Delta a_{k1})\right) + a_{k2} + F^*(\Delta a_{k1}) &= \sigma'\left(a_{k2} + \delta + F^*(\Delta a_{k1})\right) + a_{k2} + \delta + F^*(\Delta a_{k1}) - \delta \\ \sigma'\left(a_{k2} + F^*(\Delta a_{k1})\right) &= \sigma'\left(a_{k2} + \delta + F^*(\Delta a_{k1})\right) \end{aligned}$$

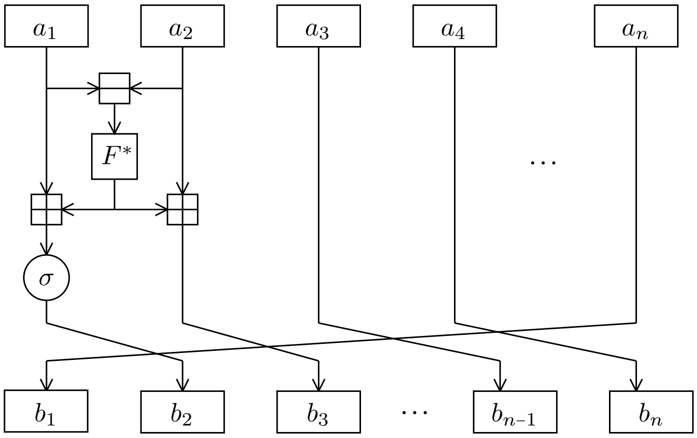


Figure 8.12: Simple scalable IDEA scheme with the right block-rotation

From (8.1),

$$\Pr[\Delta b_{k1} = \Delta b_{l1}] = \frac{1}{|\mathcal{M}|}. \tag{8.9}$$

Thus, considering these inputs in the modified scheme, we have a high probability to get different inputs to the round function for each fixed pair of plaintexts already after the first round.

> **Notation:** The modified simple scalable scheme of IDEA enhanced by the permutation σ (depicted in Figure 8.11) will be denoted by Π_1^σ, i.e. $\Pi_1^\sigma : \mathcal{M}^n \times \mathcal{M}^n \rightarrow$ for a set $\mathcal{M} = \{0, 1, \ldots, 2^m - 1\}$. We will further consider only this simple scheme, and omit the word "modified".

Note that Theorem 8.4.2 holds also for the modified scheme. Further, we would achieve the same effect of getting a different inputs to the round function already after the first round also by changing the block rotation direction from left to right as depicted in Figure 8.12.

8.4.1 Adaptive Chosen Plaintext Attack

Consider a d-limited chosen plaintext attack against the simple scheme of IDEA with pairwise different plaintext blocks $x_1, x_2, \ldots, x_d$. Let $x_k = (a_{k1}, \ldots, a_{kn})$ and $y_k = (b_{k1}, \ldots, b_{kn})$. Consider two fixed plaintexts x_k and x_l with $k \neq l$. In the previous section we already showed that choosing the plaintexts so that they differ in the first or the second part, we get the input to the next round function different with a high probability, even if the inputs to the first round function are equal. Thus, in order to get as many equal inputs to the round functions as possible, the attacker has to choose the plaintext so that they differs only in the last part, i.e. $a_{ki} = a_{li}$ for all $1 \leq i < n$ and $a_{kn} \neq a_{ln}$. In that case we need $n - 2$ rounds to get the first input to a round function different for these plaintexts (i.e. to get the different values from the last part to the second one).

In general, let $\min[k, l]$ be defined as follows:

- If there is an i ($1 \leq i < n$) such that $\Delta a_{ki} \neq \Delta a_{li}$, then $\min[k, l]$ is the number of rounds necessary to get the first difference to the input of a round function, i.e.

$$\min[k, l] = \begin{cases} 0 & \text{if } i = 1 \\ i - 2 & \text{if } i > 1 \end{cases}$$

- If $\Delta a_{ki} = \Delta a_{li}$ for all $1 \leq i < n$, then

$$\min[k, l] = 0.$$

Let S_k^r denote the output of the $(\min[k, l] + r)$-th round on the plaintext x_k. The differences between the subsequent parts are as follows:

$$\Delta S_{k1}^r = \sigma\left(S_{k2}^{r-1} + F_r^*(\Delta S_{k1}^{r-1})\right) - S_{k1}^{r-1}$$
$$\Delta S_{k(n-1)}^r = \Delta S_{kn}^{r-1} - F_r^*(\Delta S_{k1}^{r-1})$$

and for all $1 < i < n$

$$\Delta S^r_{ki} = \Delta S^{r-1}_{k(i+1)}.$$

Let $P^r[k,l,i]$ denote the probability that $\Delta S^r_{ki} = \Delta S^r_{li}$. Then for $i \in \{1, n-1\}$

$$P^{r+1}[k,l,i] \le \begin{cases} 1 & \text{if } \Delta S^r_{k1} = \Delta S^r_{l1} \\ \dfrac{1}{|\mathcal{M}|} & \text{otherwise} \end{cases} \tag{8.10}$$

and for $1 < i < n$

$$P^{r+1}[k,l,i] = \Pr[\Delta S^r_{k(i+1)} = \Delta S^r_{l(i+1)}] = P^r[k,l,i+1] \tag{8.11}$$

The following lemma shows that after the $n-2$ rounds necessary to ensure that we get distinct inputs to a round function, the inputs to the next $n-3$ rounds are distinct with a high probability.

Lemma 8.4.3 *Consider $2n-5$ rounds of the simple scalable* IDEA *(performing σ and the block rotation also after the last round). Let d be an integer, and τ be a transcript such that $X_\tau = (x_1, x_2, \ldots, x_d)$ with pairwise different inputs, and $Y_\tau = (y_1, y_2, \ldots, y_d)$. Let $x_k = (a_{k1}, \ldots, a_{kn})$ and $y_k = (b_{k1}, \ldots, b_{kn})$. Then for any fixed plaintexts x_k and x_l ($k \ne l$), the probability that their inputs to the r-th round function for $n-1 \le r \le 2n-5$ are equal is not greater than $\frac{n-2}{|\mathcal{M}|}$.*

Proof: Let $k \ne l$ be fixed, and $j = \min[k,l]$.

I. First, we analyze the case when there is an i ($1 \le i < n$) such that $\Delta a_{ki} \ne \Delta a_{li}$.

First, consider the $n-2$ rounds after the j-th one. (Note that if $j = n-2$, we do not have to take the last one into account.) The proof continues by induction.

a) From the definition of $\min[k,l]$, $\Delta S^0_{k1} \ne \Delta S^0_{l1}$, and from (8.10) we get that

$$P^1[k,l,i] \le \begin{cases} \dfrac{1}{|\mathcal{M}|} & \text{for } i \in \{1, n-1\} \\ 1 & \text{otherwise.} \end{cases}$$

b) Assume that after the $(j+r)$-th round, the probabilities are as follows:

$$\begin{aligned} P^r[k,l,1] &\le \frac{r}{|\mathcal{M}|} \\ P^r[k,l,i] &\le 1 && \text{for } 1 < i < n-r \\ P^r[k,l,i] &\le \frac{i-n+r+1}{|\mathcal{M}|} && \text{for } n-r \le i \le n-1 \end{aligned}$$

After the next round we get from (8.11)

$$P^{r+1}[k,l,i] = P^r[k,l,i+1] \le \begin{cases} 1 & \text{for } 1 < i < n-r-1 \\ \dfrac{i+2-n+r}{|\mathcal{M}|} & \text{for } n-r-1 \le i < n-1 \end{cases}$$

and for $i \in \{1, n-1\}$:

$$P^{r+1}[k,l,i] \le \frac{1}{|\mathcal{M}|} \cdot 1 + 1 \cdot p,$$

where p is the probability that the condition in (8.10) is satisfied, i.e.

$$p = \Pr[\Delta S^r_{k1} = \Delta S^r_{l1}] = P^r[k,l,1]. \tag{8.12}$$

Therefore, for $i \in \{1, n-1\}$

$$P^{r+1}[k,l,i] \le \frac{1}{|\mathcal{M}|} \cdot 1 + 1 \cdot \frac{r}{|\mathcal{M}|} \le \frac{r+1}{|\mathcal{M}|}, \tag{8.13}$$

and after the $(j+n-2)$-nd round we have

$$\begin{aligned} P^{n-2}[k,l,1] &\le \frac{n-2}{|\mathcal{M}|} \\ P^{n-2}[k,l,i] &\le \frac{i-1}{|\mathcal{M}|} && \text{for } 2 \le i \le n-1 \end{aligned}$$

If $j < n-3$, we need to perform some more rounds, however, for all $j+n-2 < r \le n-3-j$, $P^{n-2+r-1}[k,l,2] = P^{n-2}[k,l,r+1] \le \frac{r}{|\mathcal{M}|}$. Thus,

$$\begin{aligned} p &= \Pr[\Delta S_{k1}^{n-2+r-1} = \Delta S_{l1}^{n-2+r-1} \wedge \Delta S_{k2}^{n-2+r-1} = \Delta S_{l2}^{n-2+r-1}] \\ &\le \Pr[\Delta S_{k2}^{n-2+r-1} = \Delta S_{l2}^{n-2+r-1}] = P^{n-2+r-1}[k,l,2] \le \frac{r}{|\mathcal{M}|}, \end{aligned}$$

we get

$$P^{n-2+r}[k,l,1] \le \frac{r+1}{|\mathcal{M}|}. \tag{8.14}$$

In the last round $r \le n-3$, and thus for all $n-2 \le r \le 2n-5$

$$P^r[k,l,1] \le \frac{n-2}{|\mathcal{M}|}.$$

II. The other case, when for all $1 \le i < n$ $\Delta a_{ki} = \Delta a_{li}$ is very similar to the previous one, with the difference that

$$P^1[k,l,i] \begin{cases} \le \dfrac{1}{|\mathcal{M}|} & \text{for } i = 1 \\ = 1 & \text{otherwise.} \end{cases}$$

and

$$\begin{aligned} &P^r[k,l,1] \le \frac{r}{|\mathcal{M}|} \\ &P^r[k,l,i] \le 1 && \text{for } 1 < i \le n-r \\ &P^r[k,l,i] \le \frac{i-n+r+1}{|\mathcal{M}|} && \text{for } n-r < i \le n-1 \end{aligned}$$

Hence, we need one more round (i.e. $n-1$ rounds) to eliminate the last $P^r[k,l,2] = 1$. After the $(n-1)$-st round we get

$$\begin{aligned} &P^{n-1}[k,l,1] \le \frac{n-1}{|\mathcal{M}|} \\ &P^{n-1}[k,l,i] \le \frac{i}{|\mathcal{M}|} && \text{for } 2 \le i \le n-1 \end{aligned}$$

Then we need to add $n-4$ rounds. Similarly as in the previous case, for all $j+n-1 < r \le n-4$, $P^{n-1+r-1}[k,l,2] = P^{n-1}[k,l,r+1] \le \frac{r+1}{|\mathcal{M}|}$, and thus

$$P^{n-1+r}[k,l,1] \le \frac{r+2}{|\mathcal{M}|}. \tag{8.15}$$

After the last (i.e. $(n-4)$-th) round, we again get

$$P^r[k,l,1] \le \frac{n-2}{|\mathcal{M}|}$$

for all $n-2 \le r \le 2n-5$.

