Universitá degli studi di Roma Tre

Facoltá di Scienze Matematiche Fisiche e Naturali

Ciphertext-only reconstruction of LFSR-based stream ciphers

Author:
Stefano Guarino

Supervisor:
Prof. Marco Pedicini

Academic year 2008/2009

February 2010

Contents

Preface

In 1992, the Iranian intelligence agency arrested a salesman for Crypto AG, a Swiss company specialized in communications and information security, accusing him of spying for the United States and Germany.

The salesman, Hans Buehler, on his 25th trip to Iran on behalf of Crypto, was held in solitary confinement and interrogated, in his own words, "for five hours a day for nine months", until Crypto paid $1 million to win his freedom.

This episode was only the beginning of a big inquiry on Crypto AG, accused by many of his clients (Lybia, Iran and Iraq in particular) to collaborate with the N.S.A., the U.S. National Security Agency.

Their intelligence departments had evidence that, probably for decades, the algorithms Crypto sold them were totally known by the N.S.A. and some hidden functionalities were added to permit an easier attack.

As a consequence, all those countries, used to buy their security systems from western companies, began to build their own cipher machines, now totally unknown to the Occidentals.

This incredible story, that seems to come out of a 007 movie, brings us to the main focus of this thesis:

> is it possible to break a cryptosystem without knowing the type of cipher being used?

Let's make a brief overview.

When someone wants to send a message over an unprotected channel, he has to face two different problems.

First, even in a very generic acceptation of the idea of "communicating over a channel", such as when we write on a hard-disk or a CD, we have to face the eventuality that some data get distorted. The data we transmit (or write) are usually altered with a certain probability, called *noise* of the channel.

An entire branch of information theory, called *coding theory*, developed expressly to manage this trouble, giving birth to a large set of *error cor-*

recting codes, able to codify the informations in order to allow the detection and correction of potential transmission errors.

In most cases, a code consists in adding to the message a redundancy, as short as possible, calculated as a function of the message. The recipient get the message together with the redundancy, both affected by the noise, and uses the relation between them to identify the most probable original message.

The codes are usually a public convention and a particular channel is associated with a particular code, especially known to be efficient with that channel.

The second problem, much more ancient, is to protect the message from voluntary attacks, mining his confidentiality, integrity or authenticity. This is the reign of *cryptology*, the art of hiding informations.

Before a message can be sent over an unprotected channel, it has to be enciphered so that only the expected recipient can decipher it and get the original message.

For handiness, the process of ciphering a message follow a particular convention, depending on the cryptographic protocol being used, whose security is based on the knowledge of a secret parameter, called the *key*.

In 1883, in a list of his fundamental principles for military ciphers, the famous cryptographer Auguste Kerckhoffs stated what is known as the **Kerckhoffs' principle**:

> "a cryptosystem should be secure even if everything about the system, except the key, is public knowledge"

It means that the secrecy of the key should alone be sufficient for a good cipher to maintain at least confidentiality under an attack, and the strength of a cryptographical protocol shouldn't be based on the secrecy of the cipher's algorithm.

Anyway, what usually happens in practice is that a message is first enciphered and then coded, following a protocol shared by the sender and the recipient. The attacker doesn't know not only the key, but neither the specifications of the cryptosystem and the error correcting code used.

In particular, contrary to what one could think, the continuous development of intelligence agencies and espionage, with episodes like the one we reported, is driving everybody to build their own cryptosystem, making the attacker's work more and more difficult.

What we will do in this thesis is to consider a particular class of ciphers and put ourselves in the shoes of an attacker, to try to understand how many informations are achievable about the cipher and the original message, with the knowledge of the only material available in quite big quantity: ciphertext.

Part I

Introduction and Theoretical Background

Chapter 1

Introduction to cryptology

The term *cryptology*, from greek $\kappa\rho\upsilon\pi\tau o\varsigma$ (kryptos), that means "hidden", and $\lambda o\gamma o\varsigma$ (logos), that means "study", is the "science of the secret". It pools two different branches:

- *cryptography*, whose task is defensive, using mathematical results to build cryptosystems able to resist to every known attack
- *cryptanalysis*, whose task is offensive, trying to break or at least to weak the ciphers, looking for flaws in their algorithm

These two points of view are tightly linked, two sides of the same coin, since each of them needs the knowledge of the results of the other one.

The status of the cryptanalyst, anyway, is much more comfortable, because it's easy for him to prove he broke a cipher. The cryptographer, on the contrary, can only prove that his cipher is resistant to a certain number of known attacks, but the so called "provable security" is almost impossible to obtain.

Now, let's make a little overview on the history of cryptography.

1.1 Cryptography

Hiding informations to the enemies is a necessity as old as mankind.

The first way to protect a secret message, called *steganography*, consisted in hiding the message itself, for example tatooing it on the shaved head of the messenger and waiting for his hair to grow again.

Cryptography, contrarily, does not hide the message, but hides the informations contained in it, so that the enemy can't understand the message, even if he intercepts it.

We can formalize a cryptosystem as follows:

Definition 1.1. We call *cryptosystem* a quintuple $(\mathcal{P}, \mathcal{C}, \mathcal{K}, E, D)$, where:

- $\mathcal{P}$, $\mathcal{C}$ and $\mathcal{K}$ are three sets, respectively of the plaintexts, the ciphertexts, and the keys
- E and D are two classes of functions parameterized by two keys $K, K' \in \mathcal{K}$, such that $E_K : \mathcal{P} \to \mathcal{C}$ is the encryption function and $D_{K'} : \mathcal{C} \to \mathcal{P}$ is the corrisponding decryption function

The trivial way to break a system is the so-called *brute force attack*, the exhaustive search of the key by trying every possible element of the keyspace $\mathcal{K}$. This set has a very important role, because it must be large enough not to permit such an attempt.

In general, we call *attack* to the cryptosystem any strategy able to reduce the set of the possible keys and/or the cost necessary to find the right one.

Usually, given an alphabet $\mathcal{A}$, we have $\mathcal{P} = \mathcal{C} = \mathcal{A}^n$, so both the plaintext m and the ciphertext c are n-uples of elements of $\mathcal{A}$. By definition, $c = E_K(m)$ and $m = D_{K'}(c) = D_{K'}(E_K(m))$.

1.1.1 Symmetric cryptography

If the key K' is equal to the encryption key K, the scheme is called *symmetric*. The secret key K is shared by the sender and the recipient, and hidden to anyone else.

Symmetric cryptography is very ancient, since some of the oldest known ciphers are the jewish "atbash", cited in the Old Testament, the spartan "scytale" and the roman "Ceasar cipher", all examples of symmetric encryption.

The atbash and the Caesar cipher are substitutions ciphers, meaning that the encryption consists in substituting every element of the alphabet with another one. The key is simply the rule used for the substitutions:

- in the atbash the first letter of the alphabet was swapped with the last one, the second with the penultimate and so on
- in the Caesar cipher every letter was substituted with the one three positions further down in the alphabet

Let's notice that the Caesar cipher, for example, being based on a translation of the letters of the alphabet, is very unsafe, because the keyspace has only 26 elements (with the english alphabet), allowing a brute force attack.

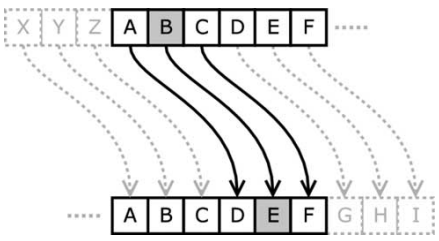


Figure 1.1: Caesar cipher

Figure 1.2: A scytale

The scytale, instead, was a wooden cylinder used togheter with a leather strip. The strip was twisted around the cilinder and the message was written in the direction of the stick, transversely to the strip. The key was the diameter of the cylinder, because twisting the strip around a cylinder of a different thickness entailed the outcome of a different message.

We will see later in details how symmetric cryptography develloped and how it is used nowadays.

1.1.2 Asymmetric cryptography

In the *asymmetric* cryptography, contrarily, every user of the cipher has his own couple of keys, the encryption key K, called *public key*, and the decryption key K', called *private key*.

The public key is public knowledge, anyone can use it to encrypt a message that only the owner of the corrispondent private key can decrypt.

Asymmetric cryptography, contrary to symmetric one, is very recent. Needing a big computational power, it was introduced only in the 70's by W. Diffie and M. Hellman.

Asymmetric protocols are based on the so called *trapdoor functions*, mathematical functions, such as the discrete logarithm, easy to calculate with the knowledge of a parameter, the *trapdoor*, but almost impossible to compute without it.

Probably the most famous asymmetric cryptosystem is the RSA, that takes his name from D. Rivest, A. Shamir and L. Adleman who developped it in 1978. The RSA grounds on the difficulty of factorizing big integers: the private key is a couple of big primes (about 512 bits each) and the public key is their product. The algorithm is shown in figure 1.3

Asymmetric cryptography, apparently easier to implement, is usually used only to cipher little messages, in particular to exchange a secret key later used for a symmetric protocol.

The reasons are primarily two:

- Cost: RSA, one of the faster asymmetric protocols, is a thousandfold slower than the DES, one of the more diffused symmetric block ciphers (see later)
- Security: the strenght of public-key cryptography is based on the *computational complexity theory*. This theory supposes the existence of

(and search for) problems impossible to solve in polynomial time, so perfect to be used in asymmetric protocols. This "non-computability", however, hasn't ever been proved. So what if someone finds a way, for example, to factorize in a faster way than what is now believed possible?

Key Generation	
Select p, q	p, q both prime, p≠q
Calculate n = p×q	
Calculate $\phi(n) = (p-1)\times(q-1)$	
Select integer e	$\gcd(\phi(n),e) = 1;\ 1<e<\phi(n)$
Calculate d	
Public key	K = {e, n}
Private key	K' = {d, n}

Encryption	
Plaintext:	M < n
Ciphertext:	$C = M^e \pmod n$

Decryption	
Ciphertext:	C
Plaintext:	$M = C^d \pmod n$

Figure 1.3: RSA algorithm

1.2 Characterization of symmetric cryptography

In this work we're only interested in symmetric cryptography. It is the kind of protocols used in military and business security, mainly for the reasons exposed above.

What follows is its basic classification in block ciphers and stream ciphers.

1.2.1 Block ciphers

A block cipher is a symmetric key cipher operating on fixed-length groups of digits, termed blocks, with an unvarying transformation. Usually this ciphers applies to binary transmissions, so we will refer to them in that context. A formal definition follows.

Definition 1.2. Let $\mathcal{A}$ be an alphabet of q symbols and $\mathcal{K}$ be the keyspace. We call *block cipher* a memoryless encryption scheme which breaks up the plaintext message $m \in \mathcal{A}^*$ into strings (called blocks) of a fixed length n, $m = m_1|m_2|\ldots|m_i|\ldots$ where each $m_i \in \mathcal{A}^n$, and encrypts one block at a

time. The encryption transformation is $E_K : \mathcal{A}^n \to \mathcal{A}^{n'}$, where $K \in \mathcal{K}$ is the key used and n' can be bigger than n.

Introduced by H. Feistel in 1973, block ciphers achieved success with the Data Encryption Standard (DES), publicly released in 1976 and adopted as a U.S. government Federal Standard.

DES originally had a block size of 64 bits and a key size of 56 bits, but, as time went on, its inadequacy became apparent and a variant of DES, called Triple DES, was widely adopted as a replacement. It triple-encrypts blocks with (usually) two different keys, resulting in a 112-bit keys and 80-bit security, and it is still considered secure.

DES has been superseded as a U.S. Federal Standard by the Advanced Encryption Standard (AES), developed by two belgian cryptographers, Joan Daemen and Vincent Rijmen, and submitted under the name Rijndael. AES has a block size of 128 bits and three possible key sizes, 128, 192 and 256 bits.

In a block cipher both algorithms, E_K for encryption and D_K for decryption, accept two inputs: an input block m_i of size n bits and a key K of size k bits, yielding an n'-bit output block. Usually $n' = n$, so for each key K, E_K is a permutation (a bijective mapping) over the set of all possible input blocks: the key selects one permutation from the $2^n!$ possible ones.

The block size, n, is typically 64 or 128 bits, although some ciphers have a variable block size. 64 bits was the most common length until the mid-1990s, when new designs began to switch to the longer 128-bit length. One of several modes of operation is generally used along with a padding scheme to allow plaintexts of arbitrary lengths to be formatted in n-blocks and encrypted. Each mode has different characteristics in regard to error propagation, ease of random access and vulnerability to certain types of attack. Typical key sizes include 40, 56, 64, 80, 128, 192 and 256 bits, but 80 bits is normally taken as the minimum key length needed to prevent brute force attacks.

Most block ciphers are constructed by repeatedly applying a simpler function, with an approach known as *iterated block cipher* or *Feistel cipher*. Each iteration is termed a *round*, and the repeated function is termed the *round function*. Typically, the number of rounds varies from 4 to 32: DES has 16 rounds, AES has 10, 12 or 14 rounds, according to the lenght of the key. The round function is usually a composition of arithmetic and logical operations (especially XOR), S-boxes (substitution boxes) and permutations.

1.2.2 Stream ciphers

A stream cipher is a symmetric key cipher where plaintext digits are combined one-to-one with a pseudorandom stream, called keystream, typically

by a modular sum (an exclusive-or (XOR) in the binary case). Formally:

Definition 1.3. Let $\mathcal{A}$ be an alphabet of q symbols and $m = m_1 m_2 \cdots m_i \cdots \in \mathcal{A}^*$ be a message. Given a set $\mathcal{K}$, let $K = k_1 k_2 \cdots k_i \cdots \in \mathcal{K}^*$ be the keystream and $E_{k_i} : \mathcal{A} \to \mathcal{A}$ be a family of simple substitutions on $\mathcal{A}$. A *stream cipher* is a memoryless encryption scheme which encrypts the plaintext message m one element at a time into the ciphertext $c = c_1 c_2 \cdots c_i \cdots \in \mathcal{A}^*$, where $\forall i \; c_i = E_{k_i}(m_i)$.

This class of ciphers is inspired by the so-called *One-Time Pad*, a stream cipher where the keystream is a random stream as long as the message, used only once and then changed everytime a new message has to be encrypted.

In 1949 Claude Shannon[15] showed that the OTP is unbreakable, or, in his words, it has the property of *perfect secrecy*, meaning that the knowledge of the ciphertext c gives absolutely no additional informations about the plaintext m.

In his work on *Information Theory*, Shannon defined a function called *entropy*, that measures the uncertainty's degree of a random variable. If X is a random variable assuming the values $x_0, x_1, \ldots, x_n$ each with probability $P(X = x_i) = p_i$, the entropy of X is defined as:

$$H(X) = \sum_{i=0}^{n} p_i \log \frac{1}{p_i}$$

where the base of the logarithm is the same base used to store informations and the unit of measurement of the entropy, usually 2.

If we think of a message as the outcome of a random variable, Shannon's entropy is a measure, in the sense of an expected value, of the amount of information contained in a message. At the same time, it measures the average information content we're missing when we do not know the value assumed by that variable.

In terms of entropy, the perfect secrecy can then be formulated as:

$$H(m) = H(m|c)$$

where $H(m)$ is the entropy of the plaintext m, while $H(m|c)$ is the conditional entropy of m given the ciphertext c.

Shannon also showed that a necessary condition for perfect secrecy is

$$H(K) \geq H(m)$$

It means that the uncertainty on the secret key must be at least as high as the one on the plaintext. If the key's lenght is k and the elements of the keystream are chosen randomly and regardless one of the others, the entropy of the key is $H(K) = k$, so Shannon's condition become $k \geq H(m)$, and this condition is obviously verified by the OTP.

The OTP is, however, impossible to realize, because it implies that the sender of the message is able to communicate to the recipient a key as long as the message he wants to transmit. That's why stream ciphers use a pseudorandom key, generated from a shorter key easy to share by the parts.

Stream ciphers are a very important class of ciphers, widely used either in the military and governmental or in the industrial security. They are very easy to implement both in hardware and software, much faster than block ciphers and, usually, have the remarkable property that an error in the trasmission of a digit only affect that digit in the ciphertext and does not propagate to other parts of the message, since the plaintext digits are encrypted one at a time.

Furthermore, Shannon's theory put strong bases for the development of stream ciphers, since it highlights how the security of such a cipher only depends on the randomicity of the keystream.

We can further classificate stream ciphers into *synchronous* and *self-synchronous* ciphers.

In a synchronous stream cipher the keystream is generated independently from the ciphertext, while in a self-synchronous one a finite state machine also uses ciphertext digits to create the keystream.

The effect of this different design is that in a self-synchronous cipher, if ciphertext digits are inserted or deleted, only a fixed number of plaintext digits are lost in the decryption, since these ciphers have the capability of re-establishing the synchronization.

Anyhow, in our work synchronous stream ciphers have a more preeminent role. We can describe them schematically as in figure 1.4.

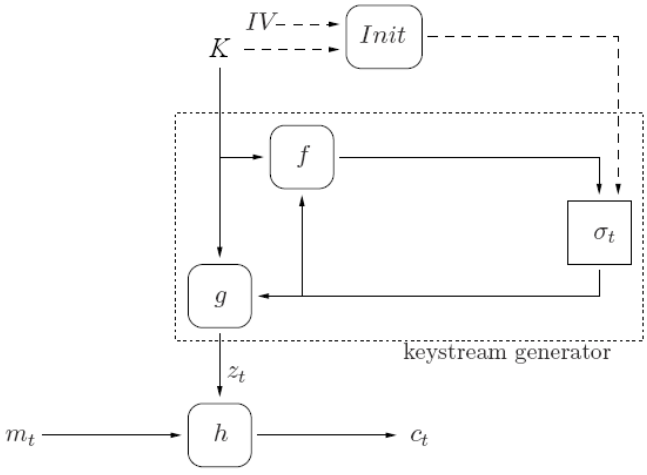


Figure 1.4: A typical synchronous stream cipher

- the function $Init$ generate an initial state σ_0 from the key K and an initial vector IV:
$$\sigma_0 = init(K, IV)$$

- in every round a function f is used to update the internal state of the cipher:
$$\sigma_t = f(K, \sigma_{t-1})$$

- another function g is then used to generate the keystream z_t as a function of the key and the internal state:
$$z_t = g(K, \sigma_t)$$

- finally the plaintext digits m_t are combined with the keystream z_t through a function h to generate the ciphertext digits c_t:
$$c_t = h(m_t, z_t)$$

Let's note that the function $h(x, y)$ must be invertible in the first variable, namely there must exist a function h^{-1} such that $\forall t\ m_t = h^{-1}(c_t, z_t)$.

Usually, the plaintext, keystream and ciphertext digits are elements from a finite field $\mathbb{F}_q$. If $h(m_t, z_t) := m_t + z_t$ is the modular sum, then the stream cipher is called *additive*. The most common case is the additive with $q = 2$, where the sum coincide with the bitwise XOR operation.

Chapter 2

Linear Feedback Shift Registers

In the description of synchronous stream ciphers we gave previously, we skipped the characterization of the keystream generator functions, probably the most important primitive of those ciphers.

Particularly interesting in our work is a basic component of many ciphers: the **Linear Feedback Shift Register** (**LFSR**).

A *shift register* of length L over $\mathbb{F}_q$ is a register with L stages, whose *state* is a sequence of L elements of $\mathbb{F}_q$ each one memorized in one of the stages. Governed by a clock, at every clock stroke the register undergoes a transition where every element "shift" to the next stage, but the last one which is outputted. The first stage, now empty, is filled with a new element calculated as a function of the previous state. If the input element of a shift register is a linear combination of its previous state, then the register is called *linear*, shorten *LFSR*.

Formally, in a generic instant t the state of the register is a L-tuple $\mathbf{s}_t = (s_t, \ldots, s_{t+L-1}) \in \mathbb{F}_q^L$.

The initialization of the register is $\mathbf{s}_0 = (s_0, \ldots, s_{L-1})$ and the sequence generated by the LFSR is defined recursively from the initial state as

$$s_{t+L} = \sum_{i=0}^{L-1} a_i s_{t+i}, \quad t \geq 0$$

where $a_i \in \mathbb{F}_q$ for all i and the sum obviously is modular in $\mathbb{F}_q$.

The stages of the register involved in the feedback function, namely those with $a_i \neq 0$, are termed *taps*.

Clearly, the element s_t generated at time t depends only on the previous state $\mathbf{s}_{t-1}$, thus the whole sequence depends only on the initial state $\mathbf{s}_0$ and on the feedback coefficients a_i.

In the next sections we will see more in details the properties of the sequences generated by a LFSR.

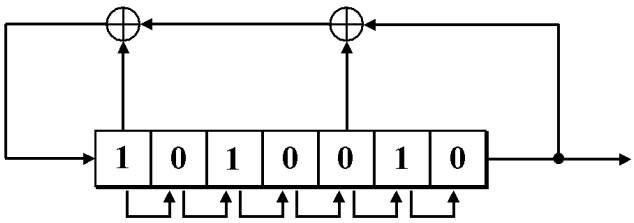


Figure 2.1: A LFSR of length 7 over $\mathbb{F}_2$. The taps are in position 2 and 6, so the feedback relation is $s_{t+7} = s_{t+2} + s_{t+6}$. The present state is 0100101.

2.1 Linear recurring sequences in $\mathbb{F}_q$

First of all a formal definition.

Definition 2.1. Let L be a positive integer, a *Lth-order linear recurring sequence* over a finite field $\mathbb{F}_q$ is a sequence $\{s_t\}_{t\in\mathbb{N}}$ of elements of $\mathbb{F}_q$ such that

$$s_{t+L} = a_{L-1}s_{t+L-1} + a_{L-2}s_{t+L-2} + \cdots + a_0 s_t + a \quad \forall t \in \mathbb{N} \tag{2.1.1}$$

with $a, a_i \in \mathbb{F}_q$, for all $i = 0, \ldots, L-1$.

Relation 2.1.1 is called the *linear recurrence relation* of the sequence, and the terms $s_0, s_1, \ldots, s_{L-1}$, which generate the whole sequence, are called the initial values. If $a = 0$ the sequence is called *homogeneous*, otherwise is called *inhomogeneous*.

Remark 2.1. An inhomogeneous Lth-order linear recurring sequence can always be seen as a homogeneous $(L+1)$th-order linear recurring sequence. In fact, given relation 2.1.1 with $a \neq 0$, we can write

$$\begin{cases} s_{t+L} = a_{L-1}s_{t+L-1} + \cdots + a_0 s_t + a \\ s_{t+L+1} = a_{L-1}s_{t+L} + \cdots + a_0 s_{t+1} + a \end{cases}$$

for all t and, subtracting the two equations, we get

$$s_{t+L+1} = (a_{L-1}+1)s_{t+L} + (a_{L-2} - a_{L-1})s_{t+L-1} + \cdots + (a_0 - a_1)s_{t+1} - a_0 s_t$$

Now, if we put $b_L = a_{L-1} + 1$, $b_{L-i} = a_{L-i-1} - a_{L-i}$ for all $i = 1, \ldots, L-1$ and $b_0 = -a_0$, we have the equivalent homogeneous $(L+1)$th-order linear recurring sequence

$$s_{t+L+1} = b_L s_{t+L} + b_{L-1}s_{t+L-1} + \cdots + b_1 s_{t+1} + b_0 s_t \quad \forall t \in \mathbb{N}$$

Hereinafter, then, we will consider only homogeneous sequences of the form

$$s_{t+L} = a_{L-1}s_{t+L-1} + a_{L-2}s_{t+L-2} + \cdots + a_0 s_t \quad \forall t \in \mathbb{N} \tag{2.1.2}$$

Definition 2.2. Given a Lth-order linear recurring sequence $\{s_t\}_{t\in\mathbb{N}}$, we call its *tth-state vector* the vector

$$\mathbf{s}_t = (s_t, s_{t+1}, \dots, s_{t+L-1}) \in \mathbb{F}_q^L$$

The vector $\mathbf{s}_0$ is called the *initial state vector* of the sequence.

As we have already remarked, a linear recurring sequence is completely determined by the initial state and the coefficients a_i, so it can be seen as a pair $(\mathbf{s}_0, \mathbf{a})$, where $\mathbf{a} = (a_0, a_1, \dots, a_{L-1}) \in \mathbb{F}_q^L$ is the vector of the coefficients of the linear recurrence relation 2.1.2.

