Investigations in the Design and Analysis of Key-Stream Generators

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
Rector Magnificus, prof.dr. R.A. van Santen, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op 18 december 2003 om 16.00 uur

door

Alexander Kholosha

geboren te Lviv (Lemberg), Oekraïne

Dit proefschrift is goedgekeurd door de promotoren:

prof.dr.ir. H.C.A. van Tilborg

en

prof.dr. T. Helleseth

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Kholosha, Alexander

Investigations in the design and analysis of key-stream generators / by Alexander Kholosha. – Eindhoven : Technische Universiteit Eindhoven, 2003.
Proefschrift. – ISBN 90-386-0782-2
NUR 919
Subject headings : cryptology / Boolean functions / shift register sequences
2000 Mathematics Subject Classification : 94A60, 94C10, 94A55, 62P99, 11T71

Printed by Eindhoven University Press
Cover by JWL Producties

Contents

Chapter 1

Introduction to Secret-Key Cipher Systems

1.1 Basics of the Subject

According to Shannon [Sha49], a family of encryption transformations $E = \{E_k \mid k \in \mathcal{K}\}$, each mapping the space of plaintexts to the space of ciphertexts, is called a *cipher system* or *cipher* in short. Parameter k is called the *key* and it is taken from key space $\mathcal{K}$. For any $k \in \mathcal{K}$ encryption transformation E_k should be invertible, which guarantees that any ciphertext corresponds to the unique plaintext. The inverse E_k^{-1} is commonly denoted by D_k and called the decryption transformation. In secret-key, or symmetric, ciphers sender and receiver share the same key k and this key is the only thing needed to implement encryption and decryption of any message in the space of plaintexts and ciphertexts. Symmetric ciphers are still by far the most widely used in all applications where the main requirement is high security, high throughput capacity and low hardware complexity.

The space of plaintexts is made up of all various-length sequences over some finite alphabet. This alphabet, depending on a particular instance, can contain Latin letters only, letters plus decimal digits, all ASCII characters or can just consist of the two binary digits 0 and 1. Except for the classical ciphers, all currently used electronic cipher systems work with binary plaintexts. Thus, any plaintext message should be encoded prior to the encryption. For practical reasons, plaintext sequences after encoding are usually divided into blocks of a fixed size. An encryption transformation is then applied separately to each block to produce the corresponding block of the ciphertext. Thus, an encryption transformation can be viewed as a finite-state deterministic automaton having the sequence of plaintext blocks at its input and outputting the sequence of ciphertext blocks. The block size and the alphabet of plaintexts and ciphertexts are usually the same. Therefore, being invertible, any encryption and decryption transformations implement one-to-one mappings and the length of an encoded message is not changed due to encryption.

When an encryption transformation implements a memoryless automaton, i.e., is a mapping, then equal plaintext blocks are mapped onto equal ciphertext blocks. In this case we are speaking about a block cipher in the Electronic Codebook (ECB) mode. The block size has to be large enough to prevent elementary methods of frequency cryptanalysis that are used to break simple substitution ciphers. In all the other cases, when encryption is made using a varying (i.e., time- or memory-dependent) transformation, we should speak about a stream cipher. Thus, the formal distinction between block and stream ciphers is the presence of memory in the encryption transformation.

According to this classification, all feedback modes of block ciphers such as Cipher Block Chaining (CBC), Cipher Feedback (CFB) and Output Feedback (OFB) modes (see [FIP81]) should be considered as stream ciphers. Due to the fact that block ciphers are rarely used in their original ECB mode of operation and all the other modes implement memory-dependent transformations, almost all practical encryption systems can be formally classified as stream ciphers. However, the true distinction of block and stream ciphers is rather based on completely different design principles and analysis techniques adopted for these two classes of symmetric ciphers. For instance, the security of CBC, CFB and OFB modes of operation solely relies on the properties of the underlying block cipher. This is the reason for considering these modes in the theory of block ciphers.

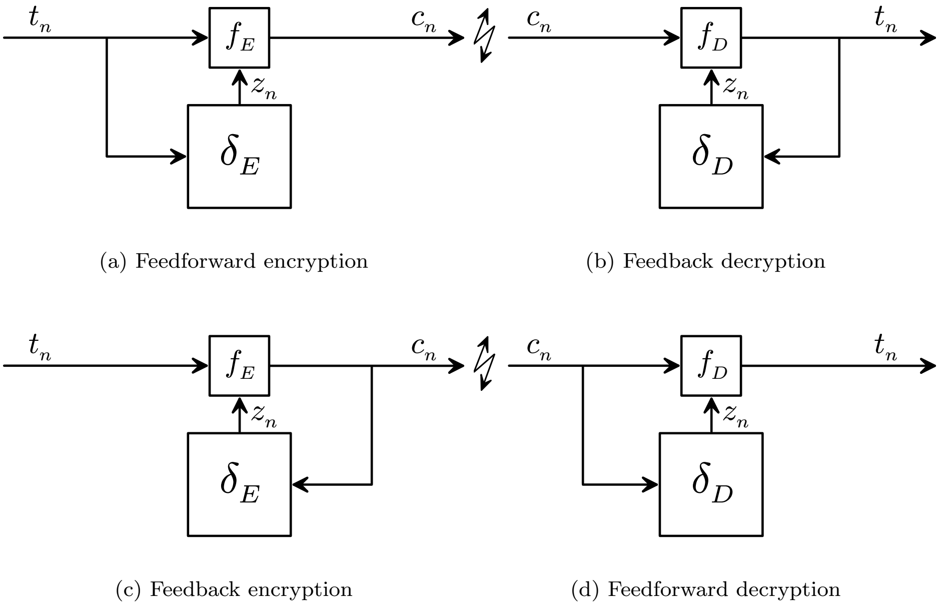
Unlike block ciphers, the block length for a stream cipher can be small or even equal to one. In this case, the encryption transformation is applied to the characters (or bits) of the plaintext. This thesis is focused mostly on stream ciphers.

It is our conviction that designing a secure and efficient stream cipher is a rather difficult task. We are basing this belief on the latest results of the NESSIE project [NES03] regarding final selection of cryptographic algorithms, where no stream cipher has been selected, since none of the six submitted ones met all the stringent security requirements.

1.2 Stream Ciphers and their Security

In what follows we give a formal definition of encryption and decryption automata that implement a stream cipher. Consider a plaintext $t = \{t_n\}_{n \geq 0}$ divided into blocks and an arbitrary key k from the key space and generate the corresponding ciphertext $c = \{c_n\}_{n \geq 0}$. This means that the plaintext and the ciphertext are related to each other by identities $E_k(t) = c$ and $D_k(c) = t$. When processing sequence t with the encryption automaton or sequence c with the decryption automaton, both automata run through the same sequence of internal states $z = \{z_n\}_{n \geq 0}$ and their output functions satisfy the identities $f_E(t_n, z_n) = c_n$ and $f_D(c_n, z_n) = t_n$. For these equalities to hold it is necessary and sufficient for function f_E to be an injective mapping of the first argument for any fixed value of the second one. The output function of both the encryption and the decryption automata is usually called a *mixer*. After applying the output function, the transition function δ updates the internal state of the automaton taking the current state and the input block as the arguments, i.e., $\delta_E(t_n, z_n) = z_{n+1}$ and $\delta_D(c_n, z_n) = z_{n+1}$. Note that key k may determine the initial state of the automata and the concrete type of the transition and output functions, but in other cases the initial state may be publicly known or be transmitted in plaintext along with the ciphertext.

By the transition function, current internal state z_n of the encryption automaton is defined by previous state z_{n-1} and input t_{n-1}. In the same way, z_{n-1} is defined by z_{n-2} and t_{n-2}, etc. Thus, in general, z_n is a function of initial state z_0 and all previously processed plaintext blocks $t_0, \ldots, t_{n-1}$. If, due to this dependence, output value c_n of the ciphertext depends not only on current input t_n but also on $t_0, \ldots, t_{n-1}$ then we are speaking about feedforward encryption (see Fig. 1.1(a)). The corresponding decryption automaton has to run through the same sequence of internal states and, therefore, its current state will depend on the output value of plaintext blocks. Such a system is called the feedback decryption (see Fig. 1.1(b)). If output value c_n of the ciphertext is a function of t_n and previously generated ciphertext blocks $c_0, \ldots, c_{n-1}$ (thus, there is no direct dependence on previously processed plaintext blocks $t_0, \ldots, t_{n-1}$) then we are speaking about the feedback encryption (see Fig. 1.1(c)). The corresponding decryption should implement a feedforward system (see Fig. 1.1(d)).



(a) Feedforward encryption

(b) Feedback decryption

(c) Feedback encryption

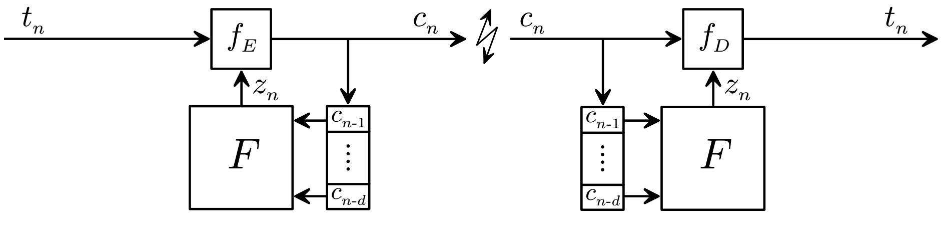
(d) Feedforward decryption

Figure 1.1: Stream cipher with feedforward (feedback) dependencies

Current internal state z_n and current output value c_n of a feedforward (feedback) encryption system can in general essentially depend on *all* the previously processed plaintext blocks (generated ciphertext blocks). If no additional restrictions are imposed on this dependence then both feedforward and feedback encryption systems are highly impractical because any single error in the communication channel causes an infinite number of errors in the decrypted text. Therefore, in practical implementations of encryption systems the dependence is usually restricted to some finite number of plaintext or ciphertext blocks. This means that for some $d > 0$, current internal state z_n is completely determined by the d previously processed plaintext blocks $t_{n-d}, \ldots, t_{n-1}$ in the feedforward mode and the d ciphertext blocks $c_{n-d}, \ldots, c_{n-1}$ in the feedback mode (see Fig. 1.2(a)). Initial state z_0 defines the initial content of d memory cells.

Consider a feedback encryption system with the dependence restricted to d ciphertext blocks and assume that some block was transmitted with error to the decryption side. Since the internal state of the decryption automaton is estimated as a function of d previously received ciphertext blocks, the current state will be estimated incorrectly as long as the erroneous ciphertext block is fed to an argument of this function (see Fig. 1.2(b)). Therefore, one block will be decrypted with an error due to the incorrect ciphertext and d subsequent blocks will be erroneous due to the incorrectly estimated internal state. This means that any single error in the communication channel causes at most $d + 1$ errors in the decrypted text. Such a stream cipher is said to have a finite error propagation equal to $d + 1$ and is called *asynchronous* or *self-synchronizing.* The decryption side of this cipher automatically re-synchronizes after d consecutive correct ciphertext blocks are received. An example of an asynchronous stream cipher is a block cipher used in CBC or CFB modes.

Note that a feedforward encryption system even with a restricted dependence will still have an error propagation equal to infinity. Indeed, as soon as any erroneous ciphertext block is received by the feedback decryption automaton, the current and all the subsequent outputs will be incorrect since the arguments of the current state function (plaintext blocks) will always be with errors. That is why such systems are not considered in practice.

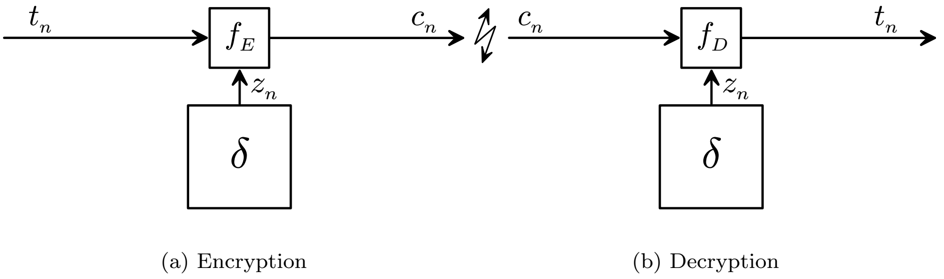


(a) Feedback encryption

(b) Feedforward decryption

Figure 1.2: Asynchronous stream cipher (F is a memoryless function)

In many encryption systems current output value c_n of the ciphertext does not depend on earlier processed plaintext $t_0, \ldots, t_{n-1}$. In this case we can assume that the next state of such an automaton depends only on the previous state and not on the input and thus the transition function can be presented as $\delta(z_{n-1}) = z_n$ both for encryption and decryption automata. Such a stream cipher is called *synchronous*. It can be decomposed into the autonomous part generating the sequence of internal states called the *key-stream generator* and the memoryless mixer implementing the output function (see Fig. 1.3). The generated sequence of states z that controls the mixer is called the *key stream*. An example of a synchronous stream cipher is a block cipher used in OFB mode. Synchronous stream ciphers provide no error propagation but in order to get the correct plaintext on the receiver side, encryption and decryption automata have to be perfectly synchronized. Therefore, additional mechanisms are needed for detecting lost synchronization and successive re-synchronization of the devices. One of the most common mixers used in synchronous stream ciphers is modular addition. In such a system both plaintext blocks and internal states of a key-stream generator are represented by non-negative integers in the same range (say, up to M) and the encryption function is of the form $f_E(t_n, z_n) = t_n + z_n \pmod M$.



(a) Encryption (b) Decryption

Figure 1.3: Synchronous stream cipher

It follows from Shannon's fundamental theorem on information-theoretic cryptography [Sha49] that the amount of information that can be sent absolutely secure over a public channel is at most as much as the information present in the key. Therefore, the only perfectly secure cipher system is the one-time-pad that corresponds to an additive mixer with a purely random key stream. Unfortunately, synchronous stream ciphers in practice have to use a key stream that is far from being purely random. This explains the fact that the basic problem in the design of key-stream generators is to define reasonable randomness criteria such that breaking a cipher which uses a key stream satisfying these criteria would involve infeasible calculations. This type of security is usually called practical security.

Moreover, an even bigger challenge is to build an efficient generator able to produce a key stream with the desired randomness characteristics. On one hand, if the key stream is generated efficiently then it can be described in some simple way, while any random sequence should not allow such a characterization. On the other hand, finding this simple description could mean breaking the key-stream generator while that should be computationally infeasible.

Statistical tests of randomness are the most natural criteria for checking whether the statistical properties of a generated sequence agree with those of the sequence consisting of uniform, independent and identically distributed random variables. Since any finite-state automaton outputs a sequence that is eventually periodic, we shall from now on assume that the key-stream sequence is binary with period p. Any p consecutive elements of such a sequence are called a cycle. Any segment consisting of repeated zeros or ones which is neither preceded nor succeeded by the same symbol is called a run. The classical randomness criteria for a periodic binary key stream are Golomb's postulates [Gol67, p. 25] which are the following.

1. In every cycle the difference between the number of ones and the number of zeros is at most 1.

2. In every cycle half of the runs have length one, one-fourth have length two, one-eighth have length three, etc., as long as the number of runs so indicated exceeds 1. Moreover, for each of these lengths there are (almost) equally many runs of zeros and of ones.

3. The auto-correlation function [LN83, pp. 463-464] is constant for all phase shifts that are not a multiple of p.

The basic property of purely random sequences is unpredictability. Note that an output sequence of a maximum-length Linear Feedback Shift Registers (LFSR) satisfies Golomb's postulates (see [HK98] and [HK99, Sect. 8.2]) but is easily predictable when using the Berlekamp-Massey algorithm [Mas69, vT00]. Therefore, being necessary for a periodic pseudo-random sequence to *look random*, these postulates are anything but sufficient for this sequence to *be considered* random. Many additional statistical tests have been proposed in order to improve the randomness criteria (see [MvOV97, Sect. 5.4] for an overview).

Different notions of security and different assumptions about the cryptanalytic context result into four principal approaches to the design of stream ciphers [Rue92, p. 68]. The *information-theoretic* approach deals with Shannon's notion of security. When following the *system-theoretic* approach the designer's goal is to make sure that none of the currently known basic cryptanalytic principles are applicable to the cipher system. The *complexity-theoretic* approach means that security of a stream cipher is based on some computationally infeasible problem that cannot be solved in polynomial time. Finally, *randomized* stream ciphers ensure that for breaking the system the cryptanalyst has to examine an infeasible amount of data. Key-stream generators are studied in this thesis following the system-theoretic approach.

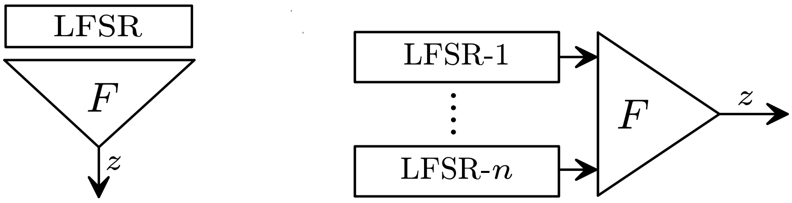
Virtually all of the currently known attacks on stream ciphers are based on the following cryptanalytic principles: substitution and approximation, divide and conquer of the key space, and statistical analysis. In order to prevent these attacks, a number of general design criteria for key-stream generators have been formulated.

The most important criteria that are necessary but not sufficient for the security of a key-stream generator are the following:

1. Long period.
2. Linear and nonlinear complexity profiles resembling the profiles (often called typical) of a purely random sequence.

   Note 1.1 According to [Rue86, p. 33], by the complexity profile we mean the dynamic behavior of the linear (nonlinear) complexity as a function of the number of bits processed. The typical linear complexity grows approximately as $n/2$. The behavior of nonlinear (maximum order) complexity of a random sequence was studied in [Jan89]. In general, estimating the nonlinear complexity is computationally hard, while methods for estimating the linear complexity are quite developed at the moment.

3. Statistical uniformity.
4. Confusion - every key stream symbol depends on all or on most of the key bits.
5. Logical functions (filtering and combining) satisfying the relevant security criteria (see Chap. 2).



(a) Nonlinear filter generator (b) Nonlinear combination generator

Figure 1.4: Basic key-stream generators

Most of the well-known key-stream generators are built following the system-theoretic approach and are based on LFSR's (see [vT00, Sect. 3.2]). The idea to use LFSR's for generating pseudo-random sequences dates back to the work of Zierler [Zie59]. Their practical significance is based on the fact that LFSR's allow efficient hardware implementation and produce sequences with a large period and good statistical properties (if the feedback polynomial is chosen to be primitive). Unfortunately, inherent linearity of these sequences results in low linear complexity that prevents immediate use as a key stream. Three basic schemes were suggested to increase the linear complexity preserving at the same time a large period and good statistical properties of the LFSR output sequence [Rue86]. These are the *nonlinear filter generator* (see Fig. 1.4(a)), the *nonlinear combination generator* (see Fig. 1.4(b)) and the *clock-controlled arrangement* (see Fig. 3.1 on page 50). These designs are usually combined in a key-stream generator to produce sequences with higher complexity and higher security.

Today, the question of security in the system-theoretic sense for stream ciphers is far from being settled. This area of cryptography is an excellent example of how the development of new cryptanalytic techniques stimulates new research and, in particular, how it changes our understanding of what security for stream ciphers means. Current security requirements for stream ciphers simply reflect the currently known cryptanalytic attacks. If, after thorough cryptanalysis, it seems that the exhaustive search of an effective key space is the only way to break the cipher then the cipher is often accepted as being secure.

Most of the attacks on stream ciphers follow the known-plaintext scenario. For synchronous stream ciphers this means that an attack is mounted on a key-stream generator when a segment of the generated sequence is known. Attacks on key-stream generators can be divided into *key-recovery attacks* where the secret key (or part of it) is targeted and *distinguishing attacks* where the goal is to distinguish the key stream from a purely random sequence. One of the most efficient classes of key-recovery attacks that can be applied to all synchronous ciphers are (*fast*) *correlation attacks.* Other key-recovery attacks that exploit the inherent linearity of LFSR's underlying a key-stream generator, are usually referred to as *algebraic attacks.* Attacks that exploit statistical irregularities in a key stream are called *statistical attacks.*

1.3 Outline of the Thesis

In the thesis we address some important problems in the design and analysis of key-stream generators for stream ciphers. Traditionally, all research in this area is carried out along two major directions.

The first direction is focused on the building blocks (e.g., feedback shift registers, logical functions, modulo N arithmetic, etc.) that constitute the generator, and on estimating the related number-theoretical characteristics of the key stream. Doing this, the main objective is to find methods for generating key-stream sequences with characteristics that provide security against algebraic attacks. Following this approach, we analyze combining and filter functions in Chap. 2 and clock-controlled LFSR's in Chap. 3.

A tensor transform introduced in Chap. 2 proves to be helpful when analyzing the security of Boolean and multivalued logical functions in cryptographic applications. Special cases of this approach not only provide easy proofs for known relations in the theory of algebraic normal, arithmetic and Walsh transforms, but also lead to some new properties of these transforms. We also propose a new type of tensor transform, the so-called probabilistic transform, giving an important insight into certain probabilistic properties of Boolean functions. Another new type of tensor transform that we propose, is the weight transform. It relates a Boolean function to the weights of its subfunctions.

In Chap. 2 we also study correlation properties of Boolean functions. We show how correlation coefficients that provide an estimate for correlation dependencies of a Boolean function, can be obtained from its weight transform. We demonstrate that the number of fixed-order product terms in the Algebraic Normal Form of a balanced Boolean function depends on its correlation coefficients. We prove that highly resilient Boolean functions cannot be approximated by a function that is nondegenerate on few variables. We also introduce in this chapter a polynomial that estimates the bias of the output distribution as a function of the input biases. The coefficients of this polynomial can be obtained by means of the probabilistic transform. Further, we suggest a characteristic for balanced Boolean functions that measures their ability to compensate a nonuniform distribution of the inputs. Resilient functions are proved to have good compensating qualities.

The other building block being analyzed in the context of the first direction is a clock-controlled LFSR. In Chap. 3 we estimate the period of its output sequence when the feedback polynomial is irreducible and the structure of the control sequence is arbitrary. A sufficient condition for this period to reach its maximal value is formulated. Some specific configurations of clock-controlled arrangements with a maximal period of the output sequence are described. Relevant recommendations for estimating the linear complexity are given. We also formulate the rules that have to be observed when constructing a clock-controlled arrangement in order to provide a close-to-uniform element distribution in the output sequence.

Moving slightly away from building blocks, in Chap. 3 we also construct a key-stream generator based on the one suggested by Geffe. Unlike the Geffe generator that has three binary input m-sequences, this generator runs over the field GF(q) and combines multiple inputs having arbitrary periods. In particular, this implies that clock-controlled shift registers can be used as inputs. The original Geffe generator cannot be used for secure key-stream generation since its combining function is zero-order correlation immune and correlation attacks can easily be launched. Using clock-controlled registers and multiple inputs makes the new generator immune against fast correlation attacks and less susceptible to basic attacks. We analyze some relevant algebraic properties of the suggested generator.

The second direction in the design and analysis of key-stream generators on which we focus are the statistical properties of a key stream. Following this approach, in Chap. 4 we develop several attacks that exploit statistical weakness in the key stream. Our first algorithm uses statistical tests based on invariant statistics. It tests a key stream for a linear recurrence perturbed with a nonuniform additive noise. For the particular case of trinomial feedback we construct a couple of invariant statistics that allow construction of computationally feasible tests.

Our second algorithm tests a ciphertext for key-stream reuse. We construct nonrandomized and randomized most powerful tests that efficiently distinguish families consisting of up to four ciphertexts obtained from different plaintexts but using the same key-stream segment. Moreover, we provide explicit algorithms for constructing parameter intervals, where these tests are uniformly most powerful.

If the cryptanalyst is dealing with blocks of key-stream or ciphertext then it may be helpful to use statistical procedures for selecting the most probable outcomes from the multinomial population. We construct new procedures that are based on the calculation of reduced frequencies. That makes them more efficient when the total number of outcomes is big compared to the amount of memory available. Using these procedures we can detect possible statistical irregularities in a generated sequence of key-stream or ciphertext blocks. If such an irregularity is detected then this can be used as a basis for the distinguishing attack on the cipher. Another useful applications can be found in frequency analysis, namely, where it is a part of a dictionary attack on ciphers and various other attacks on codes. We prove the limit theorem for the distribution of reduced frequencies.

Results presented in Chap. 2 appeared in [KvT02, Kho03], results of Chap. 3 appeared in [Kho01], and Chap. 4 in the part concerning multinomial selection procedures was published in [Kho98a].

Chapter 2

Tools for Analyzing the Security of Logical Functions Used in Key-Stream Generators

2.1 Introduction

Among the basic schemes for key-stream generators are the nonlinear filter generator and the nonlinear combination generator (see Fig. 1.4 and [Rue86]). They correspond respectively to a nonlinear transformation applied to several phases of the same linear feedback shift register (LFSR) or to the outputs of several independent LFSR's. The nonlinear transformation can be represented by a Boolean (in general, multivalued) logical function and the security of the key-stream generators heavily relies on the specific qualities of this function. If the function is not chosen properly then the whole system is susceptible to different types of correlation [Jö02] and algebraic [DXS91] attacks.

It is currently generally accepted that secure functions to be used in key-stream generators must meet the following requirements: balancedness, high nonlinearity, fulfilling the Strict Avalanche Criterion, sufficiently high algebraic degree that should hold for each individual variable and should be optimized with respect to certain correlation properties. These conditions are necessary, although it is not clear if they are sufficient to resist all kinds of attacks. The algebraic degree of a multivalued function is the degree of its Algebraic Normal Form (ANF); balancedness, nonlinearity, avalanche and correlation properties are defined by its Walsh transform [XM88, CC99, For89]; the Numerical Normal Form (NNF), that can be estimated via the arithmetic transform, characterizes probabilistic properties of Boolean functions (see Sects. 2.2, 2.5 and [KB81, CG99] for the details). Thus, the algebraic normal, arithmetic and Walsh transforms of a logical function define the most important cryptographic characteristics of the function. These are all the transforms that have been extensively studied and have been used in cryptography. The objective of this chapter is to generalize the known transforms and to develop the new ones which would provide efficient means for analyzing the security of logical functions. We understand that any useful transform applied to a logical function, firstly, should produce data giving immediate characterization of some important cryptographic properties of the function and, secondly, should allow efficient estimation.

In Sect. 2.2 we describe the general basis for a tensor transform of pseudo-Boolean functions by which we mean any function of Boolean variables taking on its values in an arbitrary field. Special cases of this approach not only provide easy proofs for known relations in the theory of algebraic normal, arithmetic and Walsh transforms, but also lead to some new properties of these transforms. We also propose a new type of tensor transform, the so-called probabilistic transform, giving an important insight into certain probabilistic properties of Boolean functions that are discussed in Sect. 2.5. Another new type of tensor transform that we propose, is the weight transform. It relates a Boolean function to the weights of its subfunctions. It is proved that coefficients of the ANF of a Boolean function depend on the values contained in its binary weight transform for the zero-valued vector.

Interrelated algebraic and correlation properties of Boolean functions are investigated in Sect. 2.3. We suggest that not-correlation-immune Boolean functions can still be cryptographically secure if only slight dependence between input bits and the output is allowed. We show how correlation coefficients that provide an estimate for correlation dependencies of a Boolean function, can be obtained from its weight transform. We demonstrate that the number of fixed-order product terms in the ANF of a balanced Boolean function depends on its correlation coefficients. We also prove that highly resilient Boolean functions cannot be approximated with a function that is nondegenerate on few number of variables.

In Sect. 2.4 we generalize a tensor transform of pseudo-Boolean functions (that was discussed in Sect. 2.2) to the multivalued case. This transform covers algebraic normal and Walsh transforms of functions over GF(q). We also give a new proof for the spectral characterization of high-order correlation immune functions over a finite field. These results appeared in [Kho03].

The arrangement with a Boolean function combining random inputs that are distributed nonuniformly with some known biases is considered in Sect. 2.5. We introduce a polynomial that estimates the bias of the output distribution as a function of input biases. The coefficients of this polynomial can be obtained by means of the probabilistic transform. Further, we suggest a characteristic for balanced Boolean functions that measures their ability to compensate a nonuniform distribution of the inputs. Resilient functions are proved to have good compensating qualities. The extended abstract of Sects. 2.2 and 2.5 appeared in [KvT02].

In Sect. 2.6 we apply the known idea of equivalence relation of Boolean functions under a transformation group to facilitate the problem of estimating the number of functions that fulfil a relevant set of security criteria. This approach is also helpful when checking whether a design criterion remains invariant under some "weak" transformations. We prove invariance of some important cryptographic characteristics of Boolean functions under weak transformations.

2.2 Tensor Transform of Pseudo-Boolean Functions

Let $M_n(P)$ denote the ring of n-dimensional square matrices over the field P. For a pair of matrices $A \in M_n(P)$ and $B \in M_m(P)$ let $A \otimes B$ denote the Kronecker product [MS96, p. 421] of these matrices and $A^{[k]}$ denote the kth Kronecker power of A. For a matrix $A \in M_{2^n}(P)$ we use notation $A = (\boldsymbol{g}_0, \ldots, \boldsymbol{g}_{2^n-1})$, where $\boldsymbol{g}_i$ $(i = 0, \ldots, 2^n - 1)$ denotes the ith column of A and the coordinates of $\boldsymbol{g}_i$ are indexed lexicographically by the elements in $\{0,1\}^n$, so

$$\boldsymbol{g}_i = \begin{pmatrix} g_i(0,\ldots,0) \\ g_i(0,\ldots,1) \\ \vdots \\ g_i(1,\ldots,1) \end{pmatrix} .$$

Let $\alpha_i = (\alpha_i^1, \ldots, \alpha_i^n)$ $(i = 0, \ldots, 2^n - 1)$ denote the n-bit binary expansion of i, where the leftmost bit is the most significant. Then $\boldsymbol{g}_i = (g_i(\alpha_0), \ldots, g_i(\alpha_{2^n-1}))^T$, where the superscript T denotes transpose of a matrix.

Lemma 2.1 *Let $A = (\boldsymbol{g}_0, \ldots, \boldsymbol{g}_{2^n-1}) \in M_{2^n}(P)$ and $A' = (\boldsymbol{g}'_0, \ldots, \boldsymbol{g}'_{2^{n-1}-1}) \in M_{2^{n-1}}(P)$. Suppose that $A = B \otimes A'$ for some matrix $B = \begin{pmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{pmatrix}$. Then for any $i \in \{0, \ldots, 2^n - 1\}$*

$$g_i(x_1, \ldots, x_n) = \begin{cases} (b_{00}\overline{x_1} + b_{10}x_1)g'_{i'}(x_2, \ldots, x_n), & \textit{if} \quad \alpha_i = (0, \alpha_{i'}), \\ (b_{01}\overline{x_1} + b_{11}x_1)g'_{i'}(x_2, \ldots, x_n), & \textit{if} \quad \alpha_i = (1, \alpha_{i'}), \end{cases}$$

where $\alpha_{i'}$ is the $(n-1)$-bit vector, binary expansion of i'.

Proof: By the definition of the Kronecker product, $A = \begin{pmatrix} b_{00}A' & b_{01}A' \\ b_{10}A' & b_{11}A' \end{pmatrix}$. Thus,

$$\begin{aligned} g_i(0, x_2, \ldots, x_n) &= \begin{cases} b_{00}g'_{i'}(x_2, \ldots, x_n), & \text{if} \quad \alpha_i = (0, \alpha_{i'}), \\ b_{01}g'_{i'}(x_2, \ldots, x_n), & \text{if} \quad \alpha_i = (1, \alpha_{i'}) \end{cases} \quad \text{and} \\ g_i(1, x_2, \ldots, x_n) &= \begin{cases} b_{10}g'_{i'}(x_2, \ldots, x_n), & \text{if} \quad \alpha_i = (0, \alpha_{i'}), \\ b_{11}g'_{i'}(x_2, \ldots, x_n), & \text{if} \quad \alpha_i = (1, \alpha_{i'}). \end{cases} \end{aligned}$$

These equations combined together prove the claimed identity. □

The following proposition easily follows from Lemma 2.1.

Proposition 2.2 *Let $A = B_1 \otimes \cdots \otimes B_n$, where $B_j = \begin{pmatrix} b_{00}^{(j)} & b_{01}^{(j)} \\ b_{10}^{(j)} & b_{11}^{(j)} \end{pmatrix}$ for $j = 1, \ldots, n$, and $A = (\boldsymbol{g}_0, \ldots, \boldsymbol{g}_{2^n-1})$. Then for any $i \in \{0, \ldots, 2^n - 1\}$*

$$g_i(x_1, \ldots, x_n) = \prod_{j=1}^{n} \left(\overline{\alpha_i^j} \left(b_{00}^{(j)}\overline{x_j} + b_{10}^{(j)}x_j \right) + \alpha_i^j \left(b_{01}^{(j)}\overline{x_j} + b_{11}^{(j)}x_j \right) \right) .$$

Let $A \in M_{2^n}(P)$ be an invertible matrix and $A = (\boldsymbol{g}_0, \ldots, \boldsymbol{g}_{2^n-1})$. Further, let pseudo-Boolean function $f(x_1, \ldots, x_n)$, mapping $\{0,1\}^n$ to P, be defined by its string of values $T^f = (f(\alpha_0), \ldots, f(\alpha_{2^n-1}))^T \in P^{2^n}$ and function $F(x_1, \ldots, x_n)$ be defined by the string of values $T^F = A^{-1}T^f = (F(\alpha_0), \ldots, F(\alpha_{2^n-1}))^T \in P^{2^n}$. Vectors T^f and T^F are considered further as column-vectors. Then $T^f = AT^F$,

$$T^f = \sum_{i=0}^{2^n-1} \boldsymbol{g}_i F(\alpha_i) \quad \text{and} \quad f(x_1, \ldots, x_n) = \sum_{i=0}^{2^n-1} g_i(x_1, \ldots, x_n) F(\alpha_i) \tag{2.1}$$

for any $(x_1, \ldots, x_n) \in \{0,1\}^n$. Equations (2.1) represent the decomposition of function f with respect to the basis vector set $(\boldsymbol{g}_0, \ldots, \boldsymbol{g}_{2^n-1})$. We will say that functions f and F are related by a *tensor transform* if matrix A of the linear transform is equal to the Kronecker product of n elementary cells of size 2 × 2.

Hereafter, by $f_{i_1,\ldots,i_m}^{\beta_1,\ldots,\beta_m}$ for any $1 \leq i_1 < \cdots < i_m \leq n$, we denote the subfunction of f obtained by fixing variables $x_{i_1}, \ldots, x_{i_m}$ with binary values $\beta_1, \ldots, \beta_m$ respectively. By $wt(\omega)$ we also denote the Hamming weight of a binary string ω and by $wt(f)$ we denote the Hamming weight of a Boolean function f, i.e., the weight of T^f.

It is well known that if B_1 and B_2 are invertible matrices over P then the Kronecker product matrix $B_1 \otimes B_2$ is invertible too and $(B_1 \otimes B_2)^{-1} = B_1^{-1} \otimes B_2^{-1}$. In particular, if $B \in M_m(P)$ is an invertible matrix and $A = B^{[n]}$ then A is invertible too and $A^{-1} = (B^{-1})^{[n]}$.

Now we will demonstrate how Proposition 2.2 substantially facilitates proving some important matrix identities for various representations of pseudo-Boolean functions. By convention, for a Boolean variable x we assume that $x^0 = \overline{x}$ and $x^1 = x$.

The Identity Transform.

Let P be an arbitrary field and set $B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and $A = B^{[n]}$, where 0 and 1 are zero and identity elements of P respectively. Then, by Proposition 2.2,

$$g_i(x_1, \ldots, x_n) = \prod_{j=1}^{n} \left(\overline{\alpha_i^j}\,\overline{x_j} + \alpha_i^j x_j \right) = \prod_{j=1}^{n} x_j^{\alpha_i^j} = x_1^{\alpha_i^1} \cdot \cdots \cdot x_n^{\alpha_i^n}$$

and

$$\begin{aligned} f(x_1, \ldots, x_n) &\overset{(2.1)}{=} \sum_{i=0}^{2^n-1} g_i(x_1, \ldots, x_n) F(\alpha_i) = \\ &= \sum_{i=0}^{2^n-1} \left(x_1^{\alpha_i^1} \cdot \cdots \cdot x_n^{\alpha_i^n} \right) F(\alpha_i) = F(x_1, \ldots, x_n) \ . \end{aligned}$$

The Algebraic Normal and Arithmetic Transforms.

Let P be an arbitrary field and set $B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ and $A = B^{[n]}$, where 0 and 1 are zero and identity elements of P respectively. Then, by Proposition 2.2,

$$g_i(x_1, \dots, x_n) = \prod_{j=1}^{n} \left(\overline{\alpha_i^j} + \alpha_i^j x_j \right) = \prod_{j=1,\dots,n:\, \alpha_i^j = 1} x_j \tag{2.2}$$

and

$$f(x_1, \dots, x_n) \overset{(2.1)}{=} \sum_{i=0}^{2^n-1} g_i(x_1, \dots, x_n) F(\alpha_i) = \sum_{i=0}^{2^n-1} \left(\prod_{j=1,\dots,n:\, \alpha_i^j = 1} x_j \right) F(\alpha_i) \ .$$

If we take $P = \mathrm{GF}(2)$ then one can easily recognize the Algebraic Normal Form (ANF) of Boolean function f on the right hand side of the last identity, where $F(\alpha_i)$ $(i = 0, \dots, 2^n - 1)$ are the coefficients of the ANF polynomial. Let P^f denote the coefficient vector of the ANF polynomial for function f and take also $R_2 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} = R_2^{-1}$, $R_{2^n} = R_2^{[n]}$. Then

$$T^f = R_{2^n} P^f \quad \text{and} \quad P^f = R_{2^n} T^f \ . \tag{2.3}$$

This transform of Boolean function f is called the *algebraic normal transform* and was introduced in [Jan89, Sect. 4.2].

On the other hand, basis vector set (2.2) for the algebraic normal transform over GF(2) can be expressed in the following form:

$$g_i(x_1, \dots, x_n) = \begin{cases} 1, & \text{if} \quad \alpha_i^j \le x_j \ (j = 1, \dots, n), \\ 0, & \text{otherwise} \ . \end{cases}$$

Therefore, by (2.3),

$$f(\boldsymbol{x}) = \sum_{i=0,\dots,2^n-1:\, \alpha_i \preceq \boldsymbol{x}} P_f(\alpha_i) \quad \text{and} \quad P_f(\alpha_i) = \sum_{\boldsymbol{x}=0,\dots,2^n-1:\, \boldsymbol{x} \preceq \alpha_i} f(\boldsymbol{x}) \ ,$$

where $\boldsymbol{x} = (x_1, \dots, x_n)$, $\preceq$ is the partial ordering on the Boolean lattice (defined as $\alpha \preceq \boldsymbol{x}$ if and only if $\alpha^j \le x_j$ for $j = 1, \dots, n$) and $P_f(\alpha_i)$ is the ith coefficient of the ANF of function f. In the sum over $\boldsymbol{x}$ the summation index is considered as an integer in the range $0, \dots, 2^n - 1$ but written in its binary expansion. The latter identity for ANF coefficients can be found in [MS96, p. 372].

If R_{2^n} is considered as a matrix over the real number field $\mathbf{R}$ and the algebraic normal transform of a Boolean function f is implemented over $\mathbf{R}$ then T^F is equal to the coefficient vector of a real-valued, exponent-free (in variables) polynomial of n variables with integer coefficients that takes on the same values as function f on the points from $\mathrm{GF}(2)^n$. Let Π_f denote the coefficient vector of such a polynomial. In this case $R_2^{-1} = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}$, $R_{2^n}^{-1} = (R_2^{-1})^{[n]}$,

$$T^f = R_{2^n}\Pi^f \quad \text{and} \quad \Pi^f = R_{2^n}^{-1}T^f \ . \tag{2.4}$$

According to [CG99], this real-valued polynomial is called the *Numerical Normal Form* of f. Corresponding transform (2.4) was introduced in [KB81] and is commonly called the *arithmetic transform.* Using Proposition 2.2 for $B = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}$ and $A = B^{[n]}$ we obtain

$$\begin{aligned} g_i(x_1,\ldots,x_n) &= \prod_{j=1}^{n}\left(\overline{\alpha_i^j}(\overline{x_j}-x_j)+\alpha_i^j x_j\right) = \\ &= \begin{cases} (-1)^{wt(x_1,\ldots,x_n)-wt(\alpha_i)}, & \text{if} \quad \alpha_i^j \le x_j \ (j=1,\ldots,n), \\ 0, & \text{otherwise} \ . \end{cases} \end{aligned}$$

Therefore, by (2.4),

$$\Pi_f(\omega) = (-1)^{wt(\omega)} \sum_{i=0,\ldots,2^n-1\colon \alpha_i \preceq \omega} (-1)^{wt(\alpha_i)} f(\alpha_i) \ ,$$

where $\Pi_f(\omega)$ is the ωth coefficient of the NNF of function f. The latter identity can be found in [CG99, Proposition 2]. The ANF coefficients of a Boolean function f can be obtained just by reducing the corresponding NNF coefficients modulo 2. The formula for calculating the NNF coefficients from the ANF is proved in [CG99, Theorem 1]. The NNF gives an important insight into certain probabilistic properties of Boolean functions that will be discussed further in Sect. 2.5. Note that the algebraic normal transform can also be implemented over the complex field $\mathbf{C}$ for any pseudo-Boolean function taking on its values in $\mathbf{C}$.

The Probabilistic Transform.

Assume that $P = \mathbf{R}$ and set $B = \frac{1}{2}\begin{pmatrix} 2 & -1 \\ 2 & 1 \end{pmatrix}$ and $A = B^{[n]}$. Then $B^{-1} = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ -2 & 2 \end{pmatrix}$ and, by Proposition 2.2,

$$\begin{aligned} g_i(x_1,\ldots,x_n) &= \prod_{j=1}^{n}\left(\overline{\alpha_i^j} + \frac{1}{2}\alpha_i^j(x_j-\overline{x_j})\right) = \\ &= \prod_{j=1,\ldots,n\colon \alpha_i^j=1} \frac{1}{2}(x_j-\overline{x_j}) \overset{(\circ)}{=} \prod_{j=1,\ldots,n\colon \alpha_i^j=1} \delta_j \ , \end{aligned}$$

where $(\circ)$ is obtained by using $\overline{x_j} = 1 - x_j$ and introducing the new variable $\delta_j := x_j - 1/2$. Therefore,

$$f(x_1, \ldots, x_n) \overset{(2.1)}{=} \sum_{i=0}^{2^n-1} g_i(x_1, \ldots, x_n) F(\alpha_i) = \sum_{i=0}^{2^n-1} \left(\prod_{j=1,\ldots,n\colon \alpha_i^j=1} \delta_j \right) F(\alpha_i) \ .$$

The right hand side of the last identity contains the real-valued, exponent-free polynomial of n variables $\delta_1, \ldots, \delta_n$ that for $\{\delta_1, \ldots, \delta_n\} \in \{-1/2, 1/2\}^n$ takes on the same values as function f on corresponding arguments $\{x_1, \ldots, x_n\}$ if identity $x_j = \delta_j + 1/2$ is assumed. Therefore, if $D_f(x_1, \ldots, x_n)$ denotes the NNF polynomial of function f then the probabilistic transform gives coefficients for polynomial $D_f(1/2+\delta_1, \ldots, 1/2+\delta_n)$ that we will denote by Δ^f. Denote also $Q_2 = \frac{1}{2}\begin{pmatrix} 2 & -1 \\ 2 & 1 \end{pmatrix}$, $Q_{2^n} = Q_2^{[n]}$. Then $Q_2^{-1} = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ -2 & 2 \end{pmatrix}$, $Q_{2^n}^{-1} = (Q_2^{-1})^{[n]}$,

$$T^f = Q_{2^n} \Delta^f \quad \text{and} \quad \Delta^f = Q_{2^n}^{-1} T^f \ . \tag{2.5}$$

We will call this transform of pseudo-Boolean function f the *probabilistic transform.* Applications of this transform will be discussed further in Sect. 2.5.