■

Theorem 8.4.4 *Let $F_1^*, F_2^*, \ldots, F_{2n-3}^*$ be independent perfect random functions on a group $\mathcal{M}$, and let d be an integer. Then*

$$AdvC^{\mathrm{ACPA}(d)}(\Pi_1^\sigma[F_1^*, \ldots, F_{2n-3}^*] \le \frac{d^2n^2+1}{2\,|\mathcal{M}|}.$$

Proof: A d-limited known-plaintext attack distinguisher has access to d plaintexts $x_1, x_2, \ldots, x_d$ and corresponding ciphertexts $y_1, y_2, \ldots, y_d$. When the oracle implements the IDEA scheme, the ciphertexts are calculated as depicted in Figure 8.13.

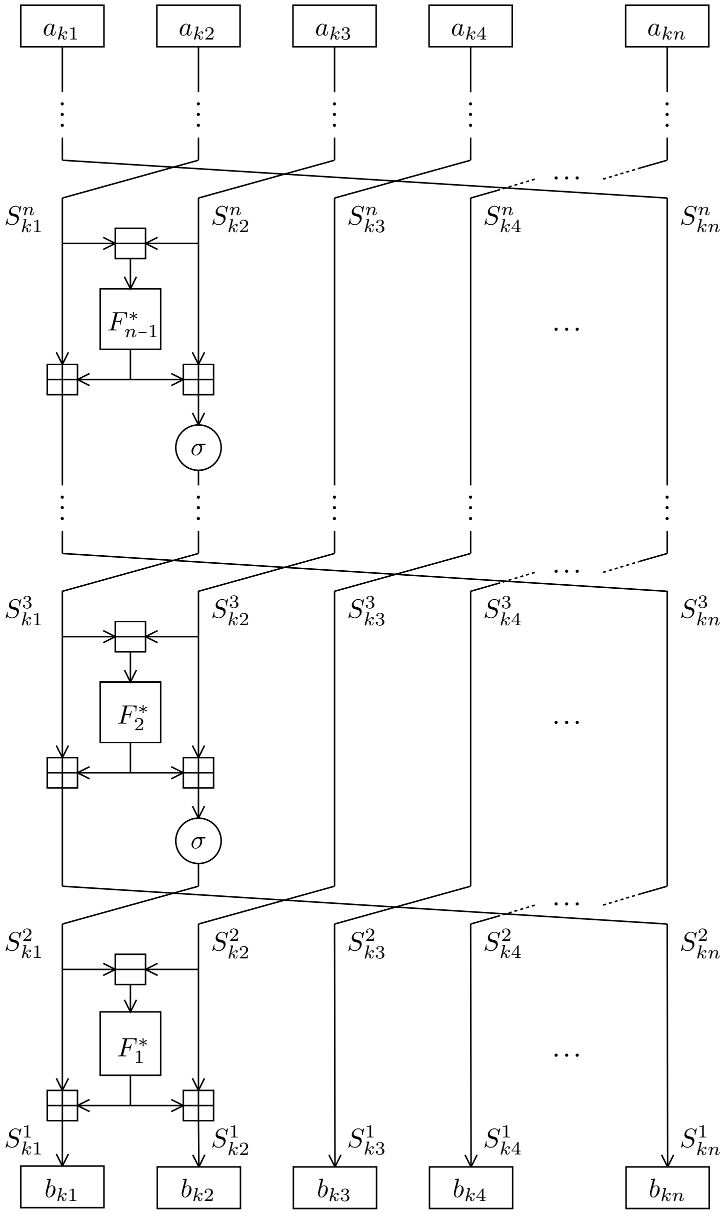


Figure 8.13: The last $n-1$ rounds of an r-round simple scalable IDEA scheme

We may assume that all inputs in X to the oracle are pairwise different. Let $\mathcal{Y} = \{Y = (y_1, \dots, y_d) \in \mathcal{M} | \forall k : y_k = (b_{k1}, \dots, b_{kn})$ and $\forall k \neq l : \ \Delta b_{k1} \neq \Delta b_{l1}\}$. Consider any fixed value of $Y \in \mathcal{Y}$.

1. If n is even then $S^1_k = y_k$ if and only if

$$
\begin{aligned}
b_{k1} &= S^1_{k1} = S^2_{k1} + F^*_1(\Delta S^2_{k1}) = S^2_{k1} + F^*_1(\Delta b_{k1})\\
\Delta b_{k1} &= \Delta S^1_{k1} = \Delta S^2_{k1} = \sigma\left(S^3_{k2} + F^*_2(\Delta S^3_{k1})\right) - S^3_{k3}\\
b_{k3} &= S^1_{k3} = S^n_{k1} + F^*_{n-1}(\Delta S^n_{k1})\\
\Delta b_{k3} &= \Delta S^1_{k3} = \Delta S^{n-1}_{kn} - F^*_{n-2}(\Delta S^{n-1}_{k1})\\
b_{k5} &= S^1_{k5} = S^{n-2}_{k1} + F^*_{n-3}(\Delta S^{n-2}_{k1})\\
\Delta b_{k5} &= \Delta S^1_{k5} = \Delta S^{n-3}_{kn} - F^*_{n-4}(\Delta S^{n-3}_{k1})\\
&\dots\\
b_{k(n-1)} &= S^1_{k(n-1)} = S^4_{k1} + F^*_3(\Delta S^4_{k1})\\
\Delta b_{k(n-1)} &= \Delta S^1_{k(n-1)} = \Delta S^3_{kn} - F^*_2(\Delta S^3_{k1})
\end{aligned}
$$

Therefore,

$$\begin{aligned}\Delta b_{k1} + \Delta b_{k(n-1)} &= \sigma\left(S^3_{k2} + F^*_2(\Delta S^3_{k1})\right) - S^3_{k3} + \Delta S^3_{kn} - F^*_2(\Delta S^3_{k1}) \\ &= \Delta S^3_{k2} + \Delta S^3_{kn} + \sigma'\left(S^3_{k2} + F^*_2(\Delta S^3_{k1})\right)\end{aligned}$$

and we can define an event E_k as follows:

$$\begin{aligned}E_k = \big[&F^*_1(\Delta S^2_{k1}) = b_{k1} - S^2_{k1} \wedge \\ &F^*_2(\Delta S^3_{k1}) = {\sigma'}^{-1}\left(\Delta b_{k1} + \Delta b_{k(n-1)} - \Delta S^3_{k2} - \Delta S^3_{kn}\right) - S^3_{k2} \wedge \\ &\forall 1 < i \leq \frac{n}{2} : F^*_{2i-1}(\Delta S^{2i}_{k1}) = b_{i(n-2i+3)} - S^{2i}_{k1} \wedge \\ &\forall 1 < i < \frac{n}{2} : F^*_{2i}(\Delta S^{2i+1}_{k1}) = \Delta S^{2i+1}_{kn} - \Delta b_{k(n-2i+3)} \\ \big]&\end{aligned}$$

There is a solution for F^*_2 if and only if the following condition (denote it by C_1) is satisfied: For all k,

$$\begin{aligned}&\Delta b_{k1} + \Delta b_{k(n-1)} - S^3_{k2} + S^3_{k3} - S^3_{kn} + S^3_{k1} \\ &= \Delta b_{k1} + \Delta b_{k(n-1)} - S^4_{k3} + S^4_{k4} - S^4_{k1} - F_3(\Delta S^4_{k1}) + \sigma\left(S^4_{k2} + F_3(\Delta S^4_{k1})\right) \\ &= \Delta b_{k1} + \Delta b_{k(n-1)} - \Delta S^4_{k1} - \Delta S^4_{k3} + \sigma'\left(S^4_{k1} + F_3(\Delta S^4_{k1})\right) \in \sigma'(\mathcal{M}).\end{aligned}$$

From (8.3),

$$\Pr[\neg C_1] = \Pr\left[\Delta b_{k1} + \Delta b_{k(n-1)} - \Delta S^3_{k2} - \Delta S^3_{kn} \notin \sigma'(\mathcal{M})\right] \leq \frac{2}{|\mathcal{M}|}. \tag{8.16}$$

2. Similarly, if n is odd then $S^1_k = y_k$ if and only if

$$\begin{aligned}b_{k1} &= S^1_{k1} = S^2_{k1} + F^*_1(\Delta S^2_{k1}) = S^2_{k1} + F^*_1(\Delta b_{k1}) \\ \Delta b_{k1} &= \Delta S^1_{k1} = \Delta S^2_{k1} = \sigma\left(S^3_{k2} + F^*_2(\Delta S^3_{k1})\right) - S^3_{k3} \\ b_{k3} &= S^1_{k3} = S^n_{k1} + F^*_{n-1}(\Delta S^n_{k1}) \\ \Delta b_{k3} &= \Delta S^1_{k3} = \Delta S^{n-1}_{kn} - F^*_{n-2}(\Delta S^{n-1}_{k1}) \\ &\ldots \\ b_{k(n-2)} &= S^1_{k(n-2)} = S^5_{k1} + F^*_4(\Delta S^5_{k1}) \\ \Delta b_{k(n-2)} &= \Delta S^1_{k(n-2)} = \Delta S^4_{kn} - F^*_3(\Delta S^4_{k1}) \\ b_{kn} &= S^1_{kn} = S^3_{k1} + F^*_2(\Delta S^3_{k1})\end{aligned}$$

Therefore,

$$\Delta b_{k1} - b_{kn} = \sigma'\left(S^3_{k2} + F^*_2(\Delta S^3_{k1})\right) - S^3_{k1} + \Delta S^3_{k2}$$

and we can define an event E_k as follows:

$$\begin{aligned}E_k = \Big[&F^*_1(\Delta b_{k1}) = b_{k1} - S^2_{k1} \wedge \\ &F^*_2(\Delta S^3_{k1}) = {\sigma'}^{-1}\left(\Delta b_{k1} - b_{kn} + S^3_{k1} - \Delta S^3_{k2}\right) - S^3_{k2} \wedge \\ &\forall 1 < i \leq \left\lfloor \frac{n}{2} \right\rfloor : F^*_{2i}(\Delta S^{2i+1}_{k1}) = b_{k(n-2i+2)} - S^{2i+1}_{k1} \wedge \\ &\forall 1 < i < \left\lfloor \frac{n}{2} \right\rfloor : F^*_{2i-1}(\Delta S^{2i}_{k1}) = \Delta S^{2i}_{kn} - \Delta b_{k(n-2i+2)} \\ \Big]&\end{aligned}$$

There is a solution for F^*_2 if and only if the following condition (C_1) is satisfied: For all k

$$\begin{aligned}&\Delta b_{k1} - b_{kn} + S^3_{k1} - S^3_{k2} + S^3_{k3} \\ &= \Delta b_{k1} - b_{kn} + \sigma\left(S^4_{k2} + F^*_3(\Delta S^4_{k1})\right) - S^4_{k3} + S^4_{k4} \in \sigma'(\mathcal{M})\end{aligned}$$

and thus

$$\Pr[\neg C_1] = \Pr\left[\exists k :\ \Delta b_{k1} - b_{kn} + S^3_{k1} - \Delta S^3_{k2} \notin \sigma'(\mathcal{M})\right] \leq \frac{1}{|\mathcal{M}|}. \tag{8.17}$$