If we consider the matrix $A \in \mathcal{M}_L(\mathbb{F}_q)$ defined as

$$A = \begin{pmatrix} 0 & 0 & \cdots & 0 & a_0 \\ 1 & 0 & \cdots & 0 & a_1 \\ 0 & 1 & \cdots & 0 & a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & a_{L-1} \end{pmatrix}$$

we can write the recurrence relation as

$$\mathbf{s}_t = \mathbf{s}_{t-1}A = \mathbf{s}_{t-2}A^2 = \cdots = \mathbf{s}_0 A^t \quad \forall t \in \mathbb{N}$$

Note that the matrix A depends only on the linear recurrence relation 2.1.2 and not on the elements of the sequence. Studying its properties, then, gives us informations on all the sequences satisfying the same relation, regardless of the initial state.

2.1.1 Periodicity properties

Recurring sequences have very interesting periodicity properties.

Definition 2.3. Let $\{s_t\}_{t\in\mathbb{N}}$ be a sequence in $\mathbb{F}_q$. If there exist two integers $r > 0$ and $t_r \geq 0$ such that

$$s_{t+r} = s_t \quad \forall t \geq t_r$$

then the sequence is called *ultimately periodic* and r is a *period* of the sequence. The smallest period $\tilde{r}$ of a sequence is called its *least period*. If $t_{\tilde{r}} = 0$ then the sequence is called *periodic*.

Lemma 2.1. Let $\{s_t\}_{t\in\mathbb{N}}$ be a ultimately periodic sequence whose least period is $\tilde{r}$. If r is another period of the sequence, then $\tilde{r} \mid r$.

Proof. Let t_r and t_0 be to integers such that $s_{t+r} = s_t$ for all $t \geq t_r$ and $s_{t+\tilde{r}} = s_t$ for all $t \geq t_0$. If $\tilde{r} \not| r$, by the euclidean division we can write

$r = m\tilde{r} + k$, for some integers $m \geq 0$ and $0 < k < \tilde{r}$. Hence, for all $t \geq \max(t_0, t_r)$ we have

$$s_t = s_{t+r} = s_{t+m\tilde{r}+k} = s_{t+(m-1)\tilde{r}+k} = \cdots = s_{t+k}$$

which means that k is a period of the sequence, but that's impossible since $k < \tilde{r}$ and $\tilde{r}$ is the least period. □

Definition 2.4. Let $\{s_t\}_{t\in\mathbb{N}}$ be a ultimately periodic sequence whose least period is $\tilde{r}$, then we call the *preperiod* $\tilde{t}$ of the sequence the smallest non-negative integer such that the equality $s_{t+r} = s_t$ holds for all $t \geq \tilde{t}$.

The following Proposition states that, as we would expect, in an ultimately periodic sequence, the preperiod $\tilde{t}$ is the same for every period of the sequence, so every periodicity begins from the same point.

Proposition 2.1. Let $\{s_t\}_{t\in\mathbb{N}}$ be a ultimately periodic sequence whose preperiod is $\tilde{t}$. Let r be any period of the sequence, then $\tilde{t}$ is the smallest non-negative integer for which the equality $s_{t+r} = s_t$ holds for all $t \geq \tilde{t}$.

Proof. Let $\tilde{r}$ be the least period of the sequence, and r any other period. We have

$$\begin{cases} s_{t+\tilde{r}} = s_t & \forall t \geq \tilde{t} \\ s_{t+r} = s_t & \forall t \geq t_r \end{cases}$$

where t_r is the smallest non-negative integer for which the second equality holds.

We want to show that $t_r = \tilde{t}$. From lemma 2.1 we know that $\tilde{r} \mid r$, so there exists an integer h such that $r = h\tilde{r}$. Now:

- $t_r \leq \tilde{t}$: clearly it holds $s_t = s_{t+\tilde{r}} = s_{t+h\tilde{r}} = s_{t+r} \quad \forall t \geq \tilde{t}$ but we know that t_r is the smallest integer for which the last equality holds, so it implies that $t_r \leq \tilde{t}$.

- $t_r \geq \tilde{t}$: by definition of $\tilde{t}$, $s_{\tilde{t}-1} \neq s_{\tilde{t}+\tilde{r}-1}$. If $t_r < \tilde{t}$, then $s_{\tilde{t}-1} = s_{\tilde{t}+r-1} = s_{\tilde{t}+h\tilde{r}-1} = s_{\tilde{t}+\tilde{r}-1}$, which contradicts the inequality above. So it must be $t_r \geq \tilde{t}$.

□

Periodicity properties are particularly interesting thanks to the following Theorem.

Theorem 2.1. *For all $L > 0$, every Lth-order linear recurring sequence in any finite field $\mathbb{F}_q$ is ultimately periodic with least period $\tilde{r} \leq q^L - 1$.*

Proof. To prove the Theorem it suffices to recall that a recurring sequence is completely determined by its initial state. Passing through the all-zero state brings obviously to the all-zero sequence, which is ultimately periodic with least period 1. Since any state of the sequence is a vector $\mathbf{s}_t \in \mathbb{F}_q^L$, clearly the number of possible non-zero states is $q^L - 1$, so that's the longest possible period. □

Remark 2.2. Note that not every ultimately periodic sequence is periodic. For example, applying the recurrence relation $s_t = s_{t-1}$ to the initial state $(0, 1)$ generates the sequence $01111\ldots$, which is ultimately periodic but not periodic, since its preperiod is the initial 0.

We can obtain periodic sequences just using the following simple result.

Proposition 2.2. Let $\{s_t\}_{t\in\mathbb{N}}$ be a linear recurring sequence in $\mathbb{F}_q$ satisfying the recurrence relation 2.1.2. If $a_0 \neq 0$ then the sequence is periodic.

Proof. Thanks to Theorem 2.1, we know the sequence to be ultimately periodic. Let $\tilde{r}$ be its least period and t_0 be its preperiod, so that $s_{t+\tilde{r}} = s_t$ for all $t \geq t_0$. If $a_0 \neq 0$ it is clearly invertible in $\mathbb{F}_q$, thus from equation 2.1.2 we have

$$s_t = a_0^{-1}(s_{t+L} - a_{L-1}s_{t+L-1} - \cdots - a_1 s_{t+1})$$

now, if we suppose $t_0 \geq 1$, we can take $t = t_0 - 1$ and obtain

$$s_{t_0-1} = a_0^{-1}(s_{t_0+L-1} - a_{L-1}s_{t_0+L-2} - \cdots - a_1 s_{t_0})$$

If, otherwise, we take $t = t_0 - 1 + \tilde{r}$ and use the definition of periodicity we have

$$\begin{aligned} s_{t_0-1+\tilde{r}} &= a_0^{-1}(s_{t_0+L-1+\tilde{r}} - a_{L-1}s_{t_0+L-2+\tilde{r}} - \cdots - a_1 s_{t_0+\tilde{r}}) \\ &= a_0^{-1}(s_{t_0+L-1} - a_{L-1}s_{t_0+L-2} - \cdots - a_1 s_{t_0}) \end{aligned}$$

but it means that $s_{t_0-1+\tilde{r}} = s_{t_0-1}$, which contradicts the definition of preperiod. □

The previous condition is only a sufficient one, but not necessary, as shown by the following example.

Example 2.1. Every sequence $(\mathbf{s}_0, \mathbf{a})$ with $\mathbf{s}_0 = (g, g)$ (where $g \in \mathbb{F}_q$) and $\mathbf{a} = (a_0, a_1) = (0, 1)$ is clearly periodic ($s_t = g$ for all $t \in \mathbb{N}$).

Finally, we show how the least period of a sequence is related to its recurrence relation.

Proposition 2.3. Let $\{s_t\}_{t\in\mathbb{N}}$ be a Lth-order recurring sequence in $\mathbb{F}_q$ with $a_0 \neq 0$ and let A be its associated matrix. Thus the least period $\tilde{r}$ of the sequence divides the order of A in the group $GL_L(\mathbb{F}_q)$ of $L \times L$ invertible matrices over $\mathbb{F}_q$: $\tilde{r} \mid ord(A)$.

Proof. First of all, we have $detA = (-1)^{L-1}a_0 \neq 0$, so actually $A \in GL_L(\mathbb{F}_q)$. Let m be the order of A in $GL_L(\mathbb{F}_q)$, namely m is the lowest integer such that $A^m = \mathbb{I}_L$. Then

$$\mathbf{s}_{t+m} = \mathbf{s}_0 A^{t+m} = \mathbf{s}_0 A^t A^m = \mathbf{s}_0 A^t = \mathbf{s}_t$$

so m is a period of the sequence and by lemma 2.1 $\tilde{r} \mid m$. □

2.1.2 Maximal period sequences

Our interest in the properties of linear recurring sequences is due to our search for a easy way to generate pseudo-randomic streams.

Linear recurring sequences can be easily generated by a LFSR, but usually they do not have good statistical properties.

The circumstances change when the sequence has a long period, so the purpose of this section is to recognize the features allowing to obtain a recurring sequence of longest possible period.

Definition 2.5. An *impulse response sequence* is a homogeneous linear recurring sequence $(\mathbf{d}, \mathbf{a})$, such that $\mathbf{d} = (0, \ldots, 0, 1)$.

Proposition 2.4. Let $\tilde{r}_s$ be the least period of a Lth-order linear recurring sequence $(\mathbf{s}_0, \mathbf{a})$ and let $\tilde{r}_d$ be the least period of the corresponding impulse response sequence $(\mathbf{d}, \mathbf{a})$. Then $\tilde{r}_s \mid \tilde{r}_d$ for all $\mathbf{s}_0 \in \mathbb{F}_q^L$.

Proof. First, let $\mathbf{d}_t$ denote the tth state vector of the impulse response sequence. Note that $\mathbf{d}_n = \mathbf{d}_m$ if and only if $A^n = A^m$. In fact $\mathbf{d}_n = \mathbf{d}_m$ if and only if $\mathbf{d}_{n+t} = \mathbf{d}_t A^n = \mathbf{d}_t A^m = \mathbf{d}_{m+t}$ for all $t \geq 0$, but the last relation holds if and only if $A^n = A^m$, since $\mathbf{d}, \mathbf{d}_1, \ldots, \mathbf{d}_{L-1}$ obviously form a basis for the L-dimensional vector space $\mathbb{F}_q^L$ over $\mathbb{F}_q$.

Now, if t_0 is the preperiod of $(\mathbf{d}, \mathbf{a})$, we have $\mathbf{d}_t = \mathbf{d}_{t+\tilde{r}_d}$ for all $t \geq t_0$. As we have seen, it implies that $A^t = A^{t+\tilde{r}_d}$ for all $t \geq t_0$, so clearly $\mathbf{s}_t = \mathbf{s}_{t+\tilde{r}_d}$ and then $\tilde{r}_d$ is a period of $(\mathbf{s}_0, \mathbf{a})$. The conclusion follows from lemma 2.1. □

Proposition 2.5. Let $(\mathbf{d}, \mathbf{a})$ be a Lth-order impulse response sequence in $\mathbb{F}_q$ with $a_0 \neq 0$ and let A be its associated matrix. Then its least period $\tilde{r}$ is equal to the order of A in $GL_L(\mathbb{F}_q)$: $\tilde{r} = ord(A)$.

Proof. Thanks to Proposition 2.3 we know that $\tilde{r} \mid ord(A)$. On the other hand, by Proposition 2.2 we know the sequence to be periodic, so $\mathbf{d}_{\tilde{r}} = \mathbf{d}$. It implies, as we have seen in the proof of Proposition 2.4, $A^{\tilde{r}} = A^0 = \mathbb{I}_L$, which implies $ord(A) \mid \tilde{r}$. □

Associated with a linear recurring sequence are two particular polynomials.

Definition 2.6. Let $\{s_t\}_{t\in\mathbb{N}}$ be a Lth-order linear recurring sequence whose recurrence relation is 2.1.2.

We call its *characteristic polynomial* the polynomial

$$f(X) = X^L - \sum_{i=1}^{L} a_{L-i}X^{L-i} = X^L - a_{L-1}X^{L-1} - \cdots - a_1X - a_0 \quad (2.1.3)$$

while its *feedback polynomial* is

$$f^*(X) = 1 - \sum_{i=1}^{L} a_{L-i}X^i = 1 - a_{L-1}X - \cdots - a_0X^L \quad (2.1.4)$$

Remark 2.3. Note that the feedback polynomial is the reciprocal of the characteristic polynomial: $f^*(X) = X^L f\left(\frac{1}{X}\right)$. Hence, every property of f^* is restatable in terms of f and vice versa. Since the two polynomials have very different shapes, the context should allow the reader to distinguish which polynomial we are talking about. Note, moreover, that these two polynomials depend only on the recurrence relation and not on the initial state. Thereafter, $\mathcal{S}(f^*)$ will denote the set of all sequences whose feedback polynomial is f^*, or analogously $\mathcal{S}(f)$ the set of all sequences whose characteristic polynomial is f.

Example 2.2. Take the recurrence relation of order 2 over $\mathbb{F}_5$ defined by $s_{t+2} = 2s_{t+1} + s_t$, whose vector of coefficients is $\mathbf{a} = (1, 2)$.

The characteristic polynomial of the sequence is hence $f(X) = X^2 - 2X - 1$, while its feedback polynomial is $f^*(X) = 1 - 2X - X^2$. Since we are in $\mathbb{F}_5$, they can be restated as $f(X) = X^2 + 3X + 4$ and $f^*(X) = 4X^2 + 3X + 1$. The relation $f^*(X) = X^2 f\left(\frac{1}{X}\right)$ is clearly verified.

Let's now recall an important definition.

Definition 2.7. Let $f(X) \in \mathbb{F}_q[X]$, we call its *order*, denoted $ord(f)$, the smallest integer k such that $X^k \equiv 1 \mod f(X)$.

Remark 2.4. If $\{s_t\}_{t\in\mathbb{N}}$ is a Lth-order linear recurring sequence over $\mathbb{F}_q$ whose characteristic polynomial is $f(X)$ and whose associated matrix is A, $f(X)$ is the characteristic polynomial of the matrix A: $f(X) = det(A - X\mathbb{I}_L)$. Analogously, A is the companion matrix of $f(X)$, so, if $a_0 \neq 0$, then A is invertible and $ord(f) = ord(A)$.

Thanks to the characteristic polynomial, we can recast some of the previous Propositions as follows.

Theorem 2.2. *Let $(\mathbf{s}_0, \mathbf{a})$ be a linear recurring sequence in $\mathbb{F}_q$ and let $f(X)$ be its characteristic polynomial. Then*

- *the least period $\tilde{r}$ divides the order of $f(X)$: $\tilde{r} \mid ord(f)$*

- *the least period of the corresponding impulse sequence* $(\mathbf{d}, \mathbf{a})$ *equals it:* $\tilde{r} = ord(f)$

Many relations between linear recurring sequences and their characteristic polynomials can be found on the basis of the following polynomial identity. We will not prove it now, the interested reader can find the original proof in chapter 8 of [10], while we will give an alternative proof later.

Theorem 2.3. *Let* $\{s_t\}_{t\in\mathbb{N}}$ *be a* L*th-order homogeneous linear recurring sequence over* $\mathbb{F}_q$*, periodic with period* r*. Let 2.1.2 be its recurrence relation and* $f(X)$ *be its characteristic polynomial. Then the identity*

$$f(X)s(X) = (1 - X^r)h(X) \tag{2.1.5}$$

holds with

$$s(X) = s_0X^{r-1} + s_1X^{r-2} + \cdots + s_{r-2}X + s_{r-1} \ \in \mathbb{F}_q[X]$$

and

$$h(X) = \sum_{j=0}^{L-1} \sum_{i=0}^{L-1-j} a_{i+j+1}s_iX^j \ \in \mathbb{F}_q[X]$$

where we set $a_L = -1$ *for convenience.*

Theorems 2.2 and 2.3 lead us to the following important result.

Proposition 2.6. Let $(\mathbf{s}_0, \mathbf{a})$ be a linear recurring sequence in $\mathbb{F}_q$ with $\mathbf{s}_0 \neq (0, 0, \ldots, 0)$ and let $f(X) \in \mathbb{F}_q[X]$ be its characteristic polynomial. If $f(X)$ is irreducible over $\mathbb{F}_q$ and $f(0) \neq 0$, then the sequence is periodic with least period $\tilde{r}$ equal to the order of $f(X)$: $\tilde{r} = ord(f)$.

Proof. First, $f(0) \neq 0$ means $a_0 \neq 0$ and the periodicity of the sequence follows from Proposition 2.2. Now, from identity 2.1.5 follows that $f(X)$ divides $(X^{\tilde{r}} - 1)h(X)$. Since $s(X)$ and $h(X)$ are non-zero polynomials and since $deg(h) < deg(f)$, the irreducibility of $f(X)$ implies that $f(X)$ divides $X^{\tilde{r}} - 1$. It means that $X^{\tilde{r}} \equiv 1 \mod f(X)$, so necessarily $ord(f) \mid \tilde{r}$. On the other hand $\tilde{r} \mid ord(f)$ by Theorem 2.2, hence $\tilde{r} = ord(f)$. □

Let's recap what we have seen until now.

Take any Lth-order recurrence relation over a finite field $\mathbb{F}_q$ and let $f(X)$ be its characteristic polynomial. The relation defines a partition of $\mathbb{F}_q^L$ in distinct "orbits": for every vector $\mathbf{v} \in \mathbb{F}_q^L$, the vectors in the same orbit of $\mathbf{v}$ are all the state vectors, and only those, of the sequence with $\mathbf{v}$ as initial vector and generated by that relation. The greatest possible least period equals the order of $f(X)$ and every other least period divides it. If $f(X)$ is irreducible, then every orbit has the same size, $ord(f)$, obviously except the orbit of the vector $(0, \ldots, 0) \in \mathbb{F}_q^L$, which is the only one element in its orbit.

As we have seen, the impulse vector $\mathbf{d} = (0, \ldots, 0, 1) \in \mathbb{F}_q^L$ always generates a sequence with period $\tilde{r} = ord(f)$, so its orbit is always the greatest one (or one of the greatest).

Example 2.3. As in example 2.2, let's take the recurrence relation $s_{t+2} = 2s_{t+1} + s_t$ over $\mathbb{F}_5$ defined by the vector $\mathbf{a} = (1, 2)$. The characteristic polynomial is $f(X) = X^2 - 2X - 1$ which is irreducible over $\mathbb{F}_5$. The 25 elements of the set $\mathbb{F}_5^2$ are, then, partitioned in 3 orbits:

$$\{(0,0)\} \qquad \text{which obviously contain only the all-zero vector}$$

$$\{(0,1),(1,2),(2,0),(0,2),(2,4),(4,0),(0,4),(4,3),(3,0),(0,3),(3,1),(1,0)\}$$

$$\{(1,1),(1,3),(3,2),(2,2),(2,1),(1,4),(4,4),(4,2),(2,3),(3,3),(3,4),(4,1)\}$$

Note that, since $ord(f) = 12$ and $f(X)$ is irreducible, as we expected the non-zero orbits both contain exactly 12 elements.

As we have already outlined, our first aim is to detect the conditions producing the longest possible period. Thanks to the previous results, we know that the best we can aspire to is a recurrence relation generating only two orbits in $\mathbb{F}_q^L$: the one with only the $(0, \ldots, 0)$ vector and another one containing all the remaining $q^L - 1$ vectors. The following result is fundamental.

Theorem 2.4. *Let $\{s_t\}_{t\in\mathbb{N}}$ be a homogeneous Lth-order linear recurring sequence whose characteristic polynomial is primitive over $\mathbb{F}_q$ and whose initial state is non-zero. Then $\{s_t\}$ is periodic and its least period is $\tilde{r} = ord(f) = q^L - 1$.*

We call such a sequence a maximal period sequence.

Proof. The Theorem is trivial. Every primitive polynomial is irreducible, so by Proposition 2.6 $\tilde{r} = ord(f)$. Moreover $ord(f) = q^L - 1$ by definition of primitive polynomial. □

Example 2.4. Let's take again $\mathbb{F}_5^2$. If we choose the recurrence relation $s_{t+2} = 2s_{t+1} + 2s_t$, we have the characteristic polynomial $f(X) = X^2 - 2X - 2$ which is known to be primitive over $\mathbb{F}_5$.

Hence, for every initial vector $\mathbf{s}_0 \neq (0, 0)$, we enter a maximal period sequence of length $\tilde{r} = 5^2 - 1 = 24$. If we started with the impulse $\mathbf{s}_0 = (0, 1)$, for example, the sequence would be:

$$\overbrace{0,1,2,1,1,4,0,3,1,3,3,2,0,4,3,4,4,1,0,2,4,2,2,3}, 0,1,2,\ldots$$

The following Proposition shows that maximal period sequences also have the other feature we were looking for: good statistical properties.

Proposition 2.7. Let $S = (\mathbf{s}_0, \mathbf{a})$ be a maximal period sequence of order L in $\mathbb{F}_q$. Let m be an integer such that $1 \leq m \leq L$ and S' any subsequence of S of length $q^L + m - q$. Then every non-zero sequence of length m appears exactly $q^{L-m} - 1$ times as a subsequence of S'. The distribution of patterns of fixed length $m \leq L$ is almost uniform.

In [18], the interested reader can find a proof of the previous Proposition and check that a sequence with such properties also satisfies the Golombs randomness postulates.

Finally, with the following Theorem we note that, at least in some special cases, the introduction of the characteristic polynomial also permits to explicitly represent the elements of a linear recurring sequence through algebraic structures over $\mathbb{F}_q$.

Theorem 2.5. *Let $\{s_t\}_{t\in\mathbb{N}}$ be a Lth-order linear recurring sequence over $\mathbb{F}_q$ whose characteristic polynomial is $f(X)$. If the roots $\alpha_1, \ldots, \alpha_L$ of $f(X)$ are all distinct, then for all $t \geq 0$ we have*

$$s_t = \sum_{j=1}^{L} \beta_j \alpha_j^t \tag{2.1.6}$$

where $\beta_1, \ldots, \beta_L$ are elements of the splitting field of $f(X)$ over $\mathbb{F}_q$, uniquely determined by the initial state of the sequence.

Proof. First of all, given the initial state $s_0, \ldots, s_{L-1}$ of the sequence and the roots $\alpha_1, \ldots, \alpha_L$ of $f(X)$, let's take the system of linear equations

$$\sum_{j=1}^{L} \alpha_j^t \beta_j = s_t \qquad \text{for all } t = 0, \ldots, L-1$$

in the unknown $\beta_1, \ldots, \beta_L$.

Clearly, the matrix V associated with the system is a Vandermonde matrix[1] and it's known that for such a matrix holds

$$detV = \prod_{1\leq i<j\leq L} (\alpha_j - \alpha_i)$$

By the hypothesis on the α_j, we get hence $detV \neq 0$, so the system admits a solution $\beta_1, \ldots, \beta_L$, whose elements are obviously in the splitting field of $f(X)$.

Now, we know that the elements of the sequence must satisfy

$$s_{t+L} = a_{L-1}s_{t+L-1} + a_{L-2}s_{t+L-2} + \cdots + a_0 s_t \quad \forall t \in \mathbb{N}$$

[1] A Vandermonde matrix, named after Alexandre-Thophile Vandermonde, is a matrix with the terms of a geometric progression in each row

If we substitute the 2.1.6, we get

$$\sum_{j=1}^{L} \beta_j \alpha_j^{t+L} - a_{L-1} \sum_{j=1}^{L} \beta_j \alpha_j^{t+L-1} - \cdots - a_0 \sum_{j=1}^{L} \beta_j \alpha_j^t = \sum_{j=1}^{L} \beta_j \alpha_j^t f(\alpha_j) = 0$$

for all $t \geq 0$, as wished. □

Note that, thanks to what we have seen until now, we know that the characteristic polynomial of a linear recurring sequence must always be chosen primitive to obtain the longest possible period.

Since a primitive polynomial is irreducible and has no multiple roots, Theorem 2.5 applies quite widely in practice.