The Walsh Transform.

According to [Rue86, p. 118], the direct and inverse *Walsh transform* operations on a real-valued pseudo-Boolean function f of n variables are defined in a point as

$$S_f(\alpha_i) = \sum_{\boldsymbol{x}=0}^{2^n-1} f(\boldsymbol{x})(-1)^{\langle \alpha_i, \boldsymbol{x} \rangle} \quad \text{and} \quad f(\boldsymbol{x}) = \frac{1}{2^n} \sum_{i=0}^{2^n-1} S_f(\alpha_i)(-1)^{\langle \alpha_i, \boldsymbol{x} \rangle} \ , \tag{2.6}$$

where $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\langle \alpha_i, \boldsymbol{x} \rangle = \alpha_i^1 x_1 \oplus \cdots \oplus \alpha_i^n x_n$ is the standard inner product over GF(2). In the sum over $\boldsymbol{x}$ in (2.6) the summation index is considered as an integer in the range $0, \ldots, 2^n - 1$ but written in its binary expansion. Coefficients $S_f(\alpha_i)$ obtained by the direct Walsh transform are called Walsh coefficients and vector $S^f = (S_f(\alpha_0), \ldots, S_f(\alpha_{2^n-1}))$ is called the Walsh transform of function f.

Assume that $P = \mathbf{R}$ and set $B = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = 2B^{-1}$ and $A = B^{[n]}$. Thus, A is a Hadamard matrix of order 2^n (see [MS96, p. 422]). Then, by Proposition 2.2,

$$g_i(x_1, \ldots, x_n) = \prod_{j=1}^{n} \left(\overline{\alpha_i^j} + \alpha_i^j (\overline{x_j} - x_j) \right) = \prod_{j=1,\ldots,n\colon \alpha_i^j=1} (\overline{x_j} - x_j) = (-1)^{\langle \alpha_i, \boldsymbol{x} \rangle}$$

and

$$f(x_1, \ldots, x_n) \overset{(2.1)}{=} \sum_{i=0}^{2^n-1} g_i(x_1, \ldots, x_n) F(\alpha_i) = \sum_{i=0}^{2^n-1} F(\alpha_i)(-1)^{\langle \alpha_i, \boldsymbol{x} \rangle} \ .$$

This identity corresponds to the inverse Walsh transform (2.6) but without the multiplicative coefficient. Therefore, in this case $F(\alpha_i) = 1/2^n S_f(\alpha_i)$, where $S_f(\alpha_i)$ is the Walsh transform of f evaluated in α_i. Let $H_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ and $H_{2^n} = H_2^{[n]}$. Then

$$T^f = \frac{1}{2^n} H_{2^n} S^f \quad \text{and} \quad S^f = H_{2^n} T^f \ . \tag{2.7}$$

Note 2.3 It is possible to generalize property (2.7) of the Walsh transform. Assume that function f is Boolean. Let r be an integer in the range $1 \le r \le n$ and let $i_1, \dots, i_r$ be a set of indices with $1 \le i_1 < \cdots < i_r \le n$. Let $k_1, \dots, k_{n-r}$ with $1 \le k_1 < \cdots < k_{n-r} \le n$ denote the indices complementing $i_1, \dots, i_r$ with respect to $\{1, \dots, n\}$. Also let the real-valued function $w(y_1, \dots, y_r)$ of r Boolean variables be defined as follows

$$w(\alpha_j^1, \dots, \alpha_j^r) = wt\left(f_{i_1,\dots,i_r}^{\alpha_j^1,\dots,\alpha_j^r}(x_{k_1}, \dots, x_{k_{n-r}})\right) = w_j$$

for $0 \le j < 2^r$, where $(\alpha_j^1, \dots, \alpha_j^r) = \alpha_j$ is the r-bit binary expansion of j. Then, by (2.7), $S^w = H_{2^r}(w_0, \dots, w_{2^r-1})^T$. On the other hand, for any $i \in \{0, \dots, 2^r - 1\}$

$$\begin{aligned} S_w(\alpha_i) &\overset{(2.6)}{=} \sum_{j=0}^{2^r-1} w(\alpha_j)(-1)^{\langle \alpha_j, \alpha_i \rangle} = \sum_{j=0}^{2^r-1} \sum_{t=0}^{2^{n-r}-1} f_{i_1,\dots,i_r}^{\alpha_j^1,\dots,\alpha_j^r}(\alpha_t)(-1)^{\langle \alpha_j, \alpha_i \rangle} = \\ &= \sum_{k=0}^{2^n-1} f(\alpha_k)(-1)^{\langle \alpha_k, \theta_i \rangle} \overset{(2.6)}{=} S_f(\theta_i) \ , \end{aligned}$$

where θ_i is the n-bit vector whose coordinates at the index positions $i_1, \dots, i_r$ are equal to $\alpha_i^1, \dots, \alpha_i^r$ respectively (where $(\alpha_i^1, \dots, \alpha_i^r) = \alpha_i$) and the remaining $(n-r)$ coordinates are set to zero. Thus,

$$H_{2^r}(w_0, \dots, w_{2^r-1})^T = (S_f(\theta_0), \dots, S_f(\theta_{2^r-1}))^T \ , \tag{2.8}$$

that is the generalization of [Sar00, Proposition 3.1], while the proof here is less complicated. If r is set equal to n then $w_j = f(\alpha_j)$, $\theta_i = \alpha_i$ and (2.8) transforms into (2.7). Identity (2.8) can be used for proving the Xiao-Massey criterion [XM88] of high-order correlation immunity for Boolean functions in terms of the Walsh transform (see Note 2.5 in Sect. 2.3).

If function f is Boolean then in some cases it is more convenient to work with the real-valued counterpart (sign function) of f, defined as $\hat{f}(\boldsymbol{x}) = 1 - 2f(\boldsymbol{x})$, and to apply the Walsh transform to $\hat{f}$. Note that $\hat{f}(\boldsymbol{x})$ is equal to the image of the element $f(\boldsymbol{x}) \in \mathrm{GF}(2)$ under the canonical additive character (see [LN83, p. 190]). Function $\hat{f}$ can be recovered by the inverse Walsh transform of $S^{\hat{f}}$. Further, since $f(\boldsymbol{x}) = 1/2 - 1/2\hat{f}(\boldsymbol{x})$, the original function f can be obtained from the Walsh transform $S^{\hat{f}}$ by the following inverse transform:

$$f(\boldsymbol{x}) = \frac{1}{2} - \frac{1}{2^{n+1}} \sum_{i=0}^{2^n-1} S_{\hat{f}}(\alpha_i)(-1)^{\langle \alpha_i, \boldsymbol{x} \rangle} \ .$$

The relationship between the Walsh transform of $f(\boldsymbol{x})$ and $\hat{f}(\boldsymbol{x})$ is given by [For89, Lemma 1] as follows

$$S_{\hat{f}}(0) = 2^n - 2S_f(0) \quad \text{and} \quad S_{\hat{f}}(w) = -2S_f(w) \quad \text{for} \quad 0 < w < 2^n \ . \tag{2.9}$$

By these identities and (2.7),

$$T^f = \left(\frac{1}{2}, \dots, \frac{1}{2}\right)^T - \frac{1}{2^{n+1}} H_{2^n} S^{\hat{f}} \quad \text{and} \quad S^{\hat{f}} = (2^n, 0, \dots, 0)^T - 2H_{2^n} T^f \tag{2.10}$$

since $H_{2^n}\left(\frac{1}{2}, 0, \dots, 0\right)^T = \left(\frac{1}{2}, \dots, \frac{1}{2}\right)^T$. On the other hand, identities, similar to (2.7), hold:

$$T^{\hat{f}} = \frac{1}{2^n} H_{2^n} S^{\hat{f}} \quad \text{and} \quad S^{\hat{f}} = H_{2^n} T^{\hat{f}} \ .$$

Combining (2.3) with (2.7) or (2.10) we obtain the following identities relating the coefficient vector of the ANF polynomial of f with the Walsh transforms S^f and $S^{\hat{f}}$:

$$\begin{aligned} P^f &= \frac{1}{2^n} R_{2^n} H_{2^n} S^f \pmod 2 = \frac{1}{2^n} \begin{pmatrix} 1 & 1 \\ 2 & 0 \end{pmatrix}^{[n]} S^f \pmod 2 \qquad (2.11) \\ P^f &= R_{2^n} \left(\left(\frac{1}{2}, \dots, \frac{1}{2}\right)^T - \frac{1}{2^{n+1}} H_{2^n} S^{\hat{f}} \right) \pmod 2 \ , \end{aligned}$$

where all operations on the right hand side are performed in $\mathbf{R}$ and the final result is reduced modulo 2. Using Proposition 2.2 for $B = \begin{pmatrix} 1 & 1 \\ 2 & 0 \end{pmatrix}$ and $A = B^{[n]}$ we obtain

$$g_i(x_1, \dots, x_n) = \prod_{j=1}^{n} \left(\overline{x_j} + 2\overline{\alpha_i^j} x_j \right) = \begin{cases} 2^{wt(x_1, \dots, x_n)}, & \text{if } \alpha_i^j \le \overline{x_j} \ (j = 1, \dots, n), \\ 0, & \text{otherwise} \ . \end{cases}$$

Therefore, by (2.11),

$$\begin{aligned} P_f(\omega) &= \frac{1}{2^{n-wt(\omega)}} \sum_{i=0,\dots,2^n-1:\, \alpha_i \preceq \overline{\omega}} S_f(\alpha_i) \pmod 2 \overset{(2.9)}{=} \\ &= 2^{wt(\omega)-1} \left(1 - \frac{1}{2^n} \sum_{i=0,\dots,2^n-1:\, \alpha_i \preceq \overline{\omega}} S_{\hat{f}}(\alpha_i) \right) \pmod 2 \ , \qquad (2.12) \end{aligned}$$

where $\overline{\omega}$ denotes the bitwise completion to 1, $\preceq$ is the earlier defined partial ordering on the Boolean lattice and $P_f(\omega)$ is the ωth coefficient of the ANF of function f.

Relation (2.12) between the ANF and Walsh coefficients can be found in [CF01, Proposition 3] but the proof above seems easier.

If (2.4) is combined with (2.7) then the resulting identities relate the NNF coefficients of f with the Walsh transform S^f:

$$\Pi^f = \frac{1}{2^n}\begin{pmatrix} 1 & 1 \\ 0 & -2 \end{pmatrix}^{[n]} S^f \quad \text{and} \quad S^f = \begin{pmatrix} 2 & 1 \\ 0 & -1 \end{pmatrix}^{[n]} \Pi^f \ . \tag{2.13}$$

Using Proposition 2.2 for $B = \begin{pmatrix} 2 & 1 \\ 0 & -1 \end{pmatrix}$ and $A = B^{[n]}$ we obtain

$$\begin{aligned} g_i(x_1, \ldots, x_n) &= \prod_{j=1}^{n} \left(2\overline{\alpha_i^j}\,\overline{x_j} + \alpha_i^j(\overline{x_j} - x_j)\right) = \\ &= \begin{cases} 2^{n-wt(\alpha_i)}(-1)^{wt(x_1,\ldots,x_n)}, & \text{if } \ x_j \le \alpha_i^j \ (j = 1, \ldots, n), \\ 0, & \text{otherwise .} \end{cases} \end{aligned}$$

Therefore, by (2.13),

$$S_f(\omega) = (-1)^{wt(\omega)} \sum_{i=0,\ldots,2^n-1:\, \omega \preceq \alpha_i} 2^{n-wt(\alpha_i)}\Pi_f(\alpha_i) \ .$$

Similarly it can be easily proved that

$$\Pi_f(\omega) = \frac{1}{2^n}(-2)^{wt(\omega)} \sum_{i=0,\ldots,2^n-1:\, \omega \preceq \alpha_i} S_f(\alpha_i) \ .$$

Finally, if (2.5) is combined with (2.7) then the resulting identities relate the probabilistic transform of f with the Walsh transform S^f:

$$\Delta^f = \frac{1}{2^n}\begin{pmatrix} 1 & 0 \\ 0 & -2 \end{pmatrix}^{[n]} S^f \quad \text{and} \quad S^f = \begin{pmatrix} 2 & 0 \\ 0 & -1 \end{pmatrix}^{[n]} \Delta^f \ . \tag{2.14}$$

Since the matrix of transform (2.14) is diagonal, coordinates of zero values in vectors Δ^f and S^f are the same. Now, using Proposition 2.2 for $B = \begin{pmatrix} 1 & 0 \\ 0 & -2 \end{pmatrix}$ and $A = B^{[n]}$ we obtain

$$g_i(x_1, \ldots, x_n) = \prod_{j=1}^{n} \left(\overline{\alpha_i^j}\,\overline{x_j} - 2\alpha_i^j x_j\right) = \begin{cases} (-2)^{wt(\alpha_i)}, & \text{if } \ x_j = \alpha_i^j \ (j = 1, \ldots, n), \\ 0, & \text{otherwise .} \end{cases}$$

Therefore, by (2.14),

$$\Delta_f(\omega) = \frac{1}{2^n}(-2)^{wt(\omega)} S_f(\omega) \overset{(2.9)}{=} \begin{cases} \frac{1}{2^n}(-2)^{wt(\omega)-1} S_{\hat{f}}(\omega), & \text{if } \ \omega \ne 0, \\ \frac{1}{2} - \frac{1}{2^{n+1}} S_{\hat{f}}(0), & \text{if } \ \omega = 0, \end{cases} \tag{2.15}$$

where $\Delta_f(\omega)$ is the ωth coordinate of the probabilistic transform of function f.

The Weight Transform.

Take $P = \mathbf{R}$ and set $D_0 = \begin{pmatrix} 0 & 1 \\ 1 & -1 \end{pmatrix}$, $D_1 = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}$ and $A = D_{\beta_1} \otimes \cdots \otimes D_{\beta_n}$ for some n-bit vector $\beta = (\beta_1, \dots, \beta_n)$. Let also

$$A^{-1} = B_1^{-1} \otimes \cdots \otimes B_n^{-1} = D_{\beta_1}^{-1} \otimes \cdots \otimes D_{\beta_n}^{-1} = (\tilde{\boldsymbol{g}}_0, \dots, \tilde{\boldsymbol{g}}_{2^n-1}) \ ,$$

where $B_j^{-1} = \begin{pmatrix} b_{00}^{(j)} & b_{01}^{(j)} \\ b_{10}^{(j)} & b_{11}^{(j)} \end{pmatrix}$, $D_0^{-1} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$ and $D_1^{-1} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$. Using Proposition 2.2 for A^{-1} and since $b_{00}^{(j)} = b_{01}^{(j)} = 1$ for any $j = 1, \dots, n$, we obtain

$$\begin{aligned} \tilde{g}_i(x_1, \dots, x_n) &= \prod_{j=1}^{n} \left(\overline{x_j} + x_j \left(\overline{\alpha_i^j} b_{10}^{(j)} + \alpha_i^j b_{11}^{(j)} \right) \right) \overset{(*)}{=} \\ &\overset{(*)}{=} \prod_{j=1}^{n} \left(\overline{x_j} + x_j \left(\overline{\alpha_i^j}\,\overline{\beta_j} + \alpha_i^j \beta_j \right) \right) = \prod_{j=1,\dots,n\colon \alpha_i^j \neq \beta_j} \overline{x_j} \ . \end{aligned} \tag{2.16}$$

Equality (*) holds because $b_{10}^{(j)} = \overline{\beta_j}$ and $b_{11}^{(j)} = \beta_j$. Thus, $\tilde{g}_i(x_1, \dots, x_n)$ is equal to one if and only if the coordinates where vectors α_i and β differ, correspond to the zero entries in vector $(x_1, \dots, x_n)$.

Assume that function f is Boolean. Then

$$\begin{aligned} F(x_1, \dots, x_n) &= \sum_{i=0}^{2^n-1} \tilde{g}_i(x_1, \dots, x_n) f(\alpha_i) = \\ &= \sum_{i=0}^{2^n-1} \left(\prod_{j=1,\dots,n\colon \alpha_i^j \neq \beta_j} \overline{x_j} \right) f(\alpha_i) = wt\left(f_{t_1,\dots,t_k}^{\beta_{t_1},\dots,\beta_{t_k}} \right) \ , \end{aligned}$$

where $k = wt(x_1, \dots, x_n)$ and $t_1, \dots, t_k$ are coordinates of nonzero entries in $(x_1, \dots, x_n)$. Here it is assumed that if $\alpha_i = \beta$ then $\prod_{j=1,\dots,n\colon \alpha_i^j \neq \beta_j} \overline{x_j} = 1$. Therefore, $wt\left(f_{1,\dots,n}^{\beta_1,\dots,\beta_n} \right) = f(\beta)$.

Let Θ_β^f denote the ordered 2^n-tuple, containing the weights of the subfunctions of f obtained by fixing all possible subsets of variables with corresponding values from vector β. Thus,

$$\Theta_\beta^f = \left\{ wt\left(f_{i_1,\dots,i_k}^{\beta_{i_1},\dots,\beta_{i_k}} \right) \mid 1 \le i_1 < \cdots < i_k \le n;\ k \in \{0, \dots, n\} \right\} \ .$$

Denote also $D_\beta = D_{\beta_1} \otimes \cdots \otimes D_{\beta_n}$. Then

$$T^f = D_\beta \, \Theta_\beta^f \quad \text{and} \quad \Theta_\beta^f = D_\beta^{-1} T^f \ . \tag{2.17}$$

We will call this transform of f the *weight transform*. In particular, if vector β consists of zeros only then $D_\beta = D_0^{[n]}$, and if it consists only of ones then $D_\beta = D_1^{[n]}$.

If we consider matrices D_0 and D_1 as matrices over the field GF(2) and perform all operations in (2.17) in this field then (2.17) will relate the string of values of function f with binary weights of its subfunctions.

Let us compare basis vector set (2.16) of the weight transform when $\beta = (0, \dots, 0)$ with basis vector set (2.2) of the inverse algebraic normal transform. It is clear that they are directly related via a simple variable complementation. Since $R_{2^n} = R_{2^n}^{-1}$, the basis vector sets of the algebraic normal transform and its inverse are equal. Therefore,

$$P_f(\alpha_i^1, \dots, \alpha_i^n) = \Theta_0^f(\overline{\alpha_i^1}, \dots, \overline{\alpha_i^n}) \pmod 2 \tag{2.18}$$

for any $i = 0, \dots, 2^n - 1$, where $(\alpha_i^1, \dots, \alpha_i^n) = \alpha_i$. This identity is easily accounted for by the well-known fact that a Boolean function has maximal algebraic degree if and only if it has an odd weight. Indeed, the right hand side of the identity contains the binary weight of the subfunction whose maximal possible order term in the ANF is equal to $\prod_{j=1,\dots,n:\, \alpha_i^j=1} x_j$ and the coefficient for this term in the ANF of f is the value on the left hand side of the identity. To construct the subfunction, relevant variables of f are being fixed only with zero values. Therefore, the term $\prod_{j=1,\dots,n:\, \alpha_i^j=1} x_j$ is either present in the ANF's of both f and the subfunction or is missing in both.

In Sect. 2.3 correlation coefficients are defined as a measure for correlation dependencies of a Boolean function. These coefficients are estimated by means of the weight transform and that suggests the importance of the weight transform for assessing cryptographic characteristics of Boolean functions.

It is important to note that the P^f, Π^f, Δ^f, S^f, $S^{\hat{f}}$ and Θ_β^f transforms of a function f can be represented by matrix equations (2.3), (2.4), (2.5), (2.7), (2.10) and (2.17), all based on the Kronecker product of appropriate elementary cells. This fact allows to use fast Fourier and Walsh transform algorithms [AHU74, Bea84] for efficient estimation of these transforms and easy transition from one transform to another. Indeed, let a and b be arbitrary 2^n-dimensional vectors over P, such that $b = (B_1 \otimes \cdots \otimes B_n)a$, where $B_j = \begin{pmatrix} b_{11}^{(j)} & b_{12}^{(j)} \\ b_{21}^{(j)} & b_{22}^{(j)} \end{pmatrix}$ $(j = 1, \dots, n)$ are arbitrary elementary cells over P. Then

$$b = \begin{pmatrix} b_{11}^{(1)} B' & b_{12}^{(1)} B' \\ b_{21}^{(1)} B' & b_{22}^{(1)} B' \end{pmatrix} a = \begin{pmatrix} b_{11}^{(1)} B'\overline{a} + b_{12}^{(1)} B'\underline{a} \\ b_{21}^{(1)} B'\overline{a} + b_{22}^{(1)} B'\underline{a} \end{pmatrix} , \tag{2.19}$$

where $B' = B_2 \otimes \cdots \otimes B_n$ and $a = (\overline{a}, \underline{a})$ is the split of a into two halves. Thus, estimation of b requires 2^{n+1} multiplications, 2^n additions in P and two transforms of order $n - 1$. It is easy to prove by induction that the total complexity of the nth-order transform is equivalent to $O(n2^n)$ arithmetic operations in P.

Also note that if a is the string of values of a function f, i.e., $a = T^f$, then $\overline{a}$ and $\underline{a}$ are strings of values of subfunctions f_1^0 and f_1^1 respectively. Thus, $B'\overline{a}$ and $B'\underline{a}$ are the adequate transforms of these subfunctions. Thus, (2.19) provides a relation between the transform of a function f and transforms of its subfunctions f_1^0 and f_1^1.

2.3 Algebraic and Correlation Properties of Boolean Functions Related to the Weight Transform

The concept of correlation immunity relates to the statistical dependence between m-tuples of inputs and the output of a cryptographic transformation. This idea is extremely important, especially for the design of stream ciphers, where filter and combination generators with not correlation immune filtering and combining functions are susceptible to ciphertext-only attacks [Sie85]. A function is said to be mth-order *correlation immune* if the distribution probability of its output is unaltered when any m of its inputs are fixed.

High-order correlation immunity for Boolean functions was first introduced in [Sie84] where the well-known *Siegenthaler's inequality* has also been first proved in [Sie84, Theorem 1]. According to this inequality, the sum of the algebraic degree and the order of correlation immunity for a Boolean function of n variables cannot exceed n. Moreover, it is upper-bounded by $n-1$ if the function is balanced (i.e., when the function takes on the zero-value exactly on a half of the domain). Balanced mth-order correlation immune functions are called m-resilient [Sar00] and any balanced function is also called 0-resilient. By Siegenthaler's inequality, high-order correlation immune functions necessarily have low algebraic degree and vice versa. In order to handle this situation one has either to find a trade-off between these two properties or somehow to weaken the requirement for a function to be correlation immune. From a practical point of view, functions with low correlation dependencies are as secure as correlation immune ones. In this case we need an estimate for correlation dependencies of a Boolean function.

Here we assume that $X_1, \ldots, X_n$ are uniform, independent and identically distributed random binary variables and $f(\boldsymbol{x})$, where $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathrm{GF}(2)^n$, is a Boolean function of n variables that is not identical 0 or 1. Let $\boldsymbol{X} = (X_1, \ldots, X_n)$. Then $f(\boldsymbol{X})$ denotes the binary random variable obtained by substituting random values X_i for the variables of f. Formally speaking, Boolean function $f(\boldsymbol{x})$ is correlation immune of order m (as defined in [Sie84]) if

$$\Pr(f(\boldsymbol{X}) = 0 \mid X_{i_1} = \beta_1, \ldots, X_{i_m} = \beta_m) = \Pr(f(\boldsymbol{X}) = 0)$$

for any choice of integer $i_1, \ldots, i_m$ with $1 \leq i_1 < \cdots < i_m \leq n$ and any m-bit tuple $(\beta_1, \ldots, \beta_m)$. The following definition of correlation coefficients generalizes the basic concept of correlation immunity.

Definition 2.4 *Let m and $i_1, \ldots, i_m$ be integers with $1 \leq m \leq n$ and $1 \leq i_1 < \cdots < i_m \leq n$. Then the set of 2^m conditional probabilities*

$$c_{i_1,\ldots,i_m}^{\beta_1,\ldots,\beta_m} = \Pr(X_{i_1} = \beta_1, \ldots, X_{i_m} = \beta_m \mid f(\boldsymbol{X}) = 0) \ ,$$

evaluated for all possible values of the m-bit tuple $(\beta_1, \ldots, \beta_m)$ and ordered lexicographically along these values, is called a vector of mth-order correlation coefficients of f, evaluated for the input subset $(i_1, \ldots, i_m)$.

By the Bayes rule, function f is mth-order correlation immune if and only if all its mth-order correlation coefficients are equal to $1/2^m$. On the other hand, for any Boolean function f,

$$\begin{aligned}\Pr(X_{i_1}=\beta_1,\ldots,X_{i_m}=\beta_m \mid f(\boldsymbol{X})=0) &= \frac{2^{n-m}-wt\left(f_{i_1,\ldots,i_m}^{\beta_1,\ldots,\beta_m}\right)}{2^n-wt(f)} \\ \Pr(X_{i_1}=\beta_1,\ldots,X_{i_m}=\beta_m \mid f(\boldsymbol{X})=1) &= \frac{wt\left(f_{i_1,\ldots,i_m}^{\beta_1,\ldots,\beta_m}\right)}{wt(f)} \ .\end{aligned} \tag{2.20}$$

Conditional probabilities in (2.20) are equal to $1/2^m$ if and only if $wt\left(f_{i_1,\ldots,i_m}^{\beta_1,\ldots,\beta_m}\right) = 2^{-m}wt(f)$. Therefore, the mth-order correlation immunity of f implies that the output of f and any m input variables, considered jointly, are statistically independent. Correlation coefficients $c_{i_1,\ldots,i_m}^{\beta_1,\ldots,\beta_m}$ are easily estimated, making use of (2.20), if the weights of function f and of the subfunctions $f_{i_1,\ldots,i_m}^{\beta_1,\ldots,\beta_m}$ are known (see Sect. 2.2 about the weight transform). For instance, 1st-order correlation coefficients satisfy the identity

$$c_i^\beta = \frac{1}{2^n-wt(f)}\left(2^{n-1}-wt\left(f_i^\beta\right)\right) \ ,$$

and $wt\left(f_i^\beta\right)$ is equal to the number of n-bit vectors $(x_1,\ldots,x_n)$ in the support of f that have the ith coordinate x_i, equal to β. By the support of f we mean the subset of $\mathrm{GF}(2)^n$ where f is equal to 1.

Note 2.5 It is well known [XM88] that a Boolean function f of n variables is mth-order correlation immune for $1 \le m \le n$, if and only if its Walsh transform coefficients are equal to zero for any nonzero vector with Hamming weight not exceeding m. This criterion is easy to prove making use of property (2.8) of the Walsh transform. Indeed, $S_f(\alpha)=0$ for any $\alpha \in \mathrm{GF}(2)^n$ with $1 \le wt(\alpha) \le m$ if and only if for any set of indices $i_1,\ldots,i_m$ with $1 \le i_1 < \cdots < i_m \le n$ Walsh coefficients $S_f(\theta_1),\ldots,S_f(\theta_{2^m-1})$ are all equal to zero. By (2.8), this is equivalent to $S_w(\beta)=0$ for any $\beta \ne 0$ (using the notation introduced in Sect. 2.2). From (2.6) it follows that the only function having zero Walsh coefficients for all nonzero arguments is the constant function and thus, for any $(y_1,\ldots,y_m) \in \mathrm{GF}(2)^m$

$$w(y_1,\ldots,y_m) = wt\left(f_{i_1,\ldots,i_m}^{y_1,\ldots,y_m}\right) \equiv \frac{1}{2^m}S_w(0) = \frac{wt(f)}{2^m} \ .$$

Now, by (2.20), this is equivalent to function f being mth-order correlation immune. In particular, this criterion implies that for $m>1$, any mth-order correlation immune function is also $(m-1)$st-order correlation immune.

Proposition 2.6 *A Boolean function f of n variables is mth-order correlation immune for $1 \le m \le n$, if and only if for every $k \in \{1,\ldots,m\}$ and any set of indices*

$i_1, \ldots, i_k$ with $1 \le i_1 < \cdots < i_k \le n$, there exists at least one k-bit tuple $(\beta_1, \ldots, \beta_k)$, such that correlation coefficient $c_{i_1,\ldots,i_k}^{\beta_1,\ldots,\beta_k}$ is equal to $1/2^k$.

Proof: By Note 2.5, if function f is mth-order correlation immune then it is kth-order correlation immune for any $k \in \{1, \ldots, m\}$. This means that for any such a k all kth-order correlation coefficients of f are equal to $1/2^k$. Thus, the condition stated in the proposition is necessary for a function to be mth-order correlation immune. To show that this condition is also sufficient, we apply induction on m.

Let $m = 1$ and assume that for any i with $1 \le i \le n$ there exists some β_i, such that the corresponding correlation coefficient $c_i^{\beta_i}$ is equal to $1/2$. Then, by (2.20), $wt(f_i^{\beta_i}) = wt(f)/2$. Therefore,

$$wt\left(f_i^{\beta_i \oplus 1}\right) = wt(f) - wt(f_i^{\beta_i}) = \frac{wt(f)}{2}$$

and

$$c_i^{\beta_i \oplus 1} = \frac{1}{2^n - wt(f)}\left(2^{n-1} - wt\left(f_i^{\beta_i \oplus 1}\right)\right) = \frac{1}{2} \ .$$

Thus, function f is 1st-order correlation immune.

Now, assuming that the proposition is true for $m = l - 1$, we prove it for $m = l$. The conditions imposed above imply that for any set of indices $i_1, \ldots, i_l$ with $1 \le i_1 < \cdots < i_l \le n$, there exists an l-bit tuple $(\beta_1, \ldots, \beta_l)$, such that $c_{i_1,\ldots,i_l}^{\beta_1,\ldots,\beta_l}$ is equal to $1/2^l$. According to the induction hypothesis, the imposed conditions are sufficient for f to be $(l-1)$st-order correlation immune and thus $c_{i_1,\ldots,i_{l-1}}^{\beta_1,\ldots,\beta_{l-1}} = 1/2^{l-1}$. Then, by (2.20), $wt\left(f_{i_1,\ldots,i_l}^{\beta_1,\ldots,\beta_l}\right) = wt(f)/2^l$ and $wt\left(f_{i_1,\ldots,i_{l-1}}^{\beta_1,\ldots,\beta_{l-1}}\right) = wt(f)/2^{l-1}$. Therefore,

$$wt\left(f_{i_1,\ldots,i_l}^{\beta_1,\ldots,\beta_{l-1},\beta_l \oplus 1}\right) = wt\left(f_{i_1,\ldots,i_{l-1}}^{\beta_1,\ldots,\beta_{l-1}}\right) - wt\left(f_{i_1,\ldots,i_l}^{\beta_1,\ldots,\beta_{l-1},\beta_l}\right) = \frac{wt(f)}{2^l}$$

and $c_{i_1,\ldots,i_l}^{\beta_1,\ldots,\beta_{l-1},\beta_l \oplus 1} = 1/2^l$. Any l-bit tuple can be obtained by consecutive inverting of required coordinates in the fixed tuple $(\beta_1, \ldots, \beta_l)$. This way it follows that for any m-bit tuple $(\gamma_1, \ldots, \gamma_l)$, the correlation coefficient $c_{i_1,\ldots,i_l}^{\gamma_1,\ldots,\gamma_l}$ is equal to $1/2^l$. Thus, function f is lth-order correlation immune. □

A Boolean function f cannot be considered cryptographically secure if there exists a function, having low algebraic degree or depending on small number of variables, that coincides with f on a larger half of the domain or, in other words, that *approximates* f (see [Can02]). Indeed, the existence of a relatively accurate, low-dimensional approximation for a cipher transformation could reduce the complexity of the exhaustive search to the dimension of the domain of this approximation. The following proposition shows that an m-resilient Boolean function cannot be approximated by any function depending on m variables only.

Proposition 2.7 *Any balanced Boolean function f of n variables is m-resilient for $1 \le m < n$, if and only if there are no approximations of f depending on m variables.*

Proof: Suppose that function $f(x_1,\ldots,x_n)$ is m-resilient for some $1 \le m < n$ and that there exists a function $g(x_{i_1},\ldots,x_{i_m})$ approximating f. Then

$$\begin{aligned}\Pr(f(\boldsymbol{X}) \neq g(X_{i_1},\ldots,X_{i_m})) &= \frac{wt(f(\boldsymbol{x}) \oplus g(x_{i_1},\ldots,x_{i_m}))}{2^n} = \\ &= \frac{\sum_{(\beta_{i_1},\ldots,\beta_{i_m})\in \mathrm{GF}(2)^m} wt\left(f_{i_1,\ldots,i_m}^{\beta_{i_1},\ldots,\beta_{i_m}} \oplus g(\beta_{i_1},\ldots,\beta_{i_m})\right)}{2^n} = \\ &= \frac{2^m 2^{n-m-1}}{2^n} = \frac{1}{2} \ .\end{aligned}$$

Thus, $g(x_{i_1},\ldots,x_{i_m})$ does not approximate f.

Suppose now that there are no approximations of f, depending on m variables. In particular, there are no linear approximations, depending on m variables, meaning that for any n-bit vector $\beta = (\beta_1,\ldots,\beta_n)$ such that $0 < wt(\beta) \le m$, $\Pr(f(\boldsymbol{X}) = (\beta_1 X_1 \oplus \cdots \oplus \beta_n X_n)) = 1/2$. On the other hand, for any nonzero β the following known [Rue86, p. 121] identity holds

$$\Pr(f(\boldsymbol{X}) = (\beta_1 X_1 \oplus \cdots \oplus \beta_n X_n)) = \frac{1}{2} - \frac{S_f(\beta)}{2^n} \ , \tag{2.21}$$

where $S_f(\beta)$ is the Walsh transform of f evaluated in β. Thus, $S_f(\beta) = 0$ and by Note 2.5, function f is mth-order correlation immune. □

For any Boolean function f of n variables let $S_m(f)$ denote the number of subfunctions obtained by fixing m variables of f with zero values and having an even weight. Let also $D_{n-m}(f)$ denote the total number of $(n-m)$th-order product terms, contained in the ANF of f. The following proposition, which easily follows from (2.18), establishes a relation between the values of $S_m(f)$ and $D_{n-m}(f)$.

Proposition 2.8 *For any Boolean function f of n variables and any positive integer $m \le n$,*

$$S_m(f) + D_{n-m}(f) = \binom{n}{m} \ .$$

Further, let $C_m(f)$ denote the number of mth-order correlation coefficient vectors of f with coordinate $c_{i_1,\ldots,i_m}^{0,\ldots,0}$ equal to $1/2^m$. From (2.20) it is clear that an nth-order correlation coefficient of a nonconstant Boolean function of n variables cannot be equal to $1/2^n$ and thus for such a function $C_n(f) = 0$.

Corollary 2.9 *For any balanced Boolean function f of n variables and any positive integer $m < n-1$,*

$$C_m(f) + D_{n-m}(f) \le \binom{n}{m} \ .$$

Moreover, equality holds if function f is such that

$$\frac{2^{n-m-1}-1}{2^{n-1}} \le c_{i_1,\ldots,i_m}^{0,\ldots,0} \le \frac{2^{n-m-1}+1}{2^{n-1}} \tag{2.22}$$

for all index values $i_1, \ldots, i_m$ *with* $1 \leq i_1 < \cdots < i_m \leq n$.

Proof: By (2.20), coordinate $c^{0,\ldots,0}_{i_1,\ldots,i_m}$ of the mth-order correlation coefficient vector of f, evaluated for the input subset $(i_1, \ldots, i_m)$, is equal to $1/2^m$ if and only if $wt\left(f^{0,\ldots,0}_{i_1,\ldots,i_m}\right) = 2^{n-m-1}$, that is an even value for any $m < n-1$. Thus, $C_m(f) \leq S_m(f)$ and the claimed inequality directly follows from Proposition 2.8.

Suppose now that condition (2.22) holds for all index values $i_1, \ldots, i_m$ with $1 \leq i_1 < \cdots < i_m \leq n$. Then, by (2.20),

$$2^{n-m-1} - 1 \leq wt\left(f^{0,\ldots,0}_{i_1,\ldots,i_m}\right) \leq 2^{n-m-1} + 1 \ .$$

Suppose also that the ANF of function f does not contain the $(n-m)$th-order product term $\prod_{j=1,\ldots,n:\, j \notin \{i_1,\ldots,i_m\}} x_j$. Then, by (2.18), the subfunction $f^{0,\ldots,0}_{i_1,\ldots,i_m}$ has an even weight, equal to 2^{n-m-1}. Then, by (2.20),

$$c^{0,\ldots,0}_{i_1,\ldots,i_m} = \frac{1}{2^{n-1}}\left(2^{n-m} - wt\left(f^{0,\ldots,0}_{i_1,\ldots,i_m}\right)\right) = \frac{1}{2^m} \ .$$

So, if condition (2.22) holds then every missing $(n-m)$th-order product term in the ANF of f gives rise to a $1/2^m$ valued coordinate of the corresponding correlation vector and thus

$$C_m(f) \geq \binom{n}{m} - D_{n-m}(f) \ .$$

Now the last inequality combined with the one argued in the first part of the corollary results in the claimed equality. □

From Corollary 2.9 and Note 2.5 it easily follows, that for $m < n-1$ and any m-resilient Boolean function f of n variables, $D_{n-k}(f) = 0$ for all $k = 1, \ldots, m$ (since $C_k(f) = \binom{n}{k}$). The maximal order product term $x_1 \cdot \cdots \cdot x_n$ is missing in the ANF of f since function f has an even weight. Therefore, the algebraic degree of f does not exceed $(n-m-1)$. So, it can be concluded that if k is the attainable algebraic degree and m is the attainable degree of resiliency for a balanced Boolean function of n variables then $k + m \leq n - 1$ (that is Siegenthaler's inequality for a balanced function).

2.4 Tensor Transform of Functions over Finite Fields

With the development of computer technology, multivalued logical functions become more and more important in cryptography, particularly in the design of stream ciphers. Therefore, efficient tools for analyzing the security of these functions are needed. Such a tool, the tensor transform, was developed in Sect. 2.2 for the pseudo-Boolean case. The objective of this section is to generalize this approach to the case of functions over GF(q).

In this section we continue to use the notation introduced in Sect. 2.2 except that binary values are replaced by q-ary ones. For a matrix $A \in M_{q^n}(P)$ we use notation $A = (\boldsymbol{g}_0, \ldots, \boldsymbol{g}_{q^n-1})$, where $\boldsymbol{g}_i$ $(i = 0, \ldots, q^n - 1)$ denotes the ith column of A and the coordinates of $\boldsymbol{g}_i$ are indexed lexicographically by the elements in $\{0, \ldots, q-1\}^n$, so

$$\boldsymbol{g}_i = \begin{pmatrix} g_i(0, \ldots, 0) \\ g_i(0, \ldots, 1) \\ \vdots \\ g_i(q-1, \ldots, q-1) \end{pmatrix} .$$

Let $\alpha_i = (\alpha_i^1, \ldots, \alpha_i^n)$ $(i = 0, \ldots, q^n - 1)$ denote the n-digit q-ary expansion of i, where the leftmost bit is the most significant. Then $\boldsymbol{g}_i = (g_i(\alpha_0), \ldots, g_i(\alpha_{q^n-1}))^T$.

Lemma 2.10 *Let $A = (\boldsymbol{g}_0, \ldots, \boldsymbol{g}_{q^n-1}) \in M_{q^n}(P)$ and $A' = (\boldsymbol{g}'_0, \ldots, \boldsymbol{g}'_{q^{n-1}-1}) \in M_{q^{n-1}}(P)$. Suppose that $A = B \otimes A'$ for some matrix $B = (b_{m,k})_{q \times q}$ $(m, k = 0, \ldots, q-1)$. Then for any $i \in \{0, \ldots, q^n - 1\}$*

$$g_i(x_1, \ldots, x_n) = \left(\sum_{m=0}^{q-1} \mathrm{I}_m(x_1) b_{m,k} \right) g'_{i'}(x_2, \ldots, x_n) \quad \textit{if} \quad \alpha_i = (k, \alpha_{i'}) ,$$

where $\alpha_{i'}$ is the $(n-1)$-digit vector, q-ary expansion of i' and $\mathrm{I}_m(x_1)$ is the indicator function of the event $\{x_1 = m\}$.

Proof: By the definition of the Kronecker product, $A = (b_{m,k}A')_{q^n \times q^n}$ $(m, k = 0, \ldots, q-1)$. Therefore, the entry $g_i(x_1, \ldots, x_n)$ of matrix A lies in the cell $b_{m,k}A'$ of size $q^{n-1} \times q^{n-1}$, where $m = x_1$ and k is equal to the most significant digit of α_i. Local coordinates of $g_i(x_1, \ldots, x_n)$ in this cell are equal to $(x_2, \ldots, x_n)$ and $\alpha_{i'}$. Thus, for any $m = 0, \ldots, q-1$

$$g_i(m, x_2, \ldots, x_n) = b_{m,k} g'_{i'}(x_2, \ldots, x_n) \quad \text{if} \quad \alpha_i = (k, \alpha_{i'}) .$$

This proves the claimed identity. □

The following proposition easily follows from Lemma 2.10.

Proposition 2.11 *Let $A = B_1 \otimes \cdots \otimes B_n$, where $B_j = \left(b_{m,k}^{(j)} \right)_{q \times q}$ $(m, k = 0, \ldots, q-1)$ for $j = 1, \ldots, n$, and $A = (\boldsymbol{g}_0, \ldots, \boldsymbol{g}_{q^n-1})$. Then for any $i \in \{0, \ldots, q^n - 1\}$*

$$g_i(x_1, \ldots, x_n) = \prod_{j=1}^{n} \left(\sum_{k=0}^{q-1} \sum_{m=0}^{q-1} \mathrm{I}_k(\alpha_i^j) \mathrm{I}_m(x_j) b_{m,k}^{(j)} \right) .$$

In this section we consider functions of n variables over $\mathrm{GF}(q)$. In order to define the string of values for such a function we need to order the elements in $\mathrm{GF}(q)^n$. Let ξ denote a primitive element of $\mathrm{GF}(q)$. Then all nonzero elements of the field are exactly $\{\xi, \xi^2, \ldots, \xi^{q-1}\}$. We can define a one-to-one correspondence between $\mathrm{GF}(q)$ and the set of integers in the range $0, \ldots, q-1$ in such a way that 0 corresponds to the zero-element of the field and $i \in \{1, \ldots, q-1\}$ corresponds to ξ^i. With this correspondence, any element in $\mathrm{GF}(q)^n$ has unique counterpart α_i for some $i \in \{0, \ldots, q^n - 1\}$ and, therefore, the elements in $\mathrm{GF}(q)^n$ can be ordered lexicographically.