All Δb_{k1} are pairwise different. If also ΔS^i_{k1} are pairwise different for all $3 \le i \le n$ and all arguments for σ'^{-1} have a preimage in $\mathcal{M}$, then

$$\Pr[\forall k : E_k] \ge \frac{1}{|\mathcal{M}|^{(n-1)d}} \ge \frac{1}{|\mathcal{M}|^{nd}}.$$

Let C_2 be the following conditions:

$$C_2 = \left[\forall\, k \ne l \;\forall\, 3 \le i \le n :\; \Delta S^i_{k1} \ne \Delta S^i_{l1}\right]$$

If both C_1 and C_2 hold, then

$$\left[\Pi^\sigma_1[F^*_1, \ldots, F^*_{2n-3}]\right]^d_{X_\tau, Y_\tau} \ge \frac{1}{|\mathcal{M}|^{nd}} \ge \left(1 - \frac{d^{\underline{2}}}{2\,|\mathcal{M}|^n}\right)[C^*]^d_{X,Y}$$

From (8.16) and (8.17), $\Pr[\neg C_1] \le \frac{2}{|\mathcal{M}|}$; and from Lemma 8.4.3, $\Pr[\neg C_2] \le \frac{d^{\underline{2}}(n-2)^2}{2\,|\mathcal{M}|}$. Thus, we can use Corollary 3.1.4 with the following parameters:

1. $\varepsilon_1 = \frac{d^{\underline{2}}}{2\,|\mathcal{M}|}$ (since $\Pr[\exists\, k \ne l :\; \Delta b_{k1} \ne \Delta b_{l1} \le \frac{d^{\underline{2}}}{2\,|\mathcal{M}|}$),
2. $\varepsilon_2 = \frac{d^{\underline{2}}}{2\,|\mathcal{M}|^n}$, and
3. $\varepsilon_3 = \frac{d^{\underline{2}}(n-2)^2+4d}{2\,|\mathcal{M}|}$,

and we get

$$\begin{aligned} AdvC^{\mathrm{ACPA}(d)}(\Pi^\sigma_1[F^*_1, \ldots, F^*_{2n-3}]) &\le \frac{d^{\underline{2}}}{2\,|\mathcal{M}|} + \frac{d^{\underline{2}}}{2\,|\mathcal{M}|^n} + \frac{d^{\underline{2}}(n-2)^2 + 4d}{2\,|\mathcal{M}|} \\ &\le \frac{d^2n^2+1}{2\,|\mathcal{M}|} \end{aligned}$$

■

Note that for the simplified scalable scheme of IDEA we achieved the minimal number of rounds — see Theorem 8.4.2.

8.4.2 Adaptive Chosen Plaintext-Ciphertext Attack

Here we evaluate the number of rounds for which the simple scalable IDEA achieves the super-pseudorandomness.

Theorem 8.4.5 *Let $F^*_1, F^*_2, \ldots, F^*_{3n-5}$ be independent perfect random functions on a group $\mathcal{M}$, and let d be an integer. Then*

$$AdvC^{\mathrm{ACPCA}(d)}(\Pi^\sigma_1[F^*_1, \ldots, F^*_{3n-5}] \le \frac{d^2n^2+1}{2\,|\mathcal{M}|}.$$

Proof: Similarly as for the basic scheme of IDEA, we may use the proof of Theorem 8.4.4, adding a new condition

$$C_3 = [\forall\, k \ne l :\; \Delta\pi_{k1} \ne \Delta\pi_{l1}]\,,$$

where $\pi_k = (\Pi^\sigma_1)^{-1}[F^*_{2n-2}, \ldots F^*_{3n-5}](y_k)$. The probability that C_3 does not hold may be analyzed in the same way as in Lemma 8.4.3, and

$$\Pr[\neg C_3] \le \frac{d^{\underline{2}}(n-2)}{2\,|\mathcal{M}|}.$$

Thus, we can use Corollary 3.1.6 with the following parameters:

1. $\varepsilon_1 = \frac{d^{\underline{2}}}{2\,|\mathcal{M}|^n}$, and
2. $\varepsilon_2 = \frac{d^{\underline{2}}(n-2)^2+d^{\underline{2}}(n-2)+4d}{2\,|\mathcal{M}|}$,

and we get

$$AdvC^{\mathrm{ACPCA}(d)}(\Pi^\sigma_1[F^*_1, \ldots, F^*_{3n-5}]) \le \frac{d^{\underline{2}}}{2\,|\mathcal{M}|^{2n}} + \frac{d^{\underline{2}}(n-2)^2 + d^{\underline{2}}(n-2) + 4d}{2\,|\mathcal{M}|} \le \frac{d^2n^2+1}{2\,|\mathcal{M}|}$$

■

8.5 Conclusions

In this chapter, we examined three schemes based on the IDEA cipher — the basic scheme of IDEA, and two of its scalable variants — with emphasis on their pseudorandomness and super-pseudorandomness.

Each of the schemes divides the input into several parts. The basic scheme of IDEA uses two parts of the same size. The scalable schemes generalize its concept for an even number (the scalable scheme) or any number (the simple scalable scheme) of parts of the same size as in the basic IDEA scheme. Assume that the parts are from a group $\mathcal{M}$. Then for a $d \ll \sqrt{|\mathcal{M}|}$, we can summarize the results of this chapter as follows:

Basic scheme of IDEA

For the basic scheme of IDEA dividing the plaintext blocks into two equally large parts from a group $\mathcal{M}$:

- 2 rounds are secure against the known plaintext attack ($AdvC^{\mathrm{KPA}(d)}(\Lambda^\sigma[F_1^*, F_2^*]) \leq \frac{d^2+d+1}{|\mathcal{M}|}$);
- 2 rounds are not secure against the chosen plaintext attack;
- 3 rounds are secure against the adaptive chosen plaintext attack ($AdvC^{\mathrm{ACPA}(d)}(\Lambda^\sigma[F_1^*, F_2^*, F_3^*]) \leq \frac{d^2+d+1}{|\mathcal{M}|}$);
- 4 rounds are secure against the adaptive chosen plaintext-ciphertext attack ($AdvC^{\mathrm{ACPCA}(d)}(\Lambda^\sigma[F_1^*, F_2^*, F_3^*, F_4^*]) \leq \frac{d^2+d+1}{|\mathcal{M}|}$).

Scalable scheme of IDEA

For the scalable scheme of IDEA dividing the plaintext blocks into $2n$ equally large parts from a group $\mathcal{M}$:

- $n-1$ rounds are not secure against the chosen plaintext attack;
- $n+2$ rounds are secure against the adaptive chosen plaintext attack ($AdvC^{\mathrm{ACPA}(d)}(\Pi_n^\sigma[F_{11}^*, \ldots, F_{(n+2)n}^*] \leq \frac{d^2(n+1)^2}{2\,|\mathcal{M}|}$);
- $2n+2$ rounds are secure against the adaptive chosen plaintext-ciphertext attack ($AdvC^{\mathrm{ACPCA}(d)}(\Pi_n^\sigma[F_{11}^*, \ldots, F_{(2n+2)n}^*] \leq \frac{d^2n^2}{|\mathcal{M}|}$).

Modifying the scheme so that the bit rotation permutation σ is performed in each sub-block of the first and the second to last round, we may improve the schemes with $n > 2$ (i.e. with at least 6 parts) as follows:

- $n+1$ rounds are secure against the adaptive chosen plaintext attack;
- $2n$ rounds are secure against the adaptive chosen plaintext-ciphertext attack.

Simple scalable scheme of IDEA

For the simple scalable schemes of IDEA dividing the plaintext blocks into n equally large parts from a group $\mathcal{M}$:

- $2n-4$ rounds are not secure against the chosen plaintext attack;
- $2n-3$ rounds are secure against the adaptive chosen plaintext attack ($AdvC^{\mathrm{ACPA}(d)}(\Pi_1^\sigma[F_1^*, F_2^*, \ldots, F_{2n-3}^*] \leq \frac{d^2n^2+1}{2\,|\mathcal{M}|}$);
- $3n-5$ rounds are secure against the adaptive chosen plaintext-ciphertext attack ($AdvC^{\mathrm{ACPCA}(d)}(\Pi_1^\sigma[F_1^*, F_2^*, \ldots, F_{3n-5}^*] \leq \frac{d^2n^2+1}{2\,|\mathcal{M}|}.$).

Scalability of IDEA through primitives

Both the scalable schemes of IDEA discussed above scale the original IDEA scheme through the structure. Here we shortly discuss its scalability through primitives.

The round function of the original IDEA scheme is depicted in Figure 8.14. It is based on the design concept of mixing operations from different algebraic groups. The round function has a $2m$-bit input, which is divided into two equally large parts (each of m bits). The $\boxplus$ operation is the addition of m-bit integers modulo 2^m, and $\odot$ is the multiplication of integers from the set $\{1, 2, \ldots, 2^m\}$ modulo $2^m + 1$ (with 2^m represented by zero). However, the multiplication operation requires $2^m + 1$ to be a prime, which is only

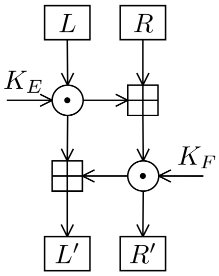


Figure 8.14: Round function of the original IDEA scheme

the case for $m \in \{1, 2, 4, 8, 16\}$. For the higher powers of 2 ($m \in \{32, 64, 128, 256\}$), $2^m + 1$ is no more a prime [6], thus we may use this structure only for the IDEA scheme with the block size up to 64 bits which is nowadays considered to be insufficient. In order to be able to scale the scheme through primitives we need another operation for $\odot$. One possible solution is to generate it from the key using a random bit generator and table representation similarly as the S-boxes of the TST cipher (see Section 7.1). Security of such a round function depends on the quality of the random bit generator. Further, we have to take into account that the inner structure of the round function is symmetric, and the $\odot$ operation is thus represented by a huge table of $2^m \times m$ bits, where $4m$ is the size of the input to the overall cipher.

Chapter 9

Comparison of the Analyzed Schemes

In this thesis, we examined three schemes: Feistel networks, TST, and IDEA. We evaluated the advantage of different attacks against these schemes. Here we summarize the obtained results, and discuss the ideal number of rounds for each scheme with respect to the individual attacks and a chosen upper bound on the advantage. We already employed this approach for the Feistel networks in Section 3.9. To evaluate the ideal number of rounds assuming the upper-bound on the advantage 2^{-l} we use Theorem 2.4.4, which gives:

$$AdvC^{\mathrm{ATK}(d)}(C[F^*_{11},\dots,F^*_{1n},\dots,F_{(tk)1},\dots,F^*_{(tk)n}]) \leq \frac{1}{2}\,(2a)^k \leq 2^{-l}$$

with the following parameters:

- $F^*_{11},\dots,F^*_{1n},\dots,F_{(tk)1},\dots,F^*_{(tk)n}$ are independent perfect random functions;
- C is one of the mentioned iterative ciphers
- n is number of primitives used per round (e.g. the Feistel network has only one primitive function per round, on the other hand the basic TST has three) — the r-th round uses the functions $F^*_{r1},\dots,F^*_{rn}$ as primitives ;
- ATK is a class of attacks;
- d is the number of plaintext/ciphertext pairs an attacker may obtain in the attack (the size of the attack);
- t is the minimal number of rounds to achieve security against the class of attacks (e.g. 3 for ACPA against the Feistel network — see Theorem 3.4.8);
- tk is the overall number of rounds;
- $0 \leq a < 1$ is the upper-bound on the advantage of an attack from the class of attacks ATK accessing at most d plaintext/ciphertext pairs against the scheme C consisting of t rounds (i.e. $a = AdvC^{\mathrm{ATK}(d)}(C[F^*_1,\dots,F^*_t])$).