Substantially, suppose we are handling a Lth order recurrence relation over $\mathbb{F}_q$ whose characteristic polynomial $f(X)$ is primitive. If we think of a linear sequence generated by $f(X)$ as something abstract, the first L elements $s_0, \ldots, s_{L-1}$ are linearly independent objects, while any other s_t, for $t \geq L$, is a linear combination of them. If we take the quotient group

$$\frac{\mathbb{F}_q[X]}{(f(X))} \cong \mathbb{F}_q[\alpha] \cong \mathbb{F}_{q^L}$$

where α is a generic root of $f(X)$, we can represent every element s_t of the sequence as a combination of elements of $\mathbb{F}_{q^L}$, since all the roots of $f(X)$ (and their powers) belong to that set. The coefficients of the combination depend only on the initialization of the sequence, once they are fixed every other element is, as expected, uniquely determined.

Example 2.5. Take the recurrence relation $s_{t+2} = s_{t+1} + s_t$ over $\mathbb{F}_2$, whose characteristic polynomial is $f(X) = X^2 - X - 1$. If $\alpha^2 - \alpha - 1 = 0$, $\mathbb{F}_4 = \mathbb{F}_2[\alpha] = \{0, 1, \alpha, \alpha + 1\}$, where α and $\alpha + 1$ are the roots of $f(X)$. Then, by Theorem 2.5, the generic element of a sequence generated by that recurrence relation is $s_t = \beta_1 \alpha^t + \beta_2 (\alpha + 1)^t$. Only the first two elements s_0 and s_1 are free and their value determines the value of the coefficients $\beta_1, \beta_2 \in \mathbb{F}_4$. Once β_1 and β_2 are fixed, every other element is uniquely determined.

For instance, if we take $s_0 = s_1 = 1$ we get the system

$$\begin{cases} \beta_1 + \beta_2 = 1 \\ \beta_1 \alpha + \beta_2 (\alpha + 1) = 1 \end{cases}$$

whose solution is $\beta_1 = \alpha, \beta_2 = \alpha + 1$. So the sequence can be expressed as $s_t = \alpha \alpha^t + (\alpha + 1)(\alpha + 1)^t = \alpha^{t+1} + (\alpha + 1)^{t+1}$ for all $t \geq 0$. Since clearly for every non-zero $\gamma \in \mathbb{F}_4$ we have $\gamma^3 = 1$, the sequence has the property

$$s_{t+3} = \alpha^{t+1+3} + (\alpha+1)^{t+1+3} = \alpha^{t+1}\alpha^3 + (\alpha+1)^{t+1}\alpha^3 = \alpha^{t+1} + (\alpha+1)^{t+1} = s_t$$

so it's trivially periodic with least period 3.

2.1.3 Linear complexity

A linear recurring sequence satisfies many other recurrence relations besides the one it is generated with. For example, if a sequence $\{s_t\}$ has least period $\tilde{r}$, so that $s_{t+\tilde{r}} = s_t$ for all $t \in \mathbb{N}$, then the same sequence obviously satisfies the relation $s_{t+h\tilde{r}} = s_t$ for every $h \in \mathbb{Z}$.

Clearly, every recurrence relation satisfied by a sequence corresponds to a different characteristic polynomial. To better describe the relation between those polynomials, we need to introduce a different approach to linear recurring sequences through the algebraic apparatus of formal power series.

Given an arbitrary sequence $\{s_t\}_{t\in\mathbb{N}}$ of elements of $\mathbb{F}_q$, we associate with it a purely formal expression called its *generating function*:

$$G(X) = s_0 + s_1 X + \cdots + s_t X^t + \cdots = \sum_{t=0}^{+\infty} s_t X^t \tag{2.1.7}$$

The underlying idea is that $G(X)$, storing all the terms of the sequence in the correct order, should somehow reflect the properties of the sequence. The name "generating function" must not be misunderstood: $G(X)$ is not a function and as a series we are not interested in its convergence, we think of it as being nothing but a hieroglyph for the sequence $\{s_t\}$.

Two such formal power series

$$B(X) = \sum_{t=0}^{+\infty} b_t X^t \qquad \text{and} \qquad C(X) = \sum_{t=0}^{+\infty} c_t X^t \tag{2.1.8}$$

over $\mathbb{F}_q$ are considered identical if and only if $b_t = c_t$ for all $t \geq 0$. The set of all formal power series over $\mathbb{F}_q$ is then in an obvious one-to-one correspondence with the set of all sequences of element of $\mathbb{F}_q$.

Hence, it may seem that we have not gained anything from the transition to formal power series, but actually we will get many interesting results thanks to the rich algebraic structure the set of all formal power series over $\mathbb{F}_q$ can be naturally endowed with.

First of all, let's note that a polynomial of finite degree L

$$p(X) = p_0 + p_1 X + \cdots + p_L X^L \ \in \mathbb{F}_q[X]$$

can be seen as a formal power series

$$p(X) = \sum_{t=0}^{+\infty} p_t X^t$$

where $p_t = 0$ for all $t > L$.

Now we can introduce the algebraic operations of addition and multiplication between formal power series as the extension of the analogous operations for polynomials.

Let $B(X)$ and $C(X)$ be defined as in 2.1.8, then we define their sum as

$$B(X) + C(X) = \sum_{t=0}^{+\infty} (b_t + c_t) X^t$$

and their product as

$$B(X)C(X) = \sum_{t=0}^{+\infty} d_t X^t \qquad \text{where } d_t = \sum_{j=0}^{t} b_j c_{t-j} \quad \text{for all } t \geq 0$$

Remark 2.5. Note that the substitution principle is not valid for formal power series, since the expression $B(a)$, where $a \in \mathbb{F}_q$ and $B(X)$ is a formal power series over $\mathbb{F}_q$, may be meaningless.

Example 2.6. Let

$$B(X) = 2 + X^2 \qquad \text{and} \qquad C(X) = \sum_{t=0}^{+\infty} X^t$$

be formal power series over $\mathbb{F}_3$. Then

$$B(X) + C(X) = 0 + X + 2X^2 + X^3 + \cdots + X^t + \cdots = \sum_{t=0}^{+\infty} d_t X^t$$

where $d_0 = 0$, $d_2 = 2$ and $d_t = 1$ for all $t \neq 0, 2$, and

$$B(X)C(X) = 2 + 2X + 0X^3 + \cdots + 0X^t + \cdots = 2 + 2X$$

Can be easily checked that addition of formal power series over $\mathbb{F}_q$ is associative and commutative, the series $0 = \sum_{t=0}^{+\infty} 0X^t$ is the identity element for addition and, given $B(X) = \sum_{t=0}^{+\infty} b_t X^t$, its additive inverse is $-B(X) = \sum_{t=0}^{+\infty} (-b_t) X^t$.

Analogously, multiplication is associative and commutative and the series $1 = 1 + \sum_{t=1}^{+\infty} 0X^t$ is the multiplicative identity. Furthermore, the distributive law is also satisfied.

Altogether, we have shown that the set of all formal power series over $\mathbb{F}_q$, furnished with this addition and multiplication, is a commutative ring with identity, called the *ring of formal power series* over $\mathbb{F}_q$ and denoted by $\mathbb{F}_q[[X]]$.

Theorem 2.6. *The ring $\mathbb{F}_q[[X]]$ of formal power series over $\mathbb{F}_q$ is an integral domain containing $\mathbb{F}_q[X]$ as a subring.*

Proof. It only remains to verify that $\mathbb{F}_q[[X]]$ has no zero-divisors. Suppose that there exists two non-zero elements $B(X)$ and $C(X)$ such that $B(X)C(X) = 0$. Let k and m be the least integers for which respectively $b_k \neq 0$ and $c_m \neq 0$. Then the coefficient of X^{k+m} in $B(X)C(X)$ is $b_k c_m \neq 0$, hence $B(X)C(X) \neq 0$. □

Now we want to identify those series $B(X) \in \mathbb{F}_q[[X]]$ that admit a multiplicative inverse. The following Theorem provides an easy characterization.

Theorem 2.7. *The formal power series*

$$B(X) = \sum_{t=0}^{+\infty} b_t X^t \ \in \mathbb{F}_q[[X]]$$

has a multiplicative inverse if and only if $b_0 \neq 0$.

Proof. Let

$$C(X) = \sum_{t=0}^{+\infty} c_t X^t \ \in \mathbb{F}_q[[X]]$$

be such that $B(X)C(X) = 1$, then the following infinite system of equation must be satisfied

$$\begin{aligned} b_0 c_0 &= 1 \\ b_0 c_1 + b_1 c_0 &= 0 \\ b_0 c_2 + b_1 c_1 + b_2 c_0 &= 0 \\ \vdots &\ \ \vdots \\ b_0 c_t + b_1 c_{t-1} + \cdots + b_t c_0 &= 0 \\ \vdots &\ \ \vdots \end{aligned}$$

If $b_0 = 0$ the system clearly admits no solutions. Otherwise, if $b_0 \neq 0$, from the first equation we know that $c_0 = b_0^{-1}$ is the multiplicative inverse of b_0 in $\mathbb{F}_q$. Analogously, from the second equation we get $c_1 = -b_0^{-1}(b_1 c_0)$ and recursively for all t from the tth equation we get

$$c_t = b_0^{-1} \sum_{j=1}^{t} b_j c_{t-j}$$

so we have constructed the multiplicative inverse $C(X)$ of $B(X)$ (and implicitly shown that the inverse in unique). □

Hereinafter, we will denote $\frac{1}{B(X)}$ the inverse of $B(X)$ and write $\frac{A(X)}{B(X)}$ instead of $A(X)\frac{1}{B(X)}$. The inverse of a series, or analogously the division

between two series, can be computed in the usual way. Simply in most cases the algorithm will be infinite.

Now let's show a basic identity for the generating function of a given sequence.

Theorem 2.8. *Let $\{s_t\}_{t\in\mathbb{N}}$ be a Lth-order homogeneous linear recurring sequence over $\mathbb{F}_q$ satisfying 2.1.2. Let $f^*(X) \in \mathbb{F}_q[X]$ be its feedback polynomial and $G(X) \in \mathbb{F}_q[[X]]$ be its generating function as defined in 2.1.7. Then the identity*

$$G(X) = \frac{g(X)}{f^*(X)} \tag{2.1.9}$$

holds with

$$g(X) = -\sum_{j=0}^{L-1}\sum_{i=0}^{j} a_{i+L-j}s_iX^j \ \in \mathbb{F}_q[X] \tag{2.1.10}$$

where we set $a_L = -1$.

Conversely, if $g(X)$ is any polynomial over $\mathbb{F}_q$ with $deg(g) < L$ and if $f^*(X) \in \mathbb{F}_q[X]$ is given by 2.1.4, then the formal power series $G(X) \in \mathbb{F}_q[[X]]$ defined by 2.1.9 is the generating function of a Lth-order homogeneous linear recurring sequence in $\mathbb{F}_q$ satisfying the linear recurrence relation 2.1.2.

Proof. First, we have

$$\begin{aligned} f^*(X)G(X) &= -\left(\sum_{t=0}^{L} a_{L-t}X^t\right)\left(\sum_{t=0}^{+\infty} s_tX^t\right) = \\ &= -\sum_{j=0}^{L-1}\left(\sum_{i=0}^{j} a_{i+L-j}s_i\right)X^j - \sum_{j=L}^{+\infty}\left(\sum_{i=j-L}^{j} a_{i+L-j}s_i\right)X^j = \\ &= g(X) - \sum_{j=L}^{+\infty}\left(\sum_{i=0}^{L} a_is_{j-L+i}\right)X^j \end{aligned} \tag{2.1.11}$$

Since $\{s_t\}$ satisfies 2.1.2, for every $j \geq L$ we can take $t = j - L$ in 2.1.2 obtaining

$$\sum_{i=0}^{L} a_is_{j-L+i} = 0$$

Thus 2.1.11 becomes

$$f^*(X)G(X) = g(X)$$

and, since $f^*(X)$ admits a multiplicative inverse in $\mathbb{F}_q[[X]]$, the identity 2.1.9 follows.

Conversely, from 2.1.11 we infer that $f^*(X)G(X)$ is equal to a polynomial of degree less than L if and only if

$$\sum_{i=0}^{L} a_i s_{j-L+i} = 0 \qquad \text{for all } j \geq L$$

But these identities just express the fact that the sequence $\{s_t\}$ of the coefficients of $G(X)$ satisfies the linear recurrence relation 2.1.2. □

We can summarize Theorem 2.8 by saying that the Lth-order homogeneous linear recurring sequences with feedback polynomial $f^*(X)$ are in one-to-one correspondence with the fractions $\frac{g(X)}{f^*(X)}$ with $deg(g) < L$, where every choice of the initial state of the sequence entails the choice of $g(x)$ as defined in 2.1.10. The identity 2.1.9 permits to compute the terms of the sequence with that initialization by long division.

Example 2.7. Consider the linear recurrence relation

$$s_{t+4} = s_{t+3} + s_{t+1} + s_t$$

over $\mathbb{F}_2$, whose feedback polynomial is

$$f^*(X) = 1 - X - X^3 - X^4 = 1 + X + X^3 + X^4 \ \in \mathbb{F}_2[X]$$

If the initial state is $(1, 1, 0, 1)$, we have $g(X) = 1 + X^2$ and by long division we compute

$$G(X) = \frac{1 + X^2}{1 + X + X^3 + X^4} = 1 + X + X^3 + X^4 + X^6 + \cdots$$

which corresponds to the sequence $\{1, 1, 0, 1, 1, 0, 1, \ldots\}$ of least period 3.

Otherwise, if we take the impulse response sequence, whose initial state is $(0, 0, 0, 1)$, we have $g(X) = X^3$ and

$$G(X) = \frac{X^3}{1 + X + X^3 + X^4} = X^3 + X^4 + X^5 + X^9 + X^{10} + X^{11} + \cdots$$

which corresponds to the sequence $\{0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, \ldots\}$ of least period 6.

Thanks to identity 2.1.9 we can give a proof of Theorem 2.3. Since the sequence $\{s_t\}$ is periodic with period r, its generating function can be written as

$$G(X) = (s_0 + s_1 X + \cdots + s_{r-1} X^{r-1})(1 + X^r + X^{2r} + \cdots) = \frac{s^*(X)}{1 - X^r}$$

where we called $s^*(X) = s_0 + s_1 X + \cdots + s_{r-1} X^{r-1}$ and we noted that $1 - X^r$ is the multiplicative inverse of $1 + X^r + X^{2r} + \cdots$.

Equating this expression with identity 2.1.9, we get

$$\frac{g(X)}{f^*(X)} = \frac{s^*(X)}{1-X^r}$$

or equivalently

$$s^*(X)f^*(X) = (1-X^r)g(X)$$

Now, if we recall the relation between characteristic and feedback polynomial of a sequence $\left(f^*(X) = X^L f\left(\frac{1}{X}\right)\right)$ and if we note that an analogous relation exists between $s^*(X)$ and the polynomial $s(X)$ defined in Theorem 2.3 $\left(s^*(X) = X^{r-1}s\left(\frac{1}{X}\right)\right)$, we get

$$s^*(X)f^*(X) = X^{r-1}s\left(\frac{1}{X}\right)X^L f\left(\frac{1}{X}\right) = (1-X^r)g(X)$$

which implies

$$\begin{aligned} s\left(\frac{1}{X}\right)f\left(\frac{1}{X}\right) &= \frac{1-X^r}{X^r}\frac{1}{X^{L-1}}g(X) = \left(\frac{1}{X^r}-1\right)\frac{1}{X^{L-1}}g(X) \\ \Leftrightarrow \quad s(X)f(X) &= (X^r-1)X^{L-1}g\left(\frac{1}{X}\right) \end{aligned}$$

and noting that, if $h(X)$ is defined as in Theorem 2.3,

$$X^{L-1}g\left(\frac{1}{X}\right) = -h(X) \tag{2.1.12}$$

leads to the desired conclusion.

The following Theorem finally describes the relation between the different linear recurrence relations valid for a given homogeneous linear recurring sequence.

Theorem 2.9. *Let $\{s_t\}_{t\in\mathbb{N}}$ be a linear recurring sequence over $\mathbb{F}_q$. Then, there exists a uniquely determined monic polynomial $m(X) \in \mathbb{F}_q[X]$ such that every other monic polynomial $f(X) \in \mathbb{F}_q[X]$ of positive degree is a characteristic polynomial of $\{s_t\}$ if and only if $m(X)$ divides $f(X)$.*

Proof. Let $f_0(X) \in \mathbb{F}_q[X]$ be the characteristic polynomial of a homogeneous linear recurrence relation satisfied by the sequence. We know that, if the sequence is periodic with period r, we can write $f_0(X)s(X) = (1-X^r)h_0(X)$, where the polynomials $s(X)$ and $h_0(X)$ are defined as in Theorem 2.3. Now, if $d(X)$ is the (monic) greatest common divisor of $f_0(X)$ and $h_0(X)$, we can write $f_0(X) = d(X)m(X)$ and $h_0(X) = b(X)d(X)$, with $m(X), b(X) \in \mathbb{F}_q[X]$. We shall prove that $m(X)$ is the desired polynomial.

Clearly $m(X)$ is monic and divides $f_0(X)$. Now, let $f(X)$ be any other characteristic polynomial of the sequence and $h(X)$ be the corresponding

polynomial in Theorem 2.3. By applying Theorem 2.8, we obtain that the ganerating function $G(X)$ of the sequence satisfies

$$G(X) = \frac{g_0(X)}{f_0^*(X)} = \frac{g(X)}{f^*(X)}$$

with $g_0(X)$ and $g(X)$ determined by 2.1.10. Therefore

$$g_0(X)f^*(X) = g(X)f_0^*(X)$$

and using 2.1.12 we get

$$\begin{aligned} h(X)f_0(X) &= -X^{deg(f)-1}g\left(\frac{1}{X}\right)X^{deg(f_0)}f_0^*\left(\frac{1}{X}\right) = \\ &= -X^{deg(f_0)-1}g_0\left(\frac{1}{X}\right)X^{deg(f)}f^*\left(\frac{1}{X}\right) = h_0(X)f(X) \end{aligned}$$

so now, dividing by $d(X)$ entails

$$h(X)m(X) = b(X)f(X)$$

Since, by definition of $d(X)$, $m(X)$ and $b(X)$ are relatively prime, then necessarily $m(X)$ divides $f(X)$.

Conversely, suppose that $f(X) \in \mathbb{F}_q[X]$ is a monic polynomial of positive degree and that it is divisible by $m(X)$, say $f(X) = c(X)m(X)$ with $c(X) \in \mathbb{F}_q[X]$. Passing to reciprocal polynomials, we have $f^*(X) = c^*(X)m^*(X)$. Furthermore, clearly $h_0(X)m(X) = b(X)f_0(X)$. Now, using 2.1.12 we get

$$\begin{aligned} g_0(X)m^*(X) &= -X^{deg(f_0)-1}h_0\left(\frac{1}{X}\right)X^{deg(m)}m\left(\frac{1}{X}\right) = \\ &= -X^{deg(m)-1}b\left(\frac{1}{X}\right)X^{deg(f_0)}f_0\left(\frac{1}{X}\right) = -X^{deg(m)-1}b\left(\frac{1}{X}\right)f_0^*(X) \end{aligned}$$

Note that, since $deg(h_0) < deg(f_0)$, then $deg(c) < deg(m)$, so the product $-X^{deg(m)-1}b\left(\frac{1}{X}\right)$ is a polynomial $a(X) \in \mathbb{F}_q[X]$. Then we can write

$$g_0(X)m^*(X) = a(X)f_0^*(X)$$

and multiplying for $c^*(X)$ we get

$$g_0(X)f^*(X) = a(X)f_0^*(X)c^*(X)$$

or analogously, dividing by $f_0^*(X)f^*(X)$ and recalling the definition of $G(X)$,

$$G(X) = \frac{g_0(X)}{f_0^*(X)} = \frac{a(X)c^*(X)}{f^*(X)}$$

Since

$$deg(a \cdot c^*) = deg(a) + deg(c^*) < deg(m) + deg(c) = deg(f)$$

the second part of Theorem 2.8 shows that $f(X)$ is a characteristic polynomial of the sequence.

Finally, by the way we have defined $m(X)$ it is clearly uniquely determined. □

Definition 2.8. The uniquely determined polynomial $m(X)$ of Theorem 2.9 is called the *minimal polynomial* of the sequence.

If $s_t = 0$ for all $t \geq 0$, the minimal polynomial is equal to the constant polynomial 1, while for any other homogeneous linear recurring sequence $m(X)$ is a monic polynomial with $deg(m) > 0$. Note that the minimal polynomial of a linear recurring sequence is the characteristic polynomial of the linear recurrence relation of least possible order satisfied by the sequence.

Example 2.8. Let's take the sequence in $\mathbb{F}_2$ defined by

$$s_{t+4} = s_{t+3} + s_{t+1} + s_t$$

and with initial state $(1, 1, 0, 1)$. To find its minimal polynomial, we proceed as in the proof of Theorem 2.9. In this case $f_0(X) = X^4 + X^3 + X + 1$ and one can easily verify that $h_0(X) = X^3 + X$. Their greatest common divisor is $d(X) = X^2 + 1$ and

$$m(X) = \frac{f_0(X)}{d(X)} = X^2 + X + 1$$

which corresponds to the linear recurrence relation

$$s_{t+2} = s_{t+1} + s_t$$

actually satisfied by the sequence.

Note that $ord(m) = 3$ is equal to the least period of the sequence. The following Proposition shows that this is true in general.

Proposition 2.8. Let $\{s_t\}_{t\in\mathbb{N}}$ be a linear recurring sequence over $\mathbb{F}_q$ and let $m(X)$ be its minimal polynomial. Then the least period $\tilde{r}$ of the sequence is equal to the order of $m(X)$: $\tilde{r} = ord(m)$.

Proof. Let t_0 be the preperiod of the sequence, then $s_{t+\tilde{r}} = s_t$ for all $t \geq t_0$ and the sequence satisfies the linear recurrence relation

$$s_{t+t_0+\tilde{r}} = s_{t+t_0} \qquad \text{for all } t \geq 0$$

Thus, according to Theorem 2.9, $m(X)$ divides $X^{t_0+\tilde{r}} - X^{t_0} = X^{t_0}(X^{\tilde{r}} - 1)$, so it must be of the form $m(X) = X^n g(X)$ where $n \leq t_0$ and $g(X)$ divides $X^{\tilde{r}} - 1$ and is such that $g(0) \neq 0$. It follows from the definition of the order of a polynomial that $ord(m) = ord(g) \mid \tilde{r}$, but by Theorem 2.2 we know that $\tilde{r} \mid ord(m)$, then necessarily $\tilde{r} = ord(m)$. □

Now we can find the least period of a linear recurring sequence without evaluating its terms, as shown in the following example. The method is particularly effective if a table of orders of polynomials is available.

Example 2.9. Let's take the sequence in $\mathbb{F}_2$ defined by

$$s_{t+5} = s_{t+1} + s_t$$

with initial state $(1, 1, 1, 0, 1)$ and proceed as in the proof of Theorem 2.9. In this case $f_0(X) = X^5 + X + 1$, $h_0(X) = X^4 + X^3 + X^2$ and their greatest common divisor is $d(X) = X^2 + X + 1$. Then the minimal polynomial of the sequence is

$$m(X) = \frac{f_0(X)}{d(X)} = X^3 + X^2 + 1$$

whose order is 7, which we know to coincide with the least period of the sequence.

Proposition 2.9. Let $f(X) \in \mathbb{F}_q[X]$ be monic and irreducible over $\mathbb{F}_q$ and let $\{s_t\}$ be a homogeneous linear recurring sequence in $\mathbb{F}_q$ not all of whose terms are 0. If $f(X)$ is a characteristic polynomial of the sequence, then it is its minimal polynomial.

Proof. It follows easily from 2.9. In fact, since the minimal polynomial $m(X)$ of the sequence divides $f(X)$, the irreducibility of $f(X)$ implies that either $m(X) = 1$ or $m(X) = f(X)$. But $m(X) = 1$ holds only for the sequence all of whose terms are 0, so necessarily $m(X) = f(X)$. □

Anyhow, a characteristic polynomial can be the minimal polynomial of a sequence even if it is not irreducible. The following Theorem gives a general criterion to verify it.