Let function $f(x_1, \ldots, x_n)$, mapping $\mathrm{GF}(q)^n$ to P, be defined by its string of values $T^f = (f(\alpha_0), \ldots, f(\alpha_{q^n-1}))^T \in P^{q^n}$. Here, x_i denotes the variable taking on its values in $\mathrm{GF}(q)$. However, using the above described correspondence, x_i can be also seen as an integer in the range $0, \ldots, q-1$. Similarly, $(x_1, \ldots, x_n)$ can either be an element of $\mathrm{GF}(q)^n$ or an integer in the range $0, \ldots, q^n - 1$ in its q-ary expansion. In the rest of this section we will use the same notation to denote both the elements of the field and the corresponding integers hoping that any ambiguity can easily be resolved by the reader in each specific case.

Further, let $A \in M_{q^n}(P)$ be an invertible matrix, $A = (\boldsymbol{g}_0, \ldots, \boldsymbol{g}_{q^n-1})$, and let function $F(x_1, \ldots, x_n)$ be defined by the string $T^F = A^{-1}T^f = (F(\alpha_0), \ldots, F(\alpha_{q^n-1}))^T \in P^{q^n}$. Vectors T^f and T^F are considered further as column-vectors. Then $T^f = AT^F$,

$$T^f = \sum_{i=0}^{q^n-1} \boldsymbol{g}_i F(\alpha_i) \quad \text{and} \quad f(x_1, \ldots, x_n) = \sum_{i=0}^{q^n-1} g_i(x_1, \ldots, x_n) F(\alpha_i) \tag{2.23}$$

for any $(x_1, \ldots, x_n) \in \mathrm{GF}(q)^n$. Equations (2.23) represent the decomposition of function f in the basis vector set $(\boldsymbol{g}_0, \ldots, \boldsymbol{g}_{q^n-1})$.

Proposition 2.11 can now be used to generalize some tensor transforms of pseudo-Boolean functions from Sect. 2.2 to the case of functions over $\mathrm{GF}(q)$.

The Identity Transform.

Let P be an arbitrary field and set $B = E_q$ - the identity matrix of size q, and $A = B^{[n]} = E_{q^n}$. Then, by Proposition 2.11,

$$g_i(x_1, \ldots, x_n) = \prod_{j=1}^{n} \left(\sum_{k=0}^{q-1} \mathrm{I}_k(\alpha_i^j) \mathrm{I}_k(x_j) \right) = \begin{cases} 1, & \text{if } \ \alpha_i = (x_1, \ldots, x_n), \\ 0, & \text{otherwise} \end{cases}$$

and

$$f(x_1, \ldots, x_n) \overset{(2.23)}{=} \sum_{i=0}^{q^n-1} g_i(x_1, \ldots, x_n) F(\alpha_i) = F(x_1, \ldots, x_n) \ .$$

The Algebraic Normal Transform.

Take $P = \mathrm{GF}(q)$ and set

$$B = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & \xi & \xi^2 & \dots & \xi^{q-2} & 1 \\ 1 & \xi^2 & \xi^4 & \dots & \xi^{2(q-2)} & 1 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & \xi^{q-2} & \xi^{(q-2)2} & \dots & \xi^{(q-2)(q-2)} & 1 \\ 1 & 1 & 1 & \dots & 1 & 1 \end{pmatrix} \tag{2.24}$$

and $A = B^{[n]}$. Then, by Proposition 2.11,

$$g_i(x_1, \dots, x_n) = \prod_{j=1}^{n} b_{x_j, \alpha_i^j} = \prod_{j=1,\dots,n:\, \alpha_i^j \neq 0} (x_j)^{\alpha_i^j} \ ,$$

where x_j on the right hand side of the last identity denotes the element of $\mathrm{GF}(q)$. Therefore,

$$f(x_1, \dots, x_n) \overset{(2.23)}{=} \sum_{i=0}^{q^n-1} g_i(x_1, \dots, x_n) F(\alpha_i) = \sum_{i=0}^{q^n-1} \left(\prod_{j=1,\dots,n:\, \alpha_i^j \neq 0} (x_j)^{\alpha_i^j} \right) F(\alpha_i) \ .$$

One can easily recognize the ANF of function f over $\mathrm{GF}(q)$ on the right hand side of the last identity, where $F(\alpha_i)$ $(i = 0, \dots, q^n - 1)$ are the coefficients of the ANF polynomial. Denote matrix (2.24) as R_q and define $\mathcal{R}_{q^n} = R_q^{[n]}$. If P^f is the coefficient vector of the ANF polynomial for function f then

$$T^f = \mathcal{R}_{q^n} P^f \quad \text{and} \quad P^f = \mathcal{R}_{q^n}^{-1} T^f \ ,$$

where $\mathcal{R}_{q^n}^{-1} = (R_q^{-1})^{[n]}$ and

$$R_q^{-1} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & -\xi^{q-2} & -\xi^{(q-2)2} & \dots & -\xi^{(q-2)(q-2)} & -1 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & -\xi^2 & -\xi^4 & \dots & -\xi^{2(q-2)} & -1 \\ 0 & -\xi & -\xi^2 & \dots & -\xi^{q-2} & -1 \\ -1 & -1 & 1 & \dots & -1 & -1 \end{pmatrix} .$$

The algebraic normal transform over $\mathrm{GF}(q)$ was introduced in [Jan89, Sect. 4.5] where it also was proved that R_q^{-1} has the specific form stated above.

The Walsh Transform.

The Walsh transform of a complex-valued function f of n variables over $\mathrm{GF}(q)$ can be defined as the n-dimensional discrete Fourier transform over the complex field

$\mathbf{C}$ (see [CC99, Sect. 2.2]). Let χ_1 denote the canonical additive character of GF(q) (so $\chi_1(a) = e^{2\pi i \mathrm{Tr}(a)/p}$ for $a \in \mathrm{GF}(q)$, where p is the characteristic of GF(q) and Tr : GF$(q) \to$ GF(p) is the absolute trace function) and $\overline{\chi}_1$ denote its complex conjugate [LN83, p. 190]. Accordingly, direct and inverse transform operations are defined in a point by the respective identities

$$S_f(\alpha) = \sum_{\boldsymbol{x} \in \mathrm{GF}(q)^n} f(\boldsymbol{x}) \chi_1(\langle \alpha, \boldsymbol{x} \rangle) \quad \text{and} \quad f(\boldsymbol{x}) = \frac{1}{q^n} \sum_{\alpha \in \mathrm{GF}(q)^n} S_f(\alpha) \overline{\chi}_1(\langle \alpha, \boldsymbol{x} \rangle) \ , \tag{2.25}$$

where $\boldsymbol{x} = (x_1, \ldots, x_n)$, $\alpha = (\alpha^1, \ldots, \alpha^n)$ and $\langle \alpha, \boldsymbol{x} \rangle = \alpha^1 x_1 + \cdots + \alpha^n x_n$ is the standard inner product over GF(q). This extends the older definition of the Walsh transform for pseudo-Boolean functions given in Sect. 2.2. The vector consisting of Walsh coefficients $S_f(\alpha)$ that are ordered lexicographically along the values of $\alpha \in \mathrm{GF}(q)^n$ is denoted as S^f and called the Walsh transform of function f.

Assume that $P = \mathbf{C}$ and set

$$B = \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & \overline{\chi}_1(\xi^2) & \overline{\chi}_1(\xi^3) & \ldots & \overline{\chi}_1(\xi^q) \\ 1 & \overline{\chi}_1(\xi^3) & \overline{\chi}_1(\xi^4) & \ldots & \overline{\chi}_1(\xi^{q+1}) \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \overline{\chi}_1(\xi^q) & \overline{\chi}_1(\xi^{q+1}) & \ldots & \overline{\chi}_1(\xi^{2q-2}) \end{pmatrix} \tag{2.26}$$

and $A = B^{[n]}$. Then, by Proposition 2.11,

$$g_i(x_1, \ldots, x_n) = \prod_{j=1}^{n} b_{x_j, \alpha_i^j} = \prod_{j=1}^{n} \overline{\chi}_1(x_j \alpha_i^j) = \overline{\chi}_1(\langle \alpha_i, \boldsymbol{x} \rangle) \ ,$$

where x_j and α_i^j on the left hand side of the last identity denote elements of GF(q). Therefore,

$$f(x_1, \ldots, x_n) \overset{(2.23)}{=} \sum_{i=0}^{q^n-1} g_i(x_1, \ldots, x_n) F(\alpha_i) = \sum_{i=0}^{q^n-1} F(\alpha_i) \overline{\chi}_1(\langle \alpha_i, \boldsymbol{x} \rangle) \ .$$

This identity corresponds to the inverse Walsh transform (2.25) but without the multiplicative coefficient. Thus, in this case $F(\alpha_i) = 1/q^n S_f(\alpha_i)$, where $S_f(\alpha_i)$ is the Walsh transform of f evaluated in α_i. Denote matrix (2.26) as H_q and define $\mathcal{H}_{q^n} = H_q^{[n]}$. Then

$$T^f = \frac{1}{q^n} \mathcal{H}_{q^n} S^f \quad \text{and} \quad S^f = q^n \mathcal{H}_{q^n}^{-1} T^f \ , \tag{2.27}$$

where $q^n \mathcal{H}_{q^n}^{-1} = (q H_q^{-1})^{[n]}$ and the inverse matrix $q H_q^{-1}$ is of the same type as (2.26) but without conjugation of characters. Indeed, the element of the product matrix $H_q H_q^{-1}$ with coordinates $(a, c) \in \mathbf{Z}_q^2$ is equal to

$$\sum_{b \in \mathrm{GF}(q)} \overline{\chi}_1(ab) \chi_1(bc) = \sum_{b \in \mathrm{GF}(q)} \chi_1(b(c-a)) = \begin{cases} q, & \text{if } \ a = c \ , \\ \sum_{b \in \mathrm{GF}(q)} \chi_1(b) = 0, & \text{otherwise} \ . \end{cases}$$

Note 2.12 Matrix identities (2.27) can be presented in a more general form based on the Walsh transform of complex-valued functions that in the binary case (when $P = \mathrm{GF}(2)$) are equal to the weight function. Let r be an integer in the range $1 \le r \le n$ and let $i_1, \ldots, i_r$ be a set of indices with $1 \le i_1 < \cdots < i_r \le n$. Let $k_1, \ldots, k_{n-r}$ with $1 \le k_1 < \cdots < k_{n-r} \le n$ denote the indices complementing $i_1, \ldots, i_r$ with respect to $\{1, \ldots, n\}$. Also let the complex-valued function $w(y_1, \ldots, y_r)$ of r variables be defined by

$$w(y_1, \ldots, y_r) = \sum_{(x_{k_1}, \ldots, x_{k_{n-r}}) \in \mathrm{GF}(q)^{n-r}} f^{y_1, \ldots, y_r}_{i_1, \ldots, i_r}(x_{k_1}, \ldots, x_{k_{n-r}}) \ ,$$

where $y_i \in \mathrm{GF}(q)$ for $i = 1, \ldots, r$. Now, using the matrix notation for the direct Walsh transform of function w and by means of (2.27), we have the identity $S^w = q^r H_{q^r}^{-1} T^w$. On the other hand, for any $\alpha \in \mathrm{GF}(q)^r$

$$\begin{aligned} S_w(\alpha) &\overset{(2.25)}{=} \sum_{\boldsymbol{y} \in \mathrm{GF}(q)^r} w(\boldsymbol{y}) \chi_1(\langle \alpha, \boldsymbol{y} \rangle) = \sum_{\boldsymbol{y} \in \mathrm{GF}(q)^r} \sum_{\boldsymbol{z} \in \mathrm{GF}(q)^{n-r}} f^{y_1, \ldots, y_r}_{i_1, \ldots, i_r}(\boldsymbol{z}) \chi_1(\langle \alpha, \boldsymbol{y} \rangle) = \\ &= \sum_{\boldsymbol{x} \in \mathrm{GF}(q)^n} f(\boldsymbol{x}) \chi_1(\langle \boldsymbol{x}, \theta_\alpha \rangle) \overset{(2.25)}{=} S_f(\theta_\alpha) \ , \end{aligned}$$

where $\boldsymbol{y} = \{y_1, \ldots, y_r\}$, $\boldsymbol{z} = \{z_1, \ldots, z_{n-r}\}$ and θ_α is the n-digit vector whose coordinates at the index positions $i_1, \ldots, i_r$ are equal to $\alpha^1, \ldots, \alpha^r$ respectively (where $(\alpha^1, \ldots, \alpha^r) = \alpha$) and the remaining $(n - r)$ coordinates are set to zero. Thus,

$$q^r H_{q^r}^{-1} T^w = (S_f(\theta_0), \ldots, S_f(\theta_{q^r - 1}))^T \ , \tag{2.28}$$

where the θ's are indexed with integer values in the range $0, \ldots, q^r - 1$ corresponding to the elements of $\mathrm{GF}(q)^r$. Identity (2.28) is the generalization of (2.8). If r is set equal to n then $w(\boldsymbol{y}) = f(\boldsymbol{y})$, $\theta_\alpha = \alpha$ and (2.28) transforms into (2.27).

Consider now a function f that takes on its values in an arbitrary finite field P. In order to define the Walsh transform for such a function we need to estimate the complex image of $f(\boldsymbol{x})$ under a nontrivial additive character of P (that will be denoted by χ). By (2.25), the pair of Walsh transform operations on f are the following:

$$\begin{aligned} S_{\hat{f}}(\alpha) &= \sum_{\boldsymbol{x} \in \mathrm{GF}(q)^n} \chi(f(\boldsymbol{x})) \chi_1(\langle \alpha, \boldsymbol{x} \rangle) \quad \text{and} \\ \chi(f(\boldsymbol{x})) &= \frac{1}{q^n} \sum_{\alpha \in \mathrm{GF}(q)^n} S_{\hat{f}}(\alpha) \overline{\chi}_1(\langle \alpha, \boldsymbol{x} \rangle) \ , \end{aligned}$$

Notation $S^{\hat{f}}$ is used here since this transform is similar the the Walsh transform of the real-valued counterpart of a Boolean function (see Sect. 2.2). Both S^f and $S^{\hat{f}}$ transforms can be computed using the nth-order fast Fourier transform algorithm with the complexity equivalent to $O(nq^n)$ arithmetic operations in $\mathbf{C}$. A more efficient algorithm that allows faster computation of $S^{\hat{f}}$ requiring just $n(p-1)p^{n+1}$ integer additions was devised in [ZCG99] for the particular case of functions over a prime field $\mathrm{GF}(p)$ that take on its values in $P = \mathrm{GF}(p)$. The Walsh transform of functions over $\mathrm{GF}(q)$ appears to be a useful tool when analyzing their cryptographic properties. In particular, the best linear approximation of f over a prime field can be easily obtained from the $S^{\hat{f}}$ transform (see [ZCG99]). The $S^{\hat{f}}$ transform also provides the following characterization of correlation immune functions over $\mathrm{GF}(q)$ with all definitions and proof presented in [CC99]. This characterization generalizes the Xiao-Massey criterion [XM88] to the case of functions over finite fields. By the weight of a vector α over $\mathrm{GF}(q)$ (denoted as $wt(\alpha)$) we mean the number of nonzero coordinates in α. The proof provided below is built along novel lines and is different from the one in [CC99].

Theorem 2.13 *The function f of n variables over* $\mathrm{GF}(q)$ *taking on its values in a finite field P is mth-order correlation immune for $1 \le m \le n$ if and only if for any nontrivial additive character χ of P and all $\alpha \in \mathrm{GF}(q)^n$ such that $1 \le wt(\alpha) \le m$ the Walsh transform of f satisfies the identity $S_{\hat{f}}(\alpha) = 0$. Moreover, f is m-resilient if and only if it additionally satisfies $S_{\hat{f}}(0) = 0$ for any nontrivial additive character χ of P.*

Proof: In a similar way as in Note 2.5 and using (2.28) one can prove that $S_{\hat{f}}(\alpha) = 0$ for any nontrivial additive character χ of P (remember that χ is used in the definition of the $S^{\hat{f}}$ transform) and for any $\alpha \in \mathrm{GF}(q)^n$ with $1 \le wt(\alpha) \le m$ if and only if for any set of indices $i_1, \ldots, i_m$ with $1 \le i_1 < \cdots < i_m \le n$ and for any nontrivial additive character χ of P

$$w(\boldsymbol{y}) = \sum_{\boldsymbol{z} \in \mathrm{GF}(q)^{n-m}} \chi\left(f_{i_1,\ldots,i_m}^{y_1,\ldots,y_m}(\boldsymbol{z})\right) \equiv \frac{S_w(0)}{q^m} = \frac{1}{q^m} \sum_{\boldsymbol{x} \in \mathrm{GF}(q)^n} \chi(f(\boldsymbol{x})) \tag{2.29}$$

for any $\boldsymbol{y} = (y_1, \ldots, y_m) \in \mathrm{GF}(q)^m$. Note that for the trivial additive character χ_0 of P obviously

$$\sum_{\boldsymbol{z} \in \mathrm{GF}(q)^{n-m}} \chi_0\left(f_{i_1,\ldots,i_m}^{y_1,\ldots,y_m}(\boldsymbol{z})\right) = \frac{1}{q^m} \sum_{\boldsymbol{x} \in \mathrm{GF}(q)^n} \chi_0(f(\boldsymbol{x})) = q^{n-m} \quad .$$

Now we need the following basic identity for additive characters (see [LN83, eq. 5.4]). For any $g, h \in P$ holds

$$\frac{1}{|P|} \sum_{\chi \in P^\wedge} \chi(g)\overline{\chi}(h) = \begin{cases} 1, & \text{if } \quad g = h \ , \\ 0, & \text{otherwise} \ , \end{cases} \tag{2.30}$$

where $P^\wedge$ denotes the group of additive characters of P. Also note that the linear system of $|P|$ equations $\sum_{\chi \in P^\wedge} \overline{\chi}(p) x_\chi = d \quad (p \in P)$ has a unique solution in $\mathbf{C}$, namely $x_{\chi_0} = d$ and $x_\chi = 0$ for $\chi \neq \chi_0$ (the system matrix is invertible with the same argument as was used to prove the invertibility of (2.26)). Therefore, (2.29) holds if and only if for any indices $i_1, \ldots, i_m$, any $(y_1, \ldots, y_m) \in \mathrm{GF}(q)^m$ and $p \in P$ the corresponding correlation coefficient is equal to

$$\begin{aligned}
&\Pr(X_{i_1} = y_1, \ldots, X_{i_m} = y_m \mid f(\boldsymbol{X}) = p) = \\
&= \frac{\#\{\boldsymbol{z} \in \mathrm{GF}(q)^{n-m} : f_{i_1,\ldots,i_m}^{y_1,\ldots,y_m}(\boldsymbol{z}) = p\}}{\#\{\boldsymbol{x} \in \mathrm{GF}(q)^n : f(\boldsymbol{x}) = p\}} \overset{(2.30)}{=} \\
&= \frac{\sum_{\boldsymbol{z}\in\mathrm{GF}(q)^{n-m}} \sum_{\chi\in P^\wedge} \chi\left(f_{i_1,\ldots,i_m}^{y_1,\ldots,y_m}(\boldsymbol{z})\right) \overline{\chi}(p)}{\sum_{\boldsymbol{x}\in\mathrm{GF}(q)^n} \sum_{\chi\in P^\wedge} \chi(f(\boldsymbol{x}))\overline{\chi}(p)} \overset{(2.29)}{=} \frac{1}{q^m}
\end{aligned}$$

and this is equivalent to function f being mth-order correlation immune.

Finally, for any *nontrivial* additive character χ of P holds

$$S_{\hat{f}}(0) = \sum_{\boldsymbol{x}\in\mathrm{GF}(q)^n} \chi(f(\boldsymbol{x})) = 0$$

if and only if for any $p \in P$

$$\#\{\boldsymbol{x} \in \mathrm{GF}(q)^n : f(\boldsymbol{x}) = p\} \overset{(2.30)}{=} \frac{1}{|P|} \sum_{\boldsymbol{x}\in\mathrm{GF}(q)^n} \sum_{\chi\in P^\wedge} \chi(f(\boldsymbol{x}))\overline{\chi}(p) = \frac{q^n}{|P|}$$

and this is equivalent to function f being balanced. □

2.5 Probabilistic Function of a Boolean Function

Consider a nonlinear combination generator with random inputs. This is the arrangement where n sequences, each one consisting of nonuniform (i.e., biased), independent and identically distributed random binary variables, are combined with a Boolean function. Random variables in the sequences are distributed identically, although the bias may be different for each of the sequences. The objective of such an arrangement is to produce an output sequence having better, compared to the input sequences, algebraic and statistical characteristics relevant to a key stream.

In this section we provide an efficient method for estimating the bias of the output sequence if the biases of the input sequences are known. Of course, the output bias also depends on the combining function that is used in the arrangement. Therefore, is seems reasonable to introduce a characteristic that would allow to compare Boolean functions against their ability for compensating a biased distribution of the input bits. Here we show that the appropriately chosen combining function can produce the output distributed with the bias that by the order of magnitude is smaller than the biases of the inputs.

Definition 2.14 *Let $f(x_1, \ldots, x_n)$ be a Boolean function of n variables. Assume that $\boldsymbol{X} = (X_1, \ldots, X_n)$ is an n-tuple consisting of independent random binary variables with $\Pr(X_i = 1) = p_i$ for $i = 1, \ldots, n$. Then function $F_f(p_1, \ldots, p_n) = \Pr(f(\boldsymbol{X}) = 1)$ is called the probabilistic function of f.*

Until recently, the problems related to the probabilistic function of a Boolean function were studied mostly in the area of random testing of digital circuits [KB81] but not in cryptography. The only paper posing these problems in a cryptographic context is [Mir02], where estimates of the maximum bias of the distribution of the output bits were made. Our approach allows to obtain an explicit polynomial expression for this bias.

From Definition 2.14 it follows that $F_f(p_1, \dots, p_n) = \sum_{\beta\colon f(\beta)=1} \Pr(\boldsymbol{X} = \beta)$ and if $\beta = (\beta_1, \dots, \beta_n)$ then $\Pr(\boldsymbol{X} = \beta) = \prod_{i=1}^{n} p_i^{\beta_i}(1-p_i)^{1-\beta_i}$. Thus, $F_f(p_1, \dots, p_n)$ is a polynomial of n variables $p_1, \dots, p_n$ with integer coefficients.

Further, let $D_f(x_1, \dots, x_n)$ denote the NNF of f, i.e., the real-valued, exponent-free (in variables) polynomial of n variables with integer coefficients such that

$$D_f(x_1, \dots, x_n) = f(x_1, \dots, x_n) \quad \text{for any} \quad (x_1, \dots, x_n) \in \mathrm{GF}(2)^n \ . \qquad (2.31)$$

Polynomial D_f can be expressed in the following canonical form

$$D_f(x_1, \dots, x_n) = \sum_{i=0}^{2^n-1} a_i \left(\prod_{j=1,\dots,n\colon \alpha_i^j=1} x_j \right) \ ,$$

where $\alpha_i = (\alpha_i^1, \dots, \alpha_i^n)$ is the n-bit binary expansion of i and $a_i \in \mathbf{Z}$. Then, since (2.31) holds, the integer coefficients a_i form a solution of the following system of linear equations

$$M(a_0, \dots, a_{2^n-1})^T = (f(0, \dots, 0), \dots, f(1, \dots, 1))^T \ ,$$

where $M = (m_{i,j})_{2^n \times 2^n}$ $(i, j = 0, \dots, 2^n - 1)$ is a nondegenerate triangular $\{0, 1\}$-matrix with $m_{i,j} = 1$ if and only if the positions of ones in the n-bit binary expansion of j are a subset of those in the binary expansion of i (in particular, it is necessary that $j \leq i$). Therefore, this system has a unique solution and that proves the *uniqueness* of the NNF polynomial D_f (see [CG99, Proposition 1]). Moreover, the coefficient vector of D_f can be obtained by means of the arithmetic transform of f (see Sect. 2.2).

Identities $\overline{x} = 1 - x$, $x_1 \wedge x_2 = x_1x_2$, $x_1 \vee x_2 = x_1 + x_2 - x_1x_2$ and $x_1 \oplus x_2 = x_1 + x_2 - 2x_1x_2$ convert elementary Boolean operations into integer expressions. Thus, using these identities any formula representing $f(x_1, \dots, x_n)$ in the basis $\{^-, \wedge, \vee, \oplus\}$ (for instance, the ANF) can be transformed into the real-valued polynomial of n variables with integer coefficients that satisfies (2.31). Moreover, if we assume that $x_i^2 \equiv x_i$ $(i = 1, \dots, n)$ then every variable in the constructed polynomial has degree at most 1 and, therefore, by the uniqueness, the polynomial is equal to D_f. That provides an alternative way for constructing the NNF polynomial D_f starting from a formula representing the Boolean function. The following proposition is similar to [KB81, Theorem 7].

Proposition 2.15 *For any Boolean function* $f(x_1, \ldots, x_n)$ *and arbitrary values* $p_1, \ldots, p_n$ *with* $0 \leq p_i \leq 1$ *for all* $i = 1, \ldots, n$

$$F_f(p_1, \ldots, p_n) = D_f(p_1, \ldots, p_n) \ .$$

Proof: To prove this identity we apply induction on n.

Let $n = 1$. Then function f is one of the following four functions of a single variable

$$f_0 \equiv 0, \quad f_1 = x_1, \quad f_2 = \overline{x_1}, \quad f_3 \equiv 1 \ .$$

But

$$\begin{aligned}
\Pr(f_0 = 1) &= 0 = D_{f_0} \\
\Pr(f_1 = 1) &= \Pr(X_1 = 1) = p_1 = D_{f_1}(p_1) \\
\Pr(f_2 = 1) &= \Pr(X_1 = 0) = 1 - p_1 = D_{f_2}(p_1) \\
\Pr(f_3 = 1) &= 1 = D_{f_3} \ .
\end{aligned}$$

Now, supposing that the proposition is true for $n = l - 1$, we prove it for $n = l$. It is easy to see that the following decomposition of function f into subfunctions holds:

$$f(x_1, \ldots, x_l) = \overline{x_1} f_1^0(x_2, \ldots, x_l) \oplus x_1 f_1^1(x_2, \ldots, x_l) \ .$$

According to the induction hypothesis, $F_{f_1^i}(p_2, \ldots, p_l) = D_{f_1^i}(p_2, \ldots, p_l)$ for $i = 0, 1$. Also note that

$$D_f(x_1, \ldots, x_l) = (1 - x_1) D_{f_1^0}(x_2, \ldots, x_l) + x_1 D_{f_1^1}(x_2, \ldots, x_l)$$

since $\overline{x_1} f_1^0(x_2, \ldots, x_l) x_1 f_1^1(x_2, \ldots, x_l) \equiv 0$ on $\mathrm{GF}(2)^n$. On the other hand, by the rule of total probability

$$F_f(p_1, \ldots, p_l) = (1 - p_1) F_{f_1^0}(p_2, \ldots, p_l) + p_1 F_{f_1^1}(p_2, \ldots, p_l) \ .$$

Thus, $F_f(p_1, \ldots, p_l) = D_f(p_1, \ldots, p_l)$ for any $p_1, \ldots, p_n$ with $0 \leq p_i \leq 1$ for all $i = 1, \ldots, n$. □

Let w_i $(i = 0, \ldots, n)$ denote the number of vectors having weight i that are also the members of the support of a Boolean function f of n variables. Then vector $(w_0, \ldots, w_n)$ is called the *weight distribution* of function f.

Assume first that $p_1 = \cdots = p_n = p = 1/2 + \delta$, where $\delta \in (-1/2, 1/2)$ is the bias of the distribution of the random variable x_i $(i = 1, \ldots, n)$. Then, since $\sum_{i=0}^n w_i = wt(f)$,

$$\begin{aligned}
F_f(p) &= \sum_{i=0}^n w_i p^i (1 - p)^{n-i} = \sum_{i=0}^n w_i \left(\frac{1}{2} + \delta\right)^i \left(\frac{1}{2} - \delta\right)^{n-i} = \\
&= d_1 \delta + d_2 \delta^2 + \cdots + d_n \delta^n + \frac{1}{2^n} wt(f) \ ,
\end{aligned}$$

where $d_1, \ldots, d_n$ are some real values. Let $\Delta_f(\delta) = F_f(1/2+\delta)-1/2$ denote the bias of the distribution of the function f output. In particular, if function f is balanced then $\Delta_f(\delta) = d_1\delta + d_2\delta^2 + \cdots + d_n\delta^n$.

In case when the values of $p_1, \ldots, p_n$ are different let $p_i = 1/2+\delta_i$ $(i = 1, \ldots, n)$. The bias of the distribution of the function f output is defined in a similar way as the polynomial of n variables

$$\Delta_f(\delta_1, \ldots, \delta_n) = F_f\left(\frac{1}{2}+\delta_1, \ldots, \frac{1}{2}+\delta_n\right) - \frac{1}{2} \ . \tag{2.32}$$

If function f is balanced then the constant term of polynomial $\Delta_f(\delta_1, \ldots, \delta_n)$ is equal to

$$\Delta_f(0, \ldots, 0) = F_f\left(\frac{1}{2}, \ldots, \frac{1}{2}\right) - \frac{1}{2} = \frac{wt(f)}{2^n} - \frac{1}{2} = 0 \ .$$

And the other way around: if the constant term of polynomial $\Delta_f(\delta_1, \ldots, \delta_n)$ is equal to zero then function f is balanced. We will call polynomial $\Delta_f(\delta_1, \ldots, \delta_n)$ the *bias polynomial* of function f.

The coefficient vector of the bias polynomial is equal to the probabilistic transform of function f (see Sect. 2.2) except for the initial coordinate of Δ^f that has to be corrected by subtracting 1/2. On the other hand, combining (2.9) and (2.14), the coefficient vector can be expressed as $-\frac{1}{2^{n+1}}\begin{pmatrix} 1 & 0 \\ 0 & -2 \end{pmatrix}^{[n]} S^{\hat{f}}$. Coefficients of the bias polynomial can also be estimated using identities (2.15) that are equivalent to [Mir02, Theorem 3.1].

Definition 2.16 *For $k \in \{1, \ldots, n\}$ a Boolean function f is called k-compensating if the bias polynomial of f does not contain product terms having degree lower than k.*

Note that any balanced Boolean function is 1-compensating. For the particular case when $p_1 = \cdots = p_n$, Definition 2.16 means that function f is k-compensating if it is balanced and $d_1 = \cdots = d_{k-1} = 0$. In other words, if the input of a k-compensating Boolean function is nonuniform with bias δ then the bias on its output by the order of magnitude is at most δ^k. The following proposition provides a method for constructing k-compensating functions.

Proposition 2.17 *Let $f(x_1, \ldots, x_n) = f_1(x_1, \ldots, x_k) \oplus f_2(x_{k+1}, \ldots, x_n)$, where $k \in \{1, \ldots, n-1\}$. Then*

$$F_f(p_1, \ldots, p_n) - \frac{1}{2} = -2\left(F_{f_1}(p_1, \ldots, p_k) - \frac{1}{2}\right)\left(F_{f_2}(p_{k+1}, \ldots, p_n) - \frac{1}{2}\right) \ ,$$

and thus $\Delta_f(\delta_1, \ldots, \delta_n) = -2\Delta_{f_1}(\delta_1, \ldots, \delta_k)\Delta_{f_2}(\delta_{k+1}, \ldots, \delta_n)$.

Proof: Since $f_1 \oplus f_2 = f_1 + f_2 - 2f_1f_2$,

$$\begin{aligned} D_f(x_1, \ldots, x_n) &= D_{f_1}(x_1, \ldots, x_k) + D_{f_2}(x_{k+1}, \ldots, x_n) - \\ &- 2D_{f_1}(x_1, \ldots, x_k)D_{f_2}(x_{k+1}, \ldots, x_n) \ . \end{aligned}$$

Therefore, by Proposition 2.15,

$$\begin{aligned} F_f(p_1,\ldots,p_n) &= F_{f_1}(p_1,\ldots,p_k) + F_{f_2}(p_{k+1},\ldots,p_n) - \\ &- 2F_{f_1}(p_1,\ldots,p_k)F_{f_2}(p_{k+1},\ldots,p_n)\ , \end{aligned}$$

that is equivalent to the statement of the proposition. □

The following corollary is obvious.

Corollary 2.18 *Let $f(x_1,\ldots,x_n)$ be a Boolean function of n variables.*

(i) If $f(x_1,\ldots,x_n) = f_1(x_1,\ldots,x_k) \oplus f_2(x_{k+1},\ldots,x_n)$, function f_1 is k_1-compensating and function f_2 is k_2-compensating then function f is (k_1+k_2)-compensating;

(ii) if $f(x_1,\ldots,x_n) = x_{i_1} \oplus \cdots \oplus x_{i_k} \oplus a_0$ then

$$F_f(p_1,\ldots,p_n) - \frac{1}{2} = (-1)^{a_0}(-2)^{k-1}\delta_{i_1}\cdot\cdots\cdot\delta_{i_k}\ .$$

In other words, an affine function consisting of k linear terms is k-compensating.

The following proposition, which is similar to [Mir02, Theorem 3.2], easily follows from (2.14) and Note 2.5. The proof provided below is not based on the previous results and is given to keep this section self-contained.

Proposition 2.19 *A Boolean function $f(x_1,\ldots,x_n)$ is k-resilient if and only if it is $(k+1)$-compensating.*

Proof: For $k=0$ the statement is obvious since a 0-resilient function is balanced by the definition and, therefore, it is 1-compensating. Now let $k>0$.

Let $\Pr(X_i=1) = p_i = 1/2 + \delta_i$ $(i=1,\ldots,n)$. By Definition 2.14,

$$\begin{aligned} &F_f(p_1,\ldots,p_n) = \Pr(f(\boldsymbol{X})=1) = \\ &= \sum_{(\beta_1,\ldots,\beta_k)\in \mathrm{GF}(2)^k} p_1^{\beta_1}\cdot\cdots\cdot p_k^{\beta_k}\Pr(f(\beta_1,\ldots,\beta_k,X_{k+1},\ldots,X_n)=1) = \\ &= \sum_{(\beta_1,\ldots,\beta_k)\in \mathrm{GF}(2)^k} p_1^{\beta_1}\cdot\cdots\cdot p_k^{\beta_k} F_f(\beta_1,\ldots,\beta_k,p_{k+1},\ldots,p_n)\ , \end{aligned} \tag{2.33}$$

where $p_i^{\beta_i} = \begin{cases} p_i, & \text{if} \quad \beta_i = 1, \\ 1-p_i, & \text{if} \quad \beta_i = 0 \end{cases}$ $(i=1,\ldots,n)$.

Function $F_f(\beta_1,\ldots,\beta_k,p_{k+1},\ldots,p_n)$ is a probabilistic function of subfunction $f^{\boldsymbol{\beta}} = f_{1,\ldots,k}^{\beta_1,\ldots,\beta_k}(x_{k+1},\ldots,x_n)$, where $\boldsymbol{\beta} = (\beta_1,\ldots,\beta_k)$, and

$$F_f(\beta_1,\ldots,\beta_k,p_{k+1},\ldots,p_n) = \Delta_{f^{\boldsymbol{\beta}}}(\delta_{k+1},\ldots,\delta_n) + \frac{1}{2}\ . \tag{2.34}$$

Therefore,

$$
\begin{aligned}
&F_f\left(\frac{1}{2}+\delta_1,\dots,\frac{1}{2}+\delta_n\right) \overset{(2.33,2.34)}{=} \\
&= \frac{1}{2} \sum_{(\beta_1,\dots,\beta_k)\in \mathrm{GF}(2)^k} \left(\frac{1}{2}+\delta_1\right)^{\beta_1} \cdot \dots \cdot \left(\frac{1}{2}+\delta_k\right)^{\beta_k} + \\
&+ \sum_{(\beta_1,\dots,\beta_k)\in \mathrm{GF}(2)^k} \left(\frac{1}{2}+\delta_1\right)^{\beta_1} \cdot \dots \cdot \left(\frac{1}{2}+\delta_k\right)^{\beta_k} \Delta_{f^{\boldsymbol{\beta}}}(\delta_{k+1},\dots,\delta_n) = \\
&= \frac{1}{2} + \sum_{(\beta_1,\dots,\beta_k)\in \mathrm{GF}(2)^k} \left(\frac{1}{2}+\delta_1\right)^{\beta_1} \cdot \dots \cdot \left(\frac{1}{2}+\delta_k\right)^{\beta_k} \Delta_{f^{\boldsymbol{\beta}}}(\delta_{k+1},\dots,\delta_n) \ . (2.35)
\end{aligned}
$$

Assume that function f is k-resilient. Then, by (2.20), its subfunction $f^{\boldsymbol{\beta}}$ is balanced for any $(\beta_1,\dots,\beta_k) \in \mathrm{GF}(2)^k$ and for the probabilistic function of $f^{\boldsymbol{\beta}}$

$$F_f(\beta_1,\dots,\beta_k,1/2,\dots,1/2) = 1/2 \ .$$

Now, by (2.34), $\Delta_{f^{\boldsymbol{\beta}}}(0,\dots,0) = F_f(\beta_1,\dots,\beta_k,1/2,\dots,1/2) - 1/2 = 0$ and thus the constant term of bias polynomial $\Delta_{f^{\boldsymbol{\beta}}}(\delta_{k+1},\dots,\delta_n)$ is equal to zero. If we look at function $F_f(1/2+\delta_1,\dots,1/2+\delta_n)$ as a polynomial of n variables then it is clear that all its product terms depend on at least one of the variables $\delta_{k+1},\dots,\delta_n$ and its constant term is equal to 1/2.

Further, by (2.20), for a k-resilient function f any subfunction $f^{\beta_{i_1},\dots,\beta_{i_k}}_{i_1,\dots,i_k}$ with $1 \le i_1 < \dots < i_k \le n$ and any $(\beta_{i_1},\dots,\beta_{i_k}) \in \mathrm{GF}(2)^k$ is balanced. In a similar way it can be proved that all product terms in F_f depend on at least one of the variables contained in the subset $\{\delta_1,\dots,\delta_n\} \setminus \{\delta_{i_1},\dots,\delta_{i_k}\}$. The minimal set containing representatives from all these subsets contains $k+1$ elements. Therefore, all product terms in F_f depend on at least $k+1$ variables. Thus, by (2.32), bias polynomial $\Delta_f(\delta_1,\dots,\delta_n)$ does not contain product terms having degree lower than $k+1$.

Now assume that function f is $(k+1)$-compensating. Then, in particular, all product terms of polynomial $\Delta_f(\delta_1,\dots,\delta_n)$ depend on at least one of the variables $\delta_{k+1},\dots,\delta_n$ and its constant term is equal to zero. Therefore,

$$
\begin{aligned}
\Delta_f(\delta_1,\dots,\delta_k,0,\dots,0) &= F_f\left(\frac{1}{2}+\delta_1,\dots,\frac{1}{2}+\delta_k,\frac{1}{2},\dots,\frac{1}{2}\right) - \frac{1}{2} \overset{(2.35)}{=} \\
&= \sum_{(\beta_1,\dots,\beta_k)\in \mathrm{GF}(2)^k} \left(\frac{1}{2}+\delta_1\right)^{\beta_1} \cdot \dots \cdot \left(\frac{1}{2}+\delta_k\right)^{\beta_k} \Delta_{f^{\boldsymbol{\beta}}}(0,\dots,0) \equiv 0
\end{aligned}
$$

and polynomial $\Delta_f(\delta_1,\dots,\delta_k,0,\dots,0)$ is identical to zero. It is easy to see that the coefficient of the multiple term $\delta_{i_1} \cdot \dots \cdot \delta_{i_t}$ in the canonical form of this polynomial is equal to

$$\frac{1}{2^{k-t}} \sum_{(\beta_1,\dots,\beta_k)\in \mathrm{GF}(2)^k} (-1)^{t-(\beta_{i_1}+\dots+\beta_{i_t})} \Delta_{f^{\boldsymbol{\beta}}}(0,\dots,0) = 0$$

and this holds for any $0 \leq t \leq k$ and $0 \leq i_1 < \cdots < i_t \leq k$. Thus, we have a system of 2^k linear equations in 2^k unknowns $\Delta_{f^{\boldsymbol{\beta}}}(0,\ldots,0)$ for $\boldsymbol{\beta} \in \mathrm{GF}(2)^k$ with the matrix consisting of elements $m_{i,j} = (-1)^{wt(\alpha_i)-\langle \alpha_i,\alpha_j \rangle}$ for $i,j = 0,\ldots,2^k-1$, where α_i and α_j are k-bit binary expansions of i and j respectively. This matrix is a Hadamard matrix [MS96, p. 44] and is nondegenerate. Therefore, this system has a unique solution, namely, $\Delta_{f^{\boldsymbol{\beta}}}(0,\ldots,0) = 0$ for any $(\beta_1,\ldots,\beta_k) \in \mathrm{GF}(2)^k$. Thus, subfunctions $f^{\boldsymbol{\beta}}$ are balanced. In a similar way it can be proved that any subfunction $f_{i_1,\ldots,i_k}^{\beta_{i_1},\ldots,\beta_{i_k}}$ with $1 \leq i_1 < \cdots < i_k \leq n$ and any $(\beta_{i_1},\ldots,\beta_{i_k}) \in \mathrm{GF}(2)^k$ is balanced. Then, by (2.20), function f is k-resilient. □

Concluding this section let us stress again that the nonuniformity (i.e., the bias) of the output of a Boolean function is equal to the value of the corresponding bias polynomial when the biases of the inputs are taken as the arguments of this polynomial. The coefficients of the bias polynomial can be efficiently estimated by the probabilistic transform of the function. If a Boolean function is k-compensating (or $(k-1)$-resilient equivalently) then the order of magnitude of the output bias is at least the kth-order product of the input biases. On the other hand, due to Siegenthaler's inequality, Proposition 2.19 means that Boolean functions with high algebraic degree have low compensating degree and vice versa. This fact underlines again the need for optimizing the algebraic degree with correlation and compensating properties when constructing secure Boolean functions.

2.6 Analyzing Cryptographic Properties of Boolean Functions Using Equivalence Relations

Consider a cipher system containing a building block that uses Boolean functions, concrete form of which constitutes the long-term key. Example of such a system is GOST block cipher with key-dependent S-boxes (see [Sch96, pp. 331-334]). When generating these long-term keys one has to chose them from the set of cryptographically secure transformations and, at the same time, it is necessary to guarantee that the key space is large enough to prevent the exhaustive search. Thus, we have to estimate the number of Boolean functions that fulfil some relevant cryptographic criteria. Searching the set of all Boolean functions of n variables can be a computationally infeasible task even if the number of variables is relatively small (starting from 6) since the total number of Boolean functions of n variables is superexponential in n. One of the ways to surmount this difficulty is to define an equivalence relation on Boolean functions under which the cryptographic criterion being considered remains invariant. Then it would be sufficient just to estimate the cardinality of the equivalence class that contains "good" functions. On the other hand, a useful criterion should remain invariant under some equivalences and this will be discussed further in this section.

Let $\mathcal{F}_n$ denote the set of all Boolean functions of n variables and S_m denote the symmetric group of order m. By a transformation group of $\mathrm{GF}(2)^n$ we mean the group consisting of bijections of $\mathrm{GF}(2)^n$ to itself.