Considering these parameters we need at least tk rounds where

$$k \geq \frac{l-1}{-\lg a - 1}$$

to achieve the requested limit on the advantage. We will further compare the schemes on the basis of a block size of the 128 bits, which is nowadays the standard size; the size of attack $d = 2$ (which ensures also security against the differential and linear analysis); and the limit for the advantage 2^{-128} (the selection of this parameter is discussed in Section 3.9).

Feistel network

For a Feistel network which divides the input in two parts, of which the smaller one has m bits:

- The threshold number of rounds for the pseudorandomness is 3 and for the super-pseudorandomness 4, when $d \ll 2^{m/2}$. [Theorem 3.4.8 and 3.8.4]
- The ideal number of rounds for achieving an advantage of less than 2^{-l} is $b\frac{l-1}{m-1-2\lg d}$ with $b = 3$ for the pseudorandomness and 4 for the super-pseudorandomness, when $d \ll 2^{m/2}$.
- Considering a balanced Feistel network with $m = 64$ and $d = 2$, we get 7 rounds for the pseudorandomness and 9 rounds for the super-pseudorandomness.
- Considering an unbalanced Feistel network with $m = 16$ and $d = 2$, we get 30 rounds for the pseudorandomness and 40 rounds for the super-pseudorandomness.

Basic TST

For a basic TST scheme which divides the input in two parts, with the m-bit left part:

- The threshold number of rounds for the pseudorandomness is 1 and for the super-pseudorandomness 2, when $d \ll 2^{m/2}$. [Theorem 7.2.3 and 7.2.6]
- The ideal number of rounds for achieving an advantage of less than 2^{-l} is $b\frac{l-1}{m-1-2\lg d}$ with $b = 1$ for the pseudorandomness and 2 for the super-pseudorandomness, when $d \ll 2^{m/2}$.
- With $m = 16$ and $d = 2$, we get 10 rounds for the pseudorandomness and 20 rounds for the super-pseudorandomness.

Simplified TST

For a simplified TST scheme which divides the input in two parts, with the m-bit left part:

- The threshold number of rounds is 2 for both the pseudorandomness and super-pseudorandomness, when $d \ll 2^{m/2}$. [Theorem 7.2.5]
- The ideal number of rounds for achieving an advantage of less than 2^{-l} is $2\frac{l-1}{m-1-2\lg d}$ for both the pseudorandomness and super-pseudorandomness, when $d \ll 2^{m/2}$.
- With $m = 16$ and $d = 2$, we get 20 for both the pseudorandomness and super-pseudorandomness.

Basic scheme of IDEA

For a basic IDEA scheme which divides the input in two m-bit parts:

- The threshold number of rounds is 3 for the pseudorandomness and 4 for the super-pseudorandomness, when $d \ll 2^{m/2}$. [Theorem 8.2.4 and 8.2.5]
- The ideal number of rounds for achieving an advantage of less than 2^{-l} is $b\frac{l-1}{m-1-2\lg(d+1)}$ with $b = 3$ for the pseudorandomness and 4 for the super-pseudorandomness, when $d \ll 2^{m/2}$.
- With $m = 64$ and $d = 2$, we get 7 rounds for the pseudorandomness and 9 rounds for the super-pseudorandomness.

Scalable scheme of IDEA

For a scalable IDEA scheme which divides the input in $2n$ m-bit parts:

- The threshold number of rounds for the pseudorandomness is $n + 2$ and $2n + 2$ for the super-pseudorandomness, when $d \ll 2^{m/2}$, which gives 4 and 6 respectively. [Theorem 8.3.5 and 8.3.6]
- The ideal number of rounds for achieving an advantage of less than 2^{-l} is $4\frac{l-1}{m-2\lg d-2\lg(n+1)}$ for the pseudorandomness and $6\frac{l-1}{m-1-2\lg d-2\lg n}$ for the super-pseudorandomness, when $d \ll 2^{m/2}$.
- With $m = 32$, $n = 2$ (two sub-blocks, i.e. 4 parts) and $d = 2$, we get 19 rounds for the pseudorandomness and 29 rounds for the super-pseudorandomness.

Simple scalable scheme of IDEA

For a simple scalable IDEA scheme which divides the input in n m-bit parts:

- The threshold number of rounds for the pseudorandomness is $2n - 3$ and $3n - 5$ for the super-pseudorandomness, when $d \ll 2^{m/2}$, which gives 5 and 7 respectively. [Theorem 8.4.4 and 8.4.5]
- The ideal number of rounds for achieving an advantage of less than 2^{-l} is $b\frac{l-1}{m-2\lg d-2\lg n}$ with $b = 5$ for the pseudorandomness and 7 for the super-pseudorandomness, when $d \ll 2^{m/2}$.
- With $m = 32$, $n = 4$ and $d = 2$, we get 23 rounds for the pseudorandomness and 32 rounds for the super-pseudorandomness.

Summary

The numbers of rounds ensuring the pseudorandomness and the super-pseudorandomness of the above mentioned schemes are summarized in the following table:

scheme	pseudorandomness		super-pseudorandomness	
	threshold number of rounds	ideal number of rounds	threshold number of rounds	ideal number of rounds
balanced Feistel	3	7	4	9
unbalanced Feistel	3	30	4	40
basic TST	1	10	2	20
simplified TST	2	20	2	20
IDEA	3	7	4	9
scalable IDEA	3	19	6	29
simple scalable IDEA	5	23	7	32

Note that purely comparing the number of rounds presented in the table is not sufficient for arbitration which cipher is better for implementation. We have to take into account also the overall number of primitives and their computational cost. For example, the original scheme of IDEA requires less rounds, but inputs and outputs of its primitives are twice as large as those of the scalable schemes. The computational cost depends also on the number of primitives in a scheme. For example, the scalable IDEA uses 2 functions per round, thus altogether 38 for pseudorandomness and 58 functions for super-pseudorandomness in the ideal case, while the simple scalable scheme uses only one function per round, i.e. 23 for pseudorandomness and 32 for super-pseudorandomness, which is significantly less. Also note that the bounds given by Theorem 2.4.4 are not tight, thus for the ideal (super-)pseudorandomness a smaller number of rounds might be sufficient (see also Section 7.7).

Chapter 10

Summary

Provable security and scalability are two desirable properties of a cipher. The first one ensures that it fulfills our expectations on securing encrypted data. The scalability makes the operation of the cipher easier, enabling adaptation of the level of security to the current requirements just by changing some parameters. In this thesis we discuss provable security of three scalable schemes (Feistel networks, TST and IDEA) in the random oracle model.

The first part of the thesis is devoted to the foundations of provable security. It introduces the security model based on two main concepts: the random oracle model and indistinguishability; and builds mathematical tools enabling to evaluate security of a cipher in this model. We discuss security against general attacks with emphasis on the adaptive chosen plaintext attack and the adaptive chosen plaintext-ciphertext attack, because ciphers resistant to these attacks were defined by Luby and Rackoff in [14] as pseudorandom and super-pseudorandom respectively. The proof technique is illustrated by the analysis of the unbalanced Feistel networks against these attacks. Besides the general attacks, we examine also composed attacks, which try to build a stronger attack based on some less efficient ones. Popular examples of the composed attacks are the differential and linear cryptanalysis, which are discussed separately. The first part concludes with the analysis of the operational modes of ciphers. They are very important for practical encryption, since in the real employment of a cipher the encrypted data are usually much longer than the basic block of the cipher. We address security of the modes defined by the FIPS 81 standard for DES, and one modification suggested by Diffie.

The second part of the thesis deals with the provable security of scalable ciphers. Two schemes are analyzed using the tools introduced in the first part. The first one, TST, demonstrates the scalability through primitives of the scheme; the other one, IDEA, the scalability through the structure of the scheme. Using the TST cipher we illustrate that careful selection of primitives can make the cipher provable secure also without full analysis. Our analysis focuses on one of the primitive functions — a hash function — and shows how slight changes in the function can influence the security of the whole scheme. Since the primitive function of the IDEA cipher is not scalable over 32 bits, we focus our analysis to scalability through structure. We introduce two schemes — one with parallel and the other one with serial structure — and evaluate their security. The second part closes with comparison of the schemes discussed in the thesis.

Appendix A

Notation

$1_{(X,Y)\in A}$	1 if $(X,Y) \in A$, 0 otherwise
$Adv_D^{\mathrm{ATK}(d)}(F_1,F_2)$	advantage of the distinguisher D realizing attack from the class of attacks ATK distinguishing between functions F_1 and F_2 [page 8]
$AdvC^{\mathrm{ATK}(d)}(C)$	advantage of the best distinguisher realizing attack from the class of attacks ATK against the cipher C (distinguishing between C and C^*) [page 8]
$AdvC_D^{\mathrm{ATK}(d)}(C\|A)$	advantage of the distinguisher D realizing attack from the class of attacks ATK against the cipher C if the oracle queries and responses satisfy the condition A [page 9]
$AdvC^{\mathrm{ATK}(d\|q)}(\mathrm{Mode}[C])$	advantage of the best distinguisher for the cipher C used in a particular mode Mode and a perfect cipher querying an oracle with up to q messages containing together up to d blocks [page 55]
$AdvF^{\mathrm{ATK}(d)}(F)$	advantage of the best distinguisher realizing attack from the class of attacks ATK against the function F (distinguishing between F and F^*) [page 8]
$AdvF_D^{\mathrm{ATK}_D(d)}(F\|A)$	advantage of the distinguisher D realizing attack from the class of attacks ATK against the function F if the oracle queries and responses satisfy the condition A [page 9]
ATK	class of attacks
ATK^+	induced attack against a function F if there is no inversion of the function calculated during an attack from the class ATK [page 14]
ATK^-	induced attack against a function F if there is only the inversion of the function (F^{-1}) calculated during an attack from the class ATK [page 14]
$\mathrm{ATK}^\pm$	induced attack against a function F if there both F and F^{-1} are calculated during an attack from the class ATK [page 14]
ATK_F	induced attack against the function F if the usage of the function is not determined [page 14]
C	cipher (random permutation) [page 10]
C^*	perfect cipher (perfect random permutation)[page 7]
$DecC^d(C)$	decorrelation bias of the cipher C according to the distance D [page 12]
$DecF^d(F)$	decorrelation bias of the function F according to the distance D [page 12]
F	random function [page 10]
F^*	perfect random function [page 7]
$[F]^d$	d-wise distribution matrix of the function F [page 10]
$\lg(x)$	$\log_2(x)$
$m^{\underline{k}}$	$m(m-1)(m-2)\cdots(m-k+1)$
$\mathcal{M}$	message space
$\mathcal{M}^+$	$\mathcal{M} \setminus \{0\}$
$\mathcal{M}^*$	$\bigcup_{i=0}^{k} \mathcal{M}^k$
$\mathcal{M}^k$	$\{(x_1,\ldots,x_k)\|\forall\, 1 \le i \le k : x_i \in \mathcal{M}\}$
$\|\mathcal{M}\|$	the number of elements of $\mathcal{M}$
$\mathrm{msb}(x)$	the most significant bit of x
S^L	left half of the block S [page 92]