Theorem 2.10. *Let $\{s_t\}_{t\in\mathbb{N}}$ be a sequence in $\mathbb{F}_q$ satisfying a Lth-order homogeneous linear recurrence relation with characteristic polynomial $f(X) \in \mathbb{F}_q[X]$. Then $f(X)$ is the minimal polynomial of the sequence if and only if the state vectors $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{L-1}$ are linearly independent over $\mathbb{F}_q$.*

Proof. First, suppose $f(X)$ is the minimal polynomial of the sequence. If there existed coefficients $b_0, b_1, \ldots, b_{L-1} \in \mathbb{F}_q$ not all of which are 0 such that $b_0\mathbf{s}_0 + b_1\mathbf{s}_1 + \cdots + b_{L-1}\mathbf{s}_{L-1} = \mathbf{0}$, we could multiply from the right by powers of the matrix A associated with $f(X)$, obtaining

$$b_0\mathbf{s}_t + b_1\mathbf{s}_{t+1} + \cdots + b_{L-1}\mathbf{s}_{t+L-1} = \mathbf{0} \qquad \text{for all } t \geq 0$$

In particular we would have

$$b_0 s_t + b_1 s_{t+1} + \cdots + b_{L-1} s_{t+L-1} = 0 \qquad \text{for all } t \geq 0$$

which define a new linear recurrence relation satisfied by the sequence. Now, $b_j = 0$ for all $1 \leq j \leq L-1$ implies $s_t = 0$ for all $t \geq 0$, which contradicts the assumption that the minimal polynomial of the sequence is $f(X)$, whose degree is L. So, let $j \geq 1$ be the largest index with $b_j \neq 0$. Then the sequence clearly satisfies a jth-order homogeneous linear recurrence relation with $j < L$, which again contradicts the assumption that the minimal polynomial of the sequence is $f(X)$. Therefore, we have shown that $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{L-1}$ are linearly independent over $\mathbb{F}_q$.

Conversely, suppose that $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{L-1}$ are linearly independent over $\mathbb{F}_q$. Since $\mathbf{s}_0 \neq 0$, the minimal polynomial has positive degree. If $f(X)$ were not the minimal polynomial, the sequence would satisfy a recurrence relation of order m with $1 \leq m < L$. But this eventuality is impossible, since if

$$s_{t+m} = b_{m-1}s_{t+m-1} + \cdots + b_0 s_t \qquad \text{for all } t \geq 0$$

for some coefficients $b_0, \ldots, b_{m-1} \in \mathbb{F}_q$, then the same identity is true for the state vectors $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_m$, contradicting the assumption of linear independence. □

Corollary 1. Let $(\mathbf{d}, \mathbf{a})$ be an impulse response sequence over $\mathbb{F}_q$. Then its minimal polynomial coincides with the characteristic polynomial of the linear recurrence relation defined by the vector $\mathbf{a}$.

Proof. The results follows trivially from Theorem 2.10, since the required linear independence property is obviously satisfied by the initial states of an impulse response sequence. □

Example 2.10. Let's go back in $\mathbb{F}_5^2$ as in example 2.3.

This time let's take the recurrence relation $s_{t+2} = s_{t+1} + s_t$, whose characteristic polynomial is $f(X) = X^2 - X - 1$, which is reducible over $\mathbb{F}_5$, being $X^2 - X - 1 = (X-3)^2$. $\mathbb{F}_5^2$ is divided in 3 orbits: one obviously contains only $(0,0)$, while the remaining 24 elements form two orbits with 4 and 20 elements respectively. The minimal polynomial of the shortest orbit is the constant polynomial $m_0(X) = 1$. The longest orbit corresponds to the impulse response sequence and, as stated by corollary 1, its minimal polynomial is $f(X)$. The last orbit corresponds to the sequence $1, 3, 4, 2, 1, 3, \ldots$ and its minimal polynomial is $m(X) = X - 3$; in fact, as the reader can easily check, it satisfies the shorter linear recurrence relation $s_{t+1} = 3s_t$.

Finally, we give a definition which applies to every sequence in a finite field $\mathbb{F}_q$. It resumes somehow the characterization of a sequence given thus far, describing its pseudo-randomness and the difficulty of recreating it or predicting its next digit.

Definition 2.9. Let $\{s_t\}_{t\in\mathbb{N}}$ be a generic sequence over $\mathbb{F}_q$. We define its *linear complexity* as the order of the shortest linear recurrence relation

satisfied by the sequence, namely the degree of its minimal polynomial. If the sequence does not satisfy any linear recurrence relation, we say its linear complexity to be ∞, while the linear complexity of the all-zero sequence is 0.

2.2 Combining linear recurring sequences

Although we have seen how to generate linear recurring sequences with maximal linear complexity and period, that's not enough to build a strong stream cipher. If we used the output of a LFSR as a keystream, the cipher would be weak under many attacks, in the first place one mounted using Berlekamp-Massey's algorithm. That algorithm, invented by Elwyn Berlekamp in 1968, is in fact able to find the minimal polynomial of a linear recurring sequence in time $\mathcal{O}(\Lambda)$ knowing only 2Λ consecutive digits, where Λ is the linear complexity of the sequence.

Furthermore, a linear recurring sequence couldn't be directly used as a keystream anyway, since, even if it has interesting randomicity properties, it also has very strong linearity properties. Combining it directly to the plaintext wouldn't supply any security against the so-called *algebraic attacks*, that recover the key of a cipher by formulating and solving a system of linear equations.

A known-plaintext attack[2], in particular, would be very simple to implement and very effective. According to Kerckhoffs' principle, we can suppose the attacker to know the structure of the LFSR, namely its length L and feedback relation $s_{t+L} = \sum_{i=0}^{L-1} a_i s_{t+i}$. His aim is hence finding the initial state of the register, which is actually the key of the cipher. If he knows any L plaintext digits $m_{t_1}, \ldots, m_{t_L}$ and the corresponding ciphertext digits, he can formulate the trivial system

$$\begin{cases} c_{t_j} = s_{t_j} + m_{t_j} \\ j = 1, \ldots, L \end{cases}$$

Recalling the recurrence relation of the sequence $\{s_t\}$, each digit s_{t_j} can be written as a linear combination of the key, so the system becomes a system of L linear equations in the L unknowns $s_0, \ldots, s_{L-1}$, solvable in $\mathcal{O}(L^3)$ operations with Gauss' algorithm.

Many other very incisive similar attacks could be moved, so we need to shatter the linearity of the sequence and make it unexploitable by the cryptanalysts.

The natural solution to these two problems, increasing the linear complexity of the sequence and shattering its linearity, consists in combining

[2] A known-plaintext attack is an attack mounted knowing some couples of plaintext and corresponding ciphertext

n registers in parallel through a non-linear function $f : \mathbb{F}_q^n \to \mathbb{F}_q$, called *combining function*.

This approach further strengthens the cipher involving a considerable increase of the keyspace, since the key of such a cipher is no longer the initialization of a register, but those of n different registers.

The scheme of this kind of ciphers is shown in figure 4.1.

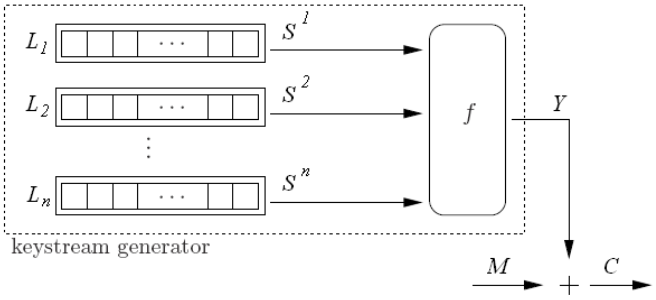


Figure 2.2: A typical LFSR combination generator

$S^i = \{s_t^i\}_{t\in\mathbb{N}}$ denotes the output of the ith register, while the output of f is the keystream $Y = \{y_t\}_{t\in\mathbb{N}}$, which is usually added digit by digit with the plaintext $M = \{m_t\}_{t\in\mathbb{N}}$ to generate the ciphertext $C = \{c_t\}_{t\in\mathbb{N}}$.

In the previous section we studied how the choice of a LFSR and its initialization affects the properties of the linear recurring sequence generated by it. We are now going to do the same work for the sequence Y: detect its properties and comprehend how they depend on the LFSRs and on the combining function f.

Even if Y is not directly generated by a LFSR, we can still talk of its period and linear complexity, since these two notions are defined for any sequence in a finite field $\mathbb{F}_q$. Furthermore, if we call N the keystream's total length, we can always imagine Y to be generated by a register whose characteristic polynomial is $g(X) = X^N - 1$ and thus take N as an upper bound for both its period and its linear complexity. What we want to do is to actually ensure that it does not exist a polynomial of lower degree that could have generated Y.

2.2.1 Multivariate functions over finite fields

When talking about multivariate functions over a finite field $\mathbb{F}_q$, first of all we need the to introduce a fundamental instrument.

Definition 2.10. Let $f : \mathbb{F}_q^n \to \mathbb{F}_q$ be any function in n unknowns over $\mathbb{F}_q$. f can always be represented as a multivariate polynomial called its *Algebraic*

Normal Form. The ANF of f is the only polynomial

$$Q_f \in \frac{\mathbb{F}_q[X_1, \ldots, X_n]}{(X_1^q - X_1, \ldots, X_n^q - X_n)}$$

such that

$$f(x) = Q_f(x) \qquad \text{for all } x = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$$

The degree of f is then defined as the total degree of the polynomial Q_f: $deg(f) = deg(Q_f)$.

This representation, which could seem not so natural at first sight, is actually very simple. It just exploits the fact that every function $f : \mathbb{F}_q^n \to \mathbb{F}_q$, assuming only q values, must be formulable as a polynomial and, since $X^q \equiv X$ in $\mathbb{F}_q$, every variable can appear in the polynomial form of f only with degree not greater than $q - 1$. Hence we can write the ANF of f as

$$Q_f = \sum_{u \in \mathbb{F}_q^n} a_u X^u$$

where we used the notation $X^u = X_1^{u_1} \cdots X_n^{u_n}$ and we assumed that $0^0 = 1$.

From here on, we will automatically suppose any function to be expressed through its ANF.

Thanks to the polynomial representation of the combining function, we can see the generic keystream digit y_t as the result of a combination of sums and products on the corresponding digits s_t^i. This allows us to describe Y just defining the operations of sum and product of two linear recurring sequences and their properties.

Boolean functions

When $q = 2$, the functions from the set $\mathbb{F}_2^n$ of n-bit words into $\mathbb{F}_2$ are called *boolean functions*.

In practical applications, this case is obviously the most frequent one. Boolean functions have been thus deeply analyzed in literature and the particularly simple structure of $\mathbb{F}_2$ permitted a refined characterization of those functions.

Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be any boolean function, we usually represent f through its ANF:

$$f(X) = \sum_{u \in \mathbb{F}_2^n} a_u X^u \tag{2.2.1}$$

Note that this time every variable X_i can appear only with degree 1.

Let $u = (u_1, \ldots, u_n) \in \mathbb{F}_2^n$ be any vector, we call its *support*, denoted $supp(u)$, the set of its non-zero components. $supp(u)$ tells us which variables appear in the corresponding monomial X^u. If $u = (01101) \in \mathbb{F}_2^5$, for example, then $supp(u) = \{2, 3, 5\}$ and $X^u = X_2 X_3 X_5$.

If we divide $f(X)$ by any monomial X^u, we obtain

$$f(X) = X^u(a_u + p_u(X)) + r_u(X) = a_u X^u + X^u p_u(X) + r_u(X)$$

where $p_u(X)$ is a polynomial with free term 0.

A term X^v is divisible by X^u if and only if $supp(u) \subseteq supp(v)$, namely if and only if $X^v = X^u \cdot X_{i_1} \cdots X_{i_k}$ for some indices $i_1, \ldots, i_k \in \{1, \ldots, n\} \setminus supp(u)$.

Hence any monomial of $p_u(X)$ can't contain any variable appearing in X^u, while a monomial in $r_u(X)$ can contain some of the variables in $supp(u)$, but obviously there must be at least one it does not contain.

Proposition 2.10. Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be any boolean function expressed in its ANF 2.2.1. It holds

$$a_u = \bigoplus_{x \preceq u} f(x) \qquad \text{for all } u \in \mathbb{F}_2^n \tag{2.2.2}$$

where $\preceq$ denotes the partial order on $\mathbb{F}_2^n$ defined by "$v \preceq u$ if and only if $v_i \leq u_i$ for all $i = 1, \ldots, n$". Namely, $v \preceq u$ if and only if $supp(v) \subseteq supp(u)$.

Proof. Let x be any vector such that $x \preceq u$.

- Since $supp(x) \subseteq supp(u)$, any 1 in x is in a position corresponding to a variable actually appearing in X^u, hence a variable not appearing in any monomial of $p_u(X)$. This means clearly that $p_u(x) = 0$ and consequently $\bigoplus_{x \preceq u} x^u p_u(x) = \bigoplus_{x \preceq u} 0 = 0$.
- Take any monomial in $r_u(X)$, say X^v. We have shown that necessarily $supp(u) \not\subseteq supp(v)$, so let $i \in supp(u) \setminus supp(v)$ and hence X_i be any variable appearing in X^u but not in X^v. Since $i \in supp(u)$, if we take the vector x' obtained by x flipping its i-th bit, clearly also $x' \preceq u$. But, since $i \notin supp(v)$, clearly the monomial X^v takes the same value when evaluated on x and x'. More generally, every monomial in $r_u(X)$ takes the same value on an even number of vectors $x \preceq u$, thus $\bigoplus_{x \preceq u} r_u(x) = 0$.
- Finally, X^u is worth 1 if and only if it is evaluated on a vector x such that $supp(u) \subseteq supp(x)$, but the only vector $x \preceq u$ with this property is u itself. Hence, $\bigoplus_{x \preceq u} x^u = 1$.

Now trivially

$$\begin{aligned} \bigoplus_{x \preceq u} f(x) &= \bigoplus_{x \preceq u} (a_u x^u + x^u p_u(x) + r_u(x)) = \\ &= a_u \bigoplus_{x \preceq u} x^u + \bigoplus_{x \preceq u} x^u p_u(x) + \bigoplus_{x \preceq u} r_u(x) = a_u \end{aligned}$$

□

Example 2.11. Take the function

$$f(X_1, X_2, X_3, X_4) = X_2 + X_2X_3 + X_1X_2X_3 + X_2X_3X_4 + 1$$

If $u = (0110)$, $X^u = X_2X_3$ and obviously $a_u = 1$. We can easily verify the Proposition:

$$\begin{aligned} \bigoplus_{x \preceq u} f(x) &= f(0000) + f(0100) + f(0010) + f(0110) = \\ &= 1 + 0 + 1 + 1 = 1 \end{aligned}$$

Furthermore, let's practically see what we have said in the proof:

$$\begin{aligned} f(X) &= X_2X_3(1 + X_1 + X_4) + 1 + X_2 = \\ &= X_2X_3 + X_2X_3(X_1 + X_4) + (1 + X_2) \end{aligned}$$

and

$$\begin{aligned} \bigoplus_{x \preceq u} x_2x_3(x_1 + x_4) &= 0 + 0 + 0 + 0 = 0 \\ \bigoplus_{x \preceq u} (1 + x_2) &= 1 + 0 + 1 + 0 = 0 \\ \bigoplus_{x \preceq u} x_2x_3 &= 0 + 0 + 0 + 1 = 1 \end{aligned}$$

Another way to describe a boolean function is to identify it with its truth table. The sets of functions from $\mathbb{F}_2^n$ into $\mathbb{F}_2$ is in fact in one-to-one correspondence with the set $\mathbb{F}_2^{2^n}$.

The correspondence is built by simply associating to any function f its truth table, namely the ordered sequence of the values taken by f on the elements of $\mathbb{F}_2^n$. The order in the sequence is induced by the order chosen for the set $\mathbb{F}_2^n$, such as the natural one obtained considering any $x \in \mathbb{F}_2^n$ as the binary representation of a number between 0 and $2^n - 1$.

According to the previous definition, $supp(f)$ should denote the set of the non-zero components in f's truth table. Anyhow, it is usually used in the classical meaning of the set $\{x \in \mathbb{F}_2^n : \ f(x) \neq 0\}$. Substantially, the two definitions agree, since any 1 in f's truth table correspond to a vector x such that $f(x) \neq 0$.

Thanks to this different perspective, we can extend to boolean functions the notions of Hamming weight and distance, defined for vectors in $\mathbb{F}_2^n$ for all n.

Definition 2.11. Let $u, v \in \mathbb{F}_2^n$ be two vectors. The *Hamming weight* of u, denoted $w_H(u)$, is defined as the number of its non-zero components, namely the cardinality of $supp(u)$:

$$w_H(u) = |\{i : \ u_i = 1\}|$$

The *Hamming distance* between u and v, denoted $d_H(u, v)$, is the number of components in which the two vectors differ, namely the Hamming weight of their bitwise modular sum:

$$d_H(u, v) = |\{i : u_i \neq v_i\}| = |\{i : u_i \oplus v_i = 1\}|$$

Now, given two functions f and g, we can naturally talk of $w_H(f)$ and $d_H(f, g)$, implicitly referring to their truth tables. It holds

$$d_H(f, g) = \sum_{x \in \mathbb{F}_2^n} (f + g)$$

A very useful instrument to study boolean functions from this point of view is the Walsh-Hadamard transform.

Definition 2.12. Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be any boolean function. We define its *Walsh-Hadamard transform* as the Fourier transform of the corresponding sign function, $\varphi_f : x \mapsto (-1)^{f(x)}$, with respect to the representation $\psi_u : x \mapsto (-1)^{u \cdot x}$:

$$\text{for all } u \in \mathbb{F}_2^n \qquad \hat{\chi}_f(u) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)} (-1)^{u \cdot x} \tag{2.2.3}$$

We recall that any function g of the form $g(x) = u \cdot x + \epsilon$, where $u \in \mathbb{F}_2^n$ and $\epsilon \in \mathbb{F}_2$, is called *affine*.

The *Fourier transform* of a function $\varphi : \mathbb{F}_2^n \to \mathbb{C}$ at a representation $\psi : \mathbb{F}_2^n \to \mathbb{C}$ is defined as

$$\hat{\varphi}(\psi) = \sum_{x \in \mathbb{F}_2^n} \varphi(x) \bar{\psi}(x) = \langle \varphi, \psi \rangle$$

The idea is to build a new instrument to compare $f(x)$ with the affine functions $u \cdot x$ for all $u \in \mathbb{F}_2^n$, which does not work anymore in modular arithmetic. Hence we need a function that can take a number of different values equal to the number of possible values of $f(x) - u \cdot x$.

The solution in the binary case is to take the primitive square root of unity, -1, and raise it to the difference between $f(x)$ and $u \cdot x$. This time the obtained function is particularly useful because it takes values in $\mathbb{R}$.

If we fix $u \in \mathbb{F}_2^n$, the generic term of the sum in 2.2.3 is

$$(-1)^{f(x)+u \cdot x} = \begin{cases} +1 & \text{if } f(x) = u \cdot x \\ -1 & \text{if } f(x) \neq u \cdot x \end{cases}$$

that can be written as $1 - 2(f(x) + u \cdot x)$. It follows that $\hat{\chi}_f(u)$ varies between 2^n, when $f(x) = u \cdot x$ for all $x \in \mathbb{F}_2^n$, and -2^n, when $f(x) = u \cdot x + 1$ for all

$x \in \mathbb{F}_2^n$, and it's 0 if $f(x)$ and $u \cdot x$ coincide on exactly half of the possible inputs.

It is then clear that $\hat{\chi}_f(u)$ estimates the Hamming distance between f and the affine functions $u \cdot x + \epsilon$. More precisely

$$\begin{aligned}
\hat{\chi}_f(u) &= \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+u\cdot x} \\
&= (-1)^{\epsilon} \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+u\cdot x+\epsilon} \\
&= (-1)^{\epsilon} \sum_{x \in \mathbb{F}_2^n} (1 - 2(f(x) + u \cdot x + \epsilon)) \\
&= (-1)^{\epsilon} \left(2^n - 2 \sum_{x \in \mathbb{F}_2^n} (f(x) + u \cdot x + \epsilon) \right) \\
&= (-1)^{\epsilon} (2^n - 2 d_H(f, u \cdot x + \epsilon))
\end{aligned}$$

hence

$$d_H(f, u \cdot x + \epsilon) = 2^{n-1} - \frac{(-1)^{\epsilon}}{2} \hat{\chi}_f(u) \tag{2.2.4}$$

The Walsh spectrum of a boolean function completely characterize it, as shown by the following Proposition.

Proposition 2.11. Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be any boolean function described through its ANF as in 2.2.1.

Hence, for all $u \in \mathbb{F}_2^n$ it holds

$$a_u = 2^{w_H(u)-1} \left(1 - \frac{1}{2^n} \sum_{v \preceq \bar{u}} \hat{\chi}_f(u) \right) \mod 2 \tag{2.2.5}$$

where $\bar{u}$ denotes the bitwise completion of u, i.e. $\bar{u}_i = 1$ if and only if $u_i = 0$.

Proof. Thanks to equation 2.2.2 we get

$$\begin{aligned}
a_u &= \sum_{x \preceq u} f(x) \mod 2 = \sum_{x \preceq u} \frac{1}{2}(1 - (-1)^{f(x)}) \mod 2 \\
&= \frac{1}{2} |\{x \preceq u\}| - \frac{1}{2} \sum_{x \preceq u} (-1)^{f(x)} \mod 2 \\
&= 2^{w_H(u)-1} - \frac{1}{2} \sum_{x \preceq u} (-1)^{f(x)} \mod 2
\end{aligned}$$

Since the normalized Fourier transform is involutive, it holds

$$\text{for all } x \in \mathbb{F}_2^n, \quad (-1)^{f(x)} = 2^{-n} \sum_{v \in \mathbb{F}_2^n} \hat{\chi}_f(v)(-1)^{v \cdot x}$$

Now, combining these two results, we have

$$\begin{aligned} a_u &= 2^{w_H(u)-1} - 2^{-n-1} \sum_{x \preceq u} \sum_{v \in \mathbb{F}_2^n} \hat{\chi}_f(v)(-1)^{v \cdot x} \mod 2 \\ &= 2^{w_H(u)-1} - 2^{-n-1} \sum_{v \in \mathbb{F}_2^n} \hat{\chi}_f(v) \left(\sum_{x \preceq u} (-1)^{v \cdot x} \right) \mod 2 \end{aligned}$$

The set $E_u = \{x \in \mathbb{F}_2^n : \ x \preceq u\}$ is a linear subspace of $\mathbb{F}_2^n$ of dimension $w_H(u)$. Its orthogonal $E_u^\perp$ clearly satisfies $E_u^\perp = E_{\bar{u}}$, hence

$$\sum_{x \preceq u} (-1)^{v \cdot x} = \begin{cases} 2^{w_H(u)} & \text{if } v \in E_{\bar{u}} \\ 0 & \text{otherwise} \end{cases}$$

In fact, if $v \notin E_{\bar{u}}$, there must exist an index i such that $v_i = u_i = 1$. Hence, for any $x \in E_u$, the vector $x' \in E_u$ obtained flipping the ith bit of x satisfies $v \cdot x \neq v \cdot x'$ and consequently $(-1)^{v \cdot x} + (-1)^{v \cdot x'} = 0$. Now finally

$$a_u = 2^{w_H(u)-1} - 2^{-n-1+w_H(u)} \sum_{v \preceq \bar{u}} \hat{\chi}_f(v) \mod 2$$

□

In this context, boolean functions are used in keystream generators to combine the outputs of some LFSR, thus we want them to have some properties essential to resist to some known attacks.

Since the main purpose is shattering the linearity of linear recurring sequences, we want the combining function to be definitely *non-linear*, namely to be as "far" as possible to any affine function.

At first sight, one could imagine to simply look at the total degree of a function, since the higher is $deg(f)$ the most f is supposed to act differently from any linear combination. If $deg(f)$ is high, in fact, some monomials in f depends on many of its input at the same time, making apparently difficult any algebraic attack.

This is a proper remark, but what we actually need most is to be sure that, given any linear combination $g(x) = u \cdot x$, f's truth table does not coincide too much neither with g or with $g + 1$.