Definition 2.20 *Let f and h be Boolean functions of n variables and G be a transformation group of $\mathrm{GF}(2)^n$. Then function f is said to be equivalent to h under the group G (or f is G-equivalent to h) if there exists a pair $(g, \sigma) \in G \times S_2$ such that $h(\boldsymbol{x}) = \sigma(f(g(\boldsymbol{x})))$ for all $\boldsymbol{x} \in \mathrm{GF}(2)^n$.*

Note also that if $\sigma \in S_2$ is not the identity permutation then applying σ results in inverting the values of $f(g(\boldsymbol{x}))$. Therefore, by Definition 2.20, any functions being the inverses of each other are always G-equivalent.

Definition 2.21 *If f is a Boolean functions of n variables and G is a transformation group of $\mathrm{GF}(2)^n$ then the set*

$$\mathrm{I}_G(f) = \{(g, \sigma) \in G \times S_2 \mid \sigma(f(g(\boldsymbol{x}))) = f(\boldsymbol{x}) \ \forall \boldsymbol{x} \in \mathrm{GF}(2)^n\}$$

is called the inertia group of function f in the group G.

It is easy to see that the relation defined is a true equivalence relation and the set $\mathrm{I}_G(f)$ is a group. Thus, the set $\mathcal{F}_n$ is partitioned into disjoint classes of G-equivalent elements. Let $[f]_G$ denote the class of G-equivalent functions containing f.

Let $\mathcal{G} = G \times S_2$ and let ϕ be the isomorphic imbedding of the group $\mathcal{G}$ into the symmetrical group of $\mathcal{F}_n$ defined as $\phi(g, \sigma) = \mathrm{A}_\sigma^g$, where $\mathrm{A}_\sigma^g(f(\boldsymbol{x})) = \sigma(f(g(\boldsymbol{x})))$. Then an inertia group $\mathrm{I}_G(f)$ is actually the stabilizer of the element f in the group $\phi(\mathcal{G})$ and $[f]_G$ is the orbit of f under the group $\phi(\mathcal{G})$. Thus,

$$|\mathcal{G}| = |\phi(\mathcal{G})| = |[f]_G| \cdot |\mathrm{I}_G(f)| \ . \tag{2.36}$$

Extreme cases occur when $|[f]_G| = 1$ or $|[f]_G| = |\mathcal{G}|$. In the first case, when the inertia group of function f in the group G coincides with the whole group $\mathcal{G}$, function f is said to be *invariant* under the group G. In the second case function f is said to have the *trivial* inertia group in the group G. It is easy to see that invariant functions do not exist. Indeed, take the group identity of G for g and bit inversion for σ, then for an invariant function f the identity $\sigma(f(g(\boldsymbol{x}))) = f(\boldsymbol{x}) \oplus 1 \equiv f(\boldsymbol{x})$ should hold, which is impossible.

The group $\phi(\mathcal{G})$ is the transformation group of $\mathcal{F}_n$ isomorphic to $\mathcal{G}$ and all pairs of G-equivalent functions are exactly the pairs equivalent under the group $\phi(\mathcal{G})$ if equivalence is defined like in [KN63]. Therefore, Definitions 2.20 and 2.21 naturally follow from the more general ones given in [KN63] for an arbitrary transformation group of $\mathcal{F}_n$. Any transformation A_σ^g can be presented as a product of two transformations: function domain permutation defined by g and function range permutation defined by σ. Permutations of the range are limited to the identity transformation and complementation. Transformations of $\mathcal{F}_n$ having the type of A_σ^g are called *substitutional.*

In Sect. 2.1 we stated the number of necessary criteria for secure Boolean functions to be used in key-stream generators. According to [MS89], a useful criterion should remain invariant under simple transformation groups. This is motivated by the fact that a function is considered insecure if it does not satisfy this minimal set of criteria or it is G-equivalent to a cryptographically insecure function, where G is some simple transformation group (e.g., the group of linear/affine transformations). This is the reason why simple transformations of the domain are also called cryptographically weak. The attacks on cryptosystems that use Boolean functions often work with the same complexity when the functions are replaced by their equivalents under weak transformations. For instance, if a function being the building block of a cipher has an affine equivalent then the cipher can be attacked with the same complexity as if it was using this affine equivalent instead of the original function. When considering nonlinearity criteria for secure Boolean functions, the general affine group and its following subgroups are usually regarded as weak transformations:

(a) the *symmetric group* S_n of order $n!$ containing all permutations acting on the variables $\{x_1, \ldots, x_n\}$, i.e.,

$$S_n = \{g_{i_1,\ldots,i_n} \mid g_{i_1,\ldots,i_n}(x_1, \ldots, x_n) = (x_{i_1}, \ldots, x_{i_n}),\ (i_1, \ldots, i_n) \in \Pi_n)\}\ ,$$

where Π_n is the set of all permutations of degree n;

(b) the *offset group* Σ_n of order 2^n containing all complementations of the variables $\{x_1, \ldots, x_n\}$, i.e.,

$$\Sigma_n = \{g^\alpha \mid g^\alpha(x_1, \ldots, x_n) = \left(x_1^{\overline{\alpha_1}}, \ldots, x_n^{\overline{\alpha_n}}\right),\ \alpha = (\alpha_1, \ldots, \alpha_n) \in \mathrm{GF}(2)^n\}\ ;$$

(c) the *Jevons group* Q^n of order $2^n n!$ containing all permutations and complementations of the variables $\{x_1, \ldots, x_n\}$, i.e.,

$$\begin{aligned} Q_n = \{g^\alpha_{i_1,\ldots,i_n} \ \mid\ & g^\alpha_{i_1,\ldots,i_n}(x_1, \ldots, x_n) = \left(x_{i_1}^{\overline{\alpha_{i_1}}}, \ldots, x_{i_n}^{\overline{\alpha_{i_n}}}\right),\ (i_1, \ldots, i_n) \in \Pi_n, \\ & \alpha = (\alpha_1, \ldots, \alpha_n) \in \mathrm{GF}(2)^n\}\ ; \end{aligned}$$

(d) the *general linear group* $\mathrm{GL}_n(2)$ of all linear transformations from $\mathrm{GF}(2)^n$ into itself, i.e.,

$$\mathrm{GL}_n(2) = \{g_A \mid g_A(x_1, \ldots, x_n) = (x_1, \ldots, x_n)A,\ A \in M_n^*(2)\}\ ,$$

where $M_n^*(2)$ is the set of n-dimensional invertible square matrices over $\mathrm{GF}(2)$;

(e) the *general affine group* $\mathrm{AGL}_n(2)$ of all affine transformations from $\mathrm{GF}(2)^n$ into itself, i.e.,

$$\begin{aligned} \mathrm{AGL}_n(2) = \{g_{A,\alpha} \ \mid\ & g_{A,\alpha}(x_1, \ldots, x_n) = (x_1, \ldots, x_n)A \oplus \alpha,\ A \in M_n^*(2), \\ & \alpha \in \mathrm{GF}(2)^n\}\ . \end{aligned}$$

The general formulae for estimating the number of equivalence classes under the groups S_n, Σ_n and Q_n were derived in [Har63] and for the groups $\mathrm{GL}_n(2)$ and $\mathrm{AGL}_n(2)$ the relevant formulae were derived in [Har64].

The general affine group contains all the other transformation groups that have been defined above, as subgroups. The lattice of these subgroups is shown in Fig. 2.1. In general, if a transformation group G' is the subgroup in a transformation group G then G'-equivalence classes are decompositions of G-equivalence classes.

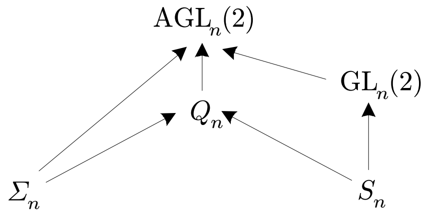


Figure 2.1: The lattice of subgroups in $\mathrm{AGL}_n(2)$

By the *generalized weight* of a Boolean function of n variables we will mean the unordered couple $(wt(f), 2^n - wt(f))$. Since G is a permutation group on $\mathrm{GF}(2)^n$, it is obvious that G-equivalent functions have the same generalized weight.

If equivalence transformations are limited just by domain permutations defined by the group G then the set of Boolean functions of n variables, invariant under the group G, corresponds to the set of functions being constant on each of the orbits of the group G. Thus, if m is the number of orbits of G then the number of invariant functions is equal to 2^m. The general linear group $\mathrm{GL}_n(2)$ has two orbits, namely D_0 containing only the zero-vector and D_1 containing the remaining $(2^n - 1)$ n-bit nonzero vectors. Thus, there exist four functions invariant under $\mathrm{GL}_n(2)$ and two of these are constant-functions. Groups Σ_n, Q_n and $\mathrm{AGL}_n(2)$ are transitive and, therefore, only constant-functions are invariant under these groups. The symmetric group S_n has $n+1$ orbits $D_0, \ldots, D_n$, where D_k is the set of all n-bit vectors having weight k. Thus, the number of functions invariant under S_n is equal to 2^{n+1}. These functions are also called *symmetric* functions of n variables.

Proposition 2.22 *If G is any of the above-defined transformation groups then G-equivalent functions have the same algebraic degree and nonlinearity. Moreover, the Walsh transforms of their real-valued counterparts are equal up to a permutation of the coefficients and the sign of the coefficients.*

Proof: It is sufficient to prove the claimed proposition only for the general affine group since this group contains all the other transformation groups that have been defined above, as subgroups. Also we may consider just two cases separately: permutations of the domain defined by $\mathrm{AGL}_n(2)$ and complementation of the function range.

Let $g_{A,\alpha} \in \mathrm{AGL}_n(2)$, $f(\boldsymbol{x})$ be a Boolean function of n variables and $h(\boldsymbol{x}) =$

$f(\boldsymbol{x}A \oplus \alpha)$. Then for any nonzero $\beta \in \mathrm{GF}(2)^n$

$$\begin{aligned}
S_h(\beta A^T) &\overset{(2.21)}{=} wt\left(h(\boldsymbol{x}) \oplus \langle \beta A^T, \boldsymbol{x}\rangle\right) - 2^{n-1} = wt(h(\boldsymbol{x}) \oplus \langle \beta, \boldsymbol{x}A\rangle) - 2^{n-1} = \\
&= wt(f(\boldsymbol{x}A \oplus \alpha) \oplus \langle \beta, \boldsymbol{x}A\rangle) - 2^{n-1} = \\
&= wt(f(\boldsymbol{y}) \oplus \langle \beta, \boldsymbol{y} \oplus \alpha\rangle) - 2^{n-1} = \\
&= \begin{cases} wt(f(\boldsymbol{y}) \oplus \langle \beta, \boldsymbol{y}\rangle) - 2^{n-1}, & \text{if} \quad \langle \alpha, \beta\rangle = 0, \\ 2^n - wt(f(\boldsymbol{y}) \oplus \langle \beta, \boldsymbol{y}\rangle) - 2^{n-1}, & \text{if} \quad \langle \alpha, \beta\rangle = 1 \end{cases} = \\
&= (-1)^{\langle \alpha, \beta\rangle} S_f(\beta) \ , \qquad (2.37)
\end{aligned}$$

where $\boldsymbol{y} = \boldsymbol{x}A \oplus \alpha$. If $\beta = 0$ then $S_h(0) = wt(h) = wt(f) = S_f(0)$. Using (2.9) it is easy to see that $S_{\hat{h}}(\beta A^T) = (-1)^{\langle \alpha, \beta\rangle} S_{\hat{f}}(\beta)$ for all $\beta \in \mathrm{GF}(2)^n$.

Thus, multiplication by the invertible matrix A^T defines a permutation of the coefficients mapping the Walsh transform of f to the Walsh transform of h. Some coefficients change their sign depending on the value of the scalar product $\langle \alpha, \beta\rangle$.

Finally, we have to check the case when $h(\boldsymbol{x}) = f(\boldsymbol{x}) \oplus 1$, i.e., when function f is complemented. Then for any nonzero $\beta \in \mathrm{GF}(2)^n$

$$\begin{aligned}
S_h(\beta) &\overset{(2.21)}{=} wt(h(\boldsymbol{x}) \oplus \langle \beta, \boldsymbol{x}\rangle) - 2^{n-1} = (2^n - wt(f(\boldsymbol{x}) \oplus \langle \beta, \boldsymbol{x}\rangle)) - 2^{n-1} = \\
&= 2^{n-1} - wt(f(\boldsymbol{x}) \oplus \langle \beta, \boldsymbol{x}\rangle) \overset{(2.21)}{=} -S_f(\beta) \ . \qquad (2.38)
\end{aligned}$$

If $\beta = 0$ then $S_h(0) = wt(h) = 2^n - wt(f) = 2^n - S_f(0)$. Using (2.9) it is easy to see that $S_{\hat{h}}(\beta) = -S_{\hat{f}}(\beta)$ for all $\beta \in \mathrm{GF}(2)^n$. Thus, a Walsh transform coefficient of the real-valued counterpart of h is equal to minus the respective Walsh transform coefficient of f.

To prove the claimed equality of algebraic degrees we first notice that when applying an equivalence transformation to f its degree cannot increase since terms of some degree k in the ANF of $f(\boldsymbol{x})$ cannot produce terms of degree higher than k in the ANF of $h(\boldsymbol{x}) = f(\boldsymbol{x}A \oplus \alpha)$. But the equivalence relation is symmetric and thus the degrees of $f(\boldsymbol{x})$ and $h(\boldsymbol{x})$ are equal (see [MS89, Theorem 2.4]). The case when $h(\boldsymbol{x}) = f(\boldsymbol{x}) \oplus 1$ is obvious.

Let us recall that the *nonlinearity* of Boolean function f of n variables (denoted as $nl(f)$) is defined as the minimum Hamming distance between f and the set of all affine functions of n variables. The nonlinearity of f can be estimated from its Walsh transform using the well-known identity

$$nl(f) = 2^{n-1} - \frac{1}{2} \max_{\alpha \in \mathrm{GF}(q)^n} |S_{\hat{f}}(\alpha)| \ . \qquad (2.39)$$

Therefore, $\mathrm{AGL}_n(2)$-equivalent functions have the same nonlinearity since their Walsh transforms consist of the same set of coefficients (assuming that values are taken in absolute magnitude). □

It easy to see that if G is any of the groups S_n, Σ_n and Q_n then G-equivalent functions have the same number of variables on which they are nondegenerate. As for the weight transform of G-equivalent functions, for any $g \in Q_n$, if $h(\boldsymbol{x}) = f(g(\boldsymbol{x}))$ then $\Theta_\beta^f = \Theta_{g(\beta)}^h$ and if $h(\boldsymbol{x}) = f(\boldsymbol{x}) \oplus 1$ then $\Theta_\beta^h = 2^n - \Theta_\beta^f$ for all $\beta \in \mathrm{GF}(2)^n$. Moreover, Q_n-equivalent functions have the same degree of correlation immunity (see [MS89, Theorem 2.5]) and by virtue of Proposition 2.19, they have the same compensating degree. Indeed, if function $h(\boldsymbol{x})$ is Q_n-equivalent to $f(\boldsymbol{x})$ then there exists a permutational matrix A and $\alpha \in \mathrm{GF}(2)^n$ such that $h(\boldsymbol{x}) = f(\boldsymbol{x}A \oplus \alpha)$ or $h(\boldsymbol{x}) = f(\boldsymbol{x}A \oplus \alpha) \oplus 1$. Then, by (2.37) and (2.38), for any nonzero $\beta \in \mathrm{GF}(2)^n$

$$S_h(\beta A^T) = \begin{cases} (-1)^{\langle \alpha,\beta \rangle} S_f(\beta), & \text{if} \quad h(\boldsymbol{x}) = f(\boldsymbol{x}A \oplus \alpha), \\ (-1)^{\langle \alpha,\beta \rangle + 1} S_f(\beta), & \text{if} \quad h(\boldsymbol{x}) = f(\boldsymbol{x}A \oplus \alpha) \oplus 1 \ . \end{cases} \tag{2.40}$$

Since A^T is a permutational matrix, vectors β and βA^T have the same weight and the criterion for the mth-order correlation immunity from Note 2.5 can be applied.

There is another important cryptographic property of Boolean functions that is worth mentioning in respect to equivalent transformations - the property to fulfil the Strict Avalanche Criterion (SAC). In [For89, Theorem 1] it was shown that a Boolean function $f(\boldsymbol{x})$ fulfils the SAC if and only if for any $i = 1, \ldots, n$ Walsh coefficients $S_{\hat{f}}(w)$ satisfy the identity

$$\sum_{\omega \in \mathrm{GF}(2)^n} (-1)^{\omega_i} S_{\hat{f}}^2(w) = 0 \ , \tag{2.41}$$

where $\omega = \{\omega_1, \ldots, \omega_n\}$. Note that the proof of Proposition 2.22, similarly to (2.40), also suggests that $S_{\hat{h}}^2(\beta A^T) = S_{\hat{f}}^2(\beta)$ for any $\beta \in \mathrm{GF}(2)^n$ and Q_n-equivalent functions $f(\boldsymbol{x})$ and $h(\boldsymbol{x})$. Thus, using (2.41), it is easy to see that the property to fulfil the SAC remains invariant under G-equivalent transformations for any of the transformation groups S_n, Σ_n or Q_n (see [For89, Theorem 3, Lemma 3]).

Now, when the appropriate equivalence relation preserving the relevant cryptographic property is defined, we need to obtain representatives from all G-equivalence classes. Let ν denote the number of G-equivalence classes that is assumed to be unknown, and $f^{(1)}, \ldots, f^{(\nu)}$ be the functions representing each of the classes. Let us enumerate the functions in $\mathcal{F}_n$ in an arbitrary way so that $\mathcal{F}_n = \{f_1, \ldots, f_{2^{2^n}}\}$. The following is the straightforward but highly impractical algorithm, based on (2.36), for finding $f^{(1)}, \ldots, f^{(\nu)}$:

1. initialize the counters i and j with 1, set $f^{(1)} = f_1$ and estimate $|\mathrm{I}_G(f^{(1)})|$;
2. if function f_{i+1} is not G-equivalent to any of the functions $f^{(1)}, \ldots, f^{(j)}$ then set $f^{(j+1)} = f_{i+1}$, estimate $|\mathrm{I}_G(f^{(j+1)})|$ and increment j, else goto Step 3;
3. if $\sum_{k=1}^{j} \frac{|G|}{|\mathrm{I}_G(f^{(k)})|} = 2^{2^n}$ then finish, else increment i and goto Step 2.

Step 2 of the algorithm implies the checking on the equivalence of Boolean functions and this turns out to be a hard computational problem. For Boolean functions presented in formula/circuit notation, in [BRS98] it was proved that the problems whether two functions are G-equivalent for any of the above-defined transformation groups are co-NP-hard members of Σ_2^p. Moreover, these problems are neither in co-NP nor Σ_2^p-complete unless the Polynomial Time Hierarchy collapses.

2.7 Examples and Conclusion

The selection of logical functions fulfilling secure design criteria of key-stream generators turns out to be the art of finding trade-offs between all the relevant cryptographic properties of the functions. This will be demonstrated in the following example, where the various tensor transforms and the equivalence relation, discussed above in this chapter, are used in the analysis of the cryptographic characteristics of Boolean functions of four variables.

Example 2.23 We shall analyze balanced Boolean functions of $n = 4$ variables. Take the general affine group $\mathrm{AGL}_4(2)$ as transformation group G and consider the equivalence of Boolean functions under this group. By Proposition 2.22, functions, equivalent under this group, have the same algebraic degree and nonlinearity; the Walsh transforms of their real-valued counterparts are equal up to a permutation and the sign of the coefficients. Using a brute force computation we find that all balanced Boolean functions of four variables fall into four classes of $\mathrm{AGL}_4(2)$-equivalence and the following functions can be taken as representatives of these classes:

$$\begin{aligned} f^{(1)}(\boldsymbol{x}) &= x_1; & f^{(2)}(\boldsymbol{x}) &= x_1x_2 \oplus x_3; \\ f^{(3)}(\boldsymbol{x}) &= x_1x_2x_3 \oplus x_4; & f^{(4)}(\boldsymbol{x}) &= x_1x_2x_3 \oplus x_1x_4 \oplus x_2 \ . \end{aligned}$$

Let us compute the Walsh transform $S^{\hat{f}}$ of these representatives. It turns out that

$$\begin{aligned} S^{\hat{f}^{(1)}} &= \{0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 16,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0\}, \\ S^{\hat{f}^{(2)}} &= \{0,\ 0,\ 8,\ 0,\ 0,\ 0,\ 8,\ 0,\ 0,\ 0,\ 8,\ 0,\ 0,\ 0,\ -8,\ 0\}, \\ S^{\hat{f}^{(3)}} &= \{0,\ 12,\ 0,\ 4,\ 0,\ 4,\ 0,\ -4,\ 0,\ 4,\ 0,\ -4,\ 0,\ -4,\ 0,\ 4\}, \\ S^{\hat{f}^{(4)}} &= \{0,\ 4,\ 0,\ -4,\ 8,\ 4,\ 0,\ 4,\ 0,\ -4,\ 0,\ 4,\ 8,\ -4,\ 0,\ -4\}. \end{aligned}$$

By Proposition 2.22 and (2.39), the nonlinearity of the functions equivalent to $f^{(1)}$ is equal to zero (all the functions in this class are linear) and the nonlinearity of the functions equivalent to $f^{(3)}$ is equal to two. All the other functions have nonlinearity equal to four.

We shall now analyze the correlation properties of these functions. The maximal order of resiliency for these functions is equal to $n - 1 = 3$. Using the spectral characterization of high-order resilient Boolean functions (see Note 2.5), we find the total of two hundred twenty two 1-resilient functions, ten 2-resilient and two 3-resilient. All linear functions that have at least two nonzero-order terms in the ANF are 1-resilient and there are twenty two functions of this type. By Siegenthaler's inequality, the remaining two hundred 1-resilient functions have algebraic degree equal to two (and thus are equivalent to $f^{(3)}$) and nonlinearity equal to four. Also by Siegenthaler's inequality and Note 2.5, 2-resilient functions are precisely the linear functions having at least three nonzero-order terms in their ANF. The only two 3-resilient functions are $x_1 \oplus x_2 \oplus x_3 \oplus x_4$ and its inverse.

Since all 1-resilient functions have algebraic degree equal to one or two which is rather low, in some cryptographic applications it might be preferable to use a non-resilient function of degree three that has the lowest possible 1st and 2nd-order correlation dependencies. The value of the correlation coefficients (see Definition 2.4) can be calculated using (2.20) and provides a clear numerical estimate for this dependence. Function f is 1-resilient if and only if for any integer i with $1 \leq i \leq 4$ and any bit value of β one has $wt\left(f_i^{\beta}\right) = 2^{-1}wt(f) = 4$. Function f is 2-resilient if and only if for any integers i_1, i_2 with $1 \leq i_1 < i_2 \leq 4$ and any bit values of β_1, β_2 one has $wt\left(f_{i_1,i_2}^{\beta_1,\beta_2}\right) = 2^{-2}wt(f) = 2$. The minimal bias of the weight of these subfunctions from the ideal value gives three possible values for the 1st-order correlation coefficients, namely: 3/8, 4/8 (corresponds to the ideal value) and 5/8 and three possible values for the 2nd-order correlation coefficients, namely: 1/8, 2/8 and 3/8. Using the weight transform, we can find 3154 functions whose 1st-order correlation coefficients lie in the closed interval $[3/8; 5/8]$ and 2nd-order correlation coefficients lie in $[1/8; 3/8]$. Out of these, 200 functions have nonlinearity equal to four and algebraic degree equal to two (apparently, these are 1-resilient functions) and 2304 functions have the same nonlinearity and degree equal to three.

Using the algebraic normal transform, we find that 1536 functions out of these 2304, have at least two 3rd-order terms in their ANF. This means that the algebraic degree for each individual variable in such a function is also equal to three. Finally, these functions can be screened further using the probabilistic transform. This transform provides the coefficients for the bias polynomial and, thereafter, the bias of the output distribution can be estimated if biases δ_i $(i = 1, \ldots, n)$ are known. The screening is done with the objective to minimize the output bias.

We conclude from the previous sections and the above example that various representations of logical functions can be constructed taking the tensor transform as a basis. These representations facilitate the study of their cryptographic properties and different representations enable different characterizations.

More precisely, the classical algebraic normal, arithmetic and Walsh transforms turn out to be a special case of the tensor transform just as two newly defined transforms, namely, probabilistic and weight transforms. The new transforms are cryptographically important since they relate a Boolean function directly to its bias polynomial and to the weights of its subfunctions. Easy proofs for some known properties of algebraic normal, arithmetic and Walsh transforms have been given and some new relations have been established using the general properties of the tensor transform. Any tensor transform is based on the Kronecker product of appropriate elementary cells. This fact allows to use fast Fourier and Walsh transform algorithms for efficient estimation of the relevant tensor transform and easy transition from one transform to another.

The requirement for a cryptographically secure Boolean function to be correlation immune can be weakened without undermining the security if only a slight dependence between input bits and the output is allowed. Correlation coefficients provide an estimate for correlation dependencies and can be obtained from the weight transform of a Boolean function. The number of $(n-m)$th-order product terms in the ANF of a Boolean function f is directly related to the number of subfunctions obtained by fixing m variables of f with zero values and having an even weight. Highly resilient Boolean functions cannot be approximated with a function that is nondegenerate on few number of variables.

The bias polynomial estimates the bias for the distribution of the value of a Boolean function if the biases of the arguments of the function are known. The compensating degree of a Boolean function is a new notion defined here. This characteristic allows to compare the functions against their ability for compensating a nonuniform distribution of the input bits. The coefficients of the bias polynomial and the compensating degree of a function can be efficiently estimated by the probabilistic transform of this function. Highly resilient Boolean functions significantly increase the order of magnitude for the bias of the distribution of the output bits compared to the bias of the inputs. However, correlation and compensating properties need to be optimized with respect to the algebraic degree when constructing secure Boolean functions. An important problem that concerns k-compensating functions is whether these are "better" than $(k-1)$-compensating functions not only asymptotically but also in an "absolute" sense. We conjecture that for any k-compensating function f and $(k-1)$-compensating function g

$$\max_{|\delta_i|\le\delta,\, i=1,\dots,n} |\Delta_f(\delta_1,\dots,\delta_n)| \le \max_{|\delta_i|\le\delta,\, i=1,\dots,n} |\Delta_g(\delta_1,\dots,\delta_n)|$$

for any $\delta \in (0, 1/2)$. We were not able to find counterexamples for this inequality among the Boolean functions of five variables or less.

Equivalence relations of Boolean functions under transformation groups can be used to facilitate estimation of the number of functions that fulfil a relevant set of security criteria. This approach is also helpful when checking whether a design criterion remains invariant under weak transformations. Some important cryptographic characteristics of Boolean functions are proved to be invariant under weak transformations.

Chapter 3

Clock-Controlled Shift Registers for Key-Stream Generation

3.1 Introduction

Linear feedback shift registers (LFSR) are known to allow fast implementation and produce sequences with large period and good statistical properties (if the feedback polynomial is chosen appropriately). But inherent linearity of these sequences results in susceptibility to algebraic attacks. That is the prime reason why LFSR's are not used directly for key-stream generation. A well-known method for increasing the linear complexity preserving at the same time a large period and good statistical properties, is a nonlinear transformation applied to several phases of the same LFSR (the nonlinear filter generator) or to the outputs of several independent LFSR's (the nonlinear combination generator) (see Fig. 1.4 and [Rue86, Rue92]). An alternative way to achieve the same goal is to control the LFSR clock. On the other hand, key-stream generators based on regularly clocked LFSR's are susceptible to basic and fast correlation attacks. Using irregular clocking reduces the danger from correlation attacks and provides practical immunity against fast correlation attacks.

The basic building block that we want to use for constructing a key-stream generator, consists of a control register CR and a clock-controlled generating register GR. A control register generates a sequence of nonnegative integers $a = \{a_i\}_{i \geq 0}$ and cycles periodically with period π. Hereafter in this chapter by period we mean the least period of a sequence, as opposed to multiple period. A generating register is an LFSR over $P = \mathrm{GF}(q)$ with *irreducible* feedback polynomial $f(x)$ of degree $m > 1$ and order M (the order is the least positive integer M for which $f(x)$ divides $x^M - 1$). Let $b = \{b(i)\}_{i \geq 0}$ denote the output sequence from the GR when clocked regularly and let α be a root of $f(x)$ in the splitting field of $f(x)$. In some cases, further in this chapter, primitiveness of $f(x)$ will be required. It is well known that M divides $\lambda = q^m - 1$ and that $M = \lambda$ if and only if f is primitive. Let also S denote $\sum_{k=0}^{\pi-1} a_k$.

In the clock-controlled mode, the output sequence $u = \{u(t)\}_{t \geq 0}$ is generated in the following way (see Fig. 3.1). The initial output is $u(0) = b(a_0)$. Further, after output $u(t-1)$ has been generated, the CR specifies the nonnegative integer a_t, the GR is shifted a_t times and then produces the next output $u(t)$. After that, the CR is shifted once to be ready for the next iteration. Thus, the general form of an output sequence element is

$$u(t) = b\left(\sum_{i=0}^{t} a_i\right) \quad \text{for} \quad t \geq 0 \ . \tag{3.1}$$

In the sequel, by irregular clocking we will mean the above type of clock control applied to the GR. According to the classification in [Rue92, p. 101], the described clock control technique is a forward clock control (as opposed to feedback clock control). A comprehensive survey on clock-controlled shift registers can be found in [GC89].

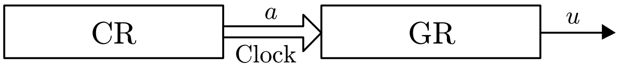


Figure 3.1: Clock-controlled arrangement

In order to ensure the security of a key-stream generator against the Berlekamp-Massey algorithm [Mas69, vT00], its output sequence should have large period and high linear complexity. On the other hand, good statistical properties of the output sequence prevent the reconstruction of statistically redundant plaintext from the known ciphertext. That is the reason why these characteristics are discussed in detail further in this chapter.

Section 3.2 contains some results about uniform decimation of linear recurring sequences in the field $P = \mathrm{GF}(q)$. These results are used in Sect. 3.3. Certain properties of sequences obtained by uniform decimation, are formulated, and a proof along novel lines is given. This theorem is a slight generalization of known results. We also derive some new conditions for sequences obtained by uniform decimation, to reach their maximum linear complexity.

The period of the output sequence generated by an arbitrary clock-controlled LFSR with an irreducible feedback polynomial and an arbitrary structure of the control sequence is estimated in Sect. 3.3. A sufficient condition for this period to reach its maximal value is formulated. Results from Sect. 3.2 are used in finding some specific configurations of clock-controlled arrangements with a maximal period of the output sequence. The special case when degree m of $f(x)$ is a prime number is studied in detail. Relevant recommendations for estimating the linear complexity are also given. The extended abstract of Sects. 3.2 and 3.3 appeared in [Kho01]. These results extend the ones earlier published in [Kho98b].

In Sect. 3.4, we discuss randomness properties of clock-controlled LFSR output sequences. The deviation of the number of occurrences of elements in a full period of u from the "ideal" value is estimated when $\gcd(S, M) = 1$. Also we estimate the autocorrelation function of the output sequence for the special case that the GR is an m-LFSR and $\gcd(S, \lambda) = 1$.

In Sect. 3.5 we construct a key-stream generator based on the one suggested by Geffe in [Gef73]. Unlike the Geffe generator that has three binary input m-sequences, this generator runs over the field $P = \mathrm{GF}(q)$ and combines multiple inputs having arbitrary periods. In particular, this implies that clock-controlled shift registers can be used as inputs. The original Geffe generator cannot be used for key-stream generation since its combining function is zero-order correlation immune and correlation attacks can easily be launched [Sie85]. Using clock-controlled registers and multiple inputs makes the new generator immune against fast correlation attacks and less susceptible to basic attacks. We analyze some relevant algebraic properties of the suggested generator. Results presented in this section were published in [Kho01].

Clock-controlled registers and their memoryless combiners are susceptible to certain types of correlation attacks, of which the complexity depends on the parameters chosen for the control register and the generating register, and on the correlation characteristics of the combining function. Section 3.6 contains a survey of correlation attacks published so far and provides relevant recommendations for selecting secure parameters of clock-controlled arrangements. Still, one can notice a general lack of empirical data on the practical efficiency of these attacks. Furthermore, these attacks can be defeated by adding a uniform noise to the key stream.

3.2 Decimation of Linear Recurring Sequences

In this section some results are presented about sequences obtained by uniform decimation of linear recurring sequences with irreducible characteristic polynomial. These results will be used further to estimate the period of a sequence generated by a clock-controlled LFSR.

Definition 3.1 *Let l and k be arbitrary nonnegative integers and $k > 0$. Then sequence $v = \{v(i)\}_{i \geq 0}$ defined by $v(i) = u(l + ki)$ for $i \geq 0$ is called the uniform (l, k)-decimation of sequence $u = \{u(i)\}_{i \geq 0}$. One also says that v is obtained by uniform (l, k)-decimation of u.*

Let $f(x)$ be an irreducible polynomial of degree $m > 0$ and order M over $P = \mathrm{GF}(q)$. Further, taking into account the fact that $Q = \mathrm{GF}(q^m)$ is the splitting field of $f(x)$, let α be a root of $f(x)$ in an extension field $Q = \mathrm{GF}(q^m)$ of P. Let $m(k)$ denote the degree of $R_k = P(\alpha^k)$ over P. Finally, let $f_k(x)$ denote the minimal polynomial of α^k over P. Note that $f_k(x)$ is irreducible in $P[x]$. Then it follows directly from the definition of extension degree that $\deg f_k(x) = m(k)$ and evidently $m(k)$ divides $m = m(1)$.

We denote the set of all homogeneous linear recurring sequences over P with characteristic polynomial $f(x)$ by $L_P(f)$. If degree of $f(x)$ is m then $L_P(f)$ is an m-dimensional vector space over P. Item (i) of the following theorem is a particular case of [Gol95a, Proposition] and has also been partially proved in [Rue86, pp. 144-

147] and [LN86, pp. 285-287]. Item (ii) generalizes [Zie59, Lemma 17] and has relation to [HK99, Sect. 8.2]. We shall present a proof of the theorem along novel lines.

Theorem 3.2 *Under the conditions imposed above, let l and k be arbitrary nonnegative integers and $k > 0$, then:*

(i) The uniform (l, k)-decimation defines a homomorphism of the vector space $L_P(f)$ onto $L_P(f_k)$ (also called epimorphism or surjective homomorphism). This epimorphism is an isomorphism if and only if $m(k) = m$.

(ii) If $f(x)$ is a primitive polynomial and if u is a nonzero sequence belonging to $L_P(f)$ then every nonzero sequence $w \in L_P(f_k)$ can be obtained as a uniform (l, k)-decimation of u using exactly $q^{m-m(k)}$ different values of $l \in \{0, \dots, \lambda - 1\}$, and the zero sequence can be obtained similarly using exactly $q^{m-m(k)} - 1$ different values of $l \in \{0, \dots, \lambda - 1\}$.

Proof:

(i) We use the representation of linear recurring sequences in finite fields in terms of trace function. By [LN83, p. 406, Theorem 8.24] if $f(x)$ is irreducible then for any $u \in L_P(f)$ there is a unique $\theta \in Q = \mathrm{GF}(q^m)$ such that $u(i) = \mathrm{Tr}_{Q/P}(\theta\alpha^i)$ $(i = 0, 1, 2, \dots)$. Since $\alpha^k \in R_k$, applying uniform (l, k)-decimation to u we get

$$\begin{aligned} v(i) &= u(l + ki) = \mathrm{Tr}_{Q/P}(\theta\alpha^l(\alpha^k)^i) = \mathrm{Tr}_{R_k/P}\left(\mathrm{Tr}_{Q/R_k}(\theta\alpha^l(\alpha^k)^i)\right) = \\ &= \mathrm{Tr}_{R_k/P}\left(\left(\mathrm{Tr}_{Q/R_k}(\theta\alpha^l)\right)(\alpha^k)^i\right) = \mathrm{Tr}_{R_k/P}(b_l(\alpha^k)^i) \quad (i = 0, 1, 2, \dots) \ , \end{aligned}$$

where $b_l = \mathrm{Tr}_{Q/R_k}(\theta\alpha^l) \in R_k$. Thus, $v \in L_P(f_k)$.

It is obvious that uniform (l, k)-decimation of a sum of sequences from $L_P(f)$ is a sum of corresponding uniform (l, k)-decimation sequences in $L_P(f_k)$. Thus, uniform decimation defines a homomorphism of $L_P(f)$ *in* $L_P(f_k)$.

Now we have to prove that this homomorphism is a surjective map. For any $w \in L_P(f_k)$ there exists a uniquely determined $\eta \in R_k$ such that $w(i) = \mathrm{Tr}_{R_k/P}(\eta(\alpha^k)^i)$ $(i = 0, 1, 2, \dots)$. Thus, w can be obtained by uniform (l, k)-decimation of a sequence from $L_P(f)$ if and only if $\eta = \mathrm{Tr}_{Q/R_k}(\theta\alpha^l)$ for some $\theta \in Q$. The number of such θ is equal to the number of solutions of the equation $\mathrm{Tr}_{Q/R_k}(x) = \eta$ in the field Q. This number is equal to $|\ker(\mathrm{Tr}_{Q/R_k})| = q^{m-m(k)} \geq 1$ since function Tr_{Q/R_k} is a nonzero linear mapping of the field Q to the field R_k.

The final statement of Item (i) follows from the fact that a homomorphism of a finite-dimensional vector space *onto* another vector space is an isomorphism if and only if their dimensions are equal.

(ii) Fix an arbitrary positive integer k. For any $w \in L_P(f_k)$ there exists a uniquely determined $\eta \in R_k$ such that $w(i) = \mathrm{Tr}_{R_k/P}(\eta(\alpha^k)^i)$ for $i = 0, 1, 2, \ldots$. From the proof of Item (i) is follows that $w = v$ if and only if $\eta = b_l = \mathrm{Tr}_{Q/R_k}(\theta\alpha^l)$. Sequence u is nonzero thus $\theta \neq 0$.

Since $f(x)$ is a primitive polynomial, α has order $\lambda = q^m - 1$. It follows that the set of elements $\{\theta\alpha^l \mid l \in 0, \ldots, \lambda - 1\}$ is equal to Q^*, the multiplicative group of the field Q. Function Tr_{Q/R_k} is a linear map of the field Q to the field R_k. The number of $l \in 0, \ldots, \lambda - 1$ such that $\eta = b_l$ is equal to the number of *nonzero* solutions of the equation $\mathrm{Tr}_{Q/R_k}(x) = \eta$ in the field Q. The total number of solutions is equal to $|\ker(\mathrm{Tr}_{Q/R_k})| = q^{m-m(k)}$. If $\eta \neq 0$, all solutions of the equation $\mathrm{Tr}_{Q/R_k}(x) = \eta$ are nonzero and the number we are looking for is equal to $q^{m-m(k)}$. If $\eta = 0$ then $x = 0$ is also a solution and the number we are looking for is equal to $q^{m-m(k)} - 1$. □

Note 3.3 Polynomial $f_k(x)$ is the minimal polynomial of α^k, so it is irreducible. Since the order of α^k (that is equal to the order of $f_k(x)$) is given by $\frac{\mathrm{ord}\,\alpha}{\gcd(k, \mathrm{ord}\,\alpha)} = \frac{M}{\gcd(k,M)}$, we conclude that $f_k(x)$ has order M if and only if k is relatively prime to M. Further, if $\gcd(k, M) = 1$ then $f_k(x)$ has degree m. Indeed, the degree of $f_k(x)$ is equal to the least value of t ($t > 0$) for which $(\alpha^k)^{q^t} = \alpha^k$ or equivalently $\alpha^{k(q^t-1)} = 1$. But $\mathrm{ord}\,\alpha = M$ and $\gcd(k, M) = 1$. It follows that $M \mid q^t - 1$ and thus that $t = m$.

Corollary 3.4 *Let* $\gcd(k, M) = 1$. *Then every uniform* (l, k)*-decimation sequence of any nonzero sequence* $u \in L_P(f)$ *is equal to a nonzero sequence belonging to* $L_P(f_k)$ *and for those nonzero sequences in* $L_P(f_k)$ *that can be obtained as a uniform* (l, k)*-decimation of* u*, the value of* $l \in \{0, \ldots, M-1\}$ *is determined uniquely.*

Proof: When applying the uniform decimation with parameters $l \geq 0$ and $k > 0$ to sequences in $L_P(f)$ we can assume that $l < M$ since all these sequences have the multiple period M. Moreover, if we fix some arbitrary value of $0 \leq \tilde{l} < M$ then for any $l > 0$, the uniform (l, k)-decimation of any nonzero sequence from $L_P(f)$ is equal to the uniform $(\tilde{l}, k)$-decimation of some other nonzero sequence from $L_P(f)$. Thus, for any fixed value of $\tilde{l}$ ($0 \leq \tilde{l} < M$), the set containing uniform (l, k)-decimation sequences of any nonzero sequence $u \in L_P(f)$, when $k > 0$ is fixed and l takes on all possible nonnegative values, is equal to the set containing uniform $(\tilde{l}, k)$-decimation sequences of some subset having cardinality M of nonzero sequences in $L_P(f)$. Now since $m = m(k)$, the statement easily follows from Item (i) of Theorem 3.2. □

Corollary 3.5 *If degree* m *of polynomial* $f(x)$ *is a prime number then* $m(k) = m$ *if and only if* k *is not a multiple of* $\frac{M}{\gcd(M,q-1)}$. *Moreover, if* $\frac{M}{\gcd(M,q-1)} \nmid k$ *then every uniform* (l, k)*-decimation sequence of any nonzero sequence* $u \in L_P(f)$ *is equal to a nonzero sequence belonging to* $L_P(f_k)$ *and for those nonzero sequences* *in $L_P(f_k)$ that can be obtained as a uniform (l,k)-decimation of u, the value of $l \in \{0,\ldots,M-1\}$ is determined uniquely.*

Proof: Since $m(k) \mid m$ and m is prime, only two alternatives are possible: either $m(k) = m$ or $m(k) = 1$, in which case $(\alpha^k)^q = \alpha^k$. So, $m(k) = 1$ if and only if M divides $k(q-1)$, i.e.,

$$\frac{M}{\gcd(M, q-1)} \,\bigg|\, k \ .$$

The rest of the proof goes the same way as in Corollary 3.4. □

Corollary 3.6 *If $f(x)$ is a primitive polynomial and $k \leq q^{m/2}$ then $m(k) = m$. Moreover, under these conditions, every uniform (l,k)-decimation sequence of any nonzero sequence $u \in L_P(f)$ is equal to a nonzero sequence belonging to $L_P(f_k)$ and every nonzero sequence $w \in L_P(f_k)$ can be obtained as a uniform (l,k)-decimation of u using a unique value of $l \in \{0,\ldots,\lambda-1\}$.*

Proof: By virtue of Theorem 3.2, Item (i), all uniform (l,k)-decimation sequences of u belong to $L_P(f_k)$. We have to prove that $m(k) = m$.