S^R	right half of the block S [page 92]
$S_k^r = (S_{k1}^r, S_{k2}^r, \ldots, S_{kn}^r)$	output of the r-th round (if $r = 0$ input) of a (simple) scalable IDEA for the k-the plaintext
X_τ	sequence of plaintexts which the distinguisher can extract from the trace τ (sequence of plaintexts occurred during an attack) [page 19]
Y_τ	sequence of ciphertexts which the distinguisher can extract from the trace τ (sequence of ciphertexts occurred during an attack) [page 19]
ΔS	$S^L - S^R$ [page 92]
ΔS_i	for a block $S = (s_1, s_2, \ldots, s_n)$: $s_i - s_{i+1}$ if $i < n$, or $s_n - s_1$ if $i = n$ [page 100]
ΔS_{ki}^r	$\Delta S_{ki}^r = S_{ki}^r - S_{k(i+1)}^r$
Θ	simplified TST scheme [page 74]
Λ^σ	scheme of IDEA [page 92]
$\pi_{x_1,y_1}([F]_{X,Y}^d)$	sub-matrix of the decorrelation matrix obtained by fixing the first plaintext/ciphertext pair to (x_1, y_1) (see also Definition C.5)
Π_n^σ	scalable scheme of IDEA [page 101]
Π_1^σ	simple scalable scheme of IDEA [page 111]
Φ	basic TST scheme [page 74]
Ψ	unbalanced Feistel network scheme
Ω	composed scheme [page 14]
τ	trace, i.e. sequence of all queries and responses occurred during an attack [page 19]

Appendix B

Acronyms

ACCA	Adaptive Chosen Ciphertext Attack
ACPA	Adaptive Chosen Plaintext Attack
ACPCA	Adaptive Chosen Plaintext-Ciphertext Attack
AES	Advanced Encryption Standard
CA	Combined Attack
CBC	Cipher Block Chaining Mode
CCA	Chosen Ciphertext Attack
CFB	Cipher Feedback Mode
CPA	Chosen Plaintext Attack
CPCA	Chosen Plaintext-Ciphertext Attack
CRT	Counter Mode
DCA	Differential Cryptanalysis
DES	Data Encryption Standard
ECB	Electronic Codebook Mode
IA	Iterated Attack
IDEA	International Data Encryption Standard
IV	Initialization Vector
KPA	Known Plaintext Attack
LCA	Linear Cryptanalysis
MSB	Most Significant Bit
OFB	Output Feedback Mode
PES	Proposed Encryption Standard
ROM	Random Oracle Model
SBC	Scalable Block Cipher
UFN	Unbalanced Feistel Network

Appendix C

Matrix Norms

The decorrelation theory uses norms on sets of matrices as a tool for calculating the distance between functions. This distance determines the advantage (strength) of an attack against a cryptographic function or permutation. Here we summarize the norms used in Chapter 3 for evaluating advantage of different types of attacks, and prove their properties.

Let $\mathcal{A}$ be a set of matrices. A mapping from $\mathcal{A}$ to the set of real numbers is a **norm** if for all matrices $A, B \in \mathcal{A}$ for which the particular operation makes sense the following properties hold:

1. $\|A\| = 0$ if and only if $A = 0$,
2. $\|u \cdot A\| = |u| \cdot \|A\|$, for any real number u,
3. $\|A + B\| \leq \|A\| + \|B\|$.

A norm is a **matrix norms** [26], if

4. $\|A \times B\| \leq \|A\| \cdot \|B\|$

Example C.1 *Let $\mathcal{A}$ be a set of all matrices. A mapping $N^f_\infty : \mathcal{A} \to \mathbb{R}$ such that for all $A \in \mathcal{A}$, $N^f_\infty(A) = \max_{X,Y} \frac{|A_{X,Y}|}{f(X,Y)}$ (with the convention that $a/0 = 0$ for any a) is a norm.*

Proof:

1. $N^f_\infty([0]) = \max_{X,Y} \frac{|0|}{f(X,Y)} = 0$
2. $N^f_\infty(u \cdot A) = \max_{X,Y} \frac{|[u \cdot A]_{X,Y}|}{f(X,Y)} = \max_{X,Y} \frac{|u \cdot A_{X,Y}|}{f(X,Y)} = |u| \cdot \max_{X,Y} \frac{|A_{X,Y}|}{f(X,Y)} = |u| \cdot N^f_\infty(A)$
3. $N^f_\infty(A + B) = \max_{X,Y} \frac{|[A+B]_{X,Y}|}{f(X,Y)} = \max_{X,Y} \frac{|A_{X,Y} + B_{X,Y}|}{f(X,Y)} \leq$
   $\max_{X,Y} \left\{ \frac{|A_{X,Y}|}{f(X,Y)} + \frac{|B_{X,Y}|}{f(X,Y)} \right\}$
   $\leq \max_{X,Y} \frac{|A_{X,Y}|}{f(X,Y)} + \max_{X,Y} \frac{|B_{X,Y}|}{f(X,Y)} = N^f_\infty(A) + N^f_\infty(B)$

■

Example C.2 *Let $\mathcal{A}$ be a set of all matrices. A mapping $\|\cdot\|_1 : \mathcal{A} \to \mathbb{R}$ such that for all $A \in \mathcal{A}$, $\|A\|_1 = \sum_{X,Y} |A_{X,Y}|$ is a matrix norm.*

Proof:

1. $\|[0]\|_1 = \sum_{X,Y} 0 = 0$
2. $\|u \cdot A\|_1 = \sum_{X,Y} |[u \cdot A]_{X,Y}| = \sum_{X,Y} |u \cdot A_{x,y}| = |u| \sum_{X,Y} |A_{x,y}| = |u| \cdot \|A\|_1$
3. $\|A + B\|_1 = \sum_{X,Y} |[A + B]_{X,Y}| = \sum_{X,Y} |A_{x,y} + B_{x,y}| \leq \sum_{X,Y} |A_{x,y}| + \sum_{X,Y} |B_{x,y}| = \|A\|_1 + \|B\|_1$
4. $\|A \times B\|_1 = \sum_{X,Y} |[A \times B]_{X,Y}| = \sum_{X,Y} |\sum_k A_{X,k} \cdot B_{k,Y}| \leq \sum_{X,Y,k} |A_{X,k} \cdot B_{k,Y}| = \sum_{X,Y,k} |A_{X,k}| \cdot |B_{k,Y}| = \sum_{X,k} |A_{X,k}| \cdot (\sum_Y |B_{k,Y}|) \leq \sum_{X,k} |A_{X,k}| \cdot \|B\|_1 = \|A\|_1 \cdot \|B\|_1$

■

Example C.3 *Let $\mathcal{A}$ be a set of all matrices. A mapping $|||\cdot|||_\infty : \mathcal{A} \to \mathbb{R}$ such that for all $A \in \mathcal{A}$, $|||A|||_\infty = \max_X \sum_Y |A_{X,Y}|$ is a matrix norm.*

Proof:

1. $\forall X : \sum_Y |[0]_{X,Y}| = \sum_Y 0 = 0$
2. $\forall X : \sum_Y |[u \cdot A]_{X,Y}| = \sum_Y |u \cdot A_{x,y}| = |u| \sum_Y |A_{x,y}|$
   $\forall X_1, X_2 : \sum_Y |A_{X_1,Y}| \leq \sum_Y |A_{X_2,Y}| \Leftrightarrow |u| \sum_Y |A_{X_1,Y}| \leq |u| \sum_Y |A_{X_2,Y}|$
   Therefore $|||u \cdot A|||_\infty = |u| \cdot |||A|||_\infty$
3. $\forall X : \sum_Y |[A+B]_{X,Y}| = \sum_Y |A_{x,y} + B_{x,y}| \leq \sum_Y |A_{x,y}| + \sum_Y |B_{x,y}| \leq$
   $|||A|||_\infty + |||B|||_\infty$
4. $\forall X : \sum_Y |[A \times B]_{X,Y}| = \sum_Y |\sum_k A_{X,k} \cdot B_{k,Y}| \leq \sum_Y \sum_k |A_{X,k} \cdot B_{k,Y}| =$
   $\sum_k \sum_Y |A_{X,k}| \cdot |B_{k,Y}| = \sum_k |A_{X,k}| \cdot (\sum_Y |B_{k,Y}|) \leq \sum_k |A_{X,k}| \cdot |||B|||_\infty \leq$
   $|||A|||_\infty \cdot |||B|||_\infty$

■

Example C.4 *Let $\mathcal{A}$ be a set of all matrices. A mapping $\|\cdot\|_m : \mathcal{A} \rightarrow \mathbb{R}$ such that for all $A \in \mathcal{A}$, $\|A\|_m = \max_k \max_{x_1,\ldots,x_k} \max_{y_{k+1},\ldots,y_d} \sum_{x_{k+1},\ldots,x_d} \sum_{y_1,\ldots y_k} |A_{X,Y}|$ is a matrix norm.*

Proof: For short, $X^{(k)}$ will denote the left k entries of X ($x_1, \ldots, x_k$), and $X^{(d-k)}$ will denote the right $d-k$ entries of X ($x_{k+1}, \ldots, x_d$).