Suppose, in fact, that f coincides with g on 90% of the possible inputs. Many statistical attacks could be mounted by simply supposing that the combining function used is actually g, but that with probability $\frac{1}{10}$ the ciphertext bit we have at our disposal is wrong.

The best way to measure how far a function f is from being linear is hence to define its *nonlinearity* as its minimum distance from any affine function:

$$nl(f) := \min_{g(x)=u \cdot x+\epsilon} d_H(f, g)$$

Thanks to identity 2.2.4, we can alternatively define it as

$$nl(f) = 2^{n-1} - \frac{1}{2} \max_{u \in \mathbb{F}_2^n} |\hat{\chi}_f(u)| \tag{2.2.6}$$

Note that, if f and g are two functions from $\mathbb{F}_2^n$ into $\mathbb{F}_2$, $d_H(f,g) = 2^n - d_H(f,g+1)$, hence evaluating $d_H(f,g)$ or $d_H(f,g+1)$ is equivalent and only one of the two calculation is necessary.

When g is constant, say $g(x) \equiv 0$ (but $g(x) \equiv 1$ would be clearly equivalent), we talk of a more particular and basic property of boolean functions, called *balance*.

A function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is balanced if its output is uniformly distributed, namely if $P(f = 0) = P(f = 1) = \frac{1}{2}$. The probability is usually simply intended as

$$P(f = \epsilon) = \frac{|\{x \in \mathbb{F}_2^n : \ f(x) = \epsilon\}|}{2^n}$$

hence f is balanced if and only if $w_H(f) = 2^{n-1}$.

The importance of this property is evident thanks to Shannon's entropy, as shown in the example.

Example 2.12. Suppose a combining function f is unbalanced, say $p_0 = P(f = 0) = \frac{1}{4}$ and $p_1 = P(f = 1) = \frac{3}{4}$. The entropy of the keystream Y is

$$\begin{aligned} H(Y) &= p_0 \cdot \log_2 \frac{1}{p_0} + p_1 \cdot \log_2 \frac{1}{p_1} \\ &= \frac{1}{4} \cdot \log_2 4 + \frac{3}{4} \cdot \log_2 \frac{4}{3} \\ &= \frac{1}{2} + \frac{3}{2} - \frac{3}{4} \cdot \log_2 3 = 2 - \frac{3}{4} \cdot \log_2 3 > 1 \end{aligned}$$

hence the generic keystream bit y_t carries more than a bit of information.

The value taken by y_t depends on the output at time t of the n LFSRs and allows to understand if the n-tuple belongs to a set of $\frac{2^n}{4}$ elements or is one of the remaining $\frac{3 \cdot 2^n}{4}$. That's why the amount of information carried by y_t is different from the one carried by a random bit. In particular, the result highlights that the gain in information when $y_t = 0$ is bigger than the loss when $y_t = 1$.

Note that, any time the plaintext is unbalanced, the same property is transmitted to the ciphertext, and that's obviously a problem.

The balance property can be easily expressed through the Walsh coefficients, since clearly a function f is balanced if and only if $\hat{\chi}_f(0) = 0$.

The property of a function to be statistically independent from linear combinations of its inputs was precisely defined by Siegenthaler in [16].

Definition 2.13. A boolean function f is *t-th order correlation-immune* if the probability distribution of its output is unaltered when the values of any t input variables are fixed.

The definition equivalently asserts that the output of f is statistically independent from any linear combination of t of its input variables. As any other property, the correlation immunity order of a function can be characterized by its Walsh spectrum. In fact

$$\begin{aligned} P(f(x) = u \cdot x) &= \frac{|\{x \in \mathbb{F}_2^n : \ f(x) = u \cdot x\}|}{2^n} \\ &= \frac{2^n - d_H(f, u \cdot x)}{2^n} \\ &= \frac{2^n - 2^{n-1} + 2^{-1}\hat{\chi}_f(u)}{2^n} \\ &= \frac{2^{n-1} + 2^{-1}\hat{\chi}_f(u)}{2^n} \\ &= \frac{1}{2} + \frac{\hat{\chi}_f(u)}{2^{n+1}} \end{aligned}$$

namely

$$P(f(x) = u \cdot x) = \frac{1}{2}\left(1 + \frac{\hat{\chi}_f(u)}{2^n}\right) \tag{2.2.7}$$

The following Proposition is now trivial.

Proposition 2.12. Let f be any boolean function. f is t-th order correlation-immune if and only if

$$\hat{\chi}_f(u) = 0 \quad \text{for all } u \in \mathbb{F}_2^n \text{ such that } 1 \leq w_H(u) \leq t$$

Proof. f is t-th order correlation-immune if and only if $P(f(x) = u \cdot x) = \frac{1}{2}$ for all $u \in \mathbb{F}_2^n$ such that $1 \leq w_H(u) \leq t$. Thus the conclusion follows easily from equation 2.2.7. □

Since any t-th order correlation-immune function is also k-th order correlation-immune for any $k \leq t$, we call correlation-immunity order of a function f the largest t such that f is t-th order correlation-immune.

A function f is finally called *t-resilient* when f is balanced and its correlation-immunity order is t.

Expressing the cryptographic properties of a boolean function through the properties of its Walsh spectrum highlights the necessity of a trade-off among them.

Parseval's Theorem, which applies to any Fourier transform, states that

$$\sum_{x \in \mathbb{F}_2^n} \left((-1)^{f(x)}\right)^2 = \frac{1}{|\mathbb{F}_2^n|} \sum_{u \in \mathbb{F}_2^n} \left(\hat{\chi}_f(u)\right)^2$$

which means

$$\sum_{u \in \mathbb{F}_2^n} \left(\hat{\chi}_f(u)\right)^2 = 2^n \sum_{x \in \mathbb{F}_2^n} 1 = 2^{2n}$$

Hence the Walsh coefficients of a function f can't be all 0 and the bigger is the number of null ones, the bigger the non-null ones are in absolute value. Consequently, the higher is the correlation-immunity order of f, the bigger $max_{u \in \mathbb{F}_2^n} |\hat{\chi}_f(u)|$ is supposed to be, hence the lower may be f's nonlinearity.

Furthermore, the following (quite trivial) results holds [16]:

- the correlation-immunity order of a function f with n variables is clearly bounded by $n-1$ and if f's correlation-immunity order is t, then $deg(f) \leq n-t$
- if f is t-resilient, then
  - if $t \leq n-2$, it holds $deg(f) \leq n-t-1$
  - if $t = n-1$, it holds $deg(f) = 1$

Finally, if we want the cipher to be fairly strong under any type of attack, the best we can aspire to is hence a combining function with correlation-immunity order t not too low and such that $|\hat{\chi}_f(u)| = c$ for all u with $w_H(u) > t$, where c is a constant.

Let's now come back to the general case of any finite field $\mathbb{F}_q$.

The functions from $\mathbb{F}_q^n$ into $\mathbb{F}_q$ and their properties have not been deeply analyzed in literature, since the q-nary case is probably considered more interesting in theory than practice.

Anyhow, the notions of nonlinearity and correlation-immunity order can be extended to any function $f : \mathbb{F}_q^n \to \mathbb{F}_q$ and they have the identical meaning.

The main difference with the binary case is that Proposition 2.10 can't be extended, since that result follows from the fact that a variable can appear in the ANF of a boolean function only with degree 1.

Furthermore, the main instrument that allowed us to characterize the boolean functions having a particular property and to understand which properties are compatible is the Walsh spectrum of the function. Unfortunately, it can be defined only for the finite fields $\mathbb{Z}_p$ of prime cardinality and anyway many of its properties are not valid anymore.

The idea of the Walsh coefficients is somehow to transform an additive group in a multiplicative one. If $f : \mathbb{F}_q^n \to \mathbb{F}_q$ and $u \in \mathbb{F}_q^n$, the difference $f(x) - u \cdot x$ can take q different values. To build a complex-valued function which can take those q values, we should take the q-th root of unity $\omega = e^{\frac{2\pi i}{q}} \in \mathbb{C}$ and elevate it to the integer associated to $f(x) - u \cdot x$. But $\omega^{f(x)-u\cdot x}$ corresponds to the product $\omega^{f(x)} \cdot \omega^{-u\cdot x}$ only if $\mathbb{F}_q$ is a cyclic group with the sum and this is true if and only if $q = p$ is prime.

This means that the Walsh-Hadamard transform of f can be usefully defined only for $\mathbb{Z}_p$ as the Fourier transform of $\varphi_f : x \mapsto \omega^{f(x)}$ at the

representation $\psi_u : x \mapsto \omega^{u \cdot x}$, namely as

$$\hat{\chi}_f(u) = \sum_{x \in \mathbb{Z}_p^n} \omega^{f(x)} \bar{\omega}^{u \cdot x} = \sum_{x \in \mathbb{Z}_p^n} \omega^{f(x) - u \cdot x}$$

where we have used $\bar{\omega} = e^{-\frac{2\pi i}{q}} = \omega^{-1}$.

Unlike the binary case, this function is actually complex-valued, hence, even if its value is strictly correlated to the relative behavior of $f(x)$ and $u \cdot x$, identities equivalent to 2.2.4, 2.2.5 and 2.2.6 cannot be stated.

These remarks, together with the complicated structure of $\mathbb{F}_q$ compared to $\mathbb{F}_2$, will translate in the impossibility to recover the combining function and in a more difficult recovery of the LFSRs in the general q-nary case.

2.2.2 Families of linear recurring sequences

Let $P(x) \in \mathbb{F}_q[X]$ be any polynomial over $\mathbb{F}_q$. We recall that $\mathcal{S}(P)$ denotes the set of all the sequences $S = \{s_t\}_{t \in \mathbb{N}}$ such that P is a characteristic polynomial of S.

We may consider on $\mathcal{S}(P)$ the operations of sum and scalar multiplication defined termwise: if V is the sequence $\{v_t\}_{t \in \mathbb{N}}$, W is the sequence $\{w_t\}_{t \in \mathbb{N}}$ and $c \in \mathbb{F}_q$, then

- $(V + W) = \{v_0 + w_0, v_1 + w_1, \ldots\} = \{v_t + w_t\}_{t \in \mathbb{N}}$
- $(cV) = \{cv_0, cv_1, \ldots\} = \{cv_t\}_{t \in \mathbb{N}}$

$\mathcal{S}(P)$ is clearly closed under these operations and is indeed a vector space over $\mathbb{F}_q$, where the role of the zero vector is played by the *all-zero* sequence, all of whose terms are 0. Since $\mathcal{S}(P)$ has q^L elements, each one produced with one of the q^L possible initializations of the register, the dimension of the vector space is L. Thanks to what we have seen in section 2.1.3, Theorem 2.9 in particular, the following result holds:

Theorem 2.11. *Let P and Q be two non constant monic polynomials over $\mathbb{F}_q$. Then $\mathcal{S}(P)$ is a subset of $S(Q)$ if and only if P divides Q.*

Now, given $P(X), Q(X) \in \mathbb{F}_q[X]$, we denote

$$\mathcal{S}(P) + \mathcal{S}(Q) = \{(V + W) : V \in \mathcal{S}(P), W \in \mathcal{S}(Q)\}$$

$$\mathcal{S}(P) \cdot \mathcal{S}(Q) = \{(V \cdot W) : V \in \mathcal{S}(P), W \in \mathcal{S}(Q)\}$$

the set of all the sequences obtained respectively by termwise addition and multiplication of two sequences in $\mathcal{S}(P)$ and $\mathcal{S}(Q)$.

The following two Theorems are fundamental, since they show that the elements of these two sets are linear recurring sequences as well and they describe us their characteristic polynomials. We will not prove them, the interested reader can find the proof and a detailed discussion of the argument in [8], [10] or [13].

Theorem 2.12. *Let $P_1, \ldots, P_n$ be n non constant monic polynomials over $\mathbb{F}_q$. Then*

$$\mathcal{S}(P_1) + \cdots + \mathcal{S}(P_n) = \mathcal{S}(Q)$$

where Q is the (monic) least common multiple of $P_1, \ldots, P_n$.

Theorem 2.13. *Let $P_1, \ldots, P_n$ be n non constant monic polynomials over $\mathbb{F}_q$. Then there exists a non constant monic polynomial Q such that*

$$\mathcal{S}(P_1) \cdots \mathcal{S}(P_n) = \mathcal{S}(Q) \quad \textit{and} \quad deg(Q) \leq \prod_{i=1}^{n} deg(P_i)$$

Furthermore, if $P_1, \ldots, P_n$ have coprime orders, then $deg(Q) = \prod_{i=1}^{n} deg(P_i)$.

Remark 2.6. Note that, given P and Q, especially if they have coprime orders, the degree of the polynomial generating the product of $\mathcal{S}(P)$ and $\mathcal{S}(Q)$ is much higher than the degree of the polynomial generating their sum. This fact will be very important later, since the degree will permit us to distinguish between the two cases (sum or product).

From here on, let $\Lambda(S)$ denote the linear complexity of a sequence S and $\rho(S)$ its period.

The following Proposition follows trivially from Theorems 2.12 and 2.13.

Proposition 2.13. Let $V = \{v_t\}_{t\in\mathbb{N}}$ and $W = \{w_t\}_{t\in\mathbb{N}}$ be two linear recurring sequences whose minimal feedback polynomial are P_0 and Q_0 respectively. Then

- their sum verifies

$$\begin{aligned} \Lambda(V+W) &\leq \Lambda(V) + \Lambda(W) \\ \rho(V+W) &= lcm(\rho(V), \rho(W)) \end{aligned}$$

  where the equality in 2.2.8 holds if and only if $gcd(P_0, Q_0) = 1$.

- their product verifies

$$\Lambda(V \cdot W) \leq \Lambda(V) \cdot \Lambda(W)$$

  where the equality holds if and only if P_0 and Q_0 are primitive and $gcd(P_0, Q_0) = 1$. In the latter case it also holds

$$\rho(V \cdot W) = \rho(V) \cdot \rho(W)$$

The ideal situation is hence when the sequences all have primitive feedback polynomials and their periods are relatively prime. Proposition 2.13 can be easily extended to n sequences and leads to the following Theorem.

Theorem 2.14. *Let $R_1, \ldots, R_n$ be n LFSR. Let their feedback polynomials P_i for $i = 1, \ldots, n$ be primitive and relatively prime.*

If L_i denotes the length of R_i (and consequently the degree of its feedback polynomial), we know L_i to be the linear complexity of R_i, since P_i is irreducible and then is the minimal polynomial of the sequence.

If the registers are combined through a function $f : \mathbb{F}_q^n \to \mathbb{F}_q$ and Y is the keystream, we have

$$\Lambda(Y) = f(L_1, \ldots, L_n)$$

where here f stands for its ANF, namely a polynomial in $\mathbb{F}_q[X_1, \ldots, X_n]$, but evaluated on the integers.

Example 2.13. Let's take for instance *Geffe* generator over $\mathbb{F}_2$, defined as the combination of three registers through the function

$$f(x_1, x_2, x_3) = x_1 \oplus x_1x_2 \oplus x_2x_3$$

The feedback polynomials of the registers are taken primitive and with degrees L_1, L_2 and L_3 relatively prime. Then the complexity of the keystream is $L_1 + L_1L_2 + L_2L_3$.

Part II

Attacks

Chapter 3

Correlation attack

In 1985, Thomas Siegenthaler proposed the first attack to a LFSRs' combination cipher exploiting the correlation property of the combining function[17].

The attack is a *ciphertext-only*, namely it is moved without any knowledge of the plaintext.

The cipher is assumed to work over $\mathbb{F}_2$ and the following specifics are assumed to be known:

- the number n of registers involved
- their length
- the non-linear boolean function f used to combine their outputs

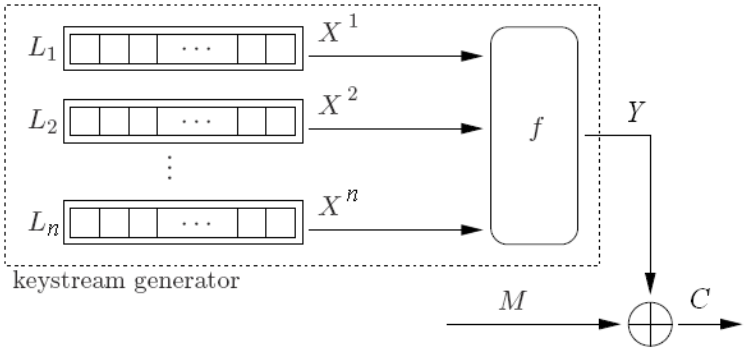


Figure 3.1: The model used by Siegenthaler

The system is described in figure 3.1. The registers are termed $L_1, \ldots, L_n$, l_i denotes the length of the ith register and $X^i = \{x_t^i\}_{t\in\mathbb{N}}$ its output stream. We call $C = \{c_t\}_{t\in\mathbb{N}}$, $M = \{m_t\}_{t\in\mathbb{N}}$ and $Y = \{y_t\}_{t\in\mathbb{N}}$ respectively the ciphertext, the plaintext and the keystream.

The generic keystream bit is obtained as $y_t = f(x_t^1, \ldots, x_t^n)$ and then XORed with the plaintext to produce the ciphertext: $c_t = m_t \oplus y_t$.

3.1 Introduction and statistical model

We can suppose that each register has his own key, composed of the initialization and the feedback relation. The attack's aim is to break the whole cipher, recovering every key.

The registers are independently attacked one at a time, with an approach known as *divide-and-conquer*.

Suppose we want to attack the ith register L_i. It can be clearly initialized in $2^{l_i} - 1$ different ways, while the possible feedback relations are $R_i = \frac{\varphi(2^{l_i}-1)}{l_i}$, so the keyspace for L_i has cardinality $R_i(2^{l_i} - 1)$. In the traditional way the total cost of a brute force attack would be

$$K = \prod_{i=1}^{n} R_i(2^{l_i} - 1)$$

since we have to try any possible combination of the n registers' keys.

With a divide-and-conquer strategy, also a brute force attack gets much faster because attacking the registers separately permits us to find the right key in

$$K' = \sum_{i=1}^{n} R_i(2^{l_i} - 1) << K$$

.

Finally, before describing the attack, we need to trace the statistical model we're in:

- The output X^i of the ith register and the keystream Y are distributed as a binary balanced random variable: $X^i, Y \sim \mathcal{B}(\frac{1}{2})$.
- The different outputs are independent and identically distributed.
- The plaintext M is the output of a binary memoryless source where $p' = P(M_t = 0) \neq \frac{1}{2}$, so it is distributed as a Bernoulli of parameter $1 - p'$: $M \sim \mathcal{B}(1 - p')$.
- We introduce an auxiliary variable $X^0 \sim \mathcal{B}(\frac{1}{2})$, independent from all the other X^i and such that $p_0 = P(c_t = x_t^0) = \frac{1}{2}$.

3.2 The attack

To attack register L_i, we look for a correlation between a generic ciphertext bit c_t and the corresponding bit x_t^i produced by the register.

We begin by calculating the probability of the two bits to be equal:

$$\begin{aligned} p_i &= P(c_t = x_t^i) = P(y_t = x_t^i)P(m_t = 0) + P(y_t \neq x_t^i)P(m_t = 1) = \\ &= q_i p' + (1 - q_i)(1 - p') = 1 - p' - q_i + 2p'q_i \end{aligned}$$

where $q_i = P(f(x_1, \ldots, x_n) = x_i)$ is the correlation between the output of f and its ith input.

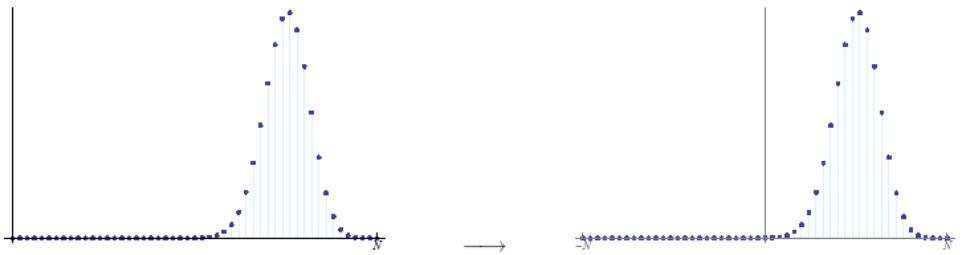
Given a portion of N bits of ciphertext, we introduce the variables z_t and α, defined as

- $z_t = c_t \oplus x_t^i$, for $t = 1, \ldots, N$
- $\alpha = \sum_{t=1}^{N} (-1)^{z_t} = \sum_{t=1}^{N} (1 - 2z_t) = N - 2\sum_{t=1}^{N} z_t$

Note that α computes somehow the correlation between C and X^i along these N bits, adding 1 every time $c_t = x_t^i$ and subtracting 1 otherwise.

Remark 3.1. Since z_t is a Bernoulli of parameter $1-p_i$, by the Central Limit Theorem, for big values of N the variable $\sum_{t=1}^{N} z_t$ can be approximated by a Gaussian with mean $\mu' = N(1 - p_i)$ and variance $\sigma'^2 = Np_i(1 - p_i)$.

This means that also α can be approximated by a Gaussian, but "stretched" in the interval $[-N, N]$, with mean $\mu = N(2p_i - 1)$ and variance $\sigma^2 = 4Np_i(1 - p_i)$, as you can see in figure 3.2.



(a) $\mathcal{B}_N\,(1 - p_i)$ (b) $N - 2\mathcal{B}_N\,(p_i)$

Figure 3.2: (a) shows the distribution of a Binomial of parameters N and $1 - p_i$; (b) shows the same binomial, multiplied by -2 and shifted by N

The idea is to test every possible key for register L_i, selecting the ones leading to a value of α sufficiently close to the expected value $N(2p_i - 1)$.

If we find more than one candidate, we will typically repeat the test using more ciphertext bits to decide which is the right one.

For each key, we generate an output of maximal length $N = 2^{l_i} - 1$ and evaluate α according to its definition.

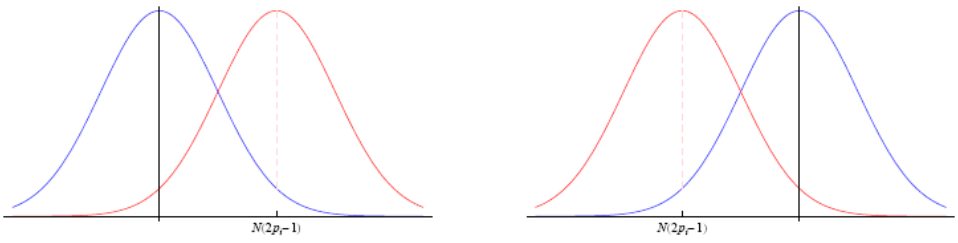
A priori we have to possible alternatives:

$\mathcal{H}_0$: If the key is not the one actually used to encrypt, then the N bits obtained running the register with that key do not coincide with the ones used to generate the keystream. In this case, α measures only the correlation between C and the auxiliary uniform variable X^0 and it's hence distributed as a normal random variable $\mathcal{N}(0, N)$, whose probability density is $P_0(x) = \frac{e^{-\frac{x^2}{2N}}}{\sqrt{2\pi N}}$

$\mathcal{H}_1$: If the tested key is the same couple of initialization and feedback connection used for that register in the encryption algorithm, then the N bits generated in the test are exactly the same used to produce the keystream. In this case, α is the correlation between C and X^i, hence, as we have seen, it's distributed as a $\mathcal{N}(\mu, \sigma^2)$, whose probability density is $P_1(x) = \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}$

Remark 3.2. Note that the two Gaussians of cases $\mathcal{H}_0$ or $\mathcal{H}_1$ (and consequently the probability density functions $P_0(x)$ and $P_1(x)$) coincide if and only if $p_i = \frac{1}{2}$, namely if $p' = \frac{1}{2}$ or $q_i = \frac{1}{2}$. When $p' = \frac{1}{2}$ the whole cipher is impossible to attack, otherwise when $p' \neq \frac{1}{2}$ we can attack every register L_i such that $q_i \neq \frac{1}{2}$.

To understand which of the two hypothesis $\mathcal{H}_0$ and $\mathcal{H}_1$ is verified, we use the obtained value of α. We fix a threshold T and we decide according as $\alpha > T$ or $\alpha < T$.