By definition, $\operatorname{ord}\alpha^k = \frac{\lambda}{\gcd(k,\lambda)} \mid (q^{m(k)} - 1)$ and $m(k) \mid m$, as was noted before. Hence, if $m(k) < m$ then $m(k) \leq \frac{m}{2}$ and therefore $\frac{\lambda}{\gcd(k,\lambda)} \leq q^{m/2} - 1$, i.e., $\gcd(k,\lambda) \geq q^{m/2} + 1$. In particular, $k \geq q^{m/2} + 1$ which contradicts the condition imposed.

Therefore, $m(k) = m$ and by Theorem 3.2, Item (ii), the zero sequence can be obtained as a uniform (l,k)-decimation of u using exactly $q^{m-m(k)} - 1 = 0$ different values of $l \in \{0,\ldots,\lambda-1\}$. So, all uniform (l,k)-decimation sequences of u are nonzero. Every nonzero sequence $w \in L_P(f_k)$ can be obtained as a uniform (l,k)-decimation of u using exactly $q^{m-m(k)} = 1$ value of $l \in \{0,\ldots,\lambda-1\}$. □

3.3 Period and Linear Complexity of Clock-Controlled LFSR

The period and the linear complexity profile [Rue86] constitute the basic algebraic properties of a key stream. The value of these parameters needs to be large enough to provide security against linear attacks [DXS91]. In this section we are going to estimate the period of the output sequence generated by a clock-controlled LFSR and find configurations of control and generating registers when the value of the period reaches its maximum. In this section we continue to use the terminology and notation introduced in Sect. 3.1.

As a generalization of Definition 3.1 of a uniform decimation, we can consider the output sequence u obtained from (3.1) as a *nonuniform* decimation of b according to the control sequence a as follows:

$$u(i + j\pi) = b(\sigma(i) + jS) \quad \text{for} \quad 0 \leq i < \pi,\ j \geq 0 \ , \tag{3.2}$$

where $S = \sum_{k=0}^{\pi-1} a_k$ and $\sigma(i) = \sum_{k=0}^{i} a_k$. Hence, any uniform (i, π)-decimation of u is a uniform $(\sigma(i), S)$-decimation of b. By Theorem 3.2, Item (i), the latter decimation belongs to $L_P(f_S(x))$. The output sequence u consists of π such sequences interleaved and belongs to $L_P(f_S(x^\pi))$.

Since the period of the sequence b divides the order M of $f(x)$, we conclude that all elements of a can be reduced modulo M without any effect on the output sequence u. So, from now on we assume without loss of generality that all elements of a are nonnegative integers less than M.

It is obvious that the minimum of the degrees of irreducible factors of $f_S(x^\pi)$ provides a lower bound for the linear complexity of the output sequence u and the lowest possible order of any irreducible factor of $f_S(x^\pi)$ gives a lower bound for the period of u.

In [GŽ88] for $P = \mathrm{GF}(2)$ and *primitive* GR feedback it was shown that the maximum linear complexity πm of an output sequence u can be obtained only if the multiplicative order of 2 modulo $\frac{\lambda}{\gcd(S,\lambda)}$ is equal to m. Furthermore, when the control sequence a and initial state vector of the GR are chosen at random and uniformly, a lower bound on the probability that the output sequence has maximum linear complexity is established. By appropriate choice of π and m this bound can be made arbitrary close to 1 with πm arbitrarily large, provided that $\pi \le 2^m$.

Since $\operatorname{ord} f_S(x) = \operatorname{ord} \alpha^S = \frac{M}{\gcd(S,M)}$ and u consists of π interleaved sequences belonging to $L_P(f_S(x))$, it easily follows from (3.2) that the period of u divides $\frac{\pi M}{\gcd(S,M)}$. From [Gol98, Lemma 1] it follows for a nonzero u that its period is a multiple of $\frac{\pi' M}{\gcd(S,M)}$ where π' is the product of all prime factors of π, not necessarily distinct, which are also factors of $\frac{M}{\gcd(S,M)}$. This provides the lower bound for the period. In particular, if every prime factor of π also divides $\frac{M}{\gcd(S,M)}$ then the period of u reaches the maximal value $\frac{\pi M}{\gcd(S,M)}$. We also note that zero output sequences can be generated even if the initial state of the GR is nonzero and $f(x)$ is primitive. This will be illustrated in Example 3.13.

By Note 3.3, if S is relatively prime to M then $f_S(x)$ is irreducible of degree m and order M. For $P = \mathrm{GF}(2)$, odd π and such an $f_S(x)$, Theorem 2 in [Cha88] provides an exact lower bound for the degree of any irreducible factor of $f_S(x^\pi)$. Namely, let $d = \frac{2^m-1}{M}$, $\pi_d = \frac{\pi}{\gcd(\pi,d)}$ and π' be the product of all prime factors of π_d, not necessarily distinct, which are also factors of M. Then the lowest possible degree of any irreducible factor of $f_S(x^\pi)$ is $\pi' m$; moreover, there is at least one irreducible factor of this degree. In particular, if $f(x)$ is primitive, $\gcd(S, \lambda) = 1$ and every prime factor of π also divides λ then $f_S(x^\pi)$ is irreducible. In this case the linear complexity of u reaches its maximal possible value πm (this is equal to the degree of $f_S(x^\pi)$).

In many cases the period of sequence u can be determined more precisely. The following theorem, that was earlier published in [Kho98b], extends [GC89, Theorem 4]. Later, in [Gol98, Theorem 2] Golić generalized this result for an arbitrary GR having an LFSR structure. We provide the proof here for its simplicity and universality of some techniques used.

Theorem 3.7 *The output sequence u defined by (3.1) is periodic. Moreover, if for $l \in \{0, \dots, M-1\}$ the uniform (l, S)-decimation sequences of b are all distinct then the period of u is equal to*

$$\tau(\pi, M, S) = \frac{\pi M}{\gcd(S, M)} \quad .$$

Proof: Put $\tau = \frac{\pi M}{\gcd(S,M)}$. We shall first prove that τ is a multiple period of u.

As was noted before, the output sequence u is a homogeneous linear recurring sequence with characteristic polynomial $f_S(x^\pi)$ and consists of π interleaved sequences belonging to $L_P(f_S(x))$, where $f_S(x)$ is the minimal polynomial of element α^S over P. Thus, the period of any such nonzero uniform S-decimation is equal to $\frac{M}{\gcd(S,M)}$ which is the multiplicative order of element α^S in P^*. Hence, the sequence u is periodic and $\tau(\pi, M, S) \mid \pi \frac{M}{\gcd(S,M)} = \tau$.

Consider two uniform π-decimation sequences of the output u, the first one starting from $u(0)$ and the second from $u(\tau(\pi, M, S))$. These decimation sequences are equal since $\tau(\pi, M, S)$ is the period of u. On the other hand, according to (3.2) the same sequences are uniform (k_0, S) and (t_0, S)-decimation sequences of b for some $k_0 = a_0 \geq 0$ and $t_0 \geq k_0$. Then, by the hypothesis of the theorem, $k_0 \equiv t_0 \pmod M$.

Further, consider two uniform π-decimation sequences of u where the first one starts from $u(1)$ and the second from $u(\tau(\pi, M, S)+1)$. These decimation sequences are equal and they are uniform (k_1, S) and (t_1, S)-decimation sequences of b for some $k_1 \geq k_0$ and $t_1 \geq t_0$. Thus, $k_1 \equiv t_1 \pmod M$. Proceeding in a similar way, consider pairs of uniform π-decimation sequences that start from $u(2)$ and $u(\tau(\pi, M, S)+2)$, from $u(3)$ and $u(\tau(\pi, M, S)+3)$ and so on. The corresponding values of k_i and t_i satisfy the equivalence

$$k_i \equiv t_i \pmod M \quad (i = 0, 1, 2, \dots) \ , \tag{3.3}$$

where $k_{i+1} \geq k_i$ and $t_{i+1} \geq t_i$.

From (3.1) we have $k_{i+1} - k_i = a_{i+1}$ and $t_{i+1} - t_i = a_{\tau(\pi,M,S)+i+1}$. It follows from the congruence relations in (3.3) and from the assumption that $0 \leq a_i < M$, that $k_{i+1} - k_i = t_{i+1} - t_i$ and thus that $a_{i+1} = a_{\tau(\pi,M,S)+i+1}$ $(i = 0, 1, 2, \dots)$. This shows that

$$\pi \mid \tau(\pi, M, S) \ . \tag{3.4}$$

It is clear that $t_i - k_i$ $(i = 0, 1, 2, \dots)$ is equal to the number of regular steps (with no clock control) the GR is making each time when the whole automaton generates $\tau(\pi, M, S)$ output elements. By virtue of (3.4), $t_i - k_i = \frac{\tau(\pi,M,S)}{\pi} S$ since if the CR makes a full period then the GR makes S steps. Thus, according to (3.3), $M \mid \frac{\tau(\pi,M,S)}{\pi} S$, from which it directly follows that

$$\frac{M}{\gcd(S,M)} \,\Bigg|\, \frac{\tau(\pi, M, S)}{\pi} \quad \text{and} \quad \tau \mid \tau(\pi, M, S) \ .$$

This proves the theorem. □

Assume that b is a nonzero sequence. Then, by Theorem 3.2, Item (i), all the uniform (l, S)-decimation sequences of b for $l \in \{0, \ldots, M-1\}$ are distinct if $m(k) = m$ (see [Gol98, Proposition 2], where a similar fact was proved for an arbitrary GR having LFSR structure).

Proposition 3.8 *Let $f(x)$ be a primitive polynomial of degree m (so it has the maximal possible order $\lambda = q^m - 1$). Then all uniform (l, S)-decimation sequences of b are distinct for $l \in \{0, \ldots, \lambda - 1\}$ if and only if for any $l \in \{0, \ldots, \lambda - 1\}$ the uniform $(l, \gcd(S, \lambda))$-decimation of b is nonzero.*

Proof: We first consider the congruence $xS \equiv y \gcd(S, \lambda) \pmod{\lambda}$ where $x \geq 0$ and $y \geq 0$. It is evident that for any fixed value of $x = 0, 1, 2, \ldots$ this congruence is solvable with respect to y and for any fixed value of $y = 0, 1, 2, \ldots$ it is solvable with respect to x. Thus, for any $l \geq 0$ a uniform (l, S)-decimation of b contains exactly the same elements as a uniform $(l, \gcd(S, \lambda))$-decimation.

Suppose now that for some $k, t \in \{0, \ldots, \lambda - 1\}$ with $k \neq t$, the uniform (k, S) and (t, S)-decimation sequences of b are equal. By Theorem 3.2, Item (ii), they can be equal if and only if $q^{m-m(S)} \geq 2$ and this is so if and only if for some $l \in \{0, \ldots, \lambda - 1\}$ the uniform (l, S)-decimation of b is zero. But then the uniform $(l, \gcd(S, \lambda))$-decimation is zero too. □

Corollary 3.9 *Let b be a nonzero sequence and suppose that one of the following two conditions holds*

(i) degree m of $f(x)$ is prime and S is not a multiple of $\frac{M}{\gcd(M, q-1)}$,

(ii) $f(x)$ is a primitive polynomial and $\gcd(S, \lambda) \leq q^{m/2}$.

Then the period of u is equal to $\tau(\pi, M, S) = \frac{\pi M}{\gcd(S, M)}$.

Proof: If condition (i) holds, we can apply Corollary 3.5 and if condition (ii) holds, we can apply Corollary 3.6. In case (ii) we additionally need Proposition 3.8 to show that for $l \in \{0, \ldots, M-1\}$ all uniform (l, S)-decimation sequences of b are distinct. The proof is finished by applying Theorem 3.7. □

Note that if $f(x)$ is primitive then one has $M = \lambda = q^m - 1$. Some other sufficient conditions to apply Theorem 3.7 can be found in [Gol98, Proposition 4].

Note 3.10 We shall now consider the case when m, the degree of $f(x)$, is a prime number and $\frac{M}{\gcd(M, q-1)} \mid S$. Then $\frac{M}{\gcd(M, S)} \mid q - 1$ and hence $\tau(\pi, M, S) \mid \pi(q-1)$ since $\tau(\pi, M, S) \mid \pi \frac{M}{\gcd(M, S)}$.

By Corollary 3.5, $m(\gcd(S, M)) = m(S) = 1$ (since $\operatorname{ord} \alpha^{\gcd(S,M)} = \operatorname{ord} \alpha^S$) and $f_S(x) = x - \alpha^S$. Let p denote the element α^S in P. Thus, the output sequence u is a homogeneous linear recurring sequence with characteristic polynomial

$f_S(x^\pi) = x^\pi - p$ and consists of π interleaved sequences having the form of a geometric progression with ratio p and initial element $u(i) = b(\sigma(i))$ $(i = 0, \dots, \pi - 1)$. We can get the $\frac{\pi M}{\gcd(M,S)}$-long period of u by taking the elements of the following array in a row-by-row order

$$\begin{array}{cccc} u(0) & u(1) & \dots & u(\pi-1) \\ u(0)p & u(1)p & \dots & u(\pi-1)p \\ \vdots & \vdots & & \vdots \\ u(0)p^j & u(1)p^j & \dots & u(\pi-1)p^j \\ \vdots & \vdots & & \vdots \\ u(0)p^{\xi-1} & u(1)p^{\xi-1} & \dots & u(\pi-1)p^{\xi-1} \end{array}, \tag{3.5}$$

where $\xi = \frac{M}{\gcd(M,S)}$. If $b(\sigma(i)) = 0$ for all $i \in \{0, \dots, \pi - 1\}$ then u is a zero sequence. Further we assume that $b(\sigma(i)) \neq 0$ for some i.

If $\pi \mid \tau(\pi, M, S)$ then $\tau(\pi, M, S) = \pi j$ where j is the smallest integer in $\{1, \dots, \frac{M}{\gcd(M,S)}\}$ with the property that $b(\sigma(i)) = \alpha^{Sj} b(\sigma(i))$ for all $i \in \{0, \dots \pi - 1\}$. Since not all of $b(\sigma(i))$ are zero, the smallest j with this property is in fact equal to $\frac{M}{\gcd(M,S)}$. Thus, $\tau(\pi, M, S) = \frac{\pi M}{\gcd(M,S)}$.

Suppose now that $\tau(\pi, M, S)$ is not a multiple of π. Since u is periodic and its period has the pattern of (3.5), there exist some $j \in \{0, \dots, \frac{M}{\gcd(M,S)} - 1\}$ and $i \in \{1, \dots, \pi - 1\}$ such that

$$\begin{pmatrix} 1 & 0 & \dots & 0 & -p^j & \dots & & 0 \\ & \ddots & \ddots & & & \ddots & \ddots & \\ 0 & & & & & & 0 & -p^j \\ -p^{j+1} & 0 & & & & & & 0 \\ 0 & \ddots & \ddots & & & & \ddots & \vdots \\ \vdots & & \ddots & \ddots & & & \ddots & 0 \\ 0 & \dots & 0 & -p^{j+1} & 0 & & & 1 \end{pmatrix} \begin{pmatrix} u(0) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ u(\pi-1) \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{pmatrix}, \tag{3.6}$$

where the ones lie on the main diagonal, the entry $-p^j$ lying in the first row has the column coordinate $i + 1$ and the entry $-p^{j+1}$ lying in the first column has the row coordinate $(\pi - i) + 1$. Let $D(\pi, -p^j, -p^{j+1}, \pi - i)$ denote the determinant of this $\pi \times \pi$ matrix ($\pi - i$ of its entries are equal to $-p^j$). It is not difficult to see that

$$D(\pi, -p^j, -p^{j+1}, \pi - i) = \begin{cases} D(i, -p^{2j+1}, -p^{j+1}, \pi - i), & \text{if} \quad i > \pi/2, \\ D(\pi - i, -p^j, -p^{2j+1}, \pi - 2i), & \text{if} \quad i < \pi/2, \\ (1 - p^{2j+1})^{\pi/2}, & \text{if} \quad i = \pi/2 \end{cases}.$$

We can apply this rule repeatedly to prove that $D(\pi, -p^j, -p^{j+1}, \pi - i) = (1 \pm p^{kj+t(j+1)})^l$ for some $k, t, l > 0$ such that $(k + t)l = \pi$. Thus, if $D(\pi, -p^j, -p^{j+1}, \pi - i) = 0$ then $p^{kj+t(j+1)} = \pm 1$ and so $\frac{M}{\gcd(S,M)} \mid 2(kj + t(j + 1))$.

If integers $j \in \{0, \ldots, \frac{M}{\gcd(M,S)} - 1\}$ and $i \in \{1, \ldots, \pi - 1\}$ exist such that $D(\pi, -p^j, -p^{j+1}, \pi - i) = 0$ then (3.6) has nonzero solutions. If, in this case, one can find a control sequence with parameters π and S and an initial state vector for the GR such that $(b(\sigma(0)), \ldots, b(\sigma(\pi - 1)))$ is a nonzero solution of (3.6) then the multiple period of u is equal to $\pi j + i$. This number is less than $\frac{\pi M}{\gcd(M,S)}$.

Note 3.11 If S is relatively prime to M, it follows from Corollary 3.4 and Theorem 3.7 that the period of u reaches the maximal value πM (this is Theorem 4 in [GC89]).

If conditions of Theorem 3.7, Proposition 3.8 and Corollary 3.9 do not hold then the period of the decimated sequence may be equal to or less than $\frac{\pi M}{\gcd(S,M)}$. This can be seen in the following examples.

Example 3.12 Let $f(x) = x^4 + x + 1$ (a primitive polynomial over $P = \mathrm{GF}(2)$) and $a = (2,3)^\infty = \{2, 3, 2, 3, \ldots\}$ be the control sequence with period $\pi = 2$. If we set the initial state vector of the GR equal to $(1,1,1,1)$ then $b = (1,1,1,1,0,0,0,1,0,0,1,1,0,1,0)^\infty$ which has period 15. The output sequence u for this clock-controlled arrangement is equal to $(1,0,1)^\infty$ with period 3.

In our case $S = 5$ and $\gcd(S, \lambda) = \gcd(5, 15) = 5$ and this exceeds $q^{m/2} = 4$. Thus, condition (ii) of Corollary 3.9 does not hold (if the condition held, the period of u would be equal to 6). Condition (i) of Corollary 3.9 is not applicable either since $m = 4$ is not prime (although S is not a multiple of $\frac{\lambda}{q-1} = 15$). Proposition 3.8 cannot be used either since the uniform $(4,5)$-decimation sequence of b is zero. The uniform $(0,5)$-decimation sequence and $(1,5)$-decimation sequence of b are equal, so Theorem 3.7 is not applicable too.

On the other hand, if the control sequence is equal to $(3,2)^\infty$ with the same value of $S = 5$ then $u = (1,0,0,1,1,1)^\infty$. In this case the period is maximal although conditions of Theorem 3.7, Proposition 3.8 and Corollary 3.9 do not hold.

Finally, if we take the control sequence equal to $(1,2)^\infty$ then $\gcd(S, \lambda) = \gcd(3, 15) = 3$ and condition (ii) of Corollary 3.9 holds. In this case, the output sequence is $(1,1,0,0,1,0,1,0,1,1)^\infty$. So, the period of u is 10 and that is equal to $\frac{\pi\lambda}{\gcd(S,\lambda)}$.

Example 3.13 Let $P = \mathrm{GF}(3)$ and $f(x) = x^3 + 2x + 1$, so $f(x)$ is a primitive polynomial over P. Let $a = (2,5,6)^\infty$ with period $\pi = 3$ be the control sequence. If we set the initial state vector of the GR equal to $(2,0,1)$ then $b = (2,0,1,1,1,0,0,2,0,2,1,2,2,1,0,2,2,2,0,0,1,0,1,2,1,1)^\infty$. The output sequence u for this clock-controlled arrangement is equal to $(1,2)^\infty$ with period 2. But if the initial state vector of the GR is equal to $(0,1,1)$ and the control sequence is equal to $(4,1,2,6)^\infty$ then the output sequence is zero. In both cases $S = 13$ and S is equal to $\frac{\lambda}{q-1}$. Thus, condition (i) of Corollary 3.9 does not hold. Indeed, if that condition would hold, the period would be equal to 6 for the first case and 8 for the second.

On the other hand, if the initial state vector of the GR is set to $(2,0,1)$ and the control sequence is equal to $(7,6)^\infty$ with the same value of $S = 13$ then $u =$

$(2,1,1,2)^\infty$. In this case the period is maximal although condition of Corollary 3.9, Item (i), does not hold.

If CR-outputs a_i take on only bit values 0 and 1 then the arrangement is called a *stop-and-go* generator and is described in [BP85]. In our notation, S for this type of generator is equal to the number of ones in the full period of a. In particular, if the CR is an m-LFSR over GF(2) with a primitive feedback polynomial of degree n and order $\pi = 2^n - 1$ then CR-outputs take on the value one 2^{n-1} times over the period and $S = 2^{n-1}$. Thus, if $q = 2$ and $f(x)$ is primitive then $\gcd(S, \lambda) = \gcd(2^{n-1}, 2^m - 1) = 1$ and by Corollary 3.9 $\tau(\pi, \lambda, S) = \pi\lambda$. For the particular case when $n = m$, we get that $\pi = \lambda$ and by [Cha88, Theorem 2] the polynomial $f_S(x^\pi)$ is irreducible. In this case the linear complexity of the output sequence has its greatest possible value $n(2^n - 1)$ equal to the degree of $f_S(x^\pi)$. Due to these features of the output sequence, it is reasonable to use it further for clock controlling another m-LFSR. It turns out to be possible to extend this system further to an arbitrary number of LFSR's. Such an arrangement is called an *m-sequence cascade* and has been considered in [GC89]. Many other types of cascades were suggested in the literature (see [GC89] for the review) but they are not the subject of this thesis.

3.4 Randomness Properties of Clock-Controlled LFSR's

The discussion presented in Sect. 3.3 leads to the conclusion that the control sequence a plays only a secondary role when the period and linear complexity of clock-controlled LFSR's are concerned. By that we mean that by using different clock sequences one can generate different output sequences having the same period and linear complexity. However, the clocking procedure has a major influence on randomness properties of the output sequence. It is obvious that if the GR generates a sequence, containing all elements of GF(q) (for instance, this holds when $f(x)$ is primitive and the GR has nonzero initial state vector) then by selecting an appropriate control sequence one can get any periodic sequence over GF(q) as the output sequence. Thus, when choosing a control sequence one should pay attention not only to the period and linear complexity of the output but one should also take randomness properties into account.

The objective of this section is finding the conditions that should be imposed upon the parameters of the clock-controlled arrangement in order to provide close-to-uniform element distribution in the output sequence and good autocorrelation properties. Hereafter we continue to use the terminology and notation introduced in Sects. 3.1 and 3.3.

As was noted above, the output sequence u consists of π interleaved sequences, all members of $L_P(f_S(x))$. If S is relatively prime to M then by virtue of Note 3.3, $f_S(x)$ is also irreducible of degree m and order M. If h is the least common multiple of M and $q-1$ then according to [LN83, p. 450],

$$\left|Z(0) - \frac{(q^{m-1}-1)M}{q^m-1}\right| \le \left(1-\frac{1}{q}\right)\left(\frac{M}{h} - \frac{M}{q^m-1}\right) q^{m/2}$$
$$\left|Z(b) - \frac{q^{m-1}M}{q^m-1}\right| \le \left(\frac{M}{h} - \frac{M}{q^m-1} + \frac{h-M}{h} q^{1/2}\right) q^{(m/2)-1} \quad \text{for} \quad b \neq 0 \ ,$$

where $Z(b)$ is the number of occurrences of element $b \in P$ in the M-long period of a linear recurring sequence belonging to $L_P(f_S(x))$. Now if we multiply the right hand parts of both inequalities by π we can estimate the deviation between the actual number of occurrences of elements $b \in P$ in the πM-long period of u (see Note 3.11) and the ideal value. If $h \approx M$ and M is sufficiently large compared to $q^m - 1$ then this deviation is comparatively small.

In particular, if $f(x)$ is primitive and $\gcd(S, \lambda) = 1$ then polynomial $f_S(x)$ is also primitive. Thus, any sequence belonging to $L_P(f_S(x))$ is an m-sequence. So, any nonzero element of P appears q^{m-1} times in its λ-long period and 0 appears $q^{m-1}-1$ times. As a consequence, any nonzero element of P appears πq^{m-1} times in the $\pi\lambda$-long period of the output sequence u and 0 appears $\pi(q^{m-1}-1)$ times (note that by Corollary 3.9, Item (ii), the period of u is equal to $\pi\lambda$). Note that the period of u should be large enough not to allow the attacker to guess the next element in the sequence by keeping track on counts and using the fact that all nonzero elements appear equally often in the whole period.

If CR-outputs a_i take on only the values 1 or 2 and $P = \mathrm{GF}(2)$ then all l-tuples of length $l \le (m+1)/2$ appear in the output sequence with the same frequency as in the original m-sequence b (as pointed out in [Rue92, p. 103]).

We shall further estimate the autocorrelation function of the output sequence of the clock-controlled LFSR. The autocorrelation function provides an important randomness test since it measures the degree of dependence between a sequence and its various phase shifts. A requirement concerning the autocorrelation is included in Golomb's randomness postulates for pseudo random sequences [Gol67, p. 25]. It thus can be adopted as a quality measure for pseudo random sequences.

According to [LN83, pp. 463-464], if $s = \{s_i\}_{i \ge 0}$ is a sequence over $\mathrm{GF}(q)$ of period r and χ is a nontrivial additive character of $\mathrm{GF}(q)$ [LN83, p. 190] then the corresponding *autocorrelation function* of s is defined by

$$C(h) = \sum_{i=0}^{r-1} \chi(s_i)\overline{\chi}(s_{i+h}) \quad \text{for} \quad h = 0, 1, \ldots, r-1 \ ,$$

where $\overline{\chi}$ denotes the complex conjugate of χ. Golomb's randomness postulate for the autocorrelation function of s requires it to be two-valued:

$$C(h) = \begin{cases} r, & \text{for} \quad h = 0, \\ K, & \text{for} \quad 0 < h < r \end{cases} . \tag{3.7}$$

Then the normalized autocorrelation $C(h)/r$ is equal to one for $h = 0$ and close to zero for $0 < h < r$ if K is small compared to r.

Assume that $f(x)$ is primitive and $\gcd(S,\lambda) = 1$. Then the autocorrelation function of the output sequence u (with period $\tau = \pi\lambda$) can be expressed as follows:

$$\begin{aligned} C(h) &= \sum_{i=0}^{\tau-1} \chi(u(i))\overline{\chi}(u(i+h)) = \sum_{i=0}^{\pi-1}\sum_{j=0}^{\lambda-1} \chi(u(i+j\pi))\overline{\chi}(u(i+j\pi+h)) \overset{(3.2)}{=} \\ &\overset{(3.2)}{=} \sum_{i=0}^{\pi-1}\sum_{j=0}^{\lambda-1} \chi(b(jS+\sigma(i)))\overline{\chi}(b(jS+\sigma(i+h))) = \\ &= \sum_{i=0}^{\pi-1} A(\sigma(i+h)-\sigma(i)) = \begin{cases} \sum_{i=0}^{\pi-1} A\left(\sum_{k=i+1}^{i+h} a_k\right), & \text{for} \quad h>0, \\ \pi A(0), & \text{for} \quad h=0 \end{cases} , \end{aligned}$$

where $A(h)$ is the autocorrelation function of an m-sequence over $\mathrm{GF}(q)$ of period $\lambda = q^m - 1$ which can be estimated using the following proposition due to Zierler [Zie59, p. 45].

Proposition 3.14 *If h is not a multiple of $t = \lambda/(q-1)$ then*

$$A(h) = -1 + q^{m-2} \sum_{a,b\in \mathrm{GF}(q)} \chi(a)\overline{\chi}(b) \ .$$

Further, there exists a primitive element ξ of $\mathrm{GF}(q)$ such that for $j = 0,1,2,\ldots$

$$A(jt) = -1 + q^{m-1} \sum_{a\in \mathrm{GF}(q)} \chi(a)\overline{\chi}(\xi^j a) \ .$$

Example 3.15 Let $q = p$ be a prime. The canonical additive character of $\mathrm{GF}(p)$ is of the form $\chi(a) = e^{2\pi i a/p}$, $a \in \mathrm{GF}(p)$. Now

$$\sum_{a,b\in \mathrm{GF}(p)} \chi(a)\overline{\chi}(b) = \sum_{j,k=0}^{p-1} e^{2\pi i j/p} e^{-2\pi i k/p} = \sum_{j=0}^{p-1} e^{2\pi i j/p} \sum_{k=0}^{p-1} e^{-2\pi i k/p} = 0 \ .$$

This implies that if h is not a multiple of t then, by Proposition 3.14, $A(h) = -1$. If h is a multiple of t and $h = jt$, let $\mu = \xi^j$. Then

$$\sum_{a\in \mathrm{GF}(q)} \chi(a)\overline{\chi}(\mu a) = \sum_{k=0}^{p-1} e^{2\pi i k/p} e^{-2\pi i \mu k/p} = \sum_{k=0}^{p-1} e^{2\pi i k(1-\mu)/p} = 0 \ ,$$

provided $\mu \neq 1$ (i.e., $h \neq 0 \pmod{\lambda}$). Thus, $A(0) = \lambda$ while $A(h) = -1$ if h is not a multiple of λ.

We conclude from the above that if the generating register in a clock-controlled arrangement is an m-LFSR over $\mathrm{GF}(p)$ (where p is prime) and $\gcd(S,\lambda) = 1$ then the autocorrelation function $C(h)$ of u is equal to $-\pi$ for all the values of $h > 0$ for which $\sum_{k=i+1}^{i+h} a_k$ is not a multiple of λ for all $i = 0, 1, \ldots, \pi - 1$. Thus, for such h the autocorrelation fulfils Golomb's postulate (3.7). The normalized autocorrelation in this case is equal to $-\lambda^{-1}$ and for large values of λ that is close to 0.

In particular, for the stop-and-go generator, when the control register is a binary m-LFSR of period $\pi = 2^n - 1$ and if $q = 2$ then

$$\begin{aligned} C(1) &= \sum_{i=0}^{\pi-1} A(a_{i+1}) = 2^{n-1}A(1) + (2^{n-1} - 1)A(0) = \\ &= (2^{n-1} - 1)(2^m - 1) - 2^{n-1} \sim 2^{n+m-1} \ . \end{aligned}$$

This fact reveals strong intersymbol dependence between the output sequence u and its 1-step phase shift. That is easily accounted for since the previous key-stream symbol is copied to the next position every time when the control register generates 0. A dependence on the key-stream symbols of the preceding symbols constitutes a considerable weakness of a key-stream generator.

3.5 Generalized Geffe Generator

Combining linear feedback shift registers with a memoryless nonlinear function F is a well-known way to increase the period and the linear complexity of the key stream, as well as to reduce the correlation between the key-stream sequence and the LFSR sequences that are used as input of F (see [Rue86, Rue92]). The key-stream generator discussed in this section is a memoryless combiner based on a specific combining function that implements a nonuniform decimation of input sequences. The key-stream sequence is obtained by irregularly interleaving the decimated sequences. Both decimation and interleaving operations are controlled by the same sequence being one of combining function inputs. This construction can be seen as a generalization of the Geffe generator from [Gef73].

First, we need to define and fix an ordering in the finite field $P = \mathrm{GF}(q)$ by numbering the elements from 0 to $q - 1$. Thus, $P = \{p_0, \ldots, p_{q-1}\}$. Let the combining function F from P^{q+1} to P be defined by $F(p_j, x_0, \ldots, x_{q-1}) = x_j$ for $j = 0, \ldots, q - 1$. Thus, the first argument of F selects which of the remaining q arguments is taken as an output of the function. Assume that a periodic sequence $a = \{a_i\}_{i \geq 0}$ over P (we will also call it the control sequence of F) with period π and linear complexity $\hat{L}$ is fed to the first argument of F and that q periodic sequences $b^j = \{b_i^j\}_{i \geq 0}$ $(j = 0, \ldots, q - 1)$ over P with period λ_j and linear complexity L_j respectively are fed to the remaining q arguments. Let $u = \{u_i\}_{i \geq 0}$ denote the output sequence generated by function F (see Fig. 3.2).

It is clear that the output sequence u is an irregularly interleaved set of q nonuniform decimation sequences of b^j $(j = 0, \ldots, q - 1)$, when both the decimation and the interleaving operations are controlled by the sequence a. When $q = 2$, the nonuniform decimation is equivalent to the shrinking operation [CKM94] controlled by $\{a_i\}_{i \geq 0}$ and $\{a_i \oplus 1\}_{i \geq 0}$, applied to sequences b^1 and b^0 respectively. The period and linear complexity of u will be estimated further in this section.

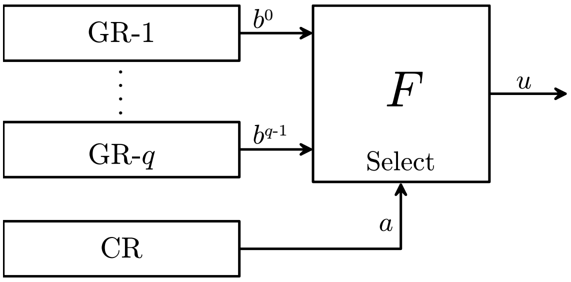


Figure 3.2: Generalized Geffe generator

Before we can continue, we need some preliminary lemmas. The first one is a special case of a fundamental result on the period of nonuniformly decimated sequences, as established in [BP81, Theorem 3].

Lemma 3.16 *Let $c = \{c_i\}_{i\geq 0}$ be a periodic sequence with the period T and let sequence $c' = \{c'_i\}_{i\geq 0}$ be a uniform d-decimation of c for some integer $d > 0$. Then c' is periodic and if T' denotes its period then*

(i) $T' \,\Big|\, \frac{T}{\gcd(T,d)}$;

(ii) if $\gcd(T, d) = 1$ *then* $T' = T$.

Let K denote the least common multiple of the periods of the sequences b^j ($j = 0, \ldots, q-1$), so $K = \text{lcm}(\lambda_0, \ldots, \lambda_{q-1})$ and let d denote $\gcd(\pi, K)$. It is obvious that K is equal to the period of the sequence of q-grams $B = \{(b_i^0, \ldots, b_i^{q-1})\}_{i\geq 0}$.

Lemma 3.17 *Suppose that sequence a contains all elements of P and that the q-gram sequence B with the period K contains a q-tuple that is equal to P in the sense of set equality. Suppose moreover that* $\gcd(\pi, K) = 1$. *Then* $\tau = \pi K$.

Proof: Under the hypothesis of the lemma, we can list a set of integers $t_j \geq 0$ ($j = 0, \ldots, q-1$) such that $a_{t_j} = p_j$. Consider q uniform (t_j, π)-decimation sequences of the output u by taking $j = 0, \ldots, q-1$. Since π is the period of the control sequence a, the (t_j, π)-decimation of u is equal to the (t_j, π)-decimation of b^j. But the hypothesis of the lemma claims that $\gcd(\pi, K) = 1$ whence it follows that $\gcd(\pi, \lambda_j) = 1$ for $j = 0, \ldots, q-1$. Hence by Lemma 3.16, Item (ii), the period of the (t_j, π)-decimation of b^j is λ_j for $j = 0, \ldots, q-1$. But since these decimation sequences are decimation sequences of u as well, by Lemma 3.16, Item (i), $\lambda_j \mid \tau$ for $j = 0, \ldots, q-1$ and thus $K \mid \tau$.

Under the hypothesis of the lemma, there exists an integer $t \geq 0$ such that the q-tuple $(b_t^0, \ldots, b_t^{q-1})$ can be obtained by permutating the elements in $(p_0, \ldots, p_{q-1})$. We now consider the uniform (t, K)-decimation of the output sequence u. Since K is the period of the q-gram sequence B, this decimation is equal to the (t, K)-decimation of a whose elements are substituted afterwards according to the rule defined by the permutation transforming $(p_0, \ldots, p_{q-1})$ into $(b_t^0, \ldots, b_t^{q-1})$. A one-to-one mapping applied to the elements of a sequence does not affect its period. Since $\gcd(\pi, K) = 1$, by Lemma 3.16, Item (ii), the period of the (t, K)-decimation of a is π. But since this decimation is a decimation of u as well, by Lemma 3.16, Item (i), $\pi \mid \tau$.

Now since $K \mid \tau$, $\pi \mid \tau$ and $\gcd(\pi, K) = 1$, we can conclude that $\pi K \mid \tau$. On the other hand, it is obvious that $\tau \mid \pi K$ and thus $\tau = \pi K$. □

Theorem 3.18 *Sequence u is periodic. Let τ denote the period of u. Then $\tau \mid \operatorname{lcm}(\pi, K)$. Moreover, if sequence a is such that each of its uniform d-decimation sequences contains all the elements of P and the q-gram sequence B is such that all its uniform d-decimation sequences contain a q-tuple that is equal to P in the sense of set equality then*

$$\left.\frac{\pi K}{\gcd(\pi, K)^2}\,\right|\,\tau \ .$$

Proof: It is obvious that in every $\operatorname{lcm}(\pi, K) = \operatorname{lcm}(\pi, \lambda_0, \ldots, \lambda_{q-1})$ steps all input sequences complete their full cycle. Since function F is memoryless, the output sequence u completes a full cycle as well in $\operatorname{lcm}(\pi, K)$ steps. Thus, u is periodic and $\tau \mid \operatorname{lcm}(\pi, K)$.

Consider the q-gram sequence B. Since all sequences b^j $(j = 0, \ldots, q-1)$ are periodic with period equal to λ_j respectively, it is obvious that the q-gram sequence B is periodic as well with period equal to $\operatorname{lcm}(\lambda_0, \ldots, \lambda_{q-1}) = K$.

Now we fix an arbitrary $t \in \{0, \ldots, d-1\}$ and consider uniform (t, d)-decimation sequences of a, u and B. Let π_t, τ_t and K_t denote the respective periods of these decimation sequences. Then, by Lemma 3.16, Item (i),

$$\pi_t \left|\, \frac{\pi}{\gcd(\pi, d)} = \frac{\pi}{d} \ , \quad \tau_t \mid \tau \quad \text{and} \quad K_t \,\right|\, \frac{K}{\gcd(K, d)} = \frac{K}{d} \ . \tag{3.8}$$

Since $\gcd(\frac{\pi}{d}, \frac{K}{d}) = 1$, it follows that $\gcd(\pi_t, K_t) = 1$.

We shall now consider the memoryless combiner described above when uniform (t, d)-decimation sequences of the respective original sequences are fed into the arguments of F. Thus, the control sequence of F has period π_t and the q-gram sequence, feeding the rest of the arguments of F, has period K_t satisfying $\gcd(\pi_t, K_t) = 1$. We note that the output sequence of F has period τ_t since it is a uniform (t, d)-decimation of sequence u. So, the conditions of Lemma 3.17 are met and thus it follows that

$$\tau_t = \pi_t K_t \ , \tag{3.9}$$

for all $t \in \{0, \ldots, d-1\}$.

By (3.8), π_t divides $\frac{\pi}{d}$ for $t = 0, \ldots, d-1$ and therefore $\operatorname{lcm}(\pi_0, \ldots, \pi_{d-1}) \,\big|\, \frac{\pi}{d}$. Sequence a can be reconstructed by interleaving d sequences obtained by (t, d)-decimating of a for $t = 0, \ldots, d-1$ and thus $d \cdot \operatorname{lcm}(\pi_0, \ldots, \pi_{d-1})$ is a multiple period of a, that is $\pi \mid d \operatorname{lcm}(\pi_0, \ldots, \pi_{d-1})$. Hence, $\operatorname{lcm}(\pi_0, \ldots, \pi_{d-1}) = \frac{\pi}{d}$. In the same way it is easy to show that $\operatorname{lcm}(K_0, \ldots, K_{d-1}) = \frac{K}{d}$.

From (3.8) it also follows that $\gcd(\pi_i, K_j) = 1$ $(i, j = 0, \ldots, d-1)$. Thus,

$$\begin{aligned}
\mathrm{lcm}(\tau_0, \ldots, \tau_{d-1}) &\overset{(3.9)}{=} \mathrm{lcm}(\pi_0 K_0, \ldots, \pi_{d-1} K_{d-1}) = \\
&= \mathrm{lcm}(\mathrm{lcm}(\pi_0, K_0), \ldots, \mathrm{lcm}(\pi_{d-1}, K_{d-1})) = \\
&= \mathrm{lcm}(\pi_0, \ldots, \pi_{d-1}, K_0, \ldots, K_{d-1}) = \\
&= \mathrm{lcm}(\mathrm{lcm}(\pi_0, \ldots, \pi_{d-1}), \mathrm{lcm}(K_0, \ldots, K_{d-1})) = \\
&= \mathrm{lcm}(\pi_0, \ldots, \pi_{d-1}) \cdot \mathrm{lcm}(K_0, \ldots, K_{d-1}) = \frac{\pi K}{d^2} \ .
\end{aligned}$$

By (3.8), τ_t divides τ for $t = 0, \ldots, d-1$. Therefore, $\mathrm{lcm}(\tau_0, \ldots, \tau_{d-1}) = \frac{\pi K}{d^2} \mid \tau$. □

The following lemma, which easily follows from [Gol95a, Proposition], will be needed to estimate the linear complexity of u.

Lemma 3.19 *Let $c = \{c_i\}_{i \geq 0}$ be a periodic sequence having linear complexity L. Then for any integer $d > 0$ there exists a polynomial $f_{(d)}(\cdot)$ of degree at most L such that $f_{(d)}$ is a characteristic polynomial for any d-decimation sequence of c.*

Proposition 3.20 *Let L denote the linear complexity of an output sequence u. Then $L \leq \pi(L_0 + \cdots + L_{q-1})$. If $q = 2$, the sequences b^0 and b^1 are nonzero, and the respective periods π, λ_0, and λ_1 are pairwise coprime then $L \geq (\hat{L} - 1)(L_0 + L_1 - 2)$.*

Proof: To prove the claimed upper bound on the linear complexity of the sequence u it is sufficient to present a polynomial $P(\cdot)$ of degree at most $\pi(L_0 + \cdots + L_{q-1})$, for which $P(u) = 0$ (i.e., the coefficients of P represent a linear relation satisfied by the elements of u, we will call any such P an annihilating polynomial of u). Consider an arbitrary uniform π-decimation of u. Since π is the period of the control sequence a, this decimation is equal to the (t_j, π)-decimation of b^j for some $j \in \{0, \ldots, q-1\}$ and $t_j \in \{0, \ldots, \lambda_j - 1\}$. Then, by Lemma 3.19, there exists a polynomial $Q_j(\cdot)$ of degree at most L_j annihilating this decimation as well as all the other π-decimation sequences of b^j. The polynomial $Q_j(\cdot)$ also annihilates the uniform π-decimation of u that we consider.

Now let $Q(\cdot)$ be the least common multiple of polynomials $Q_0(\cdot), \ldots, Q_{q-1}(\cdot)$ where $Q_j(\cdot)$ is the polynomial annihilating any π-decimation of b^j. Then $Q(\cdot)$ annihilates any π-decimation of u and thus polynomial $P(x) = Q(x^\pi)$ of degree at most $\pi(L_0 + \cdots + L_{q-1})$ annihilates u. Thus, the linear complexity of u is at most $\pi(L_0 + \cdots + L_{q-1})$.