1. $\forall k, X^{(k)}, Y^{(d-k)} : \sum_{X^{(d-k)}} \sum_{Y^{(k)}} |[0]_{X,Y}| = \sum_{X^{(d-k)}} \sum_{Y^{(k)}} 0 = 0$
2. $\forall k, X^{(k)}, Y^{(d-k)} : \sum_{X^{(d-k)}} \sum_{Y^{(k)}} |[u \cdot A]_{X,Y}| = \sum_{X^{(d-k)}} \sum_{Y^{(k)}} |u \cdot A_{x,y}| =$
   $|u| \sum_{X^{(d-k)}} \sum_{Y^{(k)}} |A_{x,y}|$
   Therefore,
   $\max_k \max_{X^{(k)}} \max_{Y^{(d-k)}} \sum_{X^{(d-k)}} \sum_{Y^{(k)}} |[u \cdot A]_{X,Y}| =$
   $|u| \cdot \max_k \max_{X^{(k)}} \max_{Y^{(d-k)}} \sum_{X^{(d-k)}} \sum_{Y^{(k)}} |A_{x,y}|$
   and $\|u \cdot A\|_m = |u| \cdot \|A\|_m$.
3. $\forall k, X^{(k)}, Y^{(d-k)} : \sum_{X^{(d-k)}} \sum_{Y^{(k)}} |[A+B]_{X,Y}| = \sum_{X^{(d-k)}} \sum_{Y^{(k)}} |A_{x,y} + B_{x,y}| \leq$
   $\sum_{X^{(d-k)}} \sum_{Y^{(k)}} |A_{x,y}| + \sum_{X^{(d-k)}} \sum_{Y^{(k)}} |B_{x,y}|$
   Therefore,
   $\max_k \max_{X^{(k)}} \max_{Y^{(d-k)}} \sum_{X^{(d-k)}} \sum_{Y^{(k)}} |[A+B]_{X,Y}| \leq$
   $\max_k \max_{X^{(k)}} \max_{Y^{(d-k)}} \sum_{X^{(d-k)}} \sum_{Y^{(k)}} |A_{x,y}| +$
   $\max_k \max_{X^{(k)}} \max_{Y^{(d-k)}} \sum_{X^{(d-k)}} \sum_{Y^{(k)}} |B_{x,y}|$,
   and $\|A+B\|_m \|A\|_m + \|B\|_m$.
4. $\forall k, X^{(d-k)}, Y^{(k)} : \sum_{X^{(d-k)}} \sum_{Y^{(k)}} |[A \times B]_{X,Y}| = \sum_{X^{(d-k)}} \sum_{Y^{(k)}} |\sum_Z A_{X,Z} \cdot B_{Z,Y}| \leq$
   $\sum_{X^{(d-k)}} \sum_{Y^{(k)}} \sum_Z |A_{X,Z} \cdot B_{Z,Y}| = \sum_{Z^{(k)}} \sum_{Y^{(k)}} \sum_{Z^{(d-k)}} \sum_{X^{(d-k)}} |B_{Z,Y}| \cdot |A_{X,Z}| =$
   $\sum_{Z^{(k)}} \sum_{Y^{(k)}} \sum_{Z^{(d-k)}} |B_{Z,Y}| \cdot (\sum_{X^{(d-k)}} |A_{X,Z}|) \leq$
   $\sum_{Z^{(k)}} \sum_{Y^{(k)}} \sum_{Z^{(d-k)}} |B_{Z,Y}| \cdot (\max_{Z^{(d-k)}} \sum_{X^{(d-k)}} |A_{X,Z}|) =$
   $\sum_{Z^{(k)}} \sum_{Y^{(k)}} (\max_{Z^{(d-k)}} \sum_{X^{(d-k)}} |A_{X,Z}|) (\sum_{Z^{(d-k)}} |B_{Z,Y}|) \leq$
   $\sum_{Z^{(k)}} \sum_{Y^{(k)}} (\max_{Z^{(d-k)}} \sum_{X^{(d-k)}} |A_{X,Z}|) (\max_{Y^{(d-k)}} \sum_{Z^{(d-k)}} |B_{Z,Y}|) =$
   $\sum_{Z^{(k)}} (\max_{Z^{(d-k)}} \sum_{X^{(d-k)}} |A_{X,Z}|) (\sum_{Y^{(k)}} \max_{Y^{(d-k)}} \sum_{Z^{(d-k)}} |B_{Z,Y}|) \leq$
   $\sum_{Z^{(k)}} (\max_{Z^{(d-k)}} \sum_{X^{(d-k)}} |A_{X,Z}|) (\max_{Z^{(k)}} \sum_{Y^{(k)}} \max_{Y^{(d-k)}} \sum_{Z^{(d-k)}} |B_{Z,Y}|) =$
   $(\max_{Z^{(k)}} \sum_{Y^{(k)}} \max_{Y^{(d-k)}} \sum_{Z^{(d-k)}} |B_{Z,Y}|) (\sum_{Z^{(k)}} \max_{Z^{(d-k)}} \sum_{X^{(d-k)}} |A_{X,Z}|) \leq$
   $(\max_{Z^{(k)}} \sum_{Y^{(k)}} \max_{Y^{(d-k)}} \sum_{Z^{(d-k)}} |B_{Z,Y}|) (\max_{X^{(k)}} \sum_{Z^{(k)}} \max_{Z^{(d-k)}} \sum_{X^{(d-k)}} |A_{X,Z}|)$
   $\leq (\max_k \max_{X^{(k)}} \sum_{Y^{(k)}} \max_{Y^{(d-k)}} \sum_{X^{(d-k)}} |A_{X,Y}|) \cdot$
   $(\max_k \max_{X^{(k)}} \sum_{Y^{(k)}} \max_{Y^{(d-k)}} \sum_{X^{(d-k)}} |B_{X,Y}|) = \|A\|_m \cdot \|B\|_m$

■

Definition C.5 *Let $\mathcal{A}$ be a set of all $|\mathcal{M}_i|^d \times |\mathcal{M}_j|^d$ matrices. A mapping $\pi_{x_1,y_1} : |\mathcal{M}_i|^d \times |\mathcal{M}_j|^d \rightarrow |\mathcal{M}_i|^{d-1} \times |\mathcal{M}_j|^{d-1}$ is defined as follows: $\forall A \in \mathcal{A}\, \forall x_2, \ldots, x_d \in \mathcal{M}_i\ \forall y_1, \ldots, y_d \in \mathcal{M}_j$:*

$$[\pi_{x_1,y_1}(A)]_{(x_2,\ldots,x_d),(y_2,\ldots,y_d)} = A_{(x_1,x_2,\ldots,x_d),(y_1,y_2,\ldots,y_d)}$$

Lemma C.6 *Let $\mathcal{A}$ be a set of all $|\mathcal{M}_i|^d \times |\mathcal{M}_j|^d$ matrices. For all $A \in \mathcal{A}$, and for all $B \in \mathcal{A}$ for which the following operations make sense:*

1. $\pi_{x_1,y_1}([uA]) = u \cdot \pi_{x_1,y_1}(A)$

2. $\pi_{x_1,y_1}([A+B]) = \pi_{x_1,y_1}(A) + \pi_{x_1,y_1}(B)$

3. $\pi_{x_1,y_1}([A \times B]) = \sum_{k_1} \pi_{x_1,k_1}(A) \times \pi_{k_1,y_1}(B)$

Proof:

1. $(\pi_{x_1,y_1}([u \cdot A]))_{(x_2,\dots x_d),(y_2,\dots y_d)} = [u \cdot A]_{(x_1,\dots x_d),(y_1,\dots y_d)} = u \cdot A_{(x_1,\dots x_d),(y_1,\dots y_d)}$
   $= u \cdot (\pi_{x_1,y_1}(A))_{(x_2,\dots x_d),(y_2,\dots y_d)}$
2. $(\pi_{x_1,y_1}([A + B]))_{(x_2,\dots x_d),(y_2,\dots y_d)} = [A + B]_{(x_1,\dots x_d),(y_1,\dots y_d)} = A_{(x_1,\dots x_d),(y_1,\dots y_d)} + B_{(x_1,\dots x_d),(y_1,\dots y_d)} = (\pi_{x_1,y_1}(A) + \pi_{x_1,y_1}(B))_{(x_2,\dots x_d),(y_2,\dots y_d)}$
3. $(\pi_{x_1,y_1}([A \times B]))_{(x_2,\dots x_d),(y_2,\dots y_d)} = [A \times B]_{(x_1,\dots x_d),(y_1,\dots y_d)} =$
   $\sum_{k_1} \sum_{k_2,\dots k_d} A_{(x_1,\dots x_d),(k_1,\dots k_d)} \cdot B_{(k_1,\dots k_d),(y_1,\dots y_d)} =$
   $\left(\sum_{k_1} \pi_{x_1,k_1}(A) \times \pi_{k_1,y_1}(B)\right)_{(x_2,\dots x_d),(y_2,\dots y_d)}$

■

Example C.7 *Let $\mathcal{A}$ be a set of $|M_i|^d \times |M_j|^d$ matrices. A mapping $\|\cdot\|_a : \mathcal{A} \to \mathbb{R}$ such that for all $A \in \mathcal{A}$,*

$$\|A\|_a = \begin{cases} \max\limits_{x_1} \sum\limits_{y_1} \|\pi_{x_1,y_1}(A)\|_a, & d > 1, \\ |||A|||_\infty, & d = 1, \end{cases}$$

i.e. $\|A\|_a = \max_{x_1} \sum_{y_1} \max_{x_2} \sum_{y_2} \dots \max_{x_d} \sum_{y_d} \left|A_{(x_1,x_2,\dots,x_d),(y_1,y_2,\dots,y_d)}\right|$ *is a matrix norm.*

Proof: By induction:

A. For $d = 1$ see Theorem A.3.

B. $d > 1$:

1. $\|[0]\|_a = \max_{x_1} \sum_{y_1} \|\pi_{x_1,y_1}(A)\|_a = \max_{x_1} \sum_{y_1} 0 = 0$
2. $\forall x_1 : \sum_{y_1} \|\pi_{x_1,y_1}([uA])\|_a = \sum_{y_1} \|u \cdot \pi_{x_1,y_1}(A)\|_a = \sum_{y_1} |u| \cdot \|\pi_{x_1,y_1}(A)\|_a$
   $= |u| \cdot \sum_{y_1} \|\pi_{x_1,y_1}(A)\|_a$
   Hence for any x_{11} and x_{12}:
   $\sum_{y_1} \|\pi_{x_{11},y_1}([uA])\|_a \leq \sum_{y_1} \|\pi_{x_{12},y_1}([uA])\|_a \Leftrightarrow \sum_{y_1} \|\pi_{x_{11},y_1}(A)\|_a \leq \sum_{y_1} \|\pi_{x_{12},y_1}(A)\|_a$.
   From this follows that $\|[u \cdot A]\|_a = |u| \cdot \|A\|_a$
3. $\forall x_1 : \sum_{y_1} \|\pi_{x_1,y_1}([A + B])\|_a = \sum_{y_1} \|\pi_{x_1,y_1}(A) + \pi_{x_1,y_1}(B)\|_a \leq \sum_{y_1} \|\pi_{x_1,y_1}(A)\|_a + \|\pi_{x_1,y_1}(B)\|_a \leq \|A\|_a + \|B\|_a$
4. $\forall x_1 : \sum_{y_1} \|A \times B\|_a = \sum_{y_1,k_1} \|\pi_{x_1,k_1}(A) \times \pi_{k_1,y_1}(B)\|_a \leq \sum_{y_1,k_1} \|\pi_{x_1,k_1}(A)\|_a \cdot \|\pi_{k_1,y_1}(B)\|_a = \sum_{k_1} \|\pi_{x_1,k_1}(A)\|_a \left(\sum_{y_1} \|\pi_{k_1,y_1}(B)\|_a\right) \leq$
   $\sum_{k_1} \|\pi_{x_1,k_1}(A)\|_a \cdot \|B\|_a \leq \|A\|_a \cdot \|B\|_a$

■

Example C.8 *Let $\mathcal{A}$ be a set of $|M_i|^d \times |M_j|^d$ matrices. A mapping $\|\cdot\|_s : \mathcal{A} \to \mathbb{R}$ such that for all $A \in \mathcal{A}$,*

$$\|A\|_s = \begin{cases} \max\left\{\max\limits_{x_1} \sum\limits_{y_1} \|\pi_{x_1,y_1}(A)\|_s,\ \max\limits_{y_1} \sum\limits_{x_1} \|\pi_{x_1,y_1}(A))\|_s\right\}, & d > 1, \\ \max\left\{|||A|||_\infty,\ |||A^T|||_\infty\right\}, & d = 1, \end{cases}$$

where A^T is a transposed matrix to A, is a matrix norm.