(a) $p_i > \frac{1}{2}$ (b) $p_i < \frac{1}{2}$

Figure 3.3: Graphs of the two probability densities: P_0 is the blue line, while P_1 is the red line.

As shown in figure 3.3, we clearly have two different situations when $p_i > \frac{1}{2}$ or $p_i < \frac{1}{2}$:

- If $p_i > \frac{1}{2}$, then $\mu > 0$ and the threshold T must be chosen in the interval $(0, \mu)$. We accept hypothesis $\mathcal{H}_0$ if $\alpha < T$ and $\mathcal{H}_1$ if $\alpha > T$.
- If $p_i < \frac{1}{2}$, contrarily $\mu < 0$ and the threshold T must be chosen in the interval $(\mu, 0)$. We accept hypothesis $\mathcal{H}_0$ if $\alpha > T$ and $\mathcal{H}_1$ if $\alpha < T$.

The conditions above are clearly symmetric and can be easily reported to the positive case resuming them as

- $\mathcal{H}_0$ is taken when $|\alpha| < |T|$
- $\mathcal{H}_1$ is taken when $|\alpha| > |T|$

We still have to discuss how to set the value T. The threshold must be chosen to minimize the error probabilities:

- $p_{fa} = P(|\alpha| > |T||\mathcal{H}_0)$, the probability to have a "false alarm", namely to select a wrong key
- $p_{nd} = P(|\alpha| < |T||\mathcal{H}_1)$, the probability of "non-detection" of a good candidate

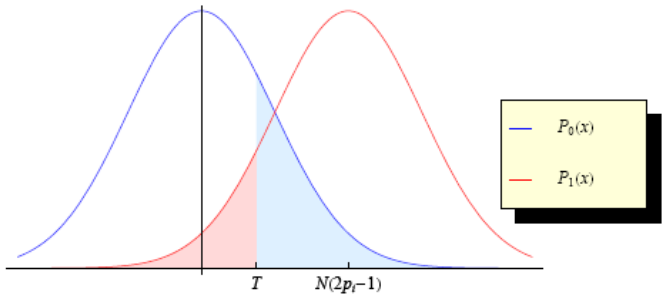
By definition of probability density function, we can compute these probabilities as

$$\begin{aligned} p_{fa} &= \int_{|T|}^{+\infty} P_0(x)dx \\ p_{nd} &= \int_{-\infty}^{|T|} P_1(x)dx \end{aligned}$$

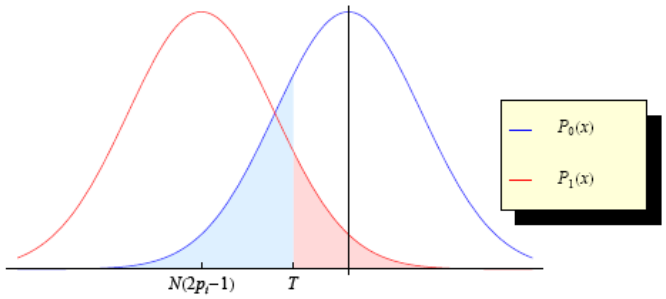
It should be easy to understand that these two probabilities are tightly linked, since when trying to minimize one of them we automatically increase the other one, as we can see in figure 3.4. Hence we need to look for a good compromise, remembering that if we detect more than one good candidate we can still perform a tighter selection to choose the right key between them, while if we miss the only good candidate the attack fails and has to be entirely repeated.

To menage more easily with these error probabilities, we can normalize the variables to reduce to the standard normal. Substituting $y = \frac{x}{\sqrt{N}}$ in the first case (P_0) and $y = \frac{x-\mu}{\sigma}$ in the second case (P_1), we get:

$$\begin{aligned} p_{fa} &= \int_{\frac{|T|}{\sqrt{N}}}^{+\infty} f(y)dy = F\left(\frac{|T|}{\sqrt{N}}\right) \\ p_{nd} &= \int_{-\infty}^{\frac{|T|-\mu}{\sigma}} f(y)dy = \int_{-\frac{|T|-\mu}{\sigma}}^{+\infty} f(y)dy = F\left(\frac{\mu - |T|}{\sigma}\right) \end{aligned}$$



(a) $p_i > \frac{1}{2}$



(b) $p_i < \frac{1}{2}$

Figure 3.4: Graphs of the values of p_{fa} (the areas in blue) and p_{nd} (the areas in red)

where $f(y) = \frac{e^{-\frac{y^2}{2}}}{\sqrt{2\pi}}$ is the standard normal density function and $F(x) = \int_x^{+\infty} f(y)dy$ is its distribution function.

Now we choose a maximal accepted value δ for p_{nd} and we obtain the condition

$$p_{nd} \leq \delta \quad \Leftrightarrow \quad F\left(\frac{\mu - |T|}{\sigma}\right) \leq \delta$$

which can easily be reversed using a standard normal table, getting

$$\frac{\mu - |T|}{\sigma} \leq F^{-1}(\delta)$$

allowing to find the threshold T as the value minimizing $|T|$ among those who verify

$$|T| \geq \mu - \sigma F^{-1}(\delta)$$

3.3 The algorithm

We have seen how to compute all the parameters we needed, the implementation of the attack is then described in details in algorithm 3.1.

Algorithm 3.1 (Correlation Attack).

Require: ciphertext $C = c_1 c_2 \dots c_N$
Require: number n and length l_i of the registers
Require: combining function f
Require: the values $p' \neq \frac{1}{2}$ and q_i, for $i = 1, \dots, n$

Choose a value δ for p_{nd}
for $i = 1, \dots, n$ **do**
  if $q_i \neq \frac{1}{2}$ **then**
    $p_i \leftarrow 1 - p' - q_i + 2p'q_i$
    $\mu \leftarrow N(2p_i - 1)$
    $\sigma \leftarrow 2\sqrt{Np_i(1-p_i)}$
    $T \leftarrow \mu - \sigma F^{-1}(\delta)$
    for each pair "feedback connection - non-zero initial state" of L_i **do**
      generate an output $X = x_1 x_2 \dots x_N$ of length $N = 2^{l_i} - 1$
      $\alpha = 0$
      for $t = 1, \dots, N$ **do**
        $z_t \leftarrow c_t \oplus x_t$
        $\alpha \leftarrow \alpha + (-1)^{z_t}$
      end for
      if $|\alpha| > |T|$ **then**
        store the pair
      end if
    end for
  end if
  if more than one pair is stored **then**
    do additional tests
  end if
end for

3.4 Comments

If we look at Siegenthaler's correlation attack nowadays, it may appear to be quite "primitive" and not much performing, but it was very revolutionary at the time it was developed.

The attack is, actually, a brute force attack on the registers taken one at a time, so it can easily be avoided with a trivial improvement: increasing the length of the registers.

As we have already seen, the keyspace's cardinality of a registers of length l is $K_l = \frac{\varphi(2^l-1)}{l}(2^l - 1)$ and as $l > 43$ we get $K_l > 2^{80}$, enough to make an exhaustive search infeasible.

Moreover, the main reason why the attack seems difficult to be implemented in practice is because of the development of the theory of correlation-immune functions. As shown by proposition 3.1, a 1-CI function is, in fact, sufficient to prevent Siegenthaler's attack, but we must remark that the interest on this property rose actually to resist to the family of attacks inspired by Siegenthaler's.

Proposition 3.1. Let the correlation immunity order of a boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be $r \geq 1$ and let $x = (x_1, \ldots, x_n)$ denote a generic element of the set $\mathbb{F}_2^n$. Then $P(f(x) = x_i) = \frac{1}{2}$ for all $i = 1, \ldots, n$.

Proof. If we use the classical definition of probability as favorable cases on possible ones, we get

$$P(f(x) = x_i) = \frac{|\{x : \ f(x) = x_i\}|}{|\mathbb{F}_2^n|} = \frac{|\{x : \ f(x) = x_i\}|}{2^n}$$

then we have to show that

$$|\{x : \ f(x) = x_i\}| = 2^{n-1}$$

For all $i = 1, \ldots, n$ we call $\mathbf{e}_i$ the vector of $\mathbb{F}_2^n$ whose ith component is 1 and whose every other component is 0. Clearly, the weight of $\mathbf{e}_i$ is $w_H(\mathbf{e}_i) = 1$ and $x \cdot \mathbf{e}_i = x_i$. From proposition 2.12, we know that

$$\hat{\chi}_f(v) = 0 \quad \text{for all } v \in \mathbb{F}_2^n : \ 0 \leq w_H(v) \leq r$$

so

$$\hat{\chi}_f(\mathbf{e_i}) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+\mathbf{e_i} \cdot x} = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+x_i} = 0$$

The last condition is valid if and only if the function $f(x)+x_i$ is balanced, namely if

$$|\{x : \ f(x) = x_i\}| = |\{x : \ f(x) \neq x_i\}| = 2^{n-1}$$

□

Although these weaknesses, correlation attack was used by Siegenthaler to break three ciphers considered solid at that time: Geffe generator, Brüer generator and Pless generator.

Moreover, the way Siegenthaler's attack exploits the correlation between a boolean function and its inputs inspired many attacks in the following 20 years, such as the very famous Fast Correlation Attack, conceived in 1989 by Meier and Staffelbach[**?**].

All those attacks implement in different ways a model known as Binary Symmetric Channel Model, which we will describe in the following chapter and which is substantially the generalization of Siegenthaler's idea.

Different correlation attacks were able to break many ciphers.

One of the most famous example is the A5/1 cipher, much implemented in GSM communications, whose design was kept secret until 1994, when it was leaked, before being completely reverse engineered in 1999. From 1997 to 2000 many tradeoff attacks appeared, some of which required a huge preprocessing phase. The turning point in the attacks was in 2003, when Ekdahl and Johansson published the first correlation attack to A5/1. The following year Maximov et al. improved the attack, finally optimized in 2006 by Barkar and Biham.

In 2006 a correlation based attack was applied also to another stream cipher much used in mobile communication, Grain cipher, which contains only 2 shift registers, a linear and a nonlinear one.

Chapter 4

Reconstruction of stream ciphers

4.1 Introduction

In chapter 2 we have introduced a particular class of stream ciphers, whose keystream is obtained combining the outputs of some LFSRs. Many other stream ciphers can be shown to be equivalent to a LFSRs combination generator or at least to a noisy versione of it[5], so the following technique applies quite widely.

We can formulate the problem as follows: when facing a counter whose stream cipher's specifications are completely unknown, how can we exploit the interceptions of his transmissions to cryptanalyse the system?

This is a very common problem, especially in the military sphere, where the features of the cryptosystem being used are usually kept secret.

The most famous historical case is probably the cryptanalysis of the PURPLE machine by the US intelligence during World War II. The cipher was used by the Japaneses to secretly communicate and the Americans didn't know anything about its specifications. The only way for them to cryptanalyse the PURPLE machine was then to first reconstruct it or an equivalent one. No one ever knew if the logic of the machine the Americans built was the same of the one used by the Japaneses, but the reconstruction worked, since they succeeded in breaking the cryptosystem. The Japaneses, for their part, never believed to the story of the reconstruction and have always been thinking that the algorithm was compromised thanks to human espionage.

In this chapter we will show how algebrical and statistical results can be actually used to reconstruct some of the primitives of a particular class of stream ciphers. We will focus on LFSRs combination generators, describing how to find out the number of registers used and their structure, namely

their length and their feedback polynomial.

In 2000, in the first part of his Ph.D thesis[5], Eric Filiol proposed a method to completely reconstruct the cipher in the most common case, the binary one.

Filiol's algorithm uses some algebraic results shown in our introduction, togheter with the main ideas of correlation attacks.

Here we propose a generalization of his method to every finite field $\mathbb{F}_q$.

The reconstruction of the registers substantially follows the same scheme and can be completely achieved.

The recovery of the combining function as done by Filiol is not feasible over $\mathbb{F}_q$, since the properties of a generic function over any finite field are much more complicated as we already noticed at the end of section 2.2.1.

4.2 Scenario

The hypothesis for the reconstruction are the following:

- We can only use ciphertext, in quite big, but realistic, quantity (some thousands of digits). That's the only material quite easy to obtain, while the knowledge of the corresponding plaintext would be irrealistic.
- The time necessary to the reconstruction algorithm could be large. This work, in fact, has to be done only once and for all, while the lifetime of this kind of ciphers is usually very long, sometimes also 10 or 20 years.
- We know the code used to represent the plaintext (such as ASCII for example) or at least we have some statistical informations on the distribution of the digits of plaintext. In particular, in the binary case we will suppose, as happens for most of the codes, that the plaintext digits is zero with probability $0.6 \leq p_0 \leq 0.7$.
- The keystream generator is a combination of LFSRs and the number of registers used is limited, typically not greater than 9.

The scheme of the cipher is shown in figure **??**.

We suppose the number of registers to be n and the combining function to be $f : \mathbb{F}_q^n \to \mathbb{F}_q$.

The ith register is termed L_i and its feedback polynomial is P_i, whose degree (equal to the length of the register) is l_i.

The output stream of L_i is denoted $S^i = \{s_t^i\}_{t\in\mathbb{N}}$ and the keystream is $Y = \{y_t\}_{t\in\mathbb{N}}$, where $y_t = f(s_t^1, \ldots, s_t^n)$.

The plaintext is $M = \{m_t\}_{t\in\mathbb{N}}$, while the ciphertext is termed $C = \{c_t\}_{t\in\mathbb{N}}$. The cipher is supposed to be additive, so the t-th digit of ciphertext is obtained by modular addition of the corresponding keystream and plaintext digits: $c_t = y_t + m_t$.

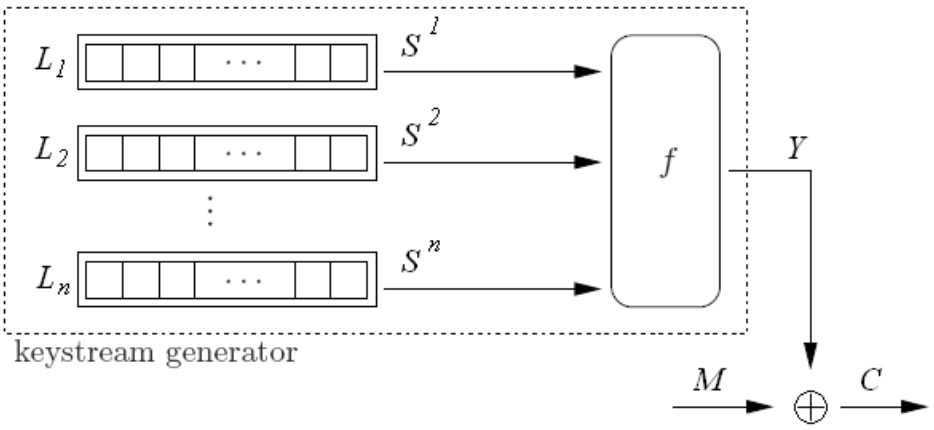


Figure 4.1: The model of the cipher we want to reconstruct

We will indistinctly use the sum either to denote the modular addition over $\mathbb{F}_q$ or for the ordinary summation over $\mathbb{Z}$, hoping that the context would help the reader to distinguish.

4.3 The BSC Model

The idea of the reconstruction is to exploit the correlation property of the function f as in Siegenthaler's attack. The difference is that this time our goal is to recover the feedback polynomials of the registers and not their inizializations.

In this section we will referr to a binary cipher, since, being the simpliest example, it is the best one to illustrate the idea.

Let $p_i = P(f(x_1, \ldots, x_n) = x_i) = P(y_t = s_t^i)$ be the correlation between f and its ith input and $p_0 = P(m_t = 0)$, then

$$p = P(c_t = s_t^i) = p_i p_0 + (1 - p_i)(1 - p_0) = 1 - p_i - p_0 + 2p_i p_0$$

We have $p_0 \neq \frac{1}{2}$ by hypothesis, so $p_i \neq \frac{1}{2}$ clearly implies $p \neq \frac{1}{2}$ too.

On the contrary, for a random sequence $S = \{s_t\}_{t \in \mathbb{N}}$, we expect to have $P(c_t = s_t) = \frac{1}{2}$, since there's not connection between the way the two sequences are generated.

The sequences generated by a random polynomial, such S, and those generated by the polynomials P_i, such as S^i, are hence expected to behave differently. This difference allows us to identify the right polynomials comparing the predictions with the results of a test.

For example, for a sufficiently big sample N, we expect the portion of bits such that $c_t = s_t^i$ to be approximately pN, while the number of bits such that $c_t = s_t$ should be around $\frac{N}{2}$.

We could, then, text every possible couple of initial vector and feedback polynomial, looking for the one generating a sequence which acts as similarly as possible to the way we expect the sequence S^i to do.

Unfortunately, this approach is clearly infeasible. First because we do not know the function f and consequently we can't evaluate the probabilities p_i, then because we do not even know the length of the registers and a research over every possible polynomial of every possible degree would require too much computational power.

Nevertheless, the idea to exploit the correlation property of the function is valid, we just need to find a way to get around these problems.

Since the ciphertext of a stream cipher is always the result of some operations performed on some basic sequences, everytime we do not know exactly which operations have been done we can use a more general approach known as *Binary Symmetric Channel model*, typical of information theory.

In our case, for example, to recover informations on the output sequences of the registers from the ciphertext, we need to "bypass" the effect of the function f and the addiction with the plaintext. This can be done as follows.

Suppose $\hat{S} = g(S^1, \dots, S^n)$, where g is a boolean function. Such a sequence is supposed to be correlated with the ciphertext, in the sense that we expect to have $\hat{p} = P(c_t = s_t) \neq \frac{1}{2}$.

We can, then, forget what actually happens in the cipher and imagine the ciphertext to be the result of the transmission of the sequence $\hat{S}$ through a Binary Symmetric Channel of parameter $1 - \hat{p}$, namely a channel which in every instant correctly transmit the bit with probability $\hat{p}$ and "flips" it with probability $1 - \hat{p}$.

For example, if we take the function $g(x_1, \dots, x_n) = x_i$, we simply have $\hat{S} = S^i$, as in figure 4.2. Since $p = P(c_t = s_t^i) = 1 - p_i - p_0 + 2p_ip_0 \neq \frac{1}{2}$, we can consider the ciphertext as the noisy version of S^i obtained sending S^i through a BSC of parameter $1 - p$, as in figure 4.3.

We call $E = \{\epsilon_t\}_{t \in \mathbb{N}}$ the noise sequence and write $c_t = s_t^i + \epsilon_t$ for all t, so the noise clearly satisfies $P(\epsilon_t = 0) = p$.

The reconstruction is now reduced to a sort of decoding problem, since we can assume C to be the received message and we are interested in detecting every possible message $\hat{S}$ that could have been originally sent over a BSC with crossover probability different from $\frac{1}{2}$.

Such a sequence, in fact, being statistically dependent of C, is necessarily of the form $\hat{S} = g(S^1, \dots, S^n)$ for some function g, and we known it to be generated by a polynomial which depends on the polynomials P_i according to the algebraic normal form of g, as we have shown in Theorems 2.12 and 2.13.

Actually, we will not look for the sequences, but directly for the polynomials, since every polynomial able to generate at least one sequence with those properties gives us informations on the polynomials P_i.

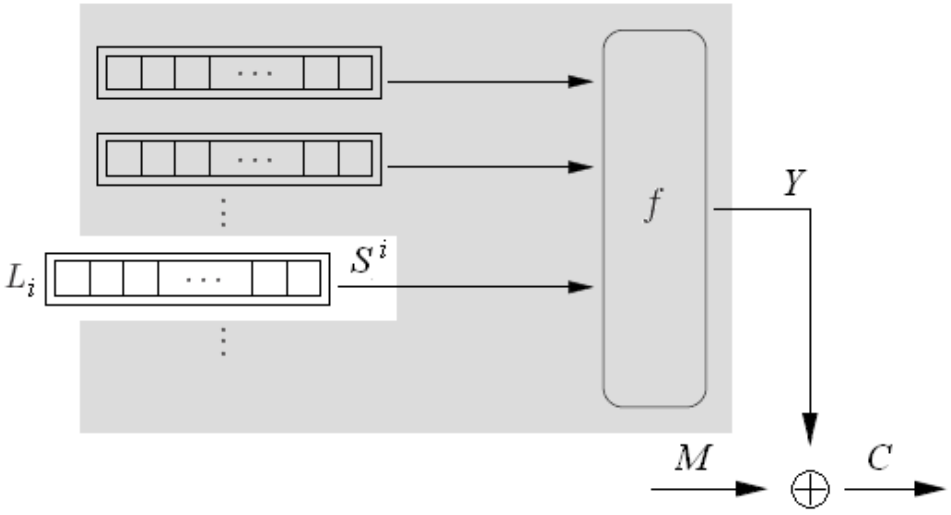


Figure 4.2: Forget the structure of the cipher and just concentrate on the sequence S^i.

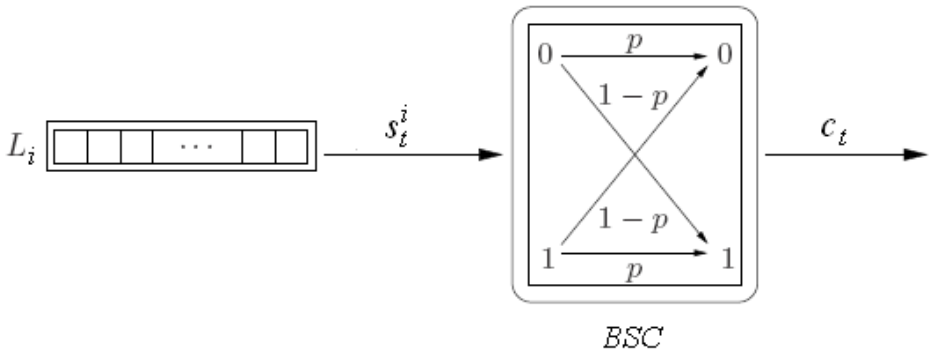


Figure 4.3: The ciphertext can be seen as the message received sending S^i over a BSC of parameter $1-p$.

4.4 Preliminary analysis

In general, we suppose the alphabet to be any finite field $\mathbb{F}_q$.

We want to extend the idea of the BSC model, since also in this case we want to express the ciphertext as a noisy version of $g(S^1, \ldots, S^n)$, where $g : \mathbb{F}_q^n \to \mathbb{F}_q$ is a generic function over $\mathbb{F}_q$.

Let be $\mathbb{F}_q = \{\alpha_0 = 0, \alpha_1, \ldots, \alpha_{q-1}\}$.

The BSC of parameter $1-p$ can be equivalently defined as a channel which in every instant adds 0 to the transmitted bit with probability p or adds 1 to it with probability $1-p$.

Therefore, the natural way to extend the BSC model to $\mathbb{F}_q$ is to consider a channel which in every instant adds α_k to the transmitted digit with probability p_k, where k varies in $\{0,\dots,q-1\}$ and $\sum_{k=0}^{q-1} p_k = 1$.

As already specified, we assume the plaintext $M = \{m_t\}_{t\in\mathbb{N}}$ to be nonuniformly distributed, namely we define

$$p_{m,k} = P(m_t = \alpha_k) \quad \text{for all } k \in \{0,\dots,q-1\}$$

where obviously $\sum_{k=0}^{q-1} p_{m,k} = 1$, and we suppose there exists at least one index k such that $p_{m,k} \neq \frac{1}{q}$.

Let g_t be the sequence obtained as $g(s_t^1,\dots,s_t^n)$, while $y_t = f(s_t^1,\dots,s_t^n)$ as usual denoted the keystream.

Now, let $E = \{\epsilon_t\}_{t\in\mathbb{N}}$ be the noise sequence, so that we write $g_t + \epsilon_t = c_t$ for all t.

The noise is distributed as follows:

$$\begin{aligned} p_k &= P(\epsilon_t = \alpha_k) = P(c_t - g_t = \alpha_k) = P(y_t + m_t - g_t = \alpha_k) = \\ &= \sum_{h=0}^{q-1} p_{m,h} P(y_t - g_t = \alpha_k - \alpha_h) \end{aligned}$$

We are thus imagining to send the message sequence $\hat{S} = g(S^1,\dots,S^n)$ over a memoryless channel affected by the noise E, which correctly transmit a digit with probability p_0 or modifies it by one of the constant values α_k $(k \neq 0)$, each with probability p_k.