The second part of the proposition follows from [Gol89, Theorem 6] since the algebraic normal form of the combining function for $q = 2$ is $F(a, x_0, x_1) = a(x_0 \oplus x_1) \oplus x_0$. Condition $q = 2$ is required since only then the algebraic normal form of F is free from powers. □

It remains an open problem how to estimate a nontrivial lower bound for the linear complexity of the output sequence u when $q > 2$.

If we assume that input sequences of the combining function F are sequences of uniform, independent and identically distributed random variables (i.e., purely random sequences) then its output sequence is purely random as well, since the combining function of the generator is balanced. Thus, the balancedness property of the combining function ensures good statistical properties of the key stream.

Sequences produced by linear feedback shift registers (clocked regularly or irregularly) could be used as inputs for function F in practical implementations of the above type of key-stream generator. Note that the combining function F of the generator is memoryless, balanced and zero-order correlation immune (its output is correlated to inputs $x_0, \ldots, x_{q-1}$ and this correlation decreases if q is increased). Thus, when all shift registers are clocked regularly, it is possible to apply the basic [Sie85] or fast [Jö02] correlation attack in order to reconstruct the initial state of shift registers that produce sequences b^j $(j = 0, \ldots, q-1)$. Therefore, it is reasonable to use large q and/or clock-controlled LFSR's to generate sequences b^j $(j = 0, \ldots, q-1)$. Using large q however does not seem very practical. We note that knowing the periods of the control and the generating registers, one can easily verify the condition of coprimality in Proposition 3.20. Memoryless combiners of clock-controlled LFSR's can also be susceptible to certain types of correlation attacks. This will be discussed further in Sect. 3.6. But the essential benefit of these combiners consists in their immunity against fast correlation attacks.

For practical implementation of the suggested generator it may be reasonable to select q as a power of 2, and to generate binary sequences a and b^j $(j = 0, \ldots, q-1)$, to feed them as input to the $(q+1)$-input combining function F. The control sequence is split into $\log_2 q$-long tuples that are used to index sequences b^j $(j = 0, \ldots, q-1)$. Following the first half of the proof of Lemma 3.17, it can be readily shown that if the control sequence splits into $\log_2 q$-tuples consisting of all q possible values and if $\gcd(\pi, K) = 1$ then $K \mid \tau$.

3.6 Correlation Attacks on Clock-Controlled Shift Registers and their Memoryless Combiners

We start by defining a statistical model for a correlation attack. In this section, we continue to use the notation introduced in Sect. 3.1. Assume that b is a purely random sequence over $P = \mathrm{GF}(2)$, i.e., it is a sequence of uniform, independent and identically distributed (i.i.d.) random variables, rather than the output of an LFSR. Also assume that the control sequence a consists of i.i.d. positive, integer valued, random variables that is independent of b. The random sequences a and b are combined according to (3.1) to generate the output random sequence u. Since the sequence a contains only positive elements, it is clear that u is a purely random sequence over P itself (for instance, this is not true for the output of the stop-and-go generator).

Irregular clocking is called *constrained* if the range of elements in a is limited by some value and *unconstrained* otherwise. The secret key is assumed to control the initial state of the generating register. The objective of a correlation attack is defined here as the reconstruction of the initial state of the GR from a given segment of the output sequence u, thereby knowing the GR length and the feedback polynomial (that can be arbitrary; so it is not necessarily linear and irreducible). The control sequence is unknown except for the probability distribution of the random variable a_i, $i \geq 0$. If $\overline{a}$ is the mean value of a_i then $p_d = 1 - 1/\overline{a}$ is called the *deletion rate.* The model for unconstrained clocking assumes independent deletions from b with probability p_d.

Let $\mathcal{D}$ be an arbitrary subset of the set of positive integers $\mathbf{Z}^+$. Then we say that a given string $Y^n = \{y_i\}_{i=0}^{n-1}$ of length n can be $\mathcal{D}$*-embedded* into a given string $X^m = \{x_i\}_{i=0}^{m-1}$ of length $m \geq n$ if there exists a string $D^n = \{d_i\}_{i=0}^{n-1}$ of length n such that all d_i's lie in $\mathcal{D}$ and $y_i = x\left(\sum_{j=0}^{i} d_i\right)$, $0 \leq i < n$. The embedding is called *constrained* if $\mathcal{D} \neq \mathbf{Z}^+$ and *unconstrained* otherwise.

Let $U^n = \{u(t)\}_{t=0}^{n-1}$ be an observed segment of the output sequence (an observed random value). We guess the initial state of the GR, and starting from this state, under regular clocking, generate an m-long segment X^m where $m \geq n$. The following hypothesis H_0 has to be tested against alternative H_1:

H_0: X^m and U^n are independent (initial state of the GR is guessed incorrectly).

H_1: X^m and U^n are correlated (initial state of the GR is guessed correctly and U^n can be obtained from X^m by the above-described statistical model).

It follows from our assumption of the statistical model that each initial state of the GR gives rise to a conditional probability distribution on the set of all output sequences. Thus, hypothesis H_0 corresponds to a uniform distribution of U^n and alternative H_1 to a conditional distribution. Given an observed segment U^n, the optimal decision strategy (minimizing the probability of decision error) is to decide on the initial state that leads to the maximum posterior probability of U^n or, equivalently, the initial state whose corresponding sequence X^m has the maximum correlation with U^n.

Thus, for a correlation attack on irregularly clocked shift register, a measure for correlation between the output string produced by irregular clocking and the output of the GR, when clocked regularly, is required. Some possibilities have been suggested in the literature: the 'edit distance' [GM91], the 'embedding property' [GO95, GO96, Gol96, CG00], and the 'joint probability' [GP93a, GO95].

The basis for the *edit distance* correlation attack is a distance measure between two sequences of different lengths, suitably defined to reflect the transformation of the GR output sequence b to the output u according to the assumed statistical model. Thus, such distance measure should allow statistical discrimination between hypothesis H_0 and alternative H_1. Hypothesis H_0 is accepted if the distance between X^m and U^n is greater than a threshold that is defined by the given probabilities of the statistical decision errors. The 'constrained Levenshtein distance' (when the only edit operation is element deletion) was suggested in [GM91] as a possible distance measure for constrained clocking, although no analytical estimation of relevant probability distributions was given. It is also not clear how close the decision rule based on edit distance is to the one based on the maximum posterior probability (which is optimal for the given statistical model). The edit distance correlation attack does not seem to be very practical since its basic tool, the edit distance, is too general.

In the *embedding correlation* attack the objective is to find all possible initial states for the GR, such that for some $m \geq n$ a given segment $\{u(t)\}_{t=0}^{n-1}$ can be $\mathcal{D}$-embedded into the m-long output sequence of the GR produced under regular clocking, where $\mathcal{D}$ is the range of elements in a. The attack is successful if there are only few of such initial states. To check whether embedding is possible, one can use the direct matching algorithm for constrained embedding [CG00], that has computational complexity $O(nm)$, or one can use algorithms for calculating the Levenshtein distance [GM91, Mih93] for constrained and unconstrained clocking, respectively, that have computational complexity $O(n(m-n))$. Embedding is possible if and only if the distance is equal to $m-n$.

In [GO95] the unconstrained embedding attack is proved to be successful if and only if the deletion rate is smaller than 1/2 and the length of the observed output sequence is greater than a value that is linear in the length of the GR (where $m = m(n)$ is chosen in such a way that $n\overline{a} \leq m(n)$ and $\lim_{n\to\infty} n/m(n) = 1/\overline{a}$). According to [GO96], if $d = \max \mathcal{D}$ and the length of X^m is chosen to be maximum possible, so equal to dn (if $a_0 = 0$), then the constrained embedding attack is successful if the length of the observed output sequence is greater than a value linear in the GR length and superexponential in d, and is not successful if this length is smaller than a value linear in the GR length and exponential in d. This proves that, by making d sufficiently large, one cannot achieve the theoretical security against the embedding attack but one can significantly improve the practical security. To determine the constrained embedding probability analytically appears to be a very difficult combinatorial problem. This problem has only been solved in [Gol96] for the specific case when $\max \mathcal{D} = 2$.

It is obvious that embedding attacks are not optimal in general since they make no use of the probability distribution of the control sequence. The statistically optimal decision rule for distinguishing H_0 and H_1 has to be based on the joint probability and that is exactly the basis for the *probabilistic correlation* attack. In this attack, one decides on the initial state with maximum joint probability of X^m and U^n. The problem of efficiently computing this probability for constrained clocking is solved in [GP93a] with computational complexity $O(n(m-n))$. The recursive algorithm, presented in [GO95], allows to estimate the joint probability for unconstrained clocking if the distribution of the control sequence is geometric with average $1/p$. The computational complexity of this algorithm is $O(n(m-n))$. The length $m(n)$ should be chosen in such a way that $\lim_{n\to\infty} n/m(n) = p$. Then it can be proved that the unconstrained probabilistic attack is successful for any $0 \leq p < 1$ provided that

$$n > r\frac{1-p}{C}, \quad \text{where} \quad C \approx \left(1 - \frac{p}{2}\right)\log(2-p) + \frac{p}{2}\log p \ .$$

The correlation attack on the Shrinking Generator [CKM94], proposed by Johansson in [Joh98], is based on a maximum a posteriori (MAP) decoding algorithm for the deletion channel. This approach can as well be readily applied to the general model of a shift register under unconstrained clocking. A deletion rate p_d is used to define the deletion channel characteristics. If $p_d = 1/2$ then the model for unconstrained clocking is equivalent to the one of the Shrinking Generator. The suboptimal MAP decoding algorithm proposed in [Joh98] is likely also to work for deletion rate values different from 1/2 but that should be further examined by simulating the attack (since part of the suboptimal MAP decoding algorithm is based on simulation results).

All above-mentioned correlation attacks on the initial state of the GR imply an exhaustive search over all possible initial states. Thus, their computational complexity remains exponential. A more efficient fast correlation attack having polynomial complexity was suggested in [Gol95b]. The primary objective of this attack is to reconstruct a segment of the control sequence a and then, when having obtained enough (little more than the length of the GR) consecutive terms of a at any point of time, it is possible to determine the initial state of the GR uniquely or almost uniquely. The feedback polynomial of the GR is now assumed to be linear. The algorithm devised in [Gol95b] consists of iterative recomputation of posterior probabilities for unknown elements of the control sequence. The convergence condition that has to hold for successful reconstruction is the following:

$$\sum_{\omega} N_{d,\omega}(1-p)^{\omega} > 1$$

for all $d \in \mathcal{D}$ whose probability is not very close to zero, where $\mathcal{D}$ is the range of elements in a, p is the deletion rate and $N_{d,\omega}$ denotes the number of polynomials having weight $\omega + 1$ that are multiples of the GR feedback polynomial and which have the additional property that the distance between at least one pair of adjacent feedback connections (taps) is equal to $d + 1$. Unfortunately, the theoretical basis for the fast correlation attack devised in [Gol95b] is rather tentative and is not supported by experimental results.

We shall finally consider memoryless combiners of clock-controlled shift registers. Assume that the combining function is zero-order correlation immune and is known to the cryptanalyst. This means that the key stream s is correlated to at least one of the inputs to the combining function. Let u denote such an input sequence generated by the corresponding GR under irregular clocking. Thus, the known key-stream segment can be seen as the result of transmitting u through the binary symmetric channel (BSC) with known error probability equal to the correlation between u and s due to the combining function. The goal of correlation attacks is to reconstruct the initial state of the corresponding GR from the known segment of the key stream.

Embedding correlation attacks are infeasible in this case but edit distance and joint probability attacks are still applicable although less efficient. The idea of these attacks was described earlier in this section. The edit distance attack for the constrained clocking case can be based on the Levenshtein distance, as suggested in [GM91]. Except element deletion, an extra edit operation, namely element substitution, should be considered due to the BSC noise. The attack based on the joint probability for constrained clocking case was devised in [GP93a].

The idea of the *decimation attack* on combination generators from [Fil00] can also be extended to mount the correlation attack on the combiner of clock-controlled registers. For the pair 'CR plus GR' that generates sequence u assume that the feedback polynomial $f(x)$ of the GR is irreducible, the period π of the CR and the sum S of the control sequence a over the period are known and consider the uniform $(0, \pi)$-decimation of the key stream s. According to (3.2), this decimation of s is equal to the additively noised uniform (a_0, S)-decimation of sequence b generated by the GR under regular clocking. By Theorem 3.2, Item (i), the latter decimation belongs to $L_P(f_S)$. Thus, applying the original correlation attack [Sie85] we can reconstruct m bits in the (a_0, S)-decimation of b (the value for a_0 has to be searched exhaustively). The initial state of the GR can be obtained then by solving a system of m linear equations.

Nonlinear combiners of clock-controlled shift registers have been extensively studied for a long while but no possibility for a fast correlation attack has been reported. Therefore, it is reasonable to assume that these schemes provide sufficient level of security.

If the combining function of clock-controlled registers is correlation immune or has memory then correlation attacks based on many-to-one string edit distance and joint probability are still feasible (see [Gol01]). The efficiency of these attacks depends on an available pair of mutually correlated feedforward linear transforms of the output sequence and input sequences respectively, in the same but now regularly clocked combiner. A large correlation coefficient, a small memory size and a small number of input sequences to the linear transform of the input increase the efficiency of the attack. A theoretical estimation of the conditions for these attacks to be successful seems to be a difficult, yet unsolved problem.

3.7 Conclusion and Open Problems

The period of the output sequence generated by a clock-controlled LFSR with an irreducible feedback polynomial and an arbitrary structure of the control sequence was estimated. A sufficient condition for this period to reach its maximal value was formulated and some specific configurations of clock-controlled arrangements where the output sequence has maximal period were described. An interesting problem is to find other sufficient conditions for the period of the output sequence to reach its maximum.

Our investigation of randomness properties of clock-controlled LFSR's brings to the conclusion that in order to provide a close-to-uniform element distribution in the output sequence, the following rules have to be observed:

- Ensure that the feedback polynomial $f(x)$ of the GR is irreducible over $\mathrm{GF}(q)$, has degree m and order M with M sufficiently large compared to $q^m - 1$.

- The least common multiple of M and $q-1$ is equal by the order of magnitude to M.
- Generate the control sequence with S being relatively prime to M.

Note that if $f(x)$ is primitive then the first two requirements are met automatically. Moreover, if

- $f(x)$ is primitive over a prime field GF(p),
- the control sequence is such that S is relatively prime to $\lambda = p^m - 1$,
- $\sum_{k=i+1}^{i+h} a_k$ is not a multiple of λ for all $i = 0, 1, \ldots, \pi - 1$

then for such values of $h > 0$ the autocorrelation function $C(h)$ of the output sequence fulfils the Golomb's postulate (see the notation in Sect. 3.1).

The classical Geffe generator is characterized by strong correlation dependencies inherent to the key stream and, therefore, is susceptible to the basic and fast correlation attacks. The suggested generalized generator combines multiple inputs taken from arbitrary periodical sources over a finite field. The number of inputs can be varied. In particular, we imply that clock-controlled shift registers can be used for generating the input sequences, which will make the generator immune against fast correlation attacks. Using the optimal number of inputs allows to reduce correlation dependencies to the level when basic attacks become impractical, at the same time ensuring efficiency of the generator. Upper and lower bounds were estimated for the period and the linear complexity of the output sequence of the generalized Geffe generator. It remains an open problem to estimate a nontrivial lower bound for the linear complexity of the output sequence in the non-binary case.

Chapter 4

Some Statistical Attacks on Stream Ciphers

4.1 Introduction

The study of different theoretical aspects of the design and analysis of key-stream generators has two major directions. The first one is focused on investigating properties of the building blocks (e.g., feedback shift registers, logical functions, modulo N arithmetic, etc.) that constitute the generator, and on estimating the related number-theoretical characteristics of the key stream. Doing this, the main objective is to find methods for generating key-stream sequences with characteristics that provide security against algebraic attacks [DXS91]. This approach is followed in Chaps. 2 and 3 of this thesis.

The second direction is focused on statistical properties of a key stream and uses various statistical tests for goodness of fit, when a hypothesis specifies the distribution in the key stream. In this case, the principle hypothesis is that the key stream is a sequence of independent random variables uniformly distributed on a finite set. Yet, in practice it is almost impossible to detect minor departures from the principle hypothesis. We can only reliably detect the most unwanted deviations of statistical characteristics of the key stream from these of the purely random sequence. The characteristics critical for the security of a key stream are uniformity, statistical homogeneity and absence of substantial dependence between the key-stream elements. This is due to the fact that any statistical irregularities of this kind, present in a key stream, can be used in the algorithms for attacking the corresponding stream cipher [Jö02]. In this chapter we develop several attacks that exploit statistical weakness in the key stream.

The problem of testing a key stream for a linear recurrence perturbed with a nonuniform additive noise is studied in Sect. 4.2. Efficient solution of this problem allows to run a distinguishing attack on the corresponding stream cipher. The goal of this attack is to distinguish the black box containing the cipher from the one producing a purely random output. If any statistical irregularities are present in the key stream then, due to the plaintext redundancy, these can be also found in the ciphertext. Therefore, both known plaintext (attack on the key stream) and ciphertext-only scenarios are possible for the distinguishing attack. The background here is similar to the one that the cryptanalyst faces when developing algorithms for fast correlation attacks on combination generators. To solve the problem, we construct the maximal invariant statistic and the invariant test. However, high computational complexity makes use of this test impractical. For the particular case of trinomial feedback we construct a couple of invariant statistics that allow construction of computationally feasible tests.

In Sect. 4.3 we construct statistical tests to distinguish families of ciphertexts obtained from different plaintexts but using the same key-stream segment (so called, overlapping families). This problem was not widely discussed in the literature but definitively has both theoretical and practical importance. Our nonrandomized and randomized most powerful tests efficiently distinguish overlapping families in the case that they contain at most four ciphertexts of different length. Moreover, we provide explicit algorithms for constructing parameter intervals where these tests are uniformly most powerful.

New statistical procedures for selecting the most probable outcomes from the multinomial population are developed in Sect. 4.4. These procedures are based on the calculation of the reduced frequencies. This makes them more efficient in the case that the total number of outcomes is big compared to the amount of memory available. Useful applications can be found in frequency analysis, namely, where it is a part of a dictionary attack on block ciphers and various other attacks on codes. We prove the limit theorem for the distribution of reduced frequencies. These results appeared in [Kho98a].

4.2 Testing a Key Stream for a Noisy Linear Recurrence

Consider a binary, nonlinear combination generator with zero-order correlation immune combining function. This implies that some LFSR sequence $a = \{a_i\}_{i\geq 0}$, being one of the inputs to the combining function, is correlated to the key-stream sequence $z = \{z_i\}_{i\geq 0}$. In the statistical model we assume that z is the sequence of independent and identically distributed (i.i.d.) random variables. According to this model, we can consider sequence z as a sequence a with noise added to it, i.e., $z_i = a_i \oplus e_i \quad (i = 0, 1, 2, \dots)$, where $e = \{e_i\}_{i\geq 0}$ is a binary sequence of i.i.d. random variables. In order to distinguish between random variables and their values we will further denote random variables by the corresponding capital letters. Thus, in this notation for some $\delta \in \{(-1, 1) \setminus 0\}$ holds

$$\Pr(Z_i = a_i) = \Pr(E_i = 0) = \frac{1-\delta}{2} \quad (i = 0, 1, 2, \dots) \ .$$

The situation is illustrated in Fig. 4.1.

The characteristic polynomial of the n-bit long LFSR, correlated to the key stream, is assumed to be known to the cryptanalyst but the initial state vector

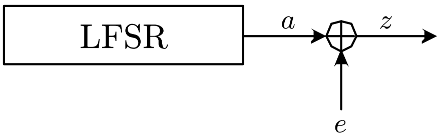


Figure 4.1: Statistical model of a combination generator

constitutes the unknown secret key of the generator. Let Γ denote the set of 2^n linear recurring sequences a in GF(2) satisfying this known recurrence

$$a_{i+n} = a_{i+k_{t-1}} \oplus \cdots \oplus a_{i+k_1} \oplus a_{i+k_0} \quad (i = 0, 1, 2, \dots) \tag{4.1}$$

but limited by their initial L elements for some $L \gg 0$. Assume that $0 = k_0 < k_1 < \cdots < k_{t-1} < n \leq L$. Assume also that a segment $(z_0, \dots, z_{L-1})$ of L key-stream digits is being observed. The problem of the cryptanalyst is to distinguish the key stream that is the noisy output of the LFSR, from the sequence of uniform, i.i.d. random variables. Further by a, z and e we will denote the initial L-long segments of the respective sequences.

The formulation of the problem above implicitly assumes that a segment of the plaintext is known to the cryptanalyst so that he could estimate the corresponding key-stream segment. However, a similar cryptanalytic method is also feasible in a ciphertext-only attack. In this case the correlation probability between an LFSR sequence and the ciphertext is defined by correlation properties of the combining function and the redundancy of the plaintext. Specific statistical characteristics of the plaintext depend on the system chosen for coding the plaintext. The plaintext distribution may differ from the uniform distribution on particular bits of the codewords (e.g., this holds for seven-bit ASCII and five-bit International Telegraph Code). In this case we need to decimate the ciphertext by the code length and consider the decimated ciphertext as the sum of the decimated LFSR sequence and the noise that results both from the combining function and the plaintext.

Consider the parameter space $\Omega = \{\theta = (a, \delta) \mid a \in \Gamma,\ \delta \in (-1, 1)\}$ and corresponding family of parameterized distributions over the sample space $\mathrm{GF}(2)^L$ of binary L-tuples as follows

$$\mathcal{P} = \{P_\theta \mid \theta \in \Omega\} \ ,$$

where

$$P_\theta(Z = z) = \left(\frac{1+\delta}{2}\right)^{wt(z \oplus a)} \left(\frac{1-\delta}{2}\right)^{L - wt(z \oplus a)} , \quad z \in \mathrm{GF}(2)^L \tag{4.2}$$

and $wt(z \oplus a)$ denotes the Hamming weight of binary string $z \oplus a$.

Let us identify the following two subsets in the parameter space Ω

$$\Omega_H = \{(a, \delta_1) \mid a \in \Gamma\} \quad \text{and} \quad \Omega_K = \{(a, 0) \mid a \in \Gamma\} \ ,$$

where δ_1 is a fixed *nonzero* value from the interval $(-1, 1)$. Then our distinguishing problem amounts to testing hypothesis H against alternative K, where

$$H = \{P_\theta \mid \theta \in \Omega_H\} \quad \text{and} \quad K = \{P_\theta \mid \theta \in \Omega_K\} \ . \tag{4.3}$$

Let G be the group of transformations of the sample space $\mathrm{GF}(2)^L$ such that $G = \{a(z) \mid a \in \Gamma\}$, where $a(z)$ is defined by $a(z) = a \oplus z$ for all $z \in \mathrm{GF}(2)^L$. Taking the definition of invariance as in [Leh97, pp. 282-284], it is not difficult to prove the following proposition.

Proposition 4.1 *The problem of testing hypothesis H against alternative K remains invariant under the group of transformations G.*

Proof: By (4.2), $P_{(a,\delta)}(Z = z) = P_{(a \oplus a',\delta)}(Z = z \oplus a')$ for every $a, a' \in \Gamma$ and any $\delta \in (-1, 1)$. It means that a random L-tuple Z has distribution $P_{(a,\delta)} \in \mathcal{P}$ if and only if the L-tuple $a'(Z) = Z \oplus a'$ has distribution $P_{(a \oplus a',\delta)} \in \mathcal{P}$. Thus, the family of distributions $\mathcal{P}$ remains invariant under the group G. The induced group of transformations $\overline{G}$ of the parameter space Ω contains the elements of the form $a'(a, \delta) = (a \oplus a', \delta)$. Thus, subsets Ω_H and Ω_K remain invariant under the group $\overline{G}$ meaning that for any $a \in \overline{G}$ the equations $a(\Omega_H) = \Omega_H$ and $a(\Omega_K) = \Omega_K$ hold. □

If a hypothesis testing problem is invariant under a group of transformations then it is natural to restrict attention to the tests invariant under this group. Test ϕ is invariant under the defined above group G if $\phi(a(z)) = \phi(z)$ for any $a \in \Gamma$ and $z \in \mathrm{GF}(2)^L$ (see the definition of invariance in [Leh97, p. 284]). Thus, tests invariant under G are free of the nuisance parameter a. This allows us to reduce the parameter space Ω and switch from the original problem (4.3) to the problem of testing hypothesis H' against alternative K', where

$$H' = \{\delta = \delta_1\} \quad \text{and} \quad K' = \{\delta = 0\} \ . \tag{4.4}$$

According to [Leh97, Theorem 1 on p. 285], a test ϕ is invariant under a transformation group G if and only if there exists a function h for which $\phi(z) = h(M(z))$ for all z, where $M(z)$ is a maximal invariant with respect to G. Note that a function is said to be *maximal invariant* if it is constant on the orbits of G but takes on a different value on each orbit. The maximal invariant function (or statistic) is constructed in the following proposition.

Proposition 4.2 *Let $s = (s_0, \ldots, s_{L-1})$ be an arbitrary L-tuple from $\mathrm{GF}(2)^L$. Assume that $(s_0, \ldots, s_{n-1})$ is the initial state vector of a linear recurring sequence and extend this sequence, according to (4.1), up to the length L, denoting the result by $a^{(s)}$. Then the function $T(s) = s \oplus a^{(s)}$ is maximal invariant under group G.*

Proof: Consider two points in $\mathrm{GF}(2)^L$ equivalent under G (denote this by $z^{(1)} \sim z^{(2)} \pmod{G}$) if there exists a transformation $a(z) \in G$ for which $a(z^{(1)}) = z^{(2)}$. This is a true equivalence relation since G is a group.

By the definition of group G,

$$z^{(1)} \sim z^{(2)} \pmod{G} \Longleftrightarrow \exists\, a \in \Gamma : \; z^{(1)} \oplus a = z^{(2)}$$

and thus

$$z^{(1)} \sim z^{(2)} \pmod{G} \Longleftrightarrow z^{(1)} \oplus z^{(2)} \in \Gamma \ .$$

Therefore, this equivalence relation is the congruence relation modulo the subgroup Γ. Equivalence classes are cosets of $\mathrm{GF}(2)^L$ modulo Γ and have the form $z \oplus \Gamma$. Therefore, these classes are all equal in size and their cardinality is given by the cardinality of Γ.

Consider $z = (z_0, \ldots, z_{L-1}) \in \mathrm{GF}(2)^L$ and $a^{(z)}$, the L-tuple from Γ having the initial state vector equal to $(z_0, \ldots, z_{n-1})$. It is obvious that for any vector $a \in \Gamma$ coset $a \oplus \Gamma$ is equal to Γ. Thus,

$$z \oplus \Gamma = z \oplus (\Gamma \oplus a^{(z)}) = (z \oplus a^{(z)}) \oplus \Gamma = z^0 \oplus \Gamma$$

and the initial n coordinates of vector z^0 are zero. Such a vector z^0 is called the basic vector of the coset and it is unique. Therefore, we may assume the n initial coordinates of an arbitrary vector $s \in \mathrm{GF}(2)^L$ to be equal to the initial state vector of the added recurring sequence. Now if we extend this initial state vector, according to (4.1), up to the length L and add the obtained vector $a^{(s)}$ to the original vector s then we will get a maximal invariant, that is the basic vector z^0 of the coset $z^0 \oplus \Gamma$ which contains vector s. □

Practical usage of invariant tests depending on maximal invariant statistic $T(s)$, constructed in Proposition 4.2, is complicated by the need to compute the following enormous sums that define the distribution of this maximal invariant

$$\sum_{a \in \Gamma} \left(\frac{1+\delta}{2} \right)^{wt(z \oplus a)} \left(\frac{1-\delta}{2} \right)^{L - wt(z \oplus a)} = \sum_{a \in \Gamma} P_{(a,\delta)}(Z = z) \ .$$

The computational complexity of this task is equivalent to $|\Gamma| = 2^n$ and for practical systems with appropriate values of L and n (say $n = 64$ and $L = 1500$) this amount of operations cannot be completed within a reasonable time period. Further, we will construct an invariant statistic that is not maximal invariant but for which it is feasible to estimate this distribution.

In the following discussion we need the LFSR feedback to be trinomial. This assumption does not impose extremely strict limitations compared to the general case since we can always find a trinomial multiple of the characteristic polynomial. The only thing we need from such a multiple is to be of a relatively low, compared to L, degree (see, e.g., [GM01, Roy02] for theoretical and algorithmic details of finding low-weight multiples of primitive polynomials over GF(2)). Thus, let recurrence (4.1) be trinomial, i.e., let $t = 2$ and

$$a_{i+n} = a_{i+k} \oplus a_i \quad (i = 0, 1, 2, \ldots) \tag{4.5}$$

for some $0 < k < n \leq L$, and consider the following ordered 10-tuple of nonnegative integers

$$C_4(k,n) = (0, k, 2k, 3k, n, n+k, n+2k, 2n, 2n+k, 3n) \ .$$

We will refer further to this sequence as to the 4-*scheme*. Longer schemes can be constructed in a similar way, although only the 4-scheme and the 5-scheme, which are the shortest ones, pose computational tasks having reasonable complexity. Longer schemes are computationally infeasible.

Let us index the elements in the tuple $C_4(k,n)$ starting from 1 and let $[j]$ denote the jth element in $C_4(k,n)$. It is obvious that for a linear recurring sequence $\{a_i\}_{i\geq 0}$ satisfying (4.5) the following identities hold

$$\begin{aligned} a_{[1]} &\oplus a_{[2]} \oplus a_{[5]} = 0 \\ a_{[2]} &\oplus a_{[3]} \oplus a_{[6]} = 0 \\ a_{[3]} &\oplus a_{[4]} \oplus a_{[7]} = 0 \\ a_{[5]} &\oplus a_{[6]} \oplus a_{[8]} = 0 \\ a_{[6]} &\oplus a_{[7]} \oplus a_{[9]} = 0 \\ a_{[8]} &\oplus a_{[9]} \oplus a_{[10]} = 0 \ . \end{aligned} \tag{4.6}$$

We will further assume that the elements in the tuple $C_4(k,n)$ are all distinct. This assumption is reasonable since that is the case for most of the recurring sequences used in practice.

Consider the following homomorphism of linear vector spaces over GF(2):

$$\Psi_{k,n} : \mathrm{GF}(2)^{10} \to \mathrm{GF}(2)^6 \quad \text{and} \quad \Psi_{k,n}(h) = (u_1, \ldots, u_6) = u$$

for any $h = (h_1, \ldots, h_{10}) \in \mathrm{GF}(2)^{10}$, where $u_1 = h_1 \oplus h_2 \oplus h_5$, $u_2 = h_2 \oplus h_3 \oplus h_6$ and so on according to (4.6). Note that if $k(u)$ is a pre-image of an element $u \in \mathrm{GF}(2)^6$ then $k(u) = h \oplus k(0)$ for any h with $\Psi_{k,n}(h) = u$, where 0 denotes the zero-vector in $\mathrm{GF}(2)^6$. Thus, $k(u)$ is the coset of $\mathrm{GF}(2)^{10}$ under the kernel of $\Psi_{k,n}$.

Let us use the homomorphism $\Psi_{k,n}$ to associate an L-tuple $z = \{a_i \oplus e_i\}_{0\leq i\leq L-1}$ with the sequence $U = \{u^{(i)}\}_{0\leq i<L-3n}$ of vectors from $\mathrm{GF}(2)^6$, where $u^{(i)} = \Psi_{k,n}(z_{i+[1]}, \ldots, z_{i+[10]})$. We will denote this fact as $\Psi_{k,n}(z) = U$. By virtue of (4.6), $\Psi_{k,n}(z) = \Psi_{k,n}(e)$ and moreover, for any $u \in \mathrm{GF}(2)^6$

$$\begin{aligned} &\Pr\left(\Psi_{k,n}(Z_{i+[1]}, \ldots, Z_{i+[10]}) = u\right) = \Pr\left\{(E_{i+[1]}, \ldots, E_{i+[10]}) \in k(u)\right\} = \\ &= \Pr(k(u)) = \sum_{h\in k(u)} \left(\frac{1+\delta}{2}\right)^{wt(h)} \left(\frac{1-\delta}{2}\right)^{10-wt(h)} \quad , \end{aligned} \tag{4.7}$$

where $Z_i = E_i \oplus a_i$. It is not difficult to estimate the mean and the variance of random vector $u^{(i)}$ and to show that

$$\mathbf{E}u^{(i)} = \left(\frac{1+\delta^3}{2}, \ldots, \frac{1+\delta^3}{2}\right); \quad \mathbf{D}u^{(i)} = \left(\frac{1-\delta^6}{2}, \ldots, \frac{1-\delta^6}{2}\right) \ .$$

It is easy to see that the initial four coordinates $h_1, \dots, h_4$ of a vector $h \in \mathrm{GF}(2)^{10}$ and vector $u \in \mathrm{GF}(2)^6$ uniquely determine the remaining coordinates of h if $h \in k(u)$. Thus, the cardinality of $k(u)$ is equal to 2^4 in the 4-scheme. In the 5-scheme the cardinality of a corresponding coset of $\mathrm{GF}(2)^{15}$ is equal to 2^5 and the number of cosets is equal to 2^{10} which is computationally feasible as well.

The constructed statistic U is invariant under the group G, although, it is not maximal invariant. With neglect of the inter-element dependence in the sequence $\{u^{(i)}\}_{0 \le i < L-3n}$, we can consider U to be a random sample from the distribution given in (4.7). Then, in order to test hypothesis (4.4) we can apply sequential analysis. It allows to minimize the number of observations, i.e., the sample size, and thus decreases the inter-element dependence in U. For the same reason it is preferable to use the 5-scheme. For instance, for the recurrence $a_{i+31} = a_{i+18} \oplus a_i$ (i.e., when $n = 31$ and $k = 18$) $C_4(18, 31) = (0, 18, 36, 54, 31, 49, 67, 62, 80, 93)$ and $C_5(18, 31) = (0, 18, 36, 54, 31, 49, 67, 62, 80, 93)$. Thus, both for the 4-scheme and 5-scheme, only five initial elements in U compose an independent sample. Practical distinguishing attacks on stream ciphers can be run using invariant tests that depend on the constructed statistics.

4.3 Testing a Ciphertext for Key-Stream Reuse

Consider a binary stream cipher that encrypts plaintext sequence $t = \{t_i\}_{i \ge 0}$ into ciphertext sequence $c = \{c_i\}_{i \ge 0}$ by coordinate-wise adding bits of key stream $z = \{z_i\}_{i \ge 0}$, i.e., $c_i = t_i \oplus z_i \quad (i = 0, 1, 2, \dots)$. For a periodic stream cipher the key stream repeats after d characters for some fixed d and a secure design of a key-stream generator implies a large value of d. However, both periodic and non-periodic stream ciphers generate the same key stream if the initial state of a key-stream generator turns out to be the same in different encrypting sessions. Such a situation, when different ciphertext messages are obtained using the same key-stream sequence is called *key-stream reuse*. Ciphertexts that reuse the same key stream are further denominated *overlapping*. Key-stream reuse results in a serious security breach for a stream cipher since, given the plaintext redundancy, overlapping ciphertexts can be decrypted using many classical techniques such as frequency analysis, dragging cribs, the probable word method, etc. [Bir99, Sin99]. In this section we will consider key-stream reuse for a family containing at most four ciphertexts of different lengths. Occurrence of larger overlapping families is very unlikely in practice.

4.3.1 Statistical Model

Assume that the following statistical model adequately describes the plaintext. Here we continue denoting random variables by the corresponding capital letters. Sequence t is regarded as a binary sequence of independent and identically distributed (i.i.d.) random variables with

$$\Pr(T_i = 0) = \frac{1-\delta}{2} \quad (i = 0, 1, 2, \dots) \quad \text{for some} \quad \delta \in \{(-1, 1) \setminus 0\} \ .$$

Our assumption makes sense since encoding of a book-style English text with some common codes like seven-bit ASCII or five-bit International Telegraph Code, results in a bit stream having a biased distribution of particular bits in the codewords. Then, decimating the encoded plaintext by the code length we get the sequence that corresponds to our statistical model.

Let $\{c^{(1)}, c^{(2)}, c^{(3)}, c^{(4)}\}$ be a quartet of overlapping ciphertexts in the sense that $c_i^{(j)} = t_i^{(j)} \oplus z_i \quad (i = 0, 1, 2, \ldots;\ j = 1, 2, 3, 4)$. Then

$$\begin{aligned}
&\Pr(C_i^{(j_1)} = C_i^{(j_2)}) = \Pr(T_i^{(j_1)} = T_i^{(j_2)}) = \frac{1+\delta^2}{2} \\
&\Pr(C_i^{(j_1)} = C_i^{(j_2)} = C_i^{(j_3)}) = \Pr(T_i^{(j_1)} = T_i^{(j_2)} = T_i^{(j_3)}) = \frac{1+3\delta^2}{4} \\
&\Pr(C_i^{(1)} = C_i^{(2)} = C_i^{(3)} = C_i^{(4)}) = \Pr(T_i^{(1)} = T_i^{(2)} = T_i^{(3)} = T_i^{(4)}) = \frac{1+6\delta^2+\delta^4}{8} \\
&\Pr(C_i^{(j_1)} = C_i^{(j_2)} = C_i^{(j_3)}) - \Pr(C_i^{(1)} = C_i^{(2)} = C_i^{(3)} = C_i^{(4)}) = \frac{1-\delta^4}{8}\ ,
\end{aligned} \tag{4.8}$$

where $i = 0, 1, 2, \ldots$ and $1 \le j_1 < j_2 < j_3 \le 4$. Here we assume that plaintexts $t^{(1)}, t^{(2)}, t^{(3)}, t^{(4)}$ are independent. If some n ciphertexts, members of an overlapping family, have the ith symbol identical then this family is said to have a *zero vertical n-gram* at position i. Probabilities (4.8) result in the following set of three distributions:

$$\begin{pmatrix} x_1 & x_2 \\ \frac{1+\delta^2}{2} & \frac{1-\delta^2}{2} \end{pmatrix} \quad \begin{pmatrix} y_1 & y_2 \\ \frac{1+3\delta^2}{4} & \frac{3(1-\delta^2)}{4} \end{pmatrix} \quad \begin{pmatrix} z_1 & z_2 & z_3 \\ \frac{1+6\delta^2+\delta^4}{8} & \frac{1-\delta^4}{2} & \frac{3(1-\delta^2)^2}{8} \end{pmatrix}\ , \tag{4.9}$$

where simple events correspond to the occurrence of

x_1 – a zero vertical bigram in a pair of overlapping ciphertexts;

y_1 – zero vertical trigram in a triple of overlapping ciphertexts;

z_1 – a zero vertical tetragram in a quartet of overlapping ciphertexts;

z_2 – a zero vertical trigram in a triple of ciphertexts that belong to the overlapping quartet but with no vertical tetragram at this position;

and x_2, y_2, z_3 are complementary events. The bottom rows contain the corresponding probabilities. The probability of z_2 is equal to the last probability in (4.8) multiplied by 4 since the subset $\{j_1, j_2, j_3\}$ can be chosen arbitrarily from $\{1, 2, 3, 4\}$. Note that the complementary event y_2 corresponds to the occurrence of a zero vertical bigram in a pair of ciphertexts that belong to the overlapping triple but with no vertical trigram at this position and complementary event z_3 corresponds to the occurrence of a zero vertical bigram in a pair of ciphertexts that belong to the overlapping quartet but with no vertical trigram or tetragram at this position.

The cryptanalytic problem that we are going to address in this section is how to distinguish a family of overlapping ciphertexts from a family containing ciphertexts that were obtained using not correlated segments of the key-stream. Assume that the family contains four ciphertexts with the shortest three ciphertexts having the lengths N_4, $N_4 + N_3$ and $N_4 + N_3 + N_2$ respectively. Therefore, N_4 vertical tetragrams, N_3 vertical trigrams and N_2 vertical bigrams are available. From a statistical point of view, we have a random sample of size $N_2 + N_3 + N_4$, where N_2 elements of the sample are distributed according to the first distribution in (4.9), N_3 elements according to the second distribution and N_4 elements according to the third if the family really contains overlapping ciphertexts. The opposite case is adequately described by the statistical model where all ciphertexts consist of uniform i.i.d. random variables. This is equivalent to the plaintext being the sequence of uniform i.i.d. random variables (i.e., $\delta = 0$). Thus, our distinguishing problem amounts to testing hypothesis H about the distribution of a random sample against alternative K with

$$H = \{\delta = 0\} \quad \text{and} \quad K = \{\delta = \delta_1\} \ ,$$

where δ_1 is a fixed *nonzero* value from the interval $(-1, 1)$. If hypothesis H is true then distributions (4.9) are of the form

$$\begin{pmatrix} x_1 & x_2 \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} \qquad \begin{pmatrix} y_1 & y_2 \\ \frac{1}{4} & \frac{3}{4} \end{pmatrix} \qquad \begin{pmatrix} z_1 & z_2 & z_3 \\ \frac{1}{8} & \frac{1}{2} & \frac{3}{8} \end{pmatrix} \ .$$

4.3.2 Most Powerful Tests

For the case of testing a simple hypothesis against a simple alternative the power of a test is defined being equal to the probability of rejecting the hypothesis if alternative is true. Using the fundamental lemma of Neyman and Pearson [Leh97, p. 74], let us construct a most powerful (MP) test for testing hypothesis H against simple alternative K. Note that a hypothesis (alternative) is called simple if it specifies only a single distribution, and otherwise it is said to be composite. Let the distributions under hypothesis H and alternative K be denoted P_0 and P_1 respectively. Then the likelihood ratio for a point x of the sample space is equal to

$$\begin{aligned}
\frac{P_1(x)}{P_0(x)} &= \frac{\left(\frac{1-\delta_1^2}{2}\right)^{n(x_2)} \left(\frac{1+\delta_1^2}{2}\right)^{N_2-n(x_2)} \left(\frac{3(1-\delta_1^2)}{4}\right)^{n(y_2)} \left(\frac{1+3\delta_1^2}{4}\right)^{N_3-n(y_2)}}{\left(\frac{1}{2}\right)^{N_2} \left(\frac{3}{4}\right)^{n(y_2)} \left(\frac{1}{4}\right)^{N_3-n(y_2)}} \times \\
&\times \frac{\left(\frac{1-\delta_1^4}{2}\right)^{n(z_2)} \left(\frac{3(1-\delta_1^2)^2}{8}\right)^{n(z_3)} \left(\frac{1+6\delta_1^2+\delta_1^4}{8}\right)^{N_4-n(z_2)-n(z_3)}}{\left(\frac{1}{2}\right)^{n(z_2)} \left(\frac{3}{8}\right)^{n(z_3)} \left(\frac{1}{8}\right)^{N_4-n(z_2)-n(z_3)}} = \\
&= \left(\frac{1-\delta_1^2}{1+\delta_1^2}\right)^{n(x_2)} \left(\frac{1-\delta_1^2}{1+3\delta_1^2}\right)^{n(y_2)} \left(\frac{1-\delta_1^4}{1+6\delta_1^2+\delta_1^4}\right)^{n(z_2)} \times \\
&\times \left(\frac{\left(1-\delta_1^2\right)^2}{1+6\delta_1^2+\delta_1^4}\right)^{n(z_3)} \left(1+\delta_1^2\right)^{N_2} \left(1+3\delta_1^2\right)^{N_3} \left(1+6\delta_1^2+\delta_1^4\right)^{N_4} \ ,
\end{aligned}$$

where $n(x_2)$, $n(y_2)$, $n(z_2)$ and $n(z_3)$ denote the number of occurrences of x_2, y_2, z_2 and z_3 respectively in the random sample x.