Proof:

A. $d = 1$:

1. $\forall x_1 : \sum_{y_1} |[0]_{x_1,y_1}| = 0.$
   $\forall y_1 : \sum_{x_1} |[0]_{x_1,y_1}| = 0.$
   Therefore, $\|[0]\|_s = \max\{0, 0\} = 0.$

2. $\forall x_1 : \sum_{y_1} |[uA]_{x_1,y_1}| = \sum_{y_1} |u \cdot A_{x_1,y_1}| = |u| \cdot \sum_{y_1} |A_{x_1,y_1}|$.
$\forall y_1 : \sum_{x_1} |[uA]_{x_1,y_1}| = \sum_{x_1} |u \cdot A_{x_1,y_1}| = |u| \cdot \sum_{x_1} |A_{x_1,y_1}|$.
Therefore,
$\max_{x_1} \sum_{y_1} |[uA]_{x_1,y_1}| = |u| \cdot \max_{x_1} \sum_{y_1} |A_{x_1,y_1}|$,
$\max_{y_1} \sum_{x_1} |[uA]_{x_1,y_1}| = |u| \cdot \max_{y_1} \sum_{x_1} |A_{x_1,y_1}|$,
and

$$\begin{aligned}
\|[uA]\|_s &= \max\left\{\max_{x_1}\sum_{y_1}|[uA]_{x_1,y_1}|,\ \max_{y_1}\sum_{x_1}|[uA]_{x_1,y_1}|\right\} \\
&= |u| \cdot \max\left\{\max_{x_1}\sum_{y_1}|A_{x_1,y_1}|,\ \max_{y_1}\sum_{x_1}|A_{x_1,y_1}|\right\}
\end{aligned}$$

3. $\forall x_1 : \sum_{y_1} |[A+B]_{x_1,y_1}| = \sum_{y_1} |A_{x_1,y_1} + B_{x_1,y_1}| \leq \sum_{y_1} |A_{x_1,k_1}| + \sum_{y_1} |B_{x_1,y_1}|$
$\leq \max_{x_1} \sum_{y_1} |A_{x_1,k_1}| + \max_{x_1} \sum_{y_1} |B_{x_1,y_1}|$
$\forall y_1 : \sum_{x_1} |[A+B]_{x_1,y_1}| = \sum_{x_1} |A_{x_1,y_1} + B_{x_1,y_1}| \leq \sum_{x_1} |A_{x_1,k_1}| + \sum_{x_1} |B_{x_1,y_1}|$
$\leq \max_{y_1} \sum_{x_1} |A_{x_1,k_1}| + \max_{y_1} \sum_{x_1} |B_{x_1,y_1}|$
Therefore,

$$\begin{aligned}
&\|[A+B]\|_s \\
&= \max\left\{\max_{x_1}\sum_{y_1}|[A+B]_{x_1,y_1}|,\ \max_{y_1}\sum_{x_1}|[A+B]_{x_1,y_1}|\right\} \\
&\leq \max\left\{\max_{x_1}\sum_{y_1}|A_{x_1,y_1}| + \max_{x_1}\sum_{y_1}|B_{x_1,y_1}|,\ \max_{y_1}\sum_{x_1}|A_{x_1,y_1}| + \max_{y_1}\sum_{x_1}|B_{x_1,y_1}|\right\} \\
&\leq \max\left\{\max_{x_1}\sum_{y_1}|A_{x_1,y_1}|,\ \max_{y_1}\sum_{x_1}|A_{x_1,y_1}|\right\} \\
&\quad + \max\left\{\max_{x_1}\sum_{y_1}|B_{x_1,y_1}|,\ \max_{y_1}\sum_{x_1}|B_{x_1,y_1}|\right\} \\
&= \|A\|_s + \|B\|_s
\end{aligned}$$

4.

$$\begin{aligned}
\forall x_1 : \sum_{y_1}|[A\times B]_{x_1,y_1}| &= \sum_{y_1}\left|\sum_{k_1}A_{x_1,k_1}\cdot B_{k_1,y_1}\right| \leq \sum_{y_1}\sum_{k_1}|A_{x_1,k_1}\cdot B_{k_1,y_1}| \\
&= \sum_{y_1}\sum_{k_1}|A_{x_1,k_1}|\cdot|B_{k_1,y_1}| = \sum_{k_1}|A_{x_1,k_1}|\left(\sum_{y_1}|B_{k_1,y_1}|\right) \\
&\leq \sum_{k_1}|A_{x_1,k_1}|\left(\max_{k_1'}\sum_{y_1}|B_{k_1',y_1}|\right) \\
&= \max_{k_1}\sum_{y_1}|B_{k_1,y_1}|\cdot\sum_{k_1}|A_{x_1,k_1}| \\
&\leq \max_{x_1}\sum_{k_1}|A_{x_1,k_1}|\cdot\max_{k_1}\sum_{y_1}|B_{k_1,y_1}|
\end{aligned}$$

Similarly, $\forall y_1 : \sum_{x_1} |[A\times B]_{x_1,y_1}| \leq \max_{k_1} \sum_{x_1} |A_{x_1,k_1}| \cdot \max_{y_1} \sum_{k_1} |B_{k_1,y_1}|$.

Therefore,

$$\begin{aligned}
&\left\|[A \times B]\right\|_s \\
&= \max\left\{\max_{x_1}\sum_{y_1}|[A\times B]_{x_1,y_1}|,\ \max_{y_1}\sum_{x_1}|[A\times B]_{x_1,y_1}|\right\} \\
&\leq \max\left\{\max_{x_1}\sum_{x_1}|A_{x_1,y_1}|\cdot\max_{x_1}\sum_{y_1}|B_{x_1,y_1}|,\ \max_{y_1}\sum_{x_1}|A_{x_1,y_1}|\cdot\max_{y_1}\sum_{x_1}|B_{x_1,y_1}|\right\} \\
&\leq \max\left\{\max_{x_1}\sum_{x_1}|A_{x_1,y_1}|,\ \max_{y_1}\sum_{x_1}|A_{x_1,y_1}|\right\} \\
&\qquad\cdot\max\left\{\max_{x_1}\sum_{y_1}|B_{x_1,y_1}|,\ \max_{y_1}\sum_{x_1}|B_{x_1,y_1}|\right\} \\
&= \left\|A\right\|_s\cdot\left\|B\right\|_s
\end{aligned}$$

B. $d > 1$:

1. $\forall x_1 : \sum_{y_1}\left\|\pi_{x_1,y_1}[0]\right\|_s = 0.$
   $\forall y_1 : \sum_{x_1}\left\|\pi_{x_1,y_1}[0]\right\|_s = 0.$
   Therefore, $\left\|[0]\right\|_s = \max\{0,0\} = 0$
2. $\forall x_1 : \sum_{y_1}\left\|\pi_{x_1,y_1}[uA]\right\|_s = \sum_{y_1}\left\|u\cdot\pi_{x_1,y_1}A\right\|_s = |u|\cdot\sum_{y_1}\left\|\pi_{x_1,y_1}A\right\|_s$
   $\forall y_1 : \sum_{x_1}\left\|\pi_{x_1,y_1}[uA]\right\|_s = \sum_{x_1}\left\|u\cdot\pi_{x_1,y_1}A\right\|_s = |u|\cdot\sum_{x_1}\left\|\pi_{x_1,y_1}A\right\|_s$
   Therefore,
   $\max_{x_1}\sum_{y_1}\left\|\pi_{x_1,y_1}[uA]\right\|_s = |u|\cdot\max_{x_1}\sum_{y_1}\left\|\pi_{x_1,y_1}A\right\|_s$,
   $\max_{y_1}\sum_{x_1}\left\|\pi_{x_1,y_1}[uA]\right\|_s = |u|\cdot\max_{y_1}\sum_{x_1}\left\|\pi_{x_1,y_1}A\right\|_s$,
   and

$$\begin{aligned}
\left\|[uA]\right\|_s &= \max\left\{\max_{x_1}\sum_{y_1}\left\|\pi_{x_1,y_1}[uA],\right\|_s\ \max_{y_1}\sum_{x_1}\left\|\pi_{x_1,y_1}[uA]\right\|_s\right\} \\
&= |u|\cdot\max\left\{\max_{x_1}\sum_{y_1}\left\|\pi_{x_1,y_1}A,\right\|_s\ \max_{y_1}\sum_{x_1}\left\|\pi_{x_1,y_1}A\right\|_s\right\}
\end{aligned}$$

3. $\forall x_1 : \sum_{y_1}\left\|\pi_{x_1,y_1}[A+B]\right\|_s = \sum_{y_1}\left\|\pi_{x_1,y_1}A + \pi_{x_1,y_1}B\right\|_s \leq \sum_{y_1}\left\|\pi_{x_1,y_1}A_{x_1,k_1}\right\|_s + \sum_{y_1}\left\|\pi_{x_1,y_1}B\right\|_s \leq \max_{x_1}\sum_{y_1}\left\|\pi_{x_1,y_1}A_{x_1,k_1}\right\|_s + \max_{x_1}\sum_{y_1}\left\|\pi_{x_1,y_1}B\right\|_s$
   $\forall y_1 : \sum_{x_1}\left\|\pi_{x_1,y_1}[A+B]\right\|_s = \sum_{x_1}\left\|\pi_{x_1,y_1}A + \pi_{x_1,y_1}B\right\|_s \leq \sum_{x_1}\left\|\pi_{x_1,y_1}A_{x_1,k_1}\right\|_s + \sum_{x_1}\left\|\pi_{x_1,y_1}B\right\|_s \leq \max_{y_1}\sum_{x_1}\left\|\pi_{x_1,y_1}A_{x_1,k_1}\right\|_s + \max_{y_1}\sum_{x_1}\left\|\pi_{x_1,y_1}B\right\|_s$
   Therefore,

$$\begin{aligned}
&\left\|[A+B]\right\|_s \\
&= \max\left\{\max_{x_1}\sum_{y_1}\left\|\pi_{x_1,y_1}[A+B]\right\|_s,\ \max_{y_1}\sum_{x_1}\left\|\pi_{x_1,y_1}[A+B]\right\|_s\right\} \\
&\leq \max\left\{\max_{x_1}\sum_{y_1}\left\|\pi_{x_1,y_1}A\right\|_s + \max_{x_1}\sum_{y_1}\left\|\pi_{x_1,y_1}B\right\|_s\ ,\right. \\
&\qquad\qquad \left.\max_{y_1}\sum_{x_1}\left\|\pi_{x_1,y_1}A\right\|_s + \max_{y_1}\sum_{x_1}\left\|\pi_{x_1,y_1}B\right\|_s\right\} \\
&\leq \max\left\{\max_{x_1}\sum_{y_1}\left\|\pi_{x_1,y_1}A\right\|_s,\ \max_{y_1}\sum_{x_1}\left\|\pi_{x_1,y_1}A\right\|_s\right\} \\
&\quad + \max\left\{\max_{x_1}\sum_{y_1}\left\|\pi_{x_1,y_1}B\right\|_s,\ \max_{y_1}\sum_{x_1}\left\|\pi_{x_1,y_1}B\right\|_s\right\} \\
&= \left\|A\right\|_s + \left\|B\right\|_s
\end{aligned}$$

4.

$$\begin{aligned}
\forall x_1 : \sum_{y_1} \left\|\pi_{x_1,y_1}[A \times B]\right\|_s &= \sum_{y_1} \left\| \sum_{k_1} \pi_{x_1,y_1} A \times \pi_{x_1,y_1} B \right\|_s \\
&\leq \sum_{y_1} \sum_{k_1} \left\|\pi_{x_1,y_1} A \times \pi_{x_1,y_1} B\right\|_s \\
&\leq \sum_{y_1} \sum_{k_1} \left\|\pi_{x_1,y_1} A\right\|_s \cdot \left\|\pi_{x_1,y_1} B\right\|_s \\
&= \sum_{k_1} \left\|\pi_{x_1,y_1} A\right\|_s \left(\sum_{y_1} \left\|\pi_{x_1,y_1} B\right\|_s \right) \\
&\leq \sum_{k_1} \left\|\pi_{x_1,y_1} A\right\|_s \left(\max_{k_1} \sum_{y_1} \left\|\pi_{x_1,y_1} B\right\|_s \right) \\
&= \max_{k_1} \sum_{y_1} \left\|\pi_{x_1,y_1} B\right\|_s \cdot \sum_{k_1} \left\|\pi_{x_1,y_1} A\right\|_s \\
&\leq \max_{x_1} \sum_{k_1} \left\|\pi_{x_1,y_1} A\right\|_s \cdot \max_{k_1} \sum_{y_1} \left\|\pi_{x_1,y_1} B\right\|_s
\end{aligned}$$