The reason why we are interested in any function g of the output of the registers is given by the theorems we have showed in Chapter 2.

In fact, let $P_1(x),\dots,P_n(x),Q(x) \in \mathbb{F}_q[x]$ be primitive polynomials, then

- if $P \mid Q$, then $\mathcal{S}(P) \subseteq \mathcal{S}(Q)$ (Theorem 2.11)
- $\mathcal{S}(P_1) + \cdots + \mathcal{S}(P_n) = \mathcal{S}(Q)$, where $Q = lcm(P_1,\dots,P_n)$ and so $deg(Q) = \sum_{i=1}^{n} deg(P_i)$ (Theorem 2.12)
- $\mathcal{S}(P_1)\cdots\mathcal{S}(P_n) = \mathcal{S}(Q)$ where Q is a polynomial such that $deg(Q) = \prod_{i=1}^{n} deg(P_i)$ (Theorem 2.13)

These results explicitly show some relations between the characteristic polynomial of any sequence obtained as a function of the sequences produced by the registers and the characteristic polynomials of the registers.

In particular, let g be any linear combination, namely $g(x) = u\cdot x$, where $x = (x_1,\dots,x_n)$ is the unknown vector and $u = (u_1,\dots,u_n)$ is a vector in $\mathbb{F}_q^n$.

As we have already seen, the set $\mathcal{S}(P)$ is closed under multiplication for a constant $c \in \mathbb{F}_q$, so S and cS are generated by the same polynomial for every sequence S.

Hence, the sets $\mathcal{S}(P)$ and $c\mathcal{S}(P)$ coincide, implying that the sequence $\hat{S} = g(S^1, \ldots, S^n) = u_1 S^1 + \cdots + u_n S^n$ is generated by $Q = lcm(P_1, \ldots, P_n)$ (Theorem 2.12) or by one of its multiples (Theorem 2.11).

At the same time, according to Remark 2.6, Theorems 2.12 and 2.13 assure that if g isn't linear the sequence $\hat{S}$ is generated by a polynomial of degree much higher than that of $lcm(P_1, \ldots, P_n)$.

The idea of the reconstruction is thus not to look directly for the polynomials P_i, but just for any polynomial Q such that there exists at least one sequence $S \in \mathcal{S}(Q)$ correlated with the ciphertext.

Such a sequence, in fact, can be seen as a function g of those generated by (at least someone of) the registers.

If $deg(Q)$ is very high compared to the expected degrees of the feedback polynomials, we don't have any information on the ANF of the function g.

But if Q belongs to a set Ω of polynomials of limited degree, g must be a linear function and we know Q to be a multiple of the polynomials of the registers involved in the linear combination.

Hence we choose a set Ω whose elements are polynomials whose degree is bounded by an integer D and mount on it a brute force attack. To reduce the complexity of the attack, we reduce the cardinality of Ω imposing that its elements must have only a small number d of non-zero coefficients.

We select any polynomial $Q \in \Omega$ such that there exists a sequence $S \in \mathcal{S}(Q)$ which is correlated with the ciphertext and we factorize everyone of the selected polynomials and look for common factors, expect them to be the polynomials P_i.

The complexity is much reduced compared with a traditional brute force attempt and the knowledge of the plaintext and the combining function is not necessary, thanks to the information theory model we use.

In particular, we save much computational complexity thanks to the approach used to verify the presence in $\mathcal{S}(Q)$ of sequences correlated with the ciphertext.

A trivial way would be trying every possible inizialization to generate every possible sequence in $\mathcal{S}(Q)$, then for each of them test its correlation with C.

The algorithm proposed by Filiol, instead, allows somehow to check every sequence at the same time, simply rearranging the method used to test correlation in Siegenthaler's attack, and we will do the same.

The idea is to check the correlation between the ciphertext and the most correlated sequence in $\mathcal{S}(Q)$. Clearly, we do not know which one it is, but we will show that this information is not necessary.

4.5 Reconstruction over $\mathbb{F}_q$

First of all, we need to formalize what we have noticed in the previous section.

Proposition 4.1. Let S be a generic sequence over $\mathbb{F}_q$ and let $x = (x_1, \ldots, x_n)$ denote a vector in $\mathbb{F}_q^n$.

If there exists $\alpha \in \mathbb{F}_q$ such that $P(c_t - s_t = \alpha) \neq \frac{1}{q}$, then S can be written as a function of the outputs $S^1, \ldots, S^n$ of the registers.

Namely, there exists a function $g : \mathbb{F}_q^n \to \mathbb{F}_q$ such that for all $t \in \mathbb{N}$

$$s_t = g(s_t^1, \ldots, s_t^n)$$

Moreover, we can write

$$P(c_t - s_t = \alpha) = \sum_{k=0}^{q-1} p_{m,k} P(f(x) - g(x) = \alpha - \alpha_k)$$

Proof. Clearly we have

$$\begin{aligned} P(c_t - s_t = \alpha) &= P(y_t + m_t - s_t = \alpha) = P(y_t = s_t + \alpha - m_t) \\ &= \sum_{k=0}^{q-1} p_{m,k} P(y_t = s_t + \alpha - \alpha_k) \end{aligned}$$

Now, since α is fixed, if $P(y_t = s_t + \alpha - \alpha_k) = \frac{1}{q}$ for all $k \in \{0, \ldots, q-1\}$, we have

$$P(c_t - s_t = \alpha) = \sum_{k=0}^{q-1} p_{m,k} \frac{1}{q} = \frac{1}{q} \sum_{k=0}^{q-1} p_{m,k} = \frac{1}{q}$$

which contradicts our hypothesis.

Then there must exist at least one $k \in \{0, \ldots, q-1\}$ such that $P(y_t = s_t + \alpha - \alpha_k) \neq \frac{1}{q}$.

Since in the last equation $\alpha - \alpha_k$ is a constant, it means that s_t and y_t are not statistically independent. Since $y_t = f(s_t^1, \ldots, s_t^n)$, it means that s_t isn't statistically independent of $s_t^1, \ldots, s_t^n$. Hence, there must exist a function g such that $s_t = g(s_t^1, \ldots, s_t^n)$.

Now trivially we can write

$$\begin{aligned} P(c_t - s_t = \alpha) &= \sum_{k=0}^{q-1} p_{m,k} P(y_t = s_t + \alpha - \alpha_k) \\ &= \sum_{k=0}^{q-1} p_{m,k} P(f(s_t^1, \ldots, s_t^n) = g(s_t^1, \ldots, s_t^n) + \alpha - \alpha_k) \\ &= \sum_{k=0}^{q-1} p_{m,k} P(f(x) - g(x) = \alpha - \alpha_k) \end{aligned}$$

□

Remark 4.1. The function g can be explicitly calculated. If we denote $\mathbf{s}_t = (s_t^1, \dots, s_t^n) \in \mathbb{F}_q^n$ the input vector of f at time t, we can introduce the sets

$$T_x = \{t : \ \mathbf{s}_t = x\} \quad \text{for all } x \in \mathbb{F}_q^n$$

of all times when x is the input of f.

For every T_x we define its subsets

$$T_x^k = \{t \in T_x : \ y_t - s_t = \alpha_k\} \quad \text{for all } k \in \{0, \dots, q-1\}$$

of the times when the difference between the sequences Y and S is α_k.

These subsets are a partition of T_x, since clearly

$$\bigcup_{k=0}^{q-1} T_x^k = T_x \qquad \text{and} \qquad T_x^k \cap T_x^{k'} = \emptyset \quad \text{for all } k \neq k'$$

By Proposition 4.1 we know that the sequences Y and S are not statistically independent, so it's impossible to have

$$|T_x^k| = \frac{|T_x|}{q} \quad \text{for all } k \in \{0, \dots, q-1\}$$

Therefore, we can just settle $g(x) = \alpha_{k'}$ where k' is such that $|T_x^{k'}| \geq |T_x^k|$ for all $k \neq k'$ (if there are more than one index with this property we just choose one of them).

It's easy to check that the so defined function g has the properties required by Proposition 4.1.

Note that some of the variables may not appear in the algebraic normal form of g.

For our purpose, the most interesting case is when the function f is $\tau - correlated$ for some integer $1 \leq \tau \leq n$, namely when, if we fix the values of τ variables, its distribuition changes.

In fact, if there exists a set of indices $i_1, \dots, i_\tau \in \{1, \dots, n\}$, τ elements $u_{i_1}, \dots, u_{i_\tau} \in \mathbb{F}_q$ and $\alpha \in \mathbb{F}_q$ such that

$$P(f(s^1, \dots, s^n) = u_{i_1} s^{i_1} + \cdots + u_{i_\tau} s^{i_\tau} + \alpha) \neq \frac{1}{q} \tag{4.5.1}$$

we can write the 4.5.1 as

$$P(f(s^1, \dots, s^n) = g(s^1, \dots, s^n) + \alpha) \neq \frac{1}{q}$$

where g is the linear function $g(x) = \mathbf{u} \cdot \mathbf{x}$ with $u \in \mathbb{F}_q^n$ and $u_j = 0$ for all $j \notin \{i_1, \dots, 1_\tau\}$.

It means, as we noticed above, that the sequence $\hat{S} = u_{i_1} S^{i_1} + \cdots + u_{i_\tau} S^{i_\tau}$, generated by the polynomial $Q = lcm(P_{i_1}, \dots, P_{i_\tau})$, is correlated with the

ciphertext C, namely the distribution of the sequence $C - \hat{S}$ isn't uniform. In particular we could find an imbalance of the sequence $C - \hat{S}$ on the value α.

Now, we just need to develop a correlation attack able to detect if a polynomial Q can generate such a sequence.

We then apply the attack to polynomials of limited degree and we obtain a selection of multiples of the polynomials of the registers.

4.5.1 Reconstruction of the registers

Let's fix two integers d and D and take the set $\Omega \subset \mathbb{F}_q[x]$ defined as

$$\Omega = \{Q \in \mathbb{F}_q[x] : \ W(Q) = d, \ deg(Q) \leq D \text{ and } a_0 = 1\}$$

where $W(Q)$ denotes the weight of Q, namely the number of non-zero coefficients of its ANF, and a_0 is the free term. We impose $a_0 = 1$ since any feedback polynomial has free term 1 according to Definition 2.6.

The generic polynomial we are going to test is then

$$Q(x) = 1 + \sum_{j=1}^{d-1} a_j x^{i_j}$$

where $a_j \in \mathbb{F}_q$ for all $j \in \{1, \ldots, d-1\}$. It corresponds to the linear feedback relation over $\mathbb{F}_q$ given by

$$s_t = \sum_{j=1}^{d-1} a_j s_{t-i_j}$$

To understand if $\mathcal{S}(Q)$ contains sequences correlated to the ciphertext C, we use the parity check equation defined by Q.

Let $\hat{S} \in \mathcal{S}(Q)$ be the sequence with maximal correlation with the ciphertext C.

$\hat{S}$ can be defined as the sequence in $\mathcal{S}(Q)$ maximizing

$$\left|\left|(P(c_t - s_t = 0), \ldots, P(c_t - s_t = \alpha_{q-1})) - \left(\frac{1}{q}, \ldots, \frac{1}{q}\right)\right|\right|$$

namely the sequence such that the distribution of $c_t - \hat{s}_t$ is the farther from the uniform one, with respect to the usual distance defined by the euclidean norm $||\cdot||$ in $\mathbb{R}^q$.

Imagining C to be a noisy version of $\hat{S}$, we can write $c_t = \hat{s}_t + \epsilon_t$ for all t, where the noise sequence $E = \{\epsilon_t\}$ has distribution

$$p_{\epsilon,k} = P(\epsilon_t = \alpha_k) = P(c_t - \hat{s}_t = \alpha_k) \qquad \text{for all } k \in \{0, \ldots, q-1\}$$

with $\sum_{k=0}^{q-1} p_{\epsilon,k} = 1$.

If we define $p_{\epsilon,k} = \frac{1}{q} + b_k$, we have

$$(p_{\epsilon,0}, \ldots, p_{\epsilon,q-1}) = \left(\frac{1}{q}, \ldots, \frac{1}{q}\right) + (b_0, \ldots, b_{q-1})$$

hence we can say that $\hat{S}$ maximize $||b||$, where $b = (b_0, \ldots, b_{q-1})$.

Now, we define the sequence

$$c'_t = \sum_{j=1}^{d-1} a_j c_{t-i_j}$$

generated by Q from the first digits of C, and we call $z_t = c'_t - c_t$.

We have

$$p_k = P(z_t = \alpha_k) = P\left(\epsilon_t - \sum_{j=1}^{d-1} a_j \epsilon_{t-i_j} = \alpha_k\right)$$

and we write

$$(p_0, \ldots, p_{q-1}) = \left(\frac{1}{q}, \ldots, \frac{1}{q}\right) + (r_0, \ldots, r_{q-1})$$

Intuitively, if $||b||$ is maximal, namely if the noise corresponding to the sequence $\hat{S}$ is the "less uniform", the linear combination $\epsilon_t - \sum_{j=1}^{d-1} a_j \epsilon_{t-i_j}$ of independent digits of E should have the same property, namely should have a distribution as far as possible from the uniform one. This means that maximal $||b||$ implies maximal $||r||$, where we put $r = (r_0, \ldots, r_{q-1})$

Now we call e_j the vector in $\mathbb{R}^q$ whose i-th component is 0 if $i \neq j$ and is 1 if $i = j$, namely the canonical bases vectors of $\mathbb{R}^q$.

We define the vectorial variable V_t as a variable taking values in $\{e_1, \ldots, e_q\}$ and such that

$$P(V_t = e_{k+1}) = P(z_t = \alpha_k) = p_k$$

for $k = 0, \ldots, q-1$.

Substantially, in every instant V_t tells us which one of the events $z_t = \alpha_k$ has occurred.

Hence the variable

$$W = \sum_{t=i_{d-1}}^{N-1} V_t = (|\{t : z_t = 0\}|, \ldots, |\{t : z_t = \alpha_{q-1}\}|)$$

is a Multinomial of parameters $N - i_{d-1}$ and $p_0, \ldots, p_{q-1}$.

By the Multivariate Central Limit Theorem, W can be approximated by a Multivariate Gaussian whose mean is the vector

$$\mu = ((N-i_{d-1})p_0, \ldots, (N-i_{d-1})p_{q-1}) = (N-i_{d-1})\left(\frac{1}{q}, \ldots, \frac{1}{q}\right) + (N-i_{d-1})r$$

The effective distribution of W dipends on the polynomial Q we are testing:

- If every sequence in $\mathcal{S}(Q)$ is independent from C, then also $\hat{S}$ is independent from C and hence $||b|| = 0$. This implies that $||r|| = 0$ too, hence W is a Multivariate Gaussian with mean $\mu_0 = (N - i_{d-1})\left(\frac{1}{q}, \ldots, \frac{1}{q}\right)$.

  We call this hypothesis $\mathcal{H}_0$.

- If there exists at least one sequence in $\mathcal{S}(Q)$ correlated with C, then $||b||$ takes the maximal possible value for a sequence in $\mathcal{S}(Q)$ and consequently the same remark holds for $||r||$. W is then a Multivariate Gaussian with mean $\mu = \mu_0 + (N - i_{d-1})r$, where $r \neq (0, \ldots, 0)$.

  We call this hypothesis $\mathcal{H}_1$.

The Gaussian of hypothesis $\mathcal{H}_1$ is thus a Gaussian whose mean is a point on the q-dimensional sphere of ray $(N - i_{d-1})||r||$ centered in μ_0.

The test consist of fixing a ray R and for any $Q \in \Omega$ compute the vector W according to its definition. Then we compute $||W - \mu_0||$ and say that

$$\begin{cases} \mathcal{H}_0 & \text{is true if} \quad ||W - \mu_0|| < R \\ \mathcal{H}_1 & \text{is true if} \quad ||W - \mu_0|| > R \end{cases}$$

In the first case we discard Q, otherwise we store it.

The probability of a "false alarm", namely to select a wrong polynomial, is the volume under the Gaussian bell centered in μ_0 outside the sphere of ray R and centered in μ_0. The probability of a "non detection", namely to discard a good polynomial, is the volume under the Gaussian bell centered in μ inside the same sphere.

To precisely compute these probabilities and to decide how to chose the ray R we can proceed as done by Filiol on $\mathbb{F}_2$ (see the next chapter for a detailed description). Substantially, we standardize the two variables reporting both to the Standard MultiNormal whose distribution function is numerically computable.which generated at least one sequence such that the corresponding variable W has mean that move away from μ_0 of at least a distance $(N - i_{d-1})||r||$.

4.6 Reconstruction over $\mathbb{F}_2$

In this section we see what happens when $q = 2$, reporting the work by Eric Filiol which inspired this thesis.

As the reader can easily see, the following algorithm is substantially a special case of the algorithm seen in section 4.5. Working over $\mathbb{F}_2$ permits to better manage with probabilities and distributions, getting some more explicit results. Furthermore, thanks to the properties of the Hamming distance and consequently to those of the Walsh spectrum of a boolean function, the recovery of the combining function can be easily achieved just exploiting some results of the reconstruction of the register.

4.6.1 Introductory results

First of all, we recall that in the binary case we assume the plaintext $M = \{m_t\}_{t\in\mathbb{N}}$ to have distribution

$$P(m_t = 0) = p_0,\ P(m_t = 1) = 1 - p_0 \qquad \text{with} \qquad 0.6 \leq p_0 \leq 0.7$$

The following proposition is equivalent to proposition 4.1, but on $\mathbb{F}_2$.

Proposition 4.2. Let $S = \{s_t\}_{t\in\mathbb{N}}$ be any binary sequence and let $x = (x_1, \ldots, x_n)$ denote a vector in $\mathbb{F}_2^n$.

If $P(c_t = s_t) \neq \frac{1}{2}$, then S can be written as a function of the outputs $S^1, \ldots, S^n$ of the registers.

Namely, there exist a boolean function $g : \mathbb{F}_2^n \to \mathbb{F}_2$ such that for all $t \in \mathbb{N}$ it holds

$$s_t = g(s_t^1, \ldots, s_t^2)$$

Moreover, we can write

$$P(c_t = s_t) = 1 - p_0 - p_g + 2p_0p_g$$

where $p_g = P(f = g)$.

Proof. Clearly, it holds

$$\begin{aligned} P(c_t = s_t) &= P(y_t \oplus m_t = s_t) \\ &= P(m_t = 0)P(y_t = s_t) + P(m_t = 1)P(y_t \neq s_t) \\ &= p_0P(y_t = s_t) + (1 - p_0)(1 - P(y_t = s_t)) \\ &= 1 - p_0 - P(y_t = s_t) + 2p_0P(y_t = s_t) \end{aligned}$$

Now, $P(y_t = s_t) = \frac{1}{2}$ would imply

$$P(c_t = s_t) = 1 - p_0 - \frac{1}{2} + p_0 = \frac{1}{2}$$

which contradicts our hypothesis.

Hence $P(y_t = s_t) \neq \frac{1}{2}$, namely s_t and y_t are not statistically independent. Since $y_t = f(s_t^1, \ldots, s_t^n)$, s_t is not statistically independent from $s_t^1, \ldots, s_t^n$, so there must exist a function g such that $s_t = g(s_t^1, \ldots, s_t^n)$.

Now trivially $P(y_t = s_t) = P(f = g) = p_g$ and we can write

$$P(c_t = s_t) = 1 - p_0 - P(y_t = s_t) + 2p_0 P(y_t = s_t) = 1 - p_0 - p_g + 2p_0 p_g$$

□

Remark 4.2. Over $\mathbb{F}_2$, the recovery of the function g of proposition 4.2 is even easier. If we denote $\mathbf{s}_t = (s_t^1, \ldots, s_t^n) \in \mathbb{F}_2^n$ the input vector of f at time t, we can introduce the sets

$$T_x = \{t : \mathbf{s}_t = x\} \quad \text{for all } x \in \mathbb{F}_2^n$$

of all times when x is the input of f.

For every T_x, we define its subsets

$$T_x^0 = \{t \in T_x : s_t = y_t\} \qquad \text{and} \qquad T_x^1 = \{t \in T_x : s_t = y_t + 1\}$$

to distinguish when $s_t = y_t$ or $s_t \neq y_t$.

These subsets are a partition of T_x, since clearly

$$T_x^0 \cup T_x^1 = T_x \qquad \text{and} \qquad T_x^0 \cap T_x^1 = \emptyset$$

By proposition 4.2, we know that the sequences S and Y are not statistically independent, so it's not possible to have

$$|T_x^0| = |T_x^1| = \frac{|T_x|}{2}$$

thus we can settle $g(x) = 0$ if $|T_x^0| \geq |T_x^1|$ and $g(x) = 1$ otherwise.

It's easy to check that the so defined function g has the properties required by proposition 4.2.

Note that some of the variables may not appear in the algebraic normal form of g.

If the function f is $\tau - correlated$, its distribution changes as soon as we fix the values of τ variables. We can exploit this property as follows.

Suppose there exist a set of indices $i_1, \ldots, i_\tau \in \{1, \ldots, n\}$ such that

$$P(f(s^1, \ldots, s^n) = s^{i_1} + \cdots + s^{i_\tau}) \neq \frac{1}{2} \tag{4.6.1}$$

we can write the 4.6.1 as

$$P(f(s^1, \ldots, s^n) = g(s^1, \ldots, s^n)) \neq \frac{1}{2}$$

where g is the linear function $g(x) = u \cdot x$ with $u = (u_1, \ldots, u_n) \in \mathbb{F}_2^n$ and $u_j = 1$ if and only if $j \in \{i_1, \ldots, 1_\tau\}$.

Hence the correlation between the ciphertext C and the sequence $\tilde{S} = S^{i_1} + \cdots + S^{i_\tau}$ provides informations about the polynomials $P_{i_1}, \ldots, P_{i_\tau}$:

- by theorem 2.12, in fact, $\tilde{S} \in \mathcal{S}(Q)$, where $Q = lcm(P_{i_1}, \ldots, P_{i_\tau})$
- by theorem 2.11, any polynomial generating the sequence $\tilde{S}$ its Q or one of its multiples

4.6.2 Reconstruction of the registers

To reconstruct the registers, first we fix two integers d and D and take the set $\Omega \subset \mathbb{F}_2[X]$ defined as

$$\Omega = \{Q \in \mathbb{F}_2[X] : \ W(Q) = d, \ deg(Q) \leq D \text{ and } a_0 = 1\}$$

where $W(Q)$ denotes the weight of Q, namely the number of non-zero coefficients of its ANF, and a_0 is the free term.

The generic polynomial we're going to test is thus

$$Q(X) = 1 + \sum_{j=1}^{d-1} X^{i_j}$$

which corresponds to the linear feedback relation over $\mathbb{F}_2$ given by

$$s_t = \bigoplus_{j=1}^{d-1} s_{t-i_j}$$

To understand if $\mathcal{S}(Q)$ contains any sequence correlated to the ciphertext C, we use the parity check equation defined by Q.

We define the sequence

$$c'_t = \bigoplus_{j=1}^{d-1} c_{t-i_j}$$

which is the one Q would have generated if applied as feedback polynomial to a register with initial state $(c_0, \ldots, c_{i_{d-i}})$.

For all $t \in \{i_{d-1}, \ldots, N-1\}$ let

$$z_t = c_t \oplus c'_t$$

be the sequence which in every instant checks if $c_t = c'_t$.

If we define $p := P(c_t \neq c'_t) = P(z_t = 1)$, for all t the variable z_t is a Bernoulli of parameter p, $z_t \sim \mathcal{B}(p)$.

To evaluate the correlation between c_t and c'_t, we compute $Z' = \sum_{t=i_{d-1}}^{N-1} z_t$, which just counts how many times $c_t \neq c'_t$.

The variables z_t being independent and identically distributed, for large values of $N - i_{d-1}$, by the Central Limit Theorem, Z' can be approximated by a Gaussian with mean $\mu' = (N - i_{d-1})p$ and variance $\sigma'^2 = (N - i_{d-1})p(1-p)$.

To better distinguish between good and bad cases, instead of Z' we introduce the variable

$$Z = \sum_{t=i_{d-1}}^{N-1} (-1)^{z_t}$$

whose standard deviation is greater than Z''s one.