Let us select and fix the level of significance α. The MP test for testing H against K at level α is (see [Leh97, p. 74])

$$\phi(x) = \begin{cases} 1, & \text{when} \quad P_1(x) > CP_0(x), \\ \gamma, & \text{when} \quad P_1(x) = CP_0(x), \\ 0, & \text{when} \quad P_1(x) < CP_0(x), \end{cases} \tag{4.10}$$

where C and γ are selected to satisfy the identity

$$P_0 \left\{ \frac{P_1(X)}{P_0(X)} > C \right\} + \gamma P_0 \left\{ \frac{P_1(X)}{P_0(X)} = C \right\} = \alpha \ .$$

Let us explain that according to the MP test ϕ, the hypothesis is accepted if a simple event x with $\phi(x) = 0$ occurs, rejected if $\phi(x) = 1$ and rejected with probability γ if $\phi(x) = \gamma$. In the last case we will say that such an event x requires an additional trial.

Let us rewrite inequalities $\frac{P_1(x)}{P_0(x)} \lesseqgtr C$ in the form

$$\left(\frac{1-\delta_1^2}{1+\delta_1^2}\right)^{n(x_2)} \left(\frac{1-\delta_1^2}{1+3\delta_1^2}\right)^{n(y_2)} \left(\frac{1-\delta_1^4}{1+6\delta_1^2+\delta_1^4}\right)^{n(z_2)} \left(\frac{\left(1-\delta_1^2\right)^2}{1+6\delta_1^2+\delta_1^4}\right)^{n(z_3)} \lesseqgtr \widetilde{C} \ , \tag{4.11}$$

where

$$\widetilde{C} = C\left(1+\delta_1^2\right)^{-N_2}\left(1+3\delta_1^2\right)^{-N_3}\left(1+6\delta_1^2+\delta_1^4\right)^{-N_4} \ . \tag{4.12}$$

As a function of $n(x_2)$, $n(y_2)$, $n(z_2)$ and $n(z_3)$, the value on the left hand side of (4.11) varies within the closed interval

$$\left[\left(\frac{1-\delta_1^2}{1+\delta_1^2}\right)^{N_2} \left(\frac{1-\delta_1^2}{1+3\delta_1^2}\right)^{N_3} \left(\frac{1-\delta_1^4}{1+6\delta_1^2+\delta_1^4}\right)^{N_4}, 1\right] \ . \tag{4.13}$$

Therefore, if

$$\widetilde{C} < \left(\frac{1-\delta_1^2}{1+\delta_1^2}\right)^{N_2} \left(\frac{1-\delta_1^2}{1+3\delta_1^2}\right)^{N_3} \left(\frac{1-\delta_1^4}{1+6\delta_1^2+\delta_1^4}\right)^{N_4}$$

then the critical region of the MP test coincides with the whole sample space and $\alpha = 1$. If $\widetilde{C} > 1$ then the critical region of the MP test is empty and $\alpha = 0$. Thus, depending upon α, the constant $\widetilde{C}$ can vary its value within the closed interval (4.13).

We introduce the following notation

$$L(n(x_2), n(y_2), n(z_2), n(z_3); \delta) =$$

$$= \left(\frac{1-\delta^2}{1+\delta^2}\right)^{n(x_2)} \left(\frac{1-\delta^2}{1+3\delta^2}\right)^{n(y_2)} \left(\frac{1-\delta^4}{1+6\delta^2+\delta^4}\right)^{n(z_2)} \left(\frac{\left(1-\delta^2\right)^2}{1+6\delta^2+\delta^4}\right)^{n(z_3)}$$

and also denote $\Delta = \frac{1+\delta^2}{1-\delta^2}$. Note that with $\delta \in (-1, 1)$ the value of Δ varies within the interval $(1, +\infty)$ and Δ, as a function of δ, strictly increases when the absolute value of δ goes from 0 to 1. Inequalities (4.11) can be rewritten in the following form:

$$\Delta_1^{-n(x_2)}(2\Delta_1 - 1)^{-n(y_2)}(2\Delta_1^2 - 1)^{-n(z_3)} \left(\frac{2\Delta_1^2 - 1}{\Delta_1}\right)^{-n(z_2)} \lesseqgtr \widetilde{C} \ , \tag{4.14}$$

where $\Delta_1 = \frac{1+\delta_1^2}{1-\delta_1^2}$.

The sample space of our random experiment is contained within the four-dimensional rectangular parallelepiped spanned by the vectors $(N_2, 0, 0, 0)$, $(0, N_3, 0, 0)$, $(0, 0, N_4, 0)$ and $(0, 0, 0, N_4)$ and each simple event $(n(x_2), n(y_2), n(z_2), n(z_3))$ belongs to the rectangular full-dimensional lattice that passes through the origin of the coordinate system. For instance, projection of the sample space onto the three-dimensional space in coordinates $(n(x_2), n(y_2), n(z_2))$ corresponds to all points of the lattice contained in the parallelepiped of size $N_2 \times N_3 \times N_4$, and projection onto the three-dimensional space in coordinates $(n(x_2), n(z_2), n(z_3))$ is shown in Fig. 4.2.

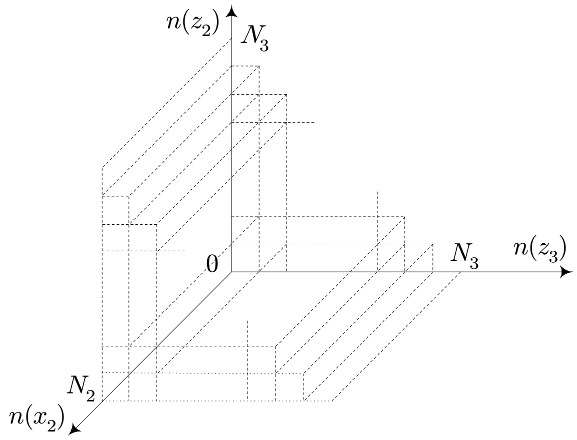


Figure 4.2: Sample space projected onto coordinates $(n(x_2), n(z_2), n(z_3))$

By taking the logarithm of inequalities (4.14) we obtain

$$-n(x_2)\ln\Delta_1 - n(y_2)\ln(2\Delta_1 - 1) - n(z_3)\ln(2\Delta_1^2 - 1) - n(z_2)\ln\frac{2\Delta_1^2 - 1}{\Delta_1} \lesseqgtr \ln\widetilde{C} \ .$$

According to (4.10), the critical region of an MP test consists of the points of the sample space that in the four-dimensional space lie “under” the hyperplane described by the following equation in coordinates $(n(x_2), n(y_2), n(z_2), n(z_3))$:

$$n(x_2)\ln\Delta_1 + n(y_2)\ln(2\Delta_1 - 1) + n(z_3)\ln(2\Delta_1^2 - 1) + n(z_2)\ln\frac{2\Delta_1^2 - 1}{\Delta_1} = -\ln\widetilde{C} \ . \tag{4.15}$$

In particular, if the critical region is not empty then it always contains the origin of the coordinate system. A contra, the region of acceptance lies “above” the plane. This hyperplane intersects the axes $n(x_2)$, $n(y_2)$, $n(z_2)$, $n(z_3)$ in the respective points having coordinates

$$\begin{array}{ll} \left(-\frac{\ln \widetilde{C}}{\ln \Delta_1}, 0, 0, 0\right); & \left(0, -\frac{\ln \widetilde{C}}{\ln(2\Delta_1 - 1)}, 0, 0\right); \\ \left(0, 0, -\left(\ln \frac{2\Delta_1^2 - 1}{\Delta_1}\right)^{-1} \ln \widetilde{C}, 0\right); & \left(0, 0, 0, -\frac{\ln \widetilde{C}}{\ln(2\Delta_1^2 - 1)}\right) \quad . \end{array} \tag{4.16}$$

The results in the next sections substantially depend on whether the MP test determined by (4.10) is randomized or nonrandomized. According to the definition, this test is nonrandomized if under distribution P_0 (i.e., when hypothesis H is true) the probability of the event $P_1(X) = CP_0(X)$ is equal to zero, otherwise, the MP test is randomized. If the same test is most powerful for all distributions contained in a composite alternative then such a test is called Uniformly Most Powerful (UMP) (see [Leh97, p. 72]). The rest of Sect. 4.3 will be focused on the construction of UMP tests.

4.3.3 Nonrandomized Most Powerful Tests

Consider first a nonrandomized test, i.e., we assume that $P_0\{P_1(X) = CP_0(X)\} = 0$. In this case hyperplane (4.15), dividing the sample space, does not contain any points of the sample space and the MP test for testing H against K is of the form

$$\phi(x) = \begin{cases} 1, & \text{when} \quad L(n(x_2), n(y_2), n(z_2), n(z_3); \delta_1) > \widetilde{C}, \\ 0, & \text{when} \quad L(n(x_2), n(y_2), n(z_2), n(z_3); \delta_1) < \widetilde{C}. \end{cases} \tag{4.17}$$

We will further assume threshold $\widetilde{C}$, defined by (4.12), to be a constant not depending on δ_1 and taking on its value within interval (4.13). We can also assume that $\widetilde{C} \neq 1$ since if $\widetilde{C} = 1$ then $L(0, 0, 0, 0; \delta_1) = 1 = \widetilde{C}$ and test (4.17) will be randomized.

Theorem 4.3 *There exists an interval $\left(\underline{\delta}, \overline{\delta}\right)$ containing δ_1, such that the MP test determined by (4.17) is UMP for testing hypothesis H against composite alternative $K^* = \left\{\delta \in \left(\underline{\delta}, \overline{\delta}\right)\right\}$.*

Proof: The proof is based on the construction of an interval $\left(\underline{\delta}, \overline{\delta}\right)$ for a given value of δ_1.

Note that the functions

$$\left(\ln \frac{1 + \delta^2}{1 - \delta^2}\right)^{-1}; \ \left(\ln \frac{1 + 3\delta^2}{1 - \delta^2}\right)^{-1}; \ \left(\ln \frac{1 + 6\delta_1^2 + \delta^4}{1 - \delta^4}\right)^{-1}; \ \left(\ln \frac{1 + 6\delta_1^2 + \delta^4}{(1 - \delta^2)^2}\right)^{-1}$$

involved in a nonzero coordinate of intersection points (4.16), strictly decrease from $+\infty$ to 0 when the absolute value of δ goes from 0 to 1. Also recall that $\widetilde{C}$ is positive, less than 1 and thus $-\ln \widetilde{C} > 0$. Therefore, intersection points of plane (4.15) with the axes contract to the origin of the coordinate system when the absolute value of δ goes from 0 to 1. Thus, the critical region does not become bigger when the absolute value of δ increases and does not reduce in size if it decreases.

Let us construct a set σ containing points of the region of acceptance that are the closest in the Euclidean metric to dividing hyperplane (4.15). This set can be defined in a following way:

$$\begin{aligned}\sigma \;&=\; \Big\{ A(b,c,d) = (x,b,c,d) \mid b \in \overline{0,N_3};\ c,d \in \overline{0,N_4};\ c+d \le N_4;\ x \in \overline{0,N_2};\\ x \;&=\; \left\lceil \left(-\ln \widetilde{C} - b\ln(2\Delta_1 - 1) - d\ln(2\Delta_1^2 - 1) - c\ln\frac{2\Delta_1^2 - 1}{\Delta_1} \right) (\ln \Delta_1)^{-1} \right\rceil \Big\}\,,\end{aligned}$$

where halved square brackets denote the ceiling function. If the value of x calculated using the above formula, turns out to be greater than N_2 then the point (x,b,c,d) is not included into σ. The cardinality of the constructed set σ does not exceed $\frac{N_4(N_4+1)(N_3+1)}{2}$.

Now, if for some $\delta' \in (-1,1)$ such that $|\delta'| < |\delta_1|$ the region of acceptance of the following test

$$\phi'(x) = \begin{cases} 1, & \text{when} \quad L(n(x_2), n(y_2), n(z_2), n(z_3); \delta') > \widetilde{C}, \\ 0, & \text{when} \quad L(n(x_2), n(y_2), n(z_2), n(z_3); \delta') < \widetilde{C} \end{cases} \qquad (4.18)$$

for testing hypothesis H against simple alternative $K' = \{\delta = \delta'\}$ contains all elements of σ then the regions of acceptance of tests ϕ' and (4.17) coincide, i.e., tests ϕ and ϕ' are equal and have the same level of significance. Moreover, if we construct the MP test for any alternative $\delta \in (\delta', \delta_1)$ (or $\delta \in (\delta_1, \delta')$) the corresponding MP test has the form of (4.18) (δ') and is equal to ϕ.

Note that the test determined by (4.18) is most powerful for testing H against K'. The presented arguments result in the following algorithm for finding the upper bound $\overline{\delta}$ for a given value of $\delta_1 \in (-1,0)$ (lower bound $\underline{\delta}$ for $\delta_1 \in (0,1)$):

1. construct set σ;
2. for each element $A = (a,b,c,d) \in \sigma$ find the corresponding value of $\delta_A > 0$ such that dividing hyperplane (4.15) contains the point having coordinates A; the corresponding value of $\Delta_A = \frac{1+\delta_A^2}{1-\delta_A^2}$ is the root of the equation

$$a\ln x + b\ln(2x-1) + d\ln(2x^2-1) + c\ln\frac{2x^2-1}{x} = -\ln\widetilde{C} \qquad (4.19)$$

   for the unknown $x > 1$; if $\delta_1 \in (-1,0)$ then take $\delta_A = -\sqrt{\frac{\Delta_A - 1}{\Delta_A + 1}}$ and if $\delta_1 \in (0,1)$ then take $\delta_A = \sqrt{\frac{\Delta_A - 1}{\Delta_A + 1}}$;
3. set $\overline{\delta}$ equal to $\min\{\delta_A \mid A \in \sigma\}$ ($\underline{\delta}$ equal to $\max\{\delta_A \mid A \in \sigma\}$).

The derivative of the left-hand side of (4.19) is positive for $x > 1$ unless $a = b = c = d = 0$. Therefore, the corresponding function strictly increases from 0 to $+\infty$ for $x > 1$ and equation (4.19) has a unique solution for any $0 < \widetilde{C} \leq 1$. If $(0,0,0,0) \in \sigma$, equation (4.19) is solvable if and only if $\widetilde{C} = 1$. But if $\widetilde{C} = 1$ then dividing hyperplane (4.15) contains the point $(0,0,0,0)$ of the sample space which means that the test is randomized. Thus, necessarily $\widetilde{C} > 1$ and if $A = (0,0,0,0) \in \sigma$ then the root of (4.19) is assumed to be $+\infty$ and the corresponding $\delta_A = \pm 1$.

In order to find the lower bound $\underline{\delta}$ for a given value of $\delta_1 \in (-1, 0)$ (upper bound $\overline{\delta}$ for $\delta_1 \in (0,1)$) it is sufficient to take the floor function instead of the ceiling function in the definition of σ and take the maximum (resp. minimum) in Step 3 of the algorithm. □

Note 4.4 A set containing points of the region of acceptance that are the closest in the Euclidean metric to dividing hyperplane (4.15) can be constructed in a way different from the one used for constructing σ. For instance, we can use

$$\begin{aligned} \sigma_1 &= \Bigg\{ A(a,b,d) = (a,b,x,d) \mid a \in \overline{0,N_2};\ b \in \overline{0,N_3};\ x,d \in \overline{0,N_4};\ x + d \leq N_4; \\ x &= \left\lceil \left(-\ln \widetilde{C} - a \ln \Delta_1 - b \ln(2\Delta_1 - 1) - d \ln(2\Delta_1^2 - 1) \right) \left(\ln \frac{2\Delta_1^2 - 1}{\Delta_1} \right)^{-1} \right\rceil \Bigg\} . \end{aligned}$$

The cardinality of this σ_1 does not exceed $(N_2+1)(N_3+1)(N_4+1)$. Or, alternatively, we can take

$$\begin{aligned} \sigma_2 &= \Bigg\{ A(a,c,d) = (a,x,c,d) \mid a \in \overline{0,N_2};\ c,d \in \overline{0,N_4};\ c + d \leq N_4;\ x \in \overline{0,N_3}; \\ x &= \left\lceil \left(-\ln \widetilde{C} - a \ln \Delta_1 - d \ln(2\Delta_1^2 - 1) - c \ln \frac{2\Delta_1^2 - 1}{\Delta_1} \right) (\ln(2\Delta_1 - 1))^{-1} \right\rceil \Bigg\} . \end{aligned}$$

Its cardinality will not exceed $\frac{N_4(N_2+1)(N_4+1)}{2}$.

Note 4.5 If a family, analyzed for key-stream reuse, contains only three ciphertexts then $N_4 = 0$ in our statistical model and the algorithm for finding the upper and lower bounds from Theorem 4.3 can be substantially simplified. In this case the sample space corresponds to all points of the flat rectangular lattice contained in the parallelepiped of size $N_2 \times N_3$. Dividing hyperplane (4.15) is a line

$$n(x_2) \ln \Delta_1 + n(y_2) \ln(2\Delta_1 - 1) = -\ln \widetilde{C} \ .$$

The situation is illustrated in Fig. 4.3. Note that the angle $\psi < \pi/4$ since $\ln \Delta_1 < \ln(2\Delta_1 - 1)$ when $\Delta_1 > 1$.

Step 2 of the algorithm requires solving equations of the form

$$a \ln x + b \ln(2x - 1) = -\ln \widetilde{C} \tag{4.20}$$

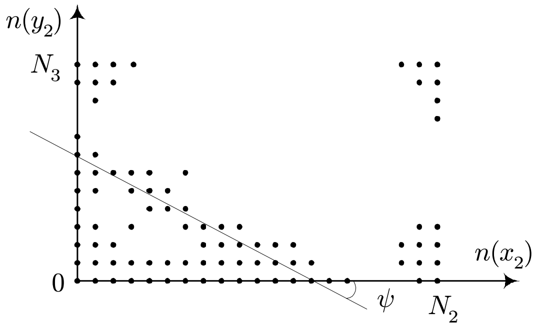


Figure 4.3: The sample space when $N_4 = 0$

for $x > 1$, where $(a, b) \in \sigma$. In fact, the particular roots of these equations are not relevant to the algorithm, it is sufficient to be able to compare the roots for different values of $(a, b) \in \sigma$. Only the equation, which root is maximal (or minimal), has to be solved. The following statement will be needed below.

Lemma 4.6 *The inequality* $(2x-1)^\alpha \lesseqgtr x^\beta$ *with* $\alpha, \beta \in \mathbb{N}$*,* $x > 1$

(i) takes on the sign '>' if $\beta \leq \alpha$*;*

(ii) takes on the sign '<' if $\beta \geq 2\alpha$*;*

(iii) takes on the sign '>' for $x \in (1, Q)$ *and sign '<' for* $x \in (Q, +\infty)$ *if* $\alpha < \beta < 2\alpha$*, where* $Q > 1$ *is uniquely determined by* $(2Q-1)^\alpha = Q^\beta$*.*

Proof: Item (i) is obvious since $2x - 1 > x$ for $x > 1$.

Assume now that $\beta > \alpha$. Let us analyze the function $f(x)$ that is the difference of the logarithms taken of both sides in inequality $(2x-1)^\alpha \lesseqgtr x^\beta$, so $f(x) = \alpha \ln(2x-1) - \beta \ln x$. Now we take the derivative of $f(x)$ and put it equal to zero:

$$f'(x) = \alpha \frac{2}{2x-1} - \beta \frac{1}{x} = 0 \quad \text{so} \quad x = \frac{\beta}{2(\beta - \alpha)} \quad .$$

Since $x = \frac{\beta}{2(\beta-\alpha)} > 1$ if and only if $\beta < 2\alpha$, the derivative $f'(x)$ is negative for $x > 1$ if $\beta \geq 2\alpha$ and, therefore, function $f(x)$ strictly decreases in this case. But $f(1) = 0$ and thus $f(x) < 0$ when $x > 1$ if $\beta \geq 2\alpha$. This proves Item (ii).

Now, if $\beta < 2\alpha$ then $\frac{\beta}{2(\beta-\alpha)} > 1$ and

$$\begin{aligned} f'(x) &> 0 \quad \text{for} \quad x \in \left(1, \frac{\beta}{2(\beta-\alpha)}\right) = I_1 \\ f'(x) &< 0 \quad \text{for} \quad x \in \left(\frac{\beta}{2(\beta-\alpha)}, +\infty\right) = I_2 \ , \end{aligned}$$

meaning that $f(x)$ strictly increases in the interval I_1 and strictly decreases to $-\infty$ on I_2. Therefore, since $f(1) = 0$, there exists the point $Q > 1$ such that $f(Q) = 0$ and $f(x) > 0$ when $x \in (1, Q)$ and $f(x) < 0$ when $x \in (Q, +\infty)$. □

We have to compare the roots of the following two equations having the form of (4.20)

$$x^a(2x-1)^b = \widetilde{C}^{-1} \quad \text{and} \quad x^c(2x-1)^d = \widetilde{C}^{-1} \ , \tag{4.21}$$

where $(a, b), (c, d) \in \sigma$ and $x > 1$. Without loss of generality it can be assumed that $c \geq a$.

Proposition 4.7 *Let Δ_1 and Δ_2 be the respective roots of equations (4.21) and $c \geq a$. Then*

(i) $\Delta_1 \geq \Delta_2$ *if* $d \geq b$*;*

(ii) $\Delta_1 < \Delta_2$ *if* $d < b$ *and* $c - a \leq b - d$*;*

(iii) if $d < b$ *and* $c-a > b-d$ *then* $\Delta_1 \geq \Delta_2$ *if and only if* $T^a(2T-1)^b \leq T^c(2T-1)^d$*, where* $T = \sqrt[c-a]{\widetilde{C}^{-1}}$.

Proof: Item (i) is obvious since $c \geq a$. If $c = a$ then Item (ii) is obvious too.

Assume now that $d < b$, $c - a > b - d$ and consider functions $y = x^a(2x-1)^b$ and $y = x^c(2x-1)^d$. Both of these functions strictly increase for $x > 1$ and their plots intersect in the points that are the roots of the equation $x^{c-a} = (2x-1)^{b-d}$. From Lemma 4.6 it follows that this equation has a root $Q > 1$ if and only if $b-d < c-a < 2(b-d)$ and this root is unique. This case is illustrated in Fig. 4.4(b) (see Lemma 4.6 Item (iii)), the case when $c-a \leq b-d$ is illustrated in Fig. 4.4(a) (see Lemma 4.6 Item (i)) and the case when $c - a \geq 2(b-d)$ is illustrated in Fig. 4.4(c) (see Lemma 4.6 Item (ii)). In the latter two cases we assume that $Q = 1$.

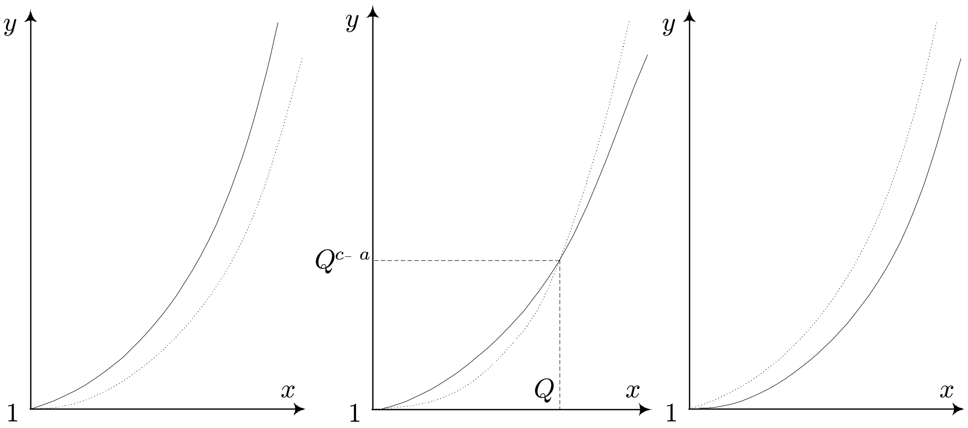
Finally, if $0 < c - a \leq b - d$ then $\Delta_1 < \Delta_2$ (this concludes the proof for Item (ii)) and if $c - a > b - d$ then

$$\begin{aligned} &\Delta_1 \geq \Delta_2 \Longleftrightarrow \widetilde{C}^{-1} \geq Q^{c-a} = (2Q-1)^{b-d} \Longleftrightarrow \\ &\Longleftrightarrow T \overset{\text{def.}}{=} \sqrt[c-a]{\widetilde{C}^{-1}} \geq Q \Longleftrightarrow T^a(2T-1)^b \leq T^c(2T-1)^d \ . \end{aligned}$$

This proves Item (iii). □

4.3.4 Randomized Most Powerful Tests

Consider now a randomized test. In that case the surface, dividing the sample space into the critical region and the region of acceptance, contains some points of the sample space and the existence of a UMP test depends on the number of points contained in this surface.



(a) when $c - a \leq b - d$ (b) $b - d < c - a < 2(b - d)$ (c) when $c - a \geq 2(b - d)$

Figure 4.4: Functions $y = x^a(2x - 1)^b$ (solid line) and $y = x^c(2x - 1)^d$ (dotted line)

Proposition 4.8 *A randomized test $\phi(x)$ for testing hypothesis H against composite alternative $K^* = \left\{\delta \in \left(\underline{\delta}, \overline{\delta}\right)\right\}$ cannot be UMP if*

$$\#\{x \,|\, \phi(x) \notin \{0, 1\}\} \geq 2 \ . \tag{4.22}$$

Proof: Suppose that test ϕ is UMP for testing H against K^*. Then for any $\delta_1 \in \left(\underline{\delta}, \overline{\delta}\right)$ it is most powerful for testing hypothesis H against simple alternative $K = \{\delta = \delta_1\}$. By the fundamental lemma of Neyman and Pearson, an MP test has the form of (4.10) and, by condition (4.22), there exist two simple events x and y such that

$$\frac{P_1(x)}{P_0(x)} = \frac{P_1(y)}{P_0(y)} = C(\delta_1) \ . \tag{4.23}$$

Moreover, this identity should hold for all $\delta_1 \in \left(\underline{\delta}, \overline{\delta}\right)$. By (4.11),

$$\begin{aligned}\frac{P_1(x)}{P_0(x)} = \frac{P_1(y)}{P_0(y)} \iff \quad & \Delta_1^{-a_1}(2\Delta_1 - 1)^{-b_1}(2\Delta_1^2 - 1)^{-d_1}\left(\frac{2\Delta_1^2 - 1}{\Delta_1}\right)^{-c_1} = \\ = \quad & \Delta_1^{-a_2}(2\Delta_1 - 1)^{-b_2}(2\Delta_1^2 - 1)^{-d_2}\left(\frac{2\Delta_1^2 - 1}{\Delta_1}\right)^{-c_2} \quad ,\end{aligned}$$

where $x = (a_1, b_1, c_1, d_1)$ and $y = (a_2, b_2, c_2, d_2)$. The above identity should hold for all $\Delta_1 \in \left(\underline{\Delta}, \overline{\Delta}\right)$ but that is impossible since this identity, considered as an algebraic equation in the unknown Δ_1, has a finite number of roots. Therefore, we arrived at a contradiction. □

Note 4.9 Identity (4.23) is possible in principle. Take, for instance, $x = (a_1, b_1, 0, 0)$, $y = (a_2, b_2, 0, 0)$. Then identity (4.23) is of the following form:

$$\frac{P_1(X_1)}{P_0(X_1)} = \Delta_1^{-a_1}(2\Delta_1 - 1)^{-b_1} = \Delta_1^{-a_2}(2\Delta_1 - 1)^{-b_2} = \frac{P_1(X_2)}{P_0(X_2)} \ .$$

Assume that $a_1 > a_2$ and $b_1 < b_2$. Then, as follows from Lemma 4.6, equation $\Delta_1^{a_1 - a_2} = (2\Delta_1 - 1)^{b_2 - b_1}$ has a root $Q > 1$ if $b_2 - b_1 < a_1 - a_2 < 2(b_2 - b_1)$.

From Proposition 4.8 it follows that in a UMP test for testing hypothesis H against composite alternative K^* there will be at most one simple event requiring an additional trial.

Proposition 4.10 *If for some nonzero $\delta_1 \in (-1, 1)$ there are no simple events having the same likelihood ratio then there exists an interval $\left(\underline{\delta}, \overline{\delta}\right)$ containing δ_1, such that for any fixed level of significance, corresponding MP test (4.10) is UMP for testing hypothesis H against composite alternative $K^* = \left\{\delta \in \left(\underline{\delta}, \overline{\delta}\right)\right\}$.*

Proof: If for some level of significance corresponding MP test (4.10) is nonrandomized then the claimed result immediately follows from Theorem 4.3. Therefore, it suffices to consider only randomized tests having a *single* point that requires the additional trial. If this point (simple event) has coordinates (A, B, C, D) then, by (4.11), $L(A, B, C, D; \delta_1) = \widetilde{C}$.

For any nonzero $\delta' \in (-1, 1)$ the following hyperplane

$$L(n(x_2), n(y_2), n(z_2), n(z_3); \delta') = L(A, B, C, D; \delta') \tag{4.24}$$

divides the sample space and defines a randomized MP test for testing hypothesis H against simple alternative $\{\delta = \delta'\}$. Such a test requires the additional trial for simple event (A, B, C, D). In order to prove the proposition, it is sufficient to find an interval $\left(\underline{\delta}, \overline{\delta}\right)$ containing δ_1 such that for every $\delta' \in \left(\underline{\delta}, \overline{\delta}\right)$ the critical region of an MP test defined by (4.24) is the same and simple event (A, B, C, D) is the only one requiring the additional trial in each of the tests.

For an arbitrary pair of simple events (a_1, b_1, c_1, d_1) and (a_2, b_2, c_2, d_2) consider the following equation in the unknown $\Delta > 1$

$$L(a_1, b_1, c_1, d_1; \Delta) = L(a_2, b_2, c_2, d_2; \Delta) \ ,$$

where

$$L(a, b, c, d; \Delta) = \Delta^{-a}(2\Delta - 1)^{-b}(2\Delta^2 - 1)^{-d}\left(\frac{2\Delta^2 - 1}{\Delta}\right)^{-c}$$

for any (a, b, c, d) with $a \in 0, \ldots, N_2$; $b \in 0, \ldots, N_3$; $c, d \in 0, \ldots, N_4$ and $c + d \leq N_4$. By (4.11), these simple events have identical likelihood ratio for some Δ if and only if this Δ belongs to the roots. An algebraic equation of this form has a finite number of roots. Moreover, since the total number of simple events is finite, there are only finitely many values of Δ that are the root of an equation of this type. Let us refer to these values as to the special points in the interval $(1, +\infty)$.

By the hypothesis of the proposition, value $\Delta_1 = \frac{1+\delta_1^2}{1-\delta_1^2}$ is not the root of any equation defined by a pair of simple events (i.e., Δ_1 is not a special point). Therefore, there exists an interval that contains Δ_1 but none of the special points. The corresponding interval around δ_1 can be taken for the interval $\left(\underline{\delta}, \overline{\delta}\right)$ that we are looking for. □

By Proposition 4.10, for any nonzero $\delta_1 \in (-1, 1)$ there exists an interval $\left(\underline{\delta}, \overline{\delta}\right)$ containing δ_1, such that any randomized MP test determined by (4.10) for testing hypothesis H against simple alternative $K = \{\delta = \delta_1\}$ that has only a single point x with $P_1(x) = CP_0(x)$, is UMP for testing hypothesis H against composite alternative $K^* = \left\{\delta \in \left(\underline{\delta}, \overline{\delta}\right)\right\}$.

4.4 Multinomial Selection Procedures Built on Reduced Frequencies

In many practical situations the cryptanalyst faces the problem of comparing a number of alternatives with the intention to select the most probable among them. In particular, these may be key-stream blocks, ciphertext blocks or codewords. For some concrete instances of this general problem the number of possible alternatives may be relatively small but the opposite may also be true, i.e., an enormous amount of alternatives may be present. Among the latter we can name frequency analysis in dictionary attacks on block ciphers with a large block size and frequency analysis of codes with a large number of codewords (see [Bir99]). In these attacks the cryptanalyst collects many blocks (or codewords) and analyzes their frequencies. In all cases of this kind we are dealing with multinomial populations with an objective to select the most probable outcomes. These problems are commonly known in the selection and ranking theory in statistics as multinomial selection problems.

4.4.1 Multinomial Selection Problems

Let us now define the selection problem rigorously. Consider a multinomial distribution on N outcomes with probabilities $p_1, \dots, p_N$, i.e.,

$$\Pr(E_j) = p_j \quad (j = 1, \dots, N) \quad \text{and} \quad \sum_{j=1}^{N} p_j = 1 \ . \tag{4.25}$$

Let the ranked multinomial probabilities p_i be denoted as $p_{[1]} \le \cdots \le p_{[N]}$. It is assumed that the probabilities p_i are known but there is no prior knowledge regarding the correspondence between outcomes and probabilities.

The general case of the multinomial selection problem is formulated as developing a statistical procedure that observes a multinomial sample and selects a subset of the most probable outcomes. The selection is deemed *correct* if the subset contains at least a preassigned number c from t outcomes having the largest probabilities (ties broken arbitrarily). The size of the selected subset can either be specified in the selection procedure or be random and determined by the sample. On the other hand, we can distinguish between fixed sample-size and sequential procedures according to whether the sample size varies or not. In particular, the objective for selection procedures with the predetermined subset size equal to s and having $c = s = t$ is to select t of the most probable outcomes, and if $c = s = t = 1$ to select the most probable outcome (i.e., an outcome with the highest probability $p_{[N]}$).

The major characteristic of any selection procedure is the *Probability of Correct Selection* (PCS). This probability is not readily computable since it depends on the true configuration of the multinomial distribution which is unknown. For any selection procedure an important question is what should be the sample size for the PCS to be at least P^* $(0 < P^* < 1)$, where P^* is specified in advance by the experimenter (so called P^*-requirement). Therefore, for any sensible selection procedure PCS should be ultimately monotone and nondecreasing in n because only in this case the value of the smallest n that fulfils the P^*-requirement can be found (see [Hwa86]).

Surveys of the results dealing with multinomial selection problems are presented in [GP85, CH86] and [GP93b, Section 4]. To the references cited in these articles we can add paper [Ivc93, Subsection 5.6] where the inverse sampling procedure with fixed subset size (i.e., sampling is continued until the frequency of some s outcomes goes over a fixed threshold) is further evaluated and paper [Amb95] where consistency of the maximum likelihood method for establishing the complete correspondence between the outcomes and their probabilities, i.e., ranking the outcomes, is analyzed.

All previously suggested selection procedures require the estimation of the values of frequencies for all the outcomes. On the other hand, in the cryptographic applications mentioned earlier the total number of possible outcomes can be so large that it becomes infeasible to store all frequency counters in the computer's internal memory. We suggest new selection procedures that do not store frequencies for rare events and, due to that, these procedures can be implemented with reduced memory requirements. Moreover, by adjusting the value for the parameter of the procedures one can find a tradeoff between the amount of memory available and the required probability of correct selection.

4.4.2 Reduced Frequencies

Our procedures for selecting the most probable outcomes from a multinomial population are based on estimation of so called *reduced frequencies* and will be described and analyzed further is this section. Let $\xi_1, \ldots, \xi_n$ be a sequence of n independent observations of the multinomial random variable distributed according to (4.25). Let us also fix an arbitrary positive integer $T \leq n - 1$, which will be a parameter of the procedure and will be further referred to as the *thinning parameter.* When estimating reduced frequencies, the outcome $\xi_i = E_j$ $(i = i, \ldots, n; j = 1, \ldots, N)$ is accounted for in the reduced frequency of the outcome E_j if and only if the next T outcomes $\xi_{i+1}, \ldots, \xi_{i+T}$ contain E_j at least once (for those i with $i + T > n$ we consider the segment $\xi_{i+1}, \ldots, \xi_n$). The last occurrence in the sequence of any outcome is never accounted for since there are no the same outcomes that follow it. Otherwise, the outcome ξ_i does not change the values of reduced frequencies. Further, reduced frequencies are accumulated in the same manner as if we were dealing with the original frequencies, that will be called absolute frequencies from now on. Reduced frequencies can be used instead of absolute frequencies in various selection procedures (see [CH86] for examples of such procedures). Let ν_i denote the reduced frequency of outcome E_i $(i = 1, \ldots, N)$ and h_i denote its absolute frequency.

For instance, let $T = 3$, $N = 2$, $n = 10$, and the observed sequence is 0111001000. In this case, if $E_1 = 0$ and $E_2 = 1$ then $\nu_1 = 4$ and $\nu_2 = 3$ because the first and the last 0's are discarded as is the last 1. On the other hand the corresponding values of absolute frequencies are $h_1 = 6$ and $h_2 = 4$.

Note that always $\nu_i \leq h_i - 1$ $(i = 1, \ldots, N)$, that reduced frequencies cannot decrease if the value of T increases, and that for the maximal value of $T = n - 1$ reduced frequencies reach their maximum $h_i - 1$. The family of distributions $P_T(t_1, \ldots, t_N) = P_T(\nu_1 = t_1, \ldots, \nu_N = t_N)$ provides the complete probabilistic characterization for reduced frequencies. Unfortunately, we were not able to find the type of the distribution in this general form but succeeded in estimating the limit distribution for the random variable

$$\frac{\sqrt{n}}{n-T}\left(\nu_i - np_i\left(1 - (1-p_i)^T\right)\right) \quad (i = 1, \ldots, N) \quad \text{as} \quad n \to \infty \ .$$

Let us fix an arbitrary outcome E_i $(i = 1, \ldots, N)$ of multinomial distribution (4.25) and set $q = p_i$, $p = 1 - q = \sum_{j \neq i} p_j$. By associating the outcome E_i with the zero-event and the complement of E_i with the one-event, multinomial scheme (4.25) is reduced to the Bernoulli distribution on two outcomes with probabilities $\Pr(0) = q$ and $\Pr(1) = p$.

Consider the set of random variables

$$\Phi(\xi_i, \ldots, \xi_{i+T}) = \sum_{j=1}^{T} \mathrm{I}_{(01^{j-1}0)}(\xi_i, \ldots, \xi_{i+j}) \quad (i = 1, \ldots, n - T - 1) \ ,$$

where $\mathrm{I}_{(l_1,\ldots,l_t)}(\xi_i, \ldots, \xi_{i+t-1})$ denotes the indicator function of the event $\{\xi_i = l_1, \ldots, \xi_{i+t-1} = l_t\}$ and $01^{j-1}0$ denotes the vector of length $j + 1$ with the initial and the last coordinates equal to zero and the rest equal to one. When the summation index j in the above sum is equal to 1 the indicator function is assumed to be $\mathrm{I}_{(00)}(\xi_i, \xi_{i+1})$. Since events specified by the indicator functions in the definition of Φ are pairwise disjoint, random variables $\Phi(\xi_i, \ldots, \xi_{i+T})$ will only take on values 0 or 1 for $i = 1, \ldots, n - T - 1$. Let us also define

$$\Phi(\xi_{n-T}, \ldots, \xi_n) = \sum_{j=1}^{T} \sum_{i=0}^{T-j} \mathrm{I}_{(01^{j-1}0)}(\xi_{n-T+i}, \ldots, \xi_{n-T+i+j}) \ .$$

When the summation index j in the above sum is equal to 1 the indicator function is assumed to be $\mathrm{I}_{(00)}(\xi_{n-T+i}, \xi_{n-T+i+1})$. It is easy to see that $\sum_{i=1}^{n-T} \Phi(\xi_i, \ldots, \xi_{i+T}) = \nu_0$. Let us now estimate the mean of random variables $\Phi(\xi_i, \ldots, \xi_{i+T})$ $(i = 1, \ldots, n-T)$ and center them.

$$\begin{aligned}
\mathbf{E}\Phi(\xi_i, \ldots, \xi_{i+T}) &= \sum_{j=1}^{T} q^2 p^{j-1} = q(1-p^T) \quad (i = 1, \ldots, n-T-1) \ , \\
\mathbf{E}\Phi(\xi_{n-T}, \ldots, \xi_n) &= \sum_{j=1}^{T} \sum_{i=0}^{T-j} q^2 p^{j-1} = Tq - p + p^{T+1} \ .
\end{aligned}$$

Let $\tilde{\Phi}(\xi_i, \ldots, \xi_{i+T})$ denote the centered version of $\Phi(\xi_i, \ldots, \xi_{i+T})$ $(i = 1, \ldots, n-T)$, namely,

$$\begin{aligned}
\tilde{\Phi}(\xi_i, \ldots, \xi_{i+T}) &= \Phi(\xi_i, \ldots, \xi_{i+T}) - q(1-p^T) \quad (i = 1, \ldots, n-T-1) \ , \\
\tilde{\Phi}(\xi_{n-T}, \ldots, \xi_n) &= \Phi(\xi_{n-T}, \ldots, \xi_n) - Tq + p - p^{T+1} \ .
\end{aligned}$$

Finding the limit distribution for the following U-statistic constitutes the primary objective for the remaining part of this subsection:

$$U_n = \frac{1}{n-T} \sum_{i=1}^{n-T} \tilde{\Phi}(\xi_i, \ldots, \xi_{i+T}) \ . \tag{4.26}$$

This statistic will be analyzed using the general technique for U-statistics developed in [Ron79]. First, we need to introduce some notions and notation.

Following [Ron79], a system (X, B), where X is a finite set and $B = (B_1, \ldots, B_r)$ is a family of subsets of X such that $\bigcup_{i=1}^{r} B_i = X$, is called a *finite hypergraph.* The elements of X are called *vertices* and the subsets B_i *edges.* The fact that B is a family means that the same subset $B_i \subseteq X$ can appear in B several times. By a *chain* of length l connecting the vertices $x_{i_1}, x_{i_{l+1}} \in X$ we mean any sequence of the form

$$x_{i_1}, B_{i_1}, x_{i_2}, B_{i_2}, \ldots, B_{i_l}, x_{i_{l+1}} \ ,$$

where the vertices $x_{i_1}, \ldots, x_{i_l}$ are all distinct, among the edges $B_{i_1}, \ldots, B_{i_l}$ there are no repetitions (if these are considered as members of the family B), and $x_{i_k}, x_{i_{k+1}} \in B_{i_k}$ for each $k = 1, \ldots, l$. A hypergraph is said to be *connected* if each pair of its vertices is connected by some chain. Connectivity components are defined as usual.

Let m be an integer greater than 1. Further, let $I_0 = \emptyset$, $I_n = \{1, \ldots, n\}$ and for $n \geq m$ define set of m-tuples $J_n \subseteq \mathbf{Z}^m$ by

$$J_n = \left\{ \left(\varepsilon^{(1)}, \ldots, \varepsilon^{(m)} \right) \mid \varepsilon^{(\nu)} \in I_n \ (\nu = 1, \ldots, m); \ 1 \leq \varepsilon^{(1)} < \cdots < \varepsilon^{(m)} \leq n \right\} \ .$$

If $i_1, \ldots, i_k$ are arbitrary k elements in J_n and i_ν $(\nu = 1, \ldots, k)$ is the m-tuple $\left(\varepsilon_\nu^{(1)}, \ldots, \varepsilon_\nu^{(m)} \right)$ then also define

$$C\{i_1, \ldots, i_k\} = \bigcup_{\nu=1}^{k} \left\{ \varepsilon_\nu^{(1)}, \ldots, \varepsilon_\nu^{(m)} \right\} \ .$$

Let K_n be an arbitrary nonempty subset of J_n and let $M_{n,k}$ denote the set of all connected hypergraphs with k edges that lie in K_n. The concrete form of the U-statistic whose limit distribution is analyzed defines which elements of J_n are included into K_n.