Similarly,
$\forall y_1 : \sum_{x_1} \left\|\pi_{x_1,y_1}[A \times B]\right\|_s \leq \max_{k_1} \sum_{x_1} \left\|\pi_{x_1,y_1} A\right\|_s \cdot \max_{y_1} \sum_{k_1} \left\|\pi_{x_1,y_1} B\right\|_s$.
Therefore,

$$\begin{aligned}
&\left\|[A \times B]\right\|_s \\
&= \max \left\{ \max_{x_1} \sum_{y_1} \left\|\pi_{x_1,y_1}[A \times B]\right\|_s, \ \max_{y_1} \sum_{x_1} \left\|\pi_{x_1,y_1}[A \times B]\right\|_s \right\} \\
&\leq \max \left\{ \max_{x_1} \sum_{y_1} \left\|\pi_{x_1,y_1} A\right\|_s \cdot \max_{x_1} \sum_{y_1} \left\|\pi_{x_1,y_1} B\right\|_s \right. , \\
&\qquad \left. \max_{y_1} \sum_{x_1} \left\|\pi_{x_1,y_1} A\right\|_s \cdot \max_{y_1} \sum_{x_1} \left\|\pi_{x_1,y_1} B\right\|_s \right\} \\
&\leq \max \left\{ \max_{x_1} \sum_{y_1} \left\|\pi_{x_1,y_1} A\right\|_s, \ \max_{y_1} \sum_{x_1} \left\|\pi_{x_1,y_1} A\right\|_s \right\} \\
&\quad \cdot \max \left\{ \max_{x_1} \sum_{y_1} \left\|\pi_{x_1,y_1} B\right\|_s, \ \max_{y_1} \sum_{x_1} \left\|\pi_{x_1,y_1} B\right\|_s \right\} \\
&= \left\|A\right\|_s \cdot \left\|B\right\|_s
\end{aligned}$$

■

The matrix norms $\|\cdot\|_1$, $|||\cdot|||_\infty$, $\|\cdot\|_m$, $\|\cdot\|_a$, and $\|\cdot\|_s$ are similar in their essence. Calculating $\|\cdot\|_1$, one adds sums of all lines. Thus also for that one the $|||\cdot|||_\infty$ takes as the maximal one. Therefore, the $\|\cdot\|_1$ gives always greater or equal values as $|||\cdot|||_\infty$. Similarly, calculating $\|\cdot\|_m$, or $\|\cdot\|_a$ one has to go also through the choice of $|||\cdot|||_\infty$, so these give also greater or equal value as $|||\cdot|||_\infty$. Since $\|\cdot\|_s$ tries all combinations of X, and Y it always gives the greatest value. The following graph shows the relationship between the mentioned matrix norms. Vectors direct from those which return smaller values to those which return greater values.

$$\begin{tikzcd}
 & \|\cdot\|_1 & \\
|||\cdot|||_\infty \arrow[ur] \arrow[r] \arrow[rdd] & \|\cdot\|_m \arrow[dr] & \\
 & & \|\cdot\|_s \\
 & \|\cdot\|_a \arrow[ur] &
\end{tikzcd}$$

Appendix D

Some Simple Lemmas

Here we present some lemmas used in several proofs, which are not related to the theory of provable security.

Lemma D.1 *Let x be a real number such that $0 < x < 1$, and d be an integer. Then*

$$(1-x)^d \geq 1 - dx.$$

Proof:

$$(1-x)^d = 1 - \binom{d}{1}x + \ldots \geq 1 - dx$$

■

Lemma D.2 *Let n and d be two positive integers such that $d < n$. Then*

$$\frac{n^{\underline{d}}}{n^d} \geq 1 - \frac{d^{\underline{2}}}{2n}.$$

Proof:

$$\frac{n^{\underline{d}}}{n^d} = \prod_{i=0}^{d-1} \frac{n-i}{n} = \prod_{i=1}^{d-1} \left(1 - \frac{i}{n}\right) \geq 1 - \sum_{i=1}^{d-1} \frac{i}{n} = 1 - \frac{d^{\underline{2}}}{2n}$$

■

Lemma D.3 *Let n and d be two positive integers such that $d < n$. Then*

$$\frac{n^{\underline{2d}}}{\left(n^{\underline{d}}\right)^2} \geq 1 - \frac{d^2}{n}.$$

Proof:

$$\frac{n^{\underline{2d}}}{\left(n^{\underline{d}}\right)^2} = \frac{(n-d)^{\underline{d}}}{n^{\underline{d}}} = \prod_{i=0}^{d-1} \frac{n-d-i}{n-i} = \prod_{i=1}^{d-1} \left(1 - \frac{d}{n-i}\right) \geq \left(1 - \frac{d}{n}\right)^d \geq 1 - \frac{d^2}{n}$$

■

Lemma D.4 *Let x be a real number such that $0 \leq x \leq 1$. Then*

$$x \geq 1 - e^{-x} \geq \left(1 - e^{-1}\right) x.$$

Proof: The exponential function is defined by the power series

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!},$$

and thus we have

$$e^{-x} = 1 - x + \sum_{n=2}^{\infty} \frac{(-1)^n x^n}{n!} \geq 1 - x$$

$$e^{-1}x = \sum_{n=0}^{\infty} \frac{(-1)^n x}{n!} = x + \sum_{n=1}^{\infty} \frac{(-1)^n x}{n!} \geq x + \sum_{n=1}^{\infty} \frac{(-1)^n x^n}{n!} = x - 1 + e^{-x}$$

■

Bibliography

[1] Mihir Bellare. Practice-oriented provable-security. In *Proceedings of First International Workshop on Information Security (ISW'97), Lecture Notes in Computer Science*, volume 1396. Springer-Verlag, 1998.

[2] Mihir Bellare, Anand Desai, E. Jokipii, and Phillip Rogaway. A concrete security treatment of symmetric encryption. In *IEEE Symposium on Foundations of Computer Science*, pages 394–403, 1997.

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

[4] Eli Biham and Adi Shamir. Differential cryptanalysis of DES-like cryptosystems. Technical Report CS90-16, The Weizmann Institute of Science, July 1990.

[5] Gilles Brassard. *Modern Cryptology*, volume 325 of *Lecture Notes in Computer Science*. Springer-Verlag, 1988.

[6] John Brillhart, D. H. Lehmer, J. L. Selfridge, Bryant Tuckerman, and Jr. S. S. Wagstaff. Factorizations of $b^n \pm 1$, $b = 2, 3, 5, 6, 7, 10, 11, 12$ up to high powers. `http://www.ams.org/online_bks/conm22/`, 2002.

[7] Valér Čanda. *Scalable Symmetric Ciphers Based on Group Bases*. PhD thesis, Institute of Experimental Mathematics, University of Essen, 2001.

[8] Valér Čanda and Tran van Trung. In *Tatracrypt 01*, volume 25, pages 39–66. Tatra Mountains Mathematical Publications, 2002.

[9] Okiok Data. Why migrating to triple-DES is not easy. `http://crypto.cs.mcgill.ca/~stiglic/Papers/tripleDES.pdf`, January 2002.

[10] Lenka Fibikova. Towards proper selection of primitives and modifications for a cryptographic scheme. In *Proceedings, Workshop "Santa's Crypto Get Together"*, pages 53–68, December 2002.

[11] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer Security*, 28(2):270–299, April 1984.

[12] Otokar Grošek, Karol Nemoga, and Ladislav Satko. A remark to Luby-Rackoff and Ueli M. Maurer pseudorandom generators. In *Tatra Mountains Mathematical Publications*, volume 20, pages 113–120. Tatra Mountains Mathematical Publications, 2000.

[13] Xuejia Lai and James L. Massey. A proposal for a new block encryption standard. In *Advances in Cryptology – EUROCRYPT '90, Lecture Notes in Computer Science*, volume 473, pages 389–404. Springer-Verlag, 1991.

[14] Michael Luby and Charles Rackoff. How to construct pseudo-random permutations from pseudorandom functions. In *Proceedings of CRYPTO '85*, pages 447–447. Springer-Verlag, 1986.

[15] Mitsuru Matsui. Linear cryptanalysis method for DES cipher. *Advanced in Cryptology — EUROCRYPT '93 Lecture Notes in Computer Science*, pages 386–397, 1994.

[16] Ueli M. Maurer. A simplified and generalized treatment of Luby-Rackoff pseudorandom permutation generator. In *Advances in Cryptology EUROCRYPT '92, Lecture Notes in Computer Science*, volume 658, pages 239–255, Berlin, 1992. Springer-Verlag.

[17] Shiho Moriai and Serge Vaudenay. Comparison of randomness provided by several schemes for block ciphers. In *Third AES Candidate Conference (AES3)*, `http://csrc.nist.gov/encryption/aes/round2/conf3/aes3papers.html`, March 2001.

[18] National Institute of Standards and Technology (NIST). FIPS 81: DES Modes of Operation. `http://csrc.nist.gov/publications/fips/`, December 1980.

[19] Bruce Schneier and John Kelsey. Unbalanced Feistel networks and block-cipher design. *Lecture Notes in Computer Science*, 1039:121–144, 1996.

[20] Claude Elwood Shannon. Communication theory of secrecy systems. In *Bell systems technical journal*, volume 28, pages 656–715, October 1949.

[21] Serge Vaudenay. Provable security for block ciphers by decorrelation. In *Symposium on Theoretical Aspects of Computer Science*, pages 249–275, 1998.

[22] Serge Vaudenay. On provable security for conventional cryptography. In *Information Security and Cryptology ICISC'99*, pages 1–16. Springer-Verlag, 1999.

[23] Serge Vaudenay. On the "Lai-Massey" scheme. In *Advances in Cryptology — ASIACRYPT '99*, volume 1716 of *Lecture Notes in Computer Science*, pages 8–19. Springer-Verlag, 1999.

[24] Serge Vaudenay. Resistance against general iterated attacks. In *Theory and Application of Cryptographic Techniques*, pages 255–271, 1999.

[25] Serge Vaudenay. Adaptive-attack norm for decorrelation and super-pseudorandomness. In *Selected Areas in Cryptography SAC '99, Lecture Notes in Computer Science*, volume 1758, pages 49–61. Springer-Verlag, 2000.

[26] Serge Vaudenay. Introduction to decorrelation theory (On-line manual). `http://lasecwww.epfl.ch/dec_manual.shtml`, June 2000.

[27] Yuliang Zheng, Tsutomu Matsumoto, and Hideki Imai. On the construction of block ciphers provably secure and not relying on any unproved hypotheses (Extended abstract). In *Advances in Cryptology — CRYPTO '89, Lecture Notes in Computer Science*, volume 435, pages 461–480. Springer-Verlag, 1989.

Index