Z, in fact, varies between $i_{d-1} - N$, when for all t $c_t \neq c'_t$, and $N - i_{d-1}$, when for all t $c_t = c'_t$. We can write it as

$$\begin{aligned} Z &= \sum_{t=i_{d-1}}^{N-1} (-1)^{z_t} = \sum_{t=i_{d-1}}^{N-1} (1 - (1 - (-1)^{z_t})) = \\ &= \sum_{t=i_{d-1}}^{N-1} 1 - \sum_{t=i_{d-1}}^{N-1} (1 - (-1)^{z_t}) = N - i_{d-1} - 2 \sum_{t=i_{d-1}}^{N-1} z_t \end{aligned}$$

since $1 - (-1)^{z_t} = \begin{cases} 0 & \text{if } z_t = 0 \\ 2 & \text{if } z_t = 1 \end{cases} = 2z_t$.

It should be evident now that also Z can be approximated by a Gaussian, but stretched in the interval $(i_{d-1} - N, N - i_{d-1})$ and with mean and variance respectively

$$\begin{aligned} \mu &= N - i_{d-1} - 2\mu' = N - i_{d-1} - 2(N - i_{d-1})p \\ &= (N - i_{d-1})(1 - 2p) \\ \sigma^2 &= 4\sigma'^2 = 4(N - i_{d-1})p(1 - p) \end{aligned}$$

Now, let $\hat{S} \in \mathcal{S}(Q)$ be the sequence in $\mathcal{S}(Q)$ with the greatest correlation with the ciphertext C. $\hat{S}$ can be defined as the sequence in $\mathcal{S}(Q)$ maximizing the distance

$$\left|\left|(P(c_t = s_t), P(c_t \neq s_t)) - \left(\frac{1}{2}, \frac{1}{2}\right)\right|\right|$$

or, equivalently, the one maximizing

$$\left|P(c_t = s_t) - \frac{1}{2}\right|$$

If we imagine the ciphertext C to be a noisy version of $\hat{S}$, we can write

$$c_t = \hat{s}_t \oplus \epsilon_t \quad \text{for all } t$$

where the noise sequence $E = \{\epsilon_t\}$ is distributed as

$$P(\epsilon_t = 0) = P(c_t = \hat{s}_t) = p_\epsilon \quad \text{and} \quad P(\epsilon_t = 1) = P(c_t \neq \hat{s}_t) = 1 - p_\epsilon$$

The condition $c_t \neq c'_t$ can now be written as

$$\begin{aligned} & c_t \neq \bigoplus_{j=1}^{d-1} c_{t-i_j} \\ \Leftrightarrow \quad & \hat{s}_t \oplus \epsilon_t \neq \bigoplus_{j=1}^{d-1} (\hat{s}_{t-i_j} \oplus \epsilon_{t-i_j}) \\ \Leftrightarrow \quad & \hat{s}_t \oplus \epsilon_t \neq \bigoplus_{j=1}^{d-1} \hat{s}_{t-i_j} \oplus \bigoplus_{j=1}^{d-1} \epsilon_{t-i_j} \\ \Leftrightarrow \quad & \epsilon_t \neq \bigoplus_{j=1}^{d-1} \epsilon_{t-i_j} \end{aligned}$$

since $\hat{S} \in \mathcal{S}(Q)$ implies $\hat{s}_t = \bigoplus_{j=1}^{d-1} \hat{s}_{t-i_j}$.

Furthermore, it holds

$$p = P(z_t = 1) = P(c_t \neq c'_t) = P\left(\epsilon_t \neq \bigoplus_{j=1}^{d-1} \epsilon_{t-i_j}\right)$$

The last condition is true if and only if the number of indices $\bar{t} \in \{t, t -$

$i_1, \ldots, t - i_{d-1}\}$ such that $\epsilon_{\bar{t}} = 1$ is odd, so p can be computed as

$$\begin{aligned}
p &= P\left(\epsilon_t \neq \bigoplus_{j=1}^{d-1} \epsilon_{t-i_j}\right) = \sum_{\substack{l=0 \\ l \text{ odd}}}^{d} \binom{d}{l} (1-p_\epsilon)^l p_\epsilon^{d-l} \\
&= \frac{1}{2}\left[2\sum_{\substack{l=0 \\ l \text{ odd}}}^{d} \binom{d}{l}(1-p_\epsilon)^l p_\epsilon^{d-l} + \sum_{\substack{l=0 \\ l \text{ even}}}^{d} \binom{d}{l}(1-p_\epsilon)^l p_\epsilon^{d-l} + \right. \\
&\quad \left. - \sum_{\substack{l=0 \\ l \text{ even}}}^{d} \binom{d}{l}(1-p_\epsilon)^l p_\epsilon^{d-l}\right] \\
&\overset{(1)}{=} \frac{1}{2}\left[\sum_{\substack{l=0 \\ l \text{ odd}}}^{d} \binom{d}{l}(1-p_\epsilon)^l p_\epsilon^{d-l} - \sum_{\substack{l=0 \\ l \text{ odd}}}^{d} \binom{d}{l}(p_\epsilon-1)^l p_\epsilon^{d-l} + \right. \\
&\quad \left. + \sum_{\substack{l=0 \\ l \text{ even}}}^{d} \binom{d}{l}(1-p_\epsilon)^l p_\epsilon^{d-l} - \sum_{\substack{l=0 \\ l \text{ even}}}^{d} \binom{d}{l}(p_\epsilon-1)^l p_\epsilon^{d-l}\right] \\
&= \frac{1}{2}\left[\sum_{l=0}^{d} \binom{d}{l}(1-p_\epsilon)^l p_\epsilon^{d-l} - \sum_{l=0}^{d} \binom{d}{l}(p_\epsilon-1)^l p_\epsilon^{d-l}\right] \\
&\overset{(2)}{=} \frac{1}{2}[1-(2p_\epsilon-1)^d]
\end{aligned}$$

where in (1) we have used

$$(1-p_\epsilon)^l = \begin{cases} -(p_\epsilon-1)^l & \text{when } l \text{ is odd} \\ (p_\epsilon-1)^l & \text{when } l \text{ is even} \end{cases}$$

while in (2) we have used Newton's binomial theorem:

$$\sum_{l=0}^{d} \binom{d}{l} x^l y^{d-l} = (x+y)^d$$

.

We have therefore defined a variable Z which depends on the polynomial Q we're testing. Its distribution varies according as Q is correlated with C or not, since:

1. If every sequence in $\mathcal{S}(Q)$ is independent from C, then neither $\hat{S}$ is correlated with C and $p_\epsilon = \frac{1}{2}$. In this case we can easily check that

also $p = \frac{1}{2}(1 - (2\frac{1}{2} - 1)^d) = \frac{1}{2}$ and then Z is a Gaussian with mean and variance respectively

$$\begin{aligned} \mu_0 &= (N - i_{d-1})\left(1 - 2\frac{1}{2}\right) = 0 \\ \sigma_0^2 &= 4(N - i_{d-1})\frac{1}{2}\left(1 - \frac{1}{2}\right) = N - i_{d-1} \end{aligned}$$

We call this hypothesis $\mathcal{H}_0$.

2. If there exists at least one sequence in $\mathcal{S}(Q)$ correlated with the ciphertext C, then, by definition of $\hat{S}$, $p_\epsilon \neq \frac{1}{2}$, or better the distance $\delta = |p_\epsilon - \frac{1}{2}|$ is the biggest possible one. In this case

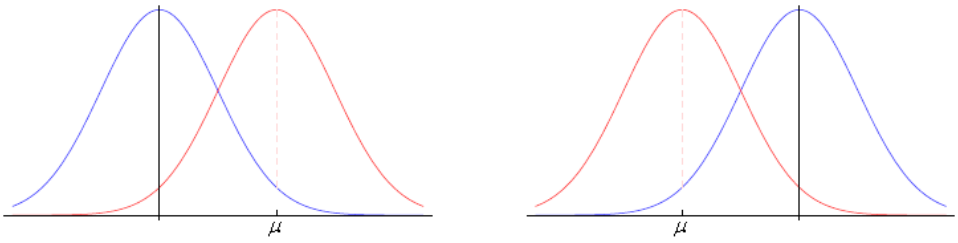
$$p = \frac{1}{2}\left[1 - \left(2\left(\frac{1}{2} \pm \delta\right) - 1\right)^d\right] = \frac{1}{2}(1 - (\pm 2\delta)^d)$$

where the sign before δ is unknown depending on the sign of $p_\epsilon - \frac{1}{2}$. As we have seen, Z has mean and variance given respectively by

$$\begin{aligned} \mu &= (N - i_{d-1})\left(1 - 2\left(\frac{1}{2}(1 - (\pm 2\delta)^d)\right)\right) = (N - i_{d-1})(\pm 2\delta)^d \\ \sigma^2 &= 4(N - i_{d-1})\frac{1}{2}(1 - (\pm 2\delta)^d)\frac{1}{2}(1 + (\pm 2\delta)^d) = (N - i_{d-1})(1 - (2\delta)^{2d}) \end{aligned}$$

We call this hypothesis $\mathcal{H}_1$.

In figure 4.4, we show the two Gaussian bells corresponding to the two hypothesis. As you can see, the situation changes symmetrically according as $p_\epsilon > \frac{1}{2}$, so that $\mu > 0$, or $p_\epsilon < \frac{1}{2}$, so that $\mu < 0$.



(a) $p_\varepsilon > \frac{1}{2}$ (b) $p_\varepsilon < \frac{1}{2}$

Figure 4.4: The Gaussian bell of hypothesis $\mathcal{H}_1$ (red) compared with the one of hypothesis $\mathcal{H}_0$ (blue) in the two cases.

The test consist in setting a threshold T and for every $Q \in \Omega$ compute the corresponding value of Z and compare it with T.

Since when testing we don't know the value of p_ϵ, for convenience we choose to take $T > 0$: if $Z > T$ or $Z < -T$ we assume hypothesis $\mathcal{H}_1$ to be true (supposing, respectively, $\mu > 0$ or $\mu < 0$) while if $-T < Z < T$ we assume as true hypothesis $\mathcal{H}_0$.

We can sum up these conditions as

$$\begin{cases} \mathcal{H}_0 & \text{is true when} \quad |Z| < T \\ \mathcal{H}_1 & \text{is true when} \quad |Z| > T \end{cases}$$

Clearly, in the first case ($\mathcal{H}_0$) we discard Q, while in the second one ($\mathcal{H}_1$) we store it.

As any statistical attack, our test may fail.

In fact, we can have a "false alarm", namely select a wrong polynomial, with probability

$$p_{fa} = P(|Z| > T \mid \mathcal{H}_0) = 2 \int_T^{+\infty} \frac{e^{-\frac{x^2}{2\sigma_0^2}}}{\sqrt{2\pi\sigma_0^2}} dx$$

since in the $\mathcal{H}_0$ case Z is a $\mathcal{N}(0, \sigma_0^2)$ and thanks to its symmetry $P(|Z| > T) = 2P(Z > T)$.

Analogously, we can have a "non-detection", namely discard a good polynomial, with probability

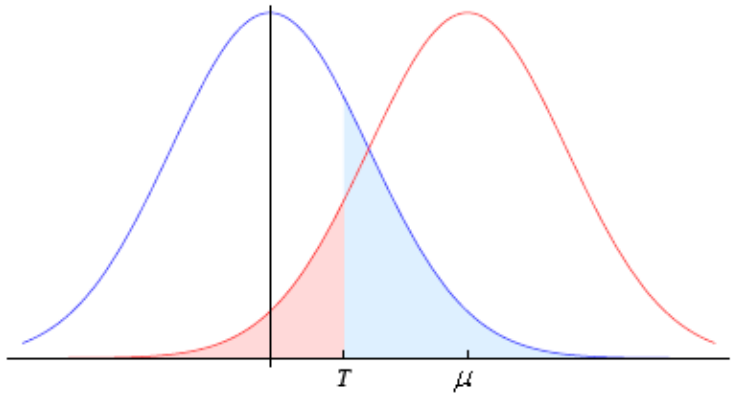
$$p_{nd} = P(|Z| < T \mid \mathcal{H}_1) = \int_{-T}^{T} \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}} dx$$

since in the $\mathcal{H}_1$ case Z is a $\mathcal{N}(\mu, \sigma^2)$ and we don't need to distinguish between $\mu < 0$ and $\mu > 0$ because, as we already noticed, the two cases are perfectly symmetrical.

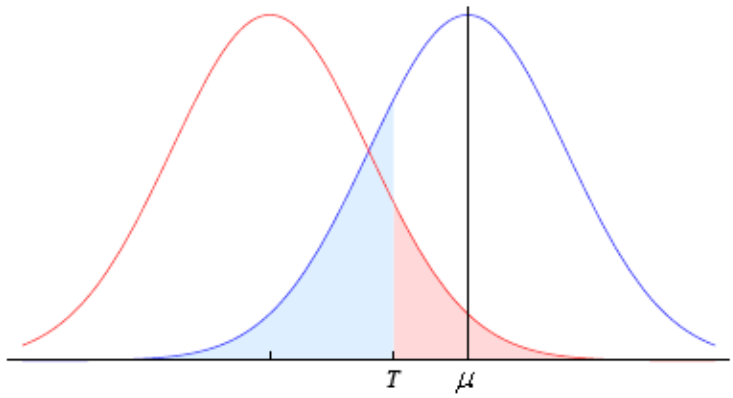
The threshold T must be actually settled to minimize these error probabilities. Unfortunately, as it's clearly visible in figure 4.5, we cannot minimize both p_{fa} and p_{nd}, since to reduce p_{nd} we should move T towards 0, resulting in increasing p_{fa}.

Anyhow, that's exactly what we will do, since for the success of the algorithm it's fundamental not to miss the good candidates. Obviously, we cannot let p_{fa} increase too much, because it would lead to a massive growth of the computational complexity, but keep p_{nd} low is crucial. Hence, we have to choose the optimal compromise between p_{nd} and p_{fa}, set T to keep p_{nd} as low as we want to, then finally take an amount N of ciphertext as big as necessary to limit p_{fa}.

To explicitly compute the values of T and N allowing the algorithm to work, let's first evaluate p_{fa} and p_{nd}.



(a) $p_{\bar{\varepsilon}} > \frac{1}{2}$



(b) $p_{\bar{\varepsilon}} < \frac{1}{2}$

Figure 4.5: The area in red is p_{fa}, while the area in blue is p_{nd}.

Let

$$\Phi(x) = \int_{-\infty}^{x} \frac{e^{-\frac{y^2}{2}}}{\sqrt{2\pi}} dy$$

be the distribution function of the standard normal.

If we standardize the variable, substituting $y = \frac{x}{\sigma_0}$ for hypothesis $\mathcal{H}_0$ and $y = \frac{x-\mu}{\sigma}$ for $\mathcal{H}_1$, we obtain

$$p_{fa} = 2\int_{\frac{T}{\sigma_0}}^{+\infty} \frac{e^{-\frac{y^2}{2}}}{\sqrt{2\pi}} dy = 2\int_{-\infty}^{-\frac{T}{\sigma_0}} \frac{e^{-\frac{y^2}{2}}}{\sqrt{2\pi}} dy = 2\Phi\left(-\frac{T}{\sigma_0}\right) \tag{4.6.2}$$

and

$$p_{nd} = \int_{\frac{-T-\mu}{\sigma}}^{\frac{T-\mu}{\sigma}} \frac{e^{-\frac{y^2}{2}}}{\sqrt{2\pi}} dy = \Phi\left(\frac{T-|\mu|}{\sigma}\right) - \Phi\left(\frac{-T-|\mu|}{\sigma}\right) \approx \Phi\left(\frac{T-|\mu|}{\sigma}\right) \tag{4.6.3}$$

since we don't know if $\mu > 0$ or $\mu < 0$ and in both cases $\Phi\left(\frac{-T-|\mu|}{\sigma}\right)$ is negligible, being $<< \Phi\left(\frac{T-|\mu|}{\sigma}\right)$.

Now we can fix the accepted values for p_{nd} and p_{f_a} and invert equations 4.6.2 and 4.6.3 (substituting the computed values of σ_0, μ and σ) using a Standard Normal Table, obtaining the system

$$\begin{cases} N - i_{d-1} = \left(\frac{-T}{\Phi^{-1}\left(\frac{p_{fa}}{2}\right)}\right)^2 \\ T = \Phi^{-1}(p_{nd})\sqrt{(N-i_{d-1})(1-(2\delta)^{2d})} + (N-i_{d-1})(2\delta)^d \end{cases}$$

Solving the system in the unknowns T and N permits to settle the best values for the parameters necessary for the algorithm.

Usually N is given (it's simply the total amount of available ciphertext), so the second equation is used to settle T, while the first one only allows to check if we have enough ciphertext digits to reduce p_{fa} as desired.

4.6.3 The algorithm

We can now expose the algorithm for the recovery of the feedback polynomials.

Since δ was defined somehow to measure the "distance" in the distribution of Z between hypothesis $\mathcal{H}_0$ and $\mathcal{H}_1$, we can fix a minimum accepted value δ_{min}. The following algorithm then examines all polynomials of degree at most D and of weight d, detecting every polynomial Q in this set such that $|P(c_t \neq c'_t) - \frac{1}{2})| \geq \delta_{min}$.

Algorithm 4.1 (Reconstruction of the registers over $\mathbb{F}_2$).

Require: N and ciphertext $C = c_0 c_1 \dots c_{N-1}$
Choose: a value p_{nd} for the probability to miss a good candidate
Choose: a value p_{fa} for the probability to select a wrong a polynomial
Choose: a minimum value δ_{min}
Choose: two parameters d and D
for each $(d-1)$-tuple $(i_1, \dots, i_{d-1})$ such that $0 < i_1 < \dots < i_{d-1} \leq D$ **do**
  $N' \leftarrow N - i_{d-1}$
  $T \leftarrow \Phi^{-1}(p_{nd})\sqrt{N'(1-(2\delta_{min})^{2d})} + N'(2\delta_{min})^d$
  $Z \leftarrow 0$

for $t = i_{d-1}, \ldots, N-1$ **do**
  $c'_t = \sum_{j=1}^{d-1} c_{t-i_j}$
  if $c_t = c'_t$ **then**
    $Z \leftarrow Z + 1$
  else
    $Z \leftarrow Z - 1$
  end if
end for
if $|Z| \geq T$ **then**
  factorize $Q(X) = 1 + \sum_{j=1}^{d-1} X^{i_j}$ and store the factors
end if
end for

4.6.4 Recovery of the combining function

Once we have implemented algorithm 4.2, the complete recovery of the coefficients of the combining function can be easily achieved.

First, the reconstruction of the registers tells us the value of n, namely the number of variables of the combining function f.

Now, any polynomial we have detected is a multiple of some feedback polynomials of the registers, thus it's of the form

$$Q = R \cdot \prod_{i \in I} P_i, \quad I \subseteq \{1, \ldots, n\}$$

where R is another polynomial in $\mathbb{F}_2[X]$.

Take any set of indices I and suppose we have collected n_I multiples of $\prod_{i \in I} P_i$, denoted Q_j for $j = 1, \ldots, n_I$.

For any Q_j we have computed and stored the corresponding value Z_j of the variable Z. The arithmetic mean of those values gives us an esteem of the expected value of Z:

$$\frac{\sum_{j=1}^{n_I} Z_j}{n_I} \approx \mathbb{E}(Z) = N(2p_\epsilon - 1)^d$$

where $p_\epsilon = P(c_t = s_t)$ with $s_t = g(s_t^1, \ldots, s_t^n)$ and $g(x) = 1_I \cdot x$[1].

We can thus esteem the value of p_ϵ and use it to esteem the value of $p_g = p(f = g)$, since by proposition 4.2 we know that

$$p_\epsilon = 1 - p_0 - p_g + 2p_0p_g$$

where $p_0 = P(m_t = 0)$.

Finally, from identity 2.2.7, we have

$$p_g = \frac{1}{2}\left(1 + \frac{\hat{\chi}_f(1_I)}{2^n}\right)$$

[1] 1_I is the vector in $\mathbb{F}_2^n$ whose i-th component is 1 if and only if $i \in I$

and we can quite precisely esteem the value of $\hat{\chi}_f(1_I)$ since the Walsh coefficients of a balanced boolean function are known to be multiple of 4.

We have thus shown how to completely recover the Walsh spectrum of the function f. The coefficients of its ANF can be easily evaluated thanks to proposition 2.11:

$$a_u = 2^{w_H(u)-1} \left(1 - \frac{1}{2^n} \sum_{v \preceq \bar{u}} \hat{\chi}_f(u) \right) \mod 2$$

Bibliography

[1] Berlekamp, E.R. : *Algebraic Coding Theory*, McGraw-Hill, New York, 1968.

[2] Canteaut, A., and Filiol, E. : *Ciphertext Only Reconstruction of Stream Ciphers Based on Combination Generators*, in: Proceedings of Fast Software Encryption 2000, ed. by SCHNEIER, B., Lecture Notes in Computer Science 1978, Springer Verlag, 2001.

[3] Canteaut, A., and Filiol, E. : *Ciphertext Only Reconstruction of LFSR-based Stream Ciphers*, Rapport de Recherche INRIA, n 3887, February 2000, Institut National de Recherche en Informatique et Automatique, Domaine de Voluceau, B.P. 105, 78153 Le Chesnay cédex.

[4] Diffie, W., and Hellman, M.E. : *New Directions in Cryptography*, IEEE Transactions in Information Theory, Vol. IT-22, Nr. 6, 1976, pp 644-654.

[5] Filiol, E. : *Techniques de Reconstruction en Cryptologie et Théorie des Codes*, Ph. D Thesis, Ecole Polytechnique, France 2001.

[6] Geffe, P.R. : *How to Protect Data with Ciphers that Are Really Hard to Break*, Electronics, pp 99-101, 1973.

[7] Golomb, S.W. : *Shift Registers Sequences*, Aegean Park Press, 1982.

[8] Herlestam, T. : *On the complexity of certain crypto generators*, in : Security, IFIP/Sec'83, North Holland, 1983.

[9] Kerckhoffs, O. : *La Cryptologie Militaire*, Louis BAUDOIN ed., Paris, 1883.

[10] Lidl, R., and Niederreiter, H. : *Finite Fields*, Encyclopedia of Mathematics and its Applications, **20**, Cambridge University Press, (2008).

[11] Meier, W., and Staffelbach, O. : *Fast Correlation Attack on Certain Stream Ciphers*, Journal of Cryptology, pp 159-176, 1989.

[12] Meier, W., and Staffelbach, O. : *Nonlinearity Criteria for Cryptographic Functions*, in: Advances in Cryptology - EUROCRYPT '89,

Lecture Notes in Computer Science 434, pp 549-562, Springer Verlag, 1990.

[13] Rueppel, R.A., and Staffelbach, O. : *Products of Linear Recurring Sequences with Maximum Complexity*, IEEE Transactions on Information Theory, Vol. 33 (1), pp 124-131, 1987.

[14] Shannon, C.E. : *A Mathematical Theory of Communication*, Bell System Journal, Vol. 27, pp 379-423 (Part I) and pp 623-656 (Part II), 1948.

[15] Shannon, C.E. : *Communication Theory of Secrecy Systems*, Bell System Technical Journal, Vol. 28, Nr. 4, pp 656-715, 1949.

[16] Siegenthaler, T. : *Correlation Immunity of Nonlinear Combining Functions for Cryptographic Applications*, IEEE Transactions on Information Theory, Vol. 35, Nr. 5, September 1984, pp 776-780.

[17] Siegenthaler, T. : *Decrypting a Class of Stream Ciphers Using Ciphertext Only*, IEEE Transactions on Computers, C-34, 1, pp 81-84, 1985.

[18] Van Oorschot, P., Menezes, A., and Vanstone, S. : *Handbook of Applied Cryptography*, CRC Press, fifth printing edition (2001). http://www.cacr.math.uwaterloo.ca/hac/

[19] Xiao, G.-Z., and Massey, J.L. : *A Spectral Characterization of Correlation-Immune Combining Functions*, IEEE Transactions on Information Theory, Vol. IT-34, Nr. 3, pp 569-571, 1988.

[20] http://en.wikipedia.org/