We shall now analyze collections Θ_n $(n = m, m+1, \dots)$ consisting of random variables θ_i with $i \in K_n$ that fulfil the following conditions:

(a) if $i_1, \dots, i_k, j_1, \dots, j_t \in K_n$ for some $k, t > 0$ are such that $C\{i_1, \dots, i_k\} \cap C\{j_1, \dots, j_t\} = \emptyset$ then random variables $\{\theta_{i_1}, \dots, \theta_{i_k}\}$, $\{\theta_{j_1}, \dots, \theta_{j_t}\}$ are independent;

(b) for any integer $k \geq 1$ there exists some positive constant C_k such that

$$\sup_{n \geq m} \max_{i \in K_n} \mathbf{E}|\theta_i|^k < C_k < +\infty \ ;$$

(c) $\mathbf{E}\theta_i = 0$ for all $i \in K_n$.

Theorem 4.11 ([Ron79]) *Assume that collections Θ_n and sets K_n $(n = m, m+1, \dots)$ satisfy (a), (b), (c) and additionally fulfil the following conditions*

(i) $|M_{n,k}| = O\left(|K_n|^k n^{1-k}\right)$ $(k = 1, 2, \dots)$;

(ii) the limit of $n|K_n|^{-2} \sum_{(i_1,i_2) \in M_{n,2}} \mathbf{E}\theta_{i_1}\theta_{i_2}$ with $n \to \infty$, where the summation is carried out over all pairs of edges $i_1, i_2 \in K_n$ that make up a connected hypergraph, exists and is positive (let $\sigma^2 > 0$ be equal to the value of this limit).

Then random variable $\frac{\sqrt{n}}{|K_n|} \sum_{i \in K_n} \theta_i$ is asymptotically normal when $n \to \infty$ with mean equal to 0 and variance equal to σ^2.

Theorem 4.12 *Let E_i $(i \in \{1, \dots, n\})$ be an arbitrary outcome of multinomial distribution (4.25), $q = p_i$ and $p = 1 - q = \sum_{j \neq i} p_j$. Further, let ν_0 be the reduced frequency of outcome E_i after n observations estimated with thinning parameter T, and put $\mu_n = nq(1 - p^T)$. Then random variable*

$$\frac{\sqrt{n}}{n-T}(\nu_0 - \mu_n)$$

is asymptotically normal when $n \to \infty$ with mean equal to 0 and variance equal to

$$\sigma^2 = 2Tq^2(1-p^T)p^T + q(1-p^T)(1-q+qp^T) \ .$$

Proof: First, we shall use Theorem 4.11 to analyze the convergence of the distribution of U-statistic (4.26). To that end, we have to check if the U-statistic fulfils the conditions of this theorem. In our case $K_n = \{(1, \dots, T+1), (2, \dots, T+2), \dots, (n-T, \dots, n)\}$. For notational simplicity we will write $K_n = \{1, \dots, n-T\}$ so that the edges be ordered like integers. Then $|K_n| = n - T$, $m = T + 1$, and $K_n \subset J_n$. Define the random variable $\theta_i = \tilde{\Phi}(\xi_i, \dots, \xi_{i+T})$ for $i \in K_n$. We make the following observations.

(a) Let $i_1,\dots,i_k,j_1,\dots,j_t \in K_n$ for some positive integers k and t. Random variables $\{\theta_{i_1},\dots,\theta_{i_k}\}$ and $\{\theta_{j_1},\dots,\theta_{j_t}\}$ are functions of random variables ξ_i for $i \in C\{i_1,\dots,i_k\}$ and $i \in C\{j_1,\dots,j_t\}$ respectively. If $C\{i_1,\dots,i_k\} \cap C\{j_1,\dots,j_t\} = \emptyset$ then $\{\theta_{i_1},\dots,\theta_{i_k}\}$ and $\{\theta_{j_1},\dots,\theta_{j_t}\}$ are independent since $\xi_1,\dots,\xi_n$ is a sequence of independent observations.

(b) From the definition of $\Phi(\xi_i,\dots,\xi_{i+T})$ it follows that $\Phi(\xi_i,\dots,\xi_{i+T}) \in \{0,1\}$ for $(i = 1,\dots,n-T-1)$ and $\Phi(\xi_{n-T},\dots,\xi_n) \in \{0,\dots,T\}$. Therefore, $\Phi(\xi_i,\dots,\xi_{i+T}) \le T$ for all $i \in K_n$. Then $\mathbf{E}\Phi(\xi_i,\dots,\xi_{i+T})^j \le T^j$ for any integer $j \ge 0$ and all $i \in K_n$, and

$$\begin{aligned}
&\mathbf{E}\left|\tilde{\Phi}(\xi_i,\dots,\xi_{i+T})\right|^k = \mathbf{E}|\Phi(\xi_i,\dots,\xi_{i+T}) - \mathbf{E}\Phi(\xi_i,\dots,\xi_{i+T})|^k \le \\
&\le \mathbf{E}(\Phi(\xi_i,\dots,\xi_{i+T}) + \mathbf{E}\Phi(\xi_i,\dots,\xi_{i+T}))^k = \\
&= \sum_{j=0}^{k}\binom{k}{j}\mathbf{E}\Phi(\xi_i,\dots,\xi_{i+T})^j(\mathbf{E}\Phi(\xi_i,\dots,\xi_{i+T}))^{k-j} \le \\
&\le \sum_{j=0}^{k}\binom{k}{j}T^j(\mathbf{E}\Phi(\xi_i,\dots,\xi_{i+T}))^{k-j} = \\
&= \begin{cases} \left(T + q(1-p^T)\right)^k, & \text{for} \quad i = 1,\dots,n-T-1\ , \\ \left(T + Tq - p + p^{T+1}\right)^k, & \text{for} \quad i = n-T\ . \end{cases}
\end{aligned}$$

Thus, for any integer $k \ge 1$ there exists some positive constant C_k such that $\sup_{n\ge m}\max_{i\in K_n}\mathbf{E}\left|\tilde{\Phi}(\xi_i,\dots,\xi_{i+T})\right|^k < C_k < +\infty$.

(c) Random variables $\tilde{\Phi}(\xi_i,\dots,\xi_{i+T})$ are centered for $i \in K_n$.

This proves that conditions (a), (b) and (c) formulated above are fulfilled for the collection of random variables $\left\{\tilde{\Phi}(\xi_i,\dots,\xi_{i+T})\right\}$ with $i \in \{1,\dots,n-T\}$. Let us now check conditions (i) and (ii) of Theorem 4.11.

(i) Consider an arbitrary, connected hypergraph with k edges $\alpha_1,\dots,\alpha_k$ that belongs to $M_{n,k}$. We will assume that $\alpha_i \le \alpha_{i+1}$ for $i = 1,\dots,k-1$. Therefore, such a hypergraph is completely defined by the value of $\alpha_1 \in K_n$ and the set of difference values $\delta_i = \alpha_{i+1} - \alpha_i$ $(i = 1,\dots,k-1)$ such that identity $\delta_1 + \dots + \delta_{k-1} = \alpha_k - \alpha_1 = d$ holds. Thus, the number of hypergraphs in $M_{n,k}$ with $\alpha_k - \alpha_1 = d$ that contain some fixed edge $\alpha_1 \in \{1,\dots,n-T-d\}$ is equal to the number of compositions of d into $k-1$ parts with no part exceeding T, i.e., the number of partitions having the form of $\delta_1 + \dots + \delta_{k-1} = d$, where the order of the parts is taken into account and $0 \le \delta_i \le T$ (the upper bound T guarantees connectivity of the hypergraph). Let this number of compositions be denoted by $c(d,k-1,T)$. Then

$$|M_{n,k}| = \sum_{d=0}^{T(k-1)}(n-T-d)c(d,k-1,T) \quad \text{and}$$

$$\frac{|M_{n,k}|}{|K_n|^k n^{1-k}} = \frac{|M_{n,k}|}{(n-T)^k n^{1-k}} \sim \frac{|M_{n,k}|}{n} \overset{n\to\infty}{\longrightarrow} \sum_{d=0}^{T(k-1)} c(d, k-1, T) > 0 \ ,$$

which proves that condition (i) is fulfilled.

(ii) We shall now estimate $\lim_{n\to\infty} n|K_n|^{-2} \sum_{(i_1,i_2)\in M_{n,2}} \mathbf{E}\theta_{i_1}\theta_{i_2}$. Note that for any pair of edges $(i_1, i_2) \in M_{n,2}$, i.e., edges that make up a connected hypergraph, one has $C\{i_1\} \cap C\{i_2\} \neq \emptyset$.

Denote vector $(\xi_i, \ldots, \xi_{i+T})$ by $\boldsymbol{\xi}_i$. Then, considering the internal structure of set K_n, the limit that we want to find can be rewritten in the following way:

$$\lim_{n\to\infty} \frac{n}{(n-T)^2} \left(\sum_{\substack{i_1,i_2 \in \overline{1,n-T-1} \\ C\{i_1\}\cap C\{i_2\}\neq\emptyset}} \mathbf{E}\tilde{\Phi}(\boldsymbol{\xi}_{i_1})\tilde{\Phi}(\boldsymbol{\xi}_{i_2}) + \sum_{i=n-2T}^{n-T} \mathbf{E}\tilde{\Phi}(\boldsymbol{\xi}_{n-T})\tilde{\Phi}(\boldsymbol{\xi}_i) \right) =$$

$$= \lim_{n\to\infty} \frac{n}{(n-T)^2} \sum_{\substack{i_1,i_2 \in \overline{1,n-T-1} \\ C\{i_1\}\cap C\{i_2\}\neq\emptyset}} \mathbf{E}\tilde{\Phi}(\boldsymbol{\xi}_{i_1})\tilde{\Phi}(\boldsymbol{\xi}_{i_2}) \ .$$

Note that for any $i, j \in \{1, \ldots, n-T-1\}$

$$\begin{aligned} \mathbf{E}\tilde{\Phi}(\boldsymbol{\xi}_i)\tilde{\Phi}(\boldsymbol{\xi}_j) &= \mathbf{E}\left(\Phi(\boldsymbol{\xi}_i) - q(1-p^T)\right)\left(\Phi(\boldsymbol{\xi}_j) - q(1-p^T)\right) = \\ &= \mathbf{E}\Phi(\boldsymbol{\xi}_i)\Phi(\boldsymbol{\xi}_j) - q^2(1-p^T)^2 \ . \end{aligned} \tag{4.27}$$

We shall now estimate mean $\mathbf{E}\Phi(\boldsymbol{\xi}_i)\Phi(\boldsymbol{\xi}_j)$ for $i, j \in \{1, \ldots, n-T-1\}$ such that $C\{i\} \cap C\{j\} \neq \emptyset$. Note that vectors $\boldsymbol{\xi}_i$ and $\boldsymbol{\xi}_j$ can overlap on k coordinates for $k \in \{1, \ldots, T+1\}$.

- If $k \in \{1, \ldots, T\}$ then $\mathbf{E}\Phi(\boldsymbol{\xi}_i)\Phi(\boldsymbol{\xi}_j)$ is equal to the total probability of vectors having the following pattern

  $$(0, i_2, \ldots, i_{T-k+1}, 0, 1, \ldots, 1, 0, i_{T-k+t+4}, \ldots, i_{2T-k+2})$$

  for any $t = 0, \ldots, T-1$ and where coordinates i_j can be arbitrary. Therefore, $\mathbf{E}\Phi(\boldsymbol{\xi}_i)\Phi(\boldsymbol{\xi}_j) = \sum_{t=0}^{T-1} q^3 p^t = q^2(1-p^T)$ and, by (4.27),

  $$\mathbf{E}\tilde{\Phi}(\boldsymbol{\xi}_i)\tilde{\Phi}(\boldsymbol{\xi}_j) = q^2(1-p^T)p^T \ .$$

- If $k = T+1$ then $\mathbf{E}\Phi(\boldsymbol{\xi}_i)\Phi(\boldsymbol{\xi}_j) = \mathbf{E}\Phi(\boldsymbol{\xi}_i) = q(1-p^T)$ and, by (4.27),

  $$\mathbf{E}\tilde{\Phi}(\boldsymbol{\xi}_i)\tilde{\Phi}(\boldsymbol{\xi}_j) = q(1-p^T)(1-q+qp^T) \ .$$

Therefore,

$$\frac{n}{(n-T)^2} \sum_{\substack{i_1,i_2 \in \overline{1,n-T-1} \\ C\{i_1\} \cap C\{i_2\} \neq \emptyset}} \mathbf{E}\tilde{\Phi}(\boldsymbol{\xi}_{i_1})\tilde{\Phi}(\boldsymbol{\xi}_{i_2}) = \frac{n}{(n-T)^2}\Bigg(2q^2(1-p^T)p^T \times$$
$$\times \sum_{k=1}^{T}(n-2T+k-1) + (n-T-1)q(1-p^T)(1-q+qp^T)\Bigg) \to$$
$$\to 2Tq^2(1-p^T)p^T + q(1-p^T)(1-q+qp^T) = \sigma^2 > 0 \quad \text{as} \quad n \to \infty \ .$$

Thus, all conditions of Theorem 4.11 are fulfilled and, therefore, $\sqrt{n}U_n \to N(0,\sigma^2)$ as $n \to \infty$. We conclude the proof by noting that

$$\begin{aligned} U_n &= \frac{1}{n-T}\sum_{i=1}^{n-T}\tilde{\Phi}(\xi_i,\ldots,\xi_{i+T}) = \\ &= \frac{1}{n-T}\left(\sum_{i=1}^{n-T}\Phi(\xi_i,\ldots,\xi_{i+T}) - (n-T-1)q(1-p^T) - Tq + p - p^{T+1}\right) = \\ &= \frac{1}{n-T}(\nu_0 - \mu_n + (T+1)q(1-p^T) - Tq + p - p^{T+1}) \end{aligned}$$

and

$$\frac{\sqrt{n}}{n-T}\left((T+1)q(1-p^T) - Tq + p - p^{T+1}\right) \to 0 \quad \text{as} \quad n \to \infty \ ,$$

that gives us the claimed result. □

Various procedures for selecting the most probable outcomes, instead of absolute frequencies, can use reduced frequencies. Obviously, the PCS of such procedures is defined by the probability distribution of reduced frequencies. Therefore, the limit distribution that we found in this section can be used for estimating the efficiency of the procedures. Selection procedures built on reduced frequencies are useful when running statistical attacks on block and stream ciphers. They help to detect possible statistical irregularities in a generated sequence of key-stream or ciphertext blocks. If such an irregularity is detected then this can be used as a basis for the distinguishing attack on the cipher that is considered as a black box. It is also reasonable to assume that any deviation in the distribution from the uniformity can be used to run a key-recovery attack but concrete implementation of the attack essentially depends on the system that comes under the cryptanalysis.

Chapter 5

Conclusion

Cryptographic algorithms provide mathematical tools for achieving several important security objectives in electronic communications. Among the most important of these objectives is ensuring privacy of the message exchange. Privacy can be achieved by using both symmetric and asymmetric techniques. Though, in the areas where the highest level of security, high throughput capacity and low implementation costs are of the utmost importance, there is no alternative to symmetric systems and to stream ciphers, in particular. Stream ciphers withstood the test of time that allowed to develop the comprehensive theory of the design and analysis of these cipher systems. In the thesis we made our own contribution to the theory of synchronous stream ciphers.

The principle part of any synchronous stream cipher is a key-stream generator. In order to satisfy the basic security requirements of long period, large linear complexity and uniform distribution properties for a key stream, the vast majority of key-stream generators are built following few basic design principles. The long period and uniformity are achieved due to the use of linear feedback shift registers (LFSR's). Nonlinearity is introduced either explicitly by applying nonlinear functions (e.g., combination and filter generators) or implicitly by controlling the clock of an LFSR. In the thesis we contribute both to the study of nonlinear cryptographic functions and to the theory of clock-controlled LFSR's.

The first part of the research was inspired by the fact that several well-known transforms (namely, Walsh, algebraic normal and arithmetic) that proved to be extremely helpful when analyzing cryptographic properties of Boolean functions, are all based on the Kronecker power of some relevant elementary cells. This allows to use fast transform algorithms for efficient estimation of the important representations of Boolean functions and easy transition from one representation to another. Our objective was to find some other useful Kronecker-type transforms. As a result, we were able to build a general approach to the transforms of the said type that allowed to define two new, probabilistic and weight, transforms.

In particular, the probabilistic transform provides an efficient way for estimating the bias for the distribution of the value of a Boolean function if the biases of the arguments are known. The newly introduced characteristic allows to compare Boolean functions against their ability for compensating a biased distribution of the input bits. It turned out that highly resilient Boolean functions significantly increase the order of magnitude for the bias of the distribution of the output bits compared to the bias of the inputs. The weight transform relates a Boolean function to the weights of its subfunctions which is helpful for estimating correlation dependencies of the function. The general approach to Kronecker-type transforms turned out also to be particularly helpful in proving relations between different transforms.

Our contribution to the theory of clock-controlled generators consists in estimating the period of the output sequence generated by a clock-controlled LFSR with an irreducible feedback polynomial and an arbitrary structure of the control sequence. These results allowed to enlarge the class of generators qualified for secure applications. We also described some specific configurations of the clock-controlled arrangement producing the output sequence with maximal period, close-to-uniform element distribution, and the two-valued autocorrelation function.

Further, we analyzed the generalized Geffe generator. Unlike the original generator by Geffe, that has three binary input m-sequences, the generalized generator runs over a finite field and combines multiple inputs having arbitrary periods. In particular, this implies that clock-controlled shift registers can be used as inputs. The original Geffe generator cannot be used for secure key-stream generation since its combining function is zero-order correlation immune and correlation attacks can easily be launched. Using clock-controlled registers and multiple inputs makes the new generator immune against fast correlation attacks and less susceptible to basic attacks. We proved some relevant algebraic properties of this generator.

In the last part of the thesis we developed cryptanalytic attacks that exploit statistical irregularities in a key stream. The primary purpose of these attacks is to build a distinguisher that would be able to tell the difference between the black box containing the cipher and the one producing a purely random output. The first distinguishing attack suggested here is statistically optimal (although, not efficient) for testing a key stream for a linear recurrence perturbed with a nonuniform additive noise. We also constructed a couple of not optimal, but computationally feasible tests. Other distinguishing attacks can be built using new procedures for selecting the most probable outcomes which use reduced frequencies. These procedures can handle large key-stream or ciphertext blocks. The whole investigation brings us to the conclusion that ensuring statistical uniformity of a key stream is not the less important task than taking care of good algebraic characteristics. Finally, we suggested an algorithm for testing a ciphertext for key-stream reuse. The constructed nonrandomized and randomized most powerful tests can efficiently distinguish families consisting of up to four ciphertexts obtained from different plaintexts but using the same key-stream segment.

Bibliography

[AHU74] Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. *The Design and Analysis of Computer Algorithms.* Addison-Wesley Series in Computer Science and Information Processing. Addison-Wesley, Amsterdam, 1974.

[Amb95] A.S. Ambrosimov. On consistency of the maximum likelihood method in ranking the outcomes of the polynomial scheme according to their probabilities. *Discrete Mathematics and Applications*, 5(3):257–268, 1995.

[Bea84] K.G. Beauchamp. *Applications of Walsh and Related Functions (with an Introduction to Sequency Theory).* Microelectronics and Signal Processing. Academic Press, London, 1984.

[Bir99] Alex Biryukov. *Methods of Cryptanalysis.* PhD thesis, Technion – Israel Institute of Technology, 1999.

[BP81] G.R. Blakley and G.B. Purdy. A necessary and sufficient condition for fundamental periods of cascade machines to be product of the fundamental periods of their constituent finite state machines. *Information Sciences: An International Journal*, 24(1):71–91, June 1981.

[BP85] Thomas Beth and Fred C. Piper. The stop-and-go generator. In Thomas Beth, Norbert Cot, and Ingemar Ingemarsson, editors, *Advances in Cryptology: Proceedings of EuroCrypt '84*, volume 209 of *Lecture Notes in Computer Science*, pages 88–92, Berlin, 1985. Springer-Verlag.

[BRS98] B. Borchert, D. Ranjan, and F. Stephan. On the computational complexity of some classical equivalence relations on Boolean functions. *Theory of Computing Systems*, 31(6):679–693, November–December 1998.

[Can02] Anne Canteaut. On the correlation between a combining function and functions of fewer variables. In *Proceedings of the 2002 IEEE Information Theory Workshop*, pages 78–81. IEEE, 2002.

[CC99] Paul Camion and Anne Canteaut. Correlation-immune and resilient functions over a finite alphabet and their applications in cryptography. *Designs, Codes and Cryptography*, 16(2):121–149, February 1999.

[CF01] Anne Canteaut and Eric Filiol. Ciphertext only reconstruction of stream ciphers based on combination generators. In Bruce Schneier, editor, *Fast Software Encryption*, volume 1978 of *Lecture Notes in Computer Science*, pages 165–180, Berlin, 2001. Springer-Verlag.

[CG99] Claude Carlet and Philippe Guillot. A new representation of Boolean functions. In Marc Fossorier, Hideki Imai, Shu Lin, and Alain Poli, editors, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, volume 1719 of *Lecture Notes in Computer Science*, pages 94–103, Berlin, 1999. Springer-Verlag.

[CG00] William G. Chambers and Dieter Gollmann. Embedding attacks on step[1..D] clock-controlled generators. *Electronic Letters*, 36(21):1771–1773, October 2000.

[CH86] Robert W. Chen and Frank K. Hwang. Least-favorable configurations in the multinomial selection problem: a survey. *American Journal of Mathematical and Management Sciences*, 6(1–2):13–25, 1986.

[Cha88] William G. Chambers. Clock-controlled shift registers in binary sequence generators. *IEE Proceedings - Computers and Digital Techniques*, 135(1):17–24, January 1988.

[CKM94] Don Coppersmith, Hugo Krawczyk, and Yishay Mansour. The shrinking generator. In Douglas R. Stinson, editor, *Advances in Cryptology - Crypto '93*, volume 773 of *Lecture Notes in Computer Science*, pages 22–39, Berlin, 1994. Springer-Verlag.

[CS02] Claude Carlet and Palash Sarkar. Spectral domain analysis of correlation immune and resilient Boolean functions. *Finite Fields and Their Applications*, 8(1):120–130, January 2002.

[DXS91] Cunsheng Ding, Guo-Zhen Xiao, and Weijuan Shan. *The Stability Theory of Stream Ciphers*, volume 561 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1991.

[Fil00] Eric Filiol. Decimation attack of stream ciphers. In Bimal K. Roy and Eiji Okamoto, editors, *Progress in Cryptology - INDOCRYPT 2000*, volume 1977 of *Lecture Notes in Computer Science*, pages 31–42, Berlin, 2000. Springer-Verlag.

[FIP81] National Institute of Standards and Technology. DES modes of operation. FIPS PUB 81, 1980 December 2.

[For89] Réjane Forré. The strict avalanche criterion: Spectral properties of Boolean functions and an extended definition. In Shafi Goldwasser, editor, *Advances in Cryptology - Crypto '88*, volume 403 of *Lecture Notes in Computer Science*, pages 450–468, Berlin, 1989. Springer-Verlag.

[GC89] Dieter Gollmann and William G. Chambers. Clock-controlled shift registers: a review. *IEEE Journal on Selected Areas in Communications*, 7(4):525–533, May 1989.

[Gef73] Philip R. Geffe. How to protect data with ciphers that are really hard to break. *Electronics*, 46(1):99–101, January 1973.

[GM91] Jovan Dj. Golić and Miodrag J. Mihaljević. A generalized correlation attack on a class of stream ciphers based on the Levenshtein distance. *Journal of Cryptology*, 3(3):201–212, 1991.

[GM01] Kishan Chand Gupta and Subhamoy Maitra. Primitive polynomials over GF(2) - a cryptologic approach. In Sihan Qing, Tatsuaki Okamoto, and Jianying Zhou, editors, *Information and Communications Security*, volume 2229 of *Lecture Notes in Computer Science*, pages 23–34, Berlin, 2001. Springer-Verlag.

[GO95] Jovan Dj. Golić and Luke O'Connor. Embedding and probabilistic correlation attacks on clock-controlled shift registers. In Alfredo De Santis, editor, *Advances in Cryptology - EuroCrypt '94*, volume 950 of *Lecture Notes in Computer Science*, pages 230–243, Berlin, 1995. Springer-Verlag.

[GO96] Jovan Dj. Golić and Luke O'Connor. A cryptanalysis of clock-controlled shift registers with multiple steps. In Ed Dawson and Jovan Dj. Golić, editors, *Cryptography: Policy and Algorithms*, volume 1029 of *Lecture Notes in Computer Science*, pages 174–185, Berlin, 1996. Springer-Verlag.

[Gol67] Solomon W. Golomb. *Shift Register Sequences.* Holden-Day series in information systems. Holden-Day, Inc., San Francisco, 1967. Revised ed., Laguna Hills: Aegean Park, 1982.

[Gol89] Jovan Dj. Golić. On the linear complexity of functions of periodic GF(q) sequences. *IEEE Transactions on Information Theory*, 35(1):69–75, January 1989.

[Gol95a] Jovan Dj. Golić. On decimation of linear recurring sequences. *Fibonacci Quarterly*, 33(5):407–411, November 1995.

[Gol95b] Jovan Dj. Golić. Towards fast correlation attacks on irregularly clocked shift registers. In Louis C. Guillou and Jean-Jacques Quisquater, editors, *Advances in Cryptology - EuroCrypt '95*, volume 921 of *Lecture Notes in Computer Science*, pages 248–262, Berlin, 1995. Springer-Verlag.

[Gol96] Jovan Dj. Golić. Constrained embedding probability for two binary strings. *SIAM Journal on Discrete Mathematics*, 9(3):360–364, August 1996.

[Gol98] Jovan Dj. Golić. Periods of interleaved and nonuniformly decimated sequences. *IEEE Transactions on Information Theory*, 44(3):1257–1260, May 1998.

[Gol01] Jovan Dj. Golić. Edit distances and probabilities for correlation attacks on clock-controlled combiners with memory. *IEEE Transactions on Information Theory*, 47(3):1032–1041, March 2001.

[GP85] Shanti S. Gupta and S. Panchapakesan. Subset selection procedures: Review and assessment. *American Journal of Mathematical and Management Sciences*, 5:235–311, 1985.

[GP93a] Jovan Dj. Golić and Slobodan V. Petrović. A generalized correlation attack with a probabilistic constrained edit distance. In Rainer A. Rueppel, editor, *Advances in Cryptology - EuroCrypt '92*, volume 658 of *Lecture Notes in Computer Science*, pages 472–476, Berlin, 1993. Springer-Verlag.

[GP93b] Shanti S. Gupta and S. Panchapakesan. Selection and screening procedures in multivariate analysis. In C.R. Rao, editor, *Multivariate Analysis: Future Directions*, volume 5 of *North-Holland Series in Statistics and Probability*, chapter 12, pages 233–262. North-Holland, Amsterdam, 1993.

[GŽ88] Jovan Dj. Golić and Miodrag V. Živković. On the linear complexity of nonuniformly decimated PN-sequences. *IEEE Transactions on Information Theory*, 34(5):1077–1079, September 1988.

[Har63] Michael A. Harrison. The number of transitivity sets of Boolean functions. *Journal of the Society for Industrial and Applied Mathematics*, 11(3):806–828, September 1963.

[Har64] Michael A. Harrison. On the classification of Boolean functions by the general linear and affine groups. *Journal of the Society for Industrial and Applied Mathematics*, 12(2):285–298, June 1964.

[HK98] Tor Helleseth and P. Vijay Kumar. Sequences with low correlation. In V.S. Pless and W.C. Huffman, editors, *Handbook of Coding Theory*, volume 2, chapter 21. Elsevier, Amsterdam, 1998.

[HK99] Tor Helleseth and P. Vijay Kumar. Pseudonoise sequences. In Jerry D. Gibson, editor, *The Mobile Communications Handbook*, The Electrical Engineering Handbook Series, chapter 8. CRC Press, London, 2nd edition, 1999.

[Hwa86] Frank K. Hwang. Fixed sample-size multinomial selection in scalar zones. *American Journal of Mathematical and Management Sciences*, 6(1–2):27–40, 1986.

[Ivc93] G.I. Ivchenko. The waiting time and related statistics in the multinomial scheme: a survey. *Discrete Mathematics and Applications*, 3(5):451–482, 1993.

[Jö02] Fredrik Jönsson. *Some Results on Fast Correlation Attacks*. PhD thesis, Lund University, 2002.

[Jan89] Cees J.A. Jansen. *Investigations on Nonlinear Streamcipher Systems: Construction and Evaluation Methods*. PhD thesis, Technical University of Delft, 1989.

[Joh98] Thomas Johansson. Reduced complexity correlation attacks on two clock-controlled generators. In Kazuo Ohta and Dingyi Pei, editors, *Advances in Cryptology - AsiaCrypt '98*, volume 1514 of *Lecture Notes in Computer Science*, pages 342–356, Berlin, 1998. Springer-Verlag.

[KB81] Sarangan Krishna Kumar and Melvin A. Breuer. Probabilistic aspects of boolean switching functions via a new transform. *Journal of the Association for Computing Machinery*, 28(3):502–520, July 1981.

[Kho98a] Alexander Kholosha. On a method of distinguishing the most probable outcomes of the polynomial scheme. *Engineering Simulation*, 15(6):791–797, 1998.

[Kho98b] Alexander Kholosha. Some problems of generating pseudo-random sequences using finite state automata. In Yu.M. Korostil, editor, *Proceedings of the Institute of Modelling Problems in Power Engineering of the National Academy of Sciences of Ukraine, Issue 1*, pages 74–90. Svit, Lviv, 1998. (in Russian).

[Kho01] Alexander Kholosha. Clock-controlled shift registers and generalized Geffe key-stream generator. In C. Pandu Rangan and Cunsheng Ding, editors, *Progress in Cryptology - INDOCRYPT 2001*, volume 2247 of *Lecture Notes in Computer Science*, pages 287–296, Berlin, 2001. Springer-Verlag.

[Kho03] Alexander Kholosha. Tensor transform of functions over finite fields. In L.M.G.M. Tolhuizen, editor, *24th Symposium on Information Theory in the Benelux*, pages 179–186, Enschede, 2003. Werkgemeenschap voor Informatie- en Communicatietheorie.

[KN63] B.M. Kloss and E.I. Nechiporuk. On classification of functions of multiple-valued logic. In *Problemy Kibernetiki, Issue 9*, pages 27–36. Fizmatgiz, Moscow, 1963. (in Russian).

[KvT02] Alexander Kholosha and Henk C.A. van Tilborg. Tensor transform of Boolean functions and related algebraic and probabilistic properties. In

Robert H. Deng, Sihan Qing, Feng Bao, and Jianying Zhou, editors, *Information and Communications Security*, volume 2513 of *Lecture Notes in Computer Science*, pages 434–446, Berlin, 2002. Springer-Verlag.

[Leh97] Erich L. Lehmann. *Testing Statistical Hypotheses.* Springer Texts in Statistics. Springer-Verlag, Berlin, 2nd edition, 1997.

[LN83] Rudolf Lidl and Harald Niederreiter. *Finite Fields*, volume 20 of *Encyclopedia of Mathematics and its Applications.* Addison-Wesley, Amsterdam, 1983.

[LN86] Rudolf Lidl and Harald Niederreiter. *Introduction to Finite Fields and Their Applications.* Cambridge University Press, Cambridge, 1986.

[Mas69] James L. Massey. Shift-register synthesis and BCH decoding. *IEEE Transactions on Information Theory*, IT-15(1):122–127, January 1969.

[Mih93] Miodrag J. Mihaljević. An approach to the initial state reconstruction of a clock-controlled shift register based on a novel distance measure. In J. Seberry and Y. Zheng, editors, *Advances in Cryptology - AusCrypt '92*, volume 718 of *Lecture Notes in Computer Science*, pages 349–356, Berlin, 1993. Springer-Verlag.

[Mir02] Kanstantsin Miranovich. Spectral analysis of Boolean functions under non-uniformity of arguments. Cryptology ePrint Archive, Report 2002/021, 2002. http://eprint.iacr.org/2002/021/.

[MS89] Willi Meier and Othmar Staffelbach. Nonlinearity criteria for cryptographic functions. In Jean-Jacques Quisquater and Joos Vandewalle, editors, *Advances in Cryptology - EuroCrypt '89*, volume 434 of *Lecture Notes in Computer Science*, pages 549–562, Berlin, 1989. Springer-Verlag.

[MS96] Florence J. MacWilliams and Neil James A. Sloane. *The Theory of Error-Correcting Codes*, volume 16 of *North-Holland Mathematical Library.* North-Holland, Amsterdam, 1996. Ninth impression.

[MvOV97] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography.* The CRC Press Series on Discrete Mathematics and its Applications. CRC Press, London, 1997.

[NES03] NESSIE. New European Schemes for Signatures, Integrity, and Encryption, 2000–2003. https://www.cosic.esat.kuleuven.ac.be/nessie/.

[Ron79] A.F. Ronzhin. On limit theorems for sums of dependent random variables. *Theory of Probability and its Applications*, 24(3):549–559, 1979.

[Roy02] Bimal K. Roy. Summarising recent results on finding multiples of primitive polynomials over GF(2). In *Proceedings of the 2002 IEEE Information Theory Workshop*, pages 82–85. IEEE, 2002.

[Rue86] Rainer A. Rueppel. *Analysis and Design of Stream Ciphers.* Communications and Control Engineering Series. Springer-Verlag, Berlin, 1986.

[Rue92] Rainer A. Rueppel. Stream ciphers. In Gustavus J. Simmons, editor, *Contemporary Cryptology: the Science of Information Integrity*, chapter 2, pages 65–134. IEEE Press, New York, 1992.

[Sar00] Palash Sarkar. Spectral domain analysis of correlation immune and resilient Boolean functions. Cryptology ePrint Archive, Report 2000/049, 2000. http://eprint.iacr.org/2000/049/. Revised version published in [CS02].

[Sch96] Bruce Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C.* John Wiley & Sons, Inc., New York, 2nd edition, 1996.

[Sha49] Claude E. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28(4):656–715, October 1949.

[Sie84] T. Siegenthaler. Correlation-immunity of nonlinear combining functions for cryptographic applications. *IEEE Transactions on Information Theory*, IT-30(5):776–780, January 1984.

[Sie85] T. Siegenthaler. Decrypting a class of stream ciphers using ciphertext only. *IEEE Transactions on Computers*, C-34(1):81–85, January 1985.

[Sin99] S. Singh. *The Code Book: the Evolution of Secrecy from Mary, Queen of Scots, to Quantum Cryptography.* Doubleday, New York, 1999.

[vT00] Henk C.A. van Tilborg. *Fundamentals of Cryptology: a Professional Reference and Interactive Tutorial.* Kluwer Academic Publishers, Dordrecht, 2000.

[XM88] Guo-Zhen Xiao and James L. Massey. A spectral characterization of correlation-immune combining functions. *IEEE Transactions on Information Theory*, 34(3):569–571, May 1988.

[ZCG99] Jinjun Zhou, Weihong Chen, and Fengxiu Gao. Best linear approximation and correlation immunity of functions over $\mathbf{Z}_m^*$. *IEEE Transactions on Information Theory*, 45(1):303–308, January 1999.

[Zie59] Neal Zierler. Linear recurring sequences. *Journal of the Society for Industrial and Applied Mathematics*, 7(1):31–48, March 1959.

Index

Samenvatting

In deze thesis behandelen we een aantal belangrijke problemen bij het ontwerpen en analyseren van sleutelstroom generatoren voor stroomvercijfering. Historisch gezien valt al het onderzoek op dit gebied uiteen in twee richtingen.

De eerste richting is gericht op building blocks (bijv. feedback schuifregisters, logische functies, rekenen modulo N, etc.) die de generator vormen, en op het schatten van de gerelateerde getaltheoretische karakteristieken van de sleutelstroom. Bij deze schatting is het belangrijkste doel het vinden van methoden om sleutelstroomrijen te genereren met karakteristieken die beveiliging tegen algebraïsche aanvallen bieden. Deze aanpak volgend, analyseren we combinerende functies en filter functies in Hoofdstuk 2 en clock-controlled LFSR's in Hoofdstuk 3.

Een tensor transformatie zoals geïntroduceerd in Hoofdstuk 2 blijkt handig te zijn bij het analyseren van de veiligheid van boolean en meerwaardige logische functies in cryptografische toepassingen. Speciale gevallen van deze aanpak bieden niet alleen makkelijke bewijzen voor bekende relaties in de theorie van algebraïsche, normaal-, en Walsh transformaties, maar leiden ook naar een aantal nieuwe eigenschappen van deze transformaties. We doen ook een voorstel betreffende een nieuw type tensor transformatie, de zogenaamde gewicht transformatie. Deze relateert een boolean functie aan de gewichten van zijn subfuncties.

In Hoofdstuk 2 bestuderen we ook de correlatie eigenschappen van boolean functies. We laten zien hoe de correlatie coëfficiënten, die een schatting bieden voor de correlatie afhankelijkheden van een boolean functie, verkregen kunnen worden van de gewicht transformatie. We laten zien dat het aantal producttermen van vaste orde in de algebraïsche normaalvorm van een gebalanceerde boolean functie (met evenveel nullen als enen) afhangt van zijn correlatie coëfficiënten. We bewijzen dat sterk robuuste boolean functies niet benaderd kunnen worden door een functie die niet gedegenereerd is op een paar variabelen. Ook introduceren we in dit hoofdstuk een polynoom dat de bias van de output distributie als een functie van de input biases schat. De coëfficiënten van dit polynoom kunnen verkregen worden door middel van de probabilistische transformatie. Verder doen we een suggestie voor een karakteristiek voor gebalanceerde boolean functies dat hun vermogen om een niet uniforme distributie van de inputs te compenseren meet. Van robuuste functies is bewezen dat ze goede compenserende kwaliteiten hebben.

Het andere building block dat geanalyseerd wordt in de context van de eerste richting is een clock-controlled LFSR. In Hoofdstuk 3 schatten we de periode van

zijn output rij wanneer het feedback polynoom irreducibel is en de structuur van de control sequence willekeurig is. Een voldoende voorwaarde voor het maximaal zijn van deze periode wordt geformuleerd. Een aantal specifieke configuraties van clock-controlled arrangementen met maximale output rij periode worden beschreven. Relevante aanbevelingen voor het schatten van de lineaire complexiteit worden gegeven. Ook formuleren we de regels die in acht genomen moeten worden bij het construeren van een clock-controlled arrangement om een bijna- uniforme distributie van elementen in de output rij te geven.

Ons enigszins verplaatsend van de building blocks, construeren we in Hoofdstuk 3 ook een sleutelstroom generator gebaseerd op degene voorgesteld door Geffe. Anders dan de Geffe generator, die drie binaire input m-rijen heeft, werkt deze generator over het lichaam GF(q) en combineert het meerdere inputs van willekeurige perioden. In het bijzonder impliceert dit dat clock-controlled schuifregisters gebruikt kunnen worden als inputs. De originele Geffe generator kan niet gebruikt worden voor sleutelstroom generatie omdat zijn combinerende functie nulde orde correlatie immuun is en correlatie aanvallen gemakkelijk te lanceren zijn. Het gebruik van clock-controlled registers en meerdere inputs maakt de nieuwe generator immuun voor snelle correlatie aanvallen en minder gevoelig voor primaire aanvallen. We analyseren een aantal relevante algebraïsche eigenschappen van de voorgestelde generator.

De tweede richting in het ontwerpen en analyseren van sleutelstroom generatoren waar we ons op concentreren zijn de statistische eigenschappen van een sleutelstroom. Deze aanpak volgend, ontwikkelen we in Hoofdstuk 4 verscheidene aanvallen die profiteren van statistische zwakheid in de sleutelstroom. Onze eerste algoritme gebruikt statistische tests gebaseerd op invariante statistieken. Het test een sleutelstroom op lineaire recurrentie verstoord door niet uniforme additieve ruis. Voor het geval van trinomiale feedback construeren we een aantal invariante statistieken die de constructie van computationeel uitvoerbare tests mogelijk maken.

Onze tweede algoritme test een vercijferde tekst op hergebruik van sleutelstroom. We construeren de sterkste niet gerandomizeerde en gerandomizeerde tests die efficient onderscheid kunnen maken tussen families, bestaande uit maximaal vier vercijferde teksten, verkregen van verschillende originele teksten, maar gebruik makend van hetzelfde sleutelstroom segment. Bovendien geven we expliciete algoritmen voor het construeren van parameter intervallen, waar deze tests uniform het meest krachtig zijn.

Als de cryptoanalist te maken heeft met blokken sleutelstroom of cijferschrift kan het nuttig zijn statistische procedures te gebruiken voor het selecteren van de meest waarschijnlijke uitkomsten van de multinomiale populatie. We construeren nieuwe procedures die gebaseerd zijn op de berekening van gereduceerde frequenties. Dat maakt ze meer efficient als het totale aantal uitkomsten groot is vergeleken met de hoeveelheid geheugen dat beschikbaar is. Nuttige toepassingen kunnen gevonden worden in frequentie analyse, bijvoorbeeld waar het een deel is van een woordenboek aanval op cijfers en verscheidene andere aanvallen op codes. We bewijzen de limietstelling voor de distributie van gereduceerde frequenties.

Acknowledgements

I have spent the last three wonderful years doing research with the Cryptography and Coding group at the Technische Universiteit Eindhoven. During this relatively short period of time I had the greatest pleasure to work with many scientists who contributed a lot to my professional development.

First of all I would like to express my gratitude to Henk van Tilborg who made it possible for me to come to the Netherlands, work in his group, and who agreed to supervise my research. I am grateful for the freedom he gave me in choosing my favorite topics in the vast field of cryptography. I would like to thank my second promoter, Tor Helleseth, for agreeing to take this responsibility.

Thanks are also due to the members of my reading committee Arjen Lenstra and Joos Vandewalle for valuable comments that I have received. My biggest appreciation to Arjen Lenstra for his patience when reading the thesis that resulted in the long list of comments. Some of them meant just a small technical editing and others required a considerable brain work. By the time the thesis reached its final form it has improved considerably and it is the merit of my reviewers. Also, many thanks to the previous generations of PhD students who continually refined the used LaTeX style – the infinite task to which I made my own contribution as well.

Many thanks to Ellen Jochemsz who helped me with really a lot of things including the Dutch summary for the thesis. It is a pity that I am leaving and losing such a great officemate. I would also like to thank my colleagues at the University that were excellent partners in sport, best companions at social events, and just good friends of mine.

It is my pleasure also to mention my old and true friends, the Fijens family, thanks to whom I feel at home staying in this country. The last but not least important is the support given to me from my wife Lida, my parents and Reks that I am feeling every minute of my life. Thank you for your love and patience.