Stream Ciphers and Linear Complexity

Hu Qi

April 2007

Acknowledgements

As a graduate student of Mathematics Department in National University of Singapore, I have quite a fruitful and happy experience. My life would not be the same without my supervisor and friends.

Many thanks to my supervisor, Professor Harald Niederreiter. Being a leading researcher, he is more like an intelligent gentleman for me. I am grateful for Professor Niederreiter's directions on my research and study, encouragement on my work, great patience and kindly help for my thesis and graduate application.

I also would like to express my sincere thanks to Professor Alan Jon Berrick, Professor Tay Yong Chiang, Professor Chin Chee Whye, and Professor Ma Siu Lun. They are very nice. I really appreciate their instructions on my mathematics study and research, advices for my future academic development and warm help.

My great gratitude goes to my parents. Many thanks for their encouragement and support which make me strong.

Finally, many thanks to my good friends although their names are not mentioned here. It is them that make my life colorful, interesting and happy.

Contents

List of Figures

Abstract

The thesis mainly reviews the mathematical analysis of the security of stream ciphers. Firstly, we will introduce the background of stream ciphers with their design principles and theoretical security from the information theory viewpoint. Then we will introduce the algebraic tools for the analysis of linear recurring sequences followed by discussions on the two kinds of basic nonlinear filters. Also, we will discuss the randomness of the sequences over $\mathbb{F}_q$ and justify the use of linear complexity profiles to measure the randomness of the key streams generated by linear feedback shift registers. An exploration of the probabilistic properties of sequences over $\mathbb{F}_q$ is included too. Finally, we define an important parameter k-error linear complexity to measure the security of the key streams and discuss the lower bounds for this parameter of periodic sequences over $\mathbb{F}_q$.

Chapter 1

Stream Ciphers and Their Realizations

In this chapter, we will give a general picture of stream ciphers. Contents include their brief history, basic principles and their theoretical background from the viewpoint of information theory. In the end, we will introduce some realizations of stream ciphers published in the open literature.

1.1 Introduction to Stream Ciphers

Beginning with a short summary of the development of stream ciphers in the past 50 years, we will introduce the basic concepts and ideas of stream ciphers, including the design principles, the synchronous and self-synchronous problems and how the key generators work.

1. The Brief History of Stream Ciphers

The birth of stream ciphers should be attributed to the invention of electronic communication technologies. At the end of the 19th century, several scientists such as Tesla and Marconi contributed to the invention of "radio", which was called "wireless telegraphy" at that time. And after about 30 years of development, radio technologies became more mature and began to be used in the daily life of humans. The long distance communication changed a lot compared to the 19th century because humans could use radio to deliver electronic signals carrying messages instantly. In about 1920, radio was already widely used in both military and commercial areas to exchange information.

Because the information was transmitted by electronic signals in the open air, everyone who had a receiver could get these signals and translate them to plaintext easily. So it was not secure to deliver secret messages directly by radios. Then, the rotor-based electromechanical encryption device was introduced and adopted all over the world (see Shamir [33]) to solve this secret information exchange problem. However, memories in these devices were quite expensive so that they only could have internal states being kept. But as for the user data, the machine itself could not store them. It was cost that forced the encryption being processed character by character.

In 1949, Shannon proved the perfect security of the Vernam one-time pad cipher in his famous paper [36]. So from that time, it was known as the theoretically unbreakable cryptosystem. The effect of Shannon's paper in 1949 was that the support and popularity of stream ciphers increased dramatically. In a long period since that time, almost all units in the world, such as the military and diplomatic services, commercial and spy organizations, and telecommunication providers, used stream ciphers to exchange their secret messages.

In the 1960s, transistor-based encryption devices were introduced. They were fast but still had little memories. Computers also were invented and had applications in that time. But they were more used in cryptanalysis than in cryptography. So steam ciphers, encrypting each character of the plaintext due to the unavailability of external memories, remained popular. Another milestone of stream ciphers is the invention of the linear feedback shift register. Then the stream ciphers could be precisely analyzed and controlled by mathematical theory. Lots of research was done in the following decades since LFSR was invented. Also it could be implemented and computed easily and fast. Therefore, stream ciphers continued their popularity. Until now, most military and diplomatic organizations still keep their tradition to use stream ciphers to exchange important and top secret intelligence, despite the great popularity of block ciphers in modern commercial areas.

Nowadays, lots of research efforts are devoted to stream ciphers. Generally, they could be divided into two parts. The first one is the research for military and diplomatic purposes. Scholars serving for this purpose conduct their research, develop and analyze their stream cipher cryptosystems without publishing results in the open literature. Although we have no information about their research, we are sure of one thing: there are many cryptography researchers supported by their nations' special foundations and the departments of defence. The second part is the open research. Besides lots of individual stream cipher cryptographers in universities and industrial companies, there are some research organizations that attract worldwide top cryptographers to design new stream cipher cryptosystems, analyze their security, and contribute to their applications. A good example is the ECRYPT Stream Ciphers Project. Recently, the demand for standardized stream ciphers increased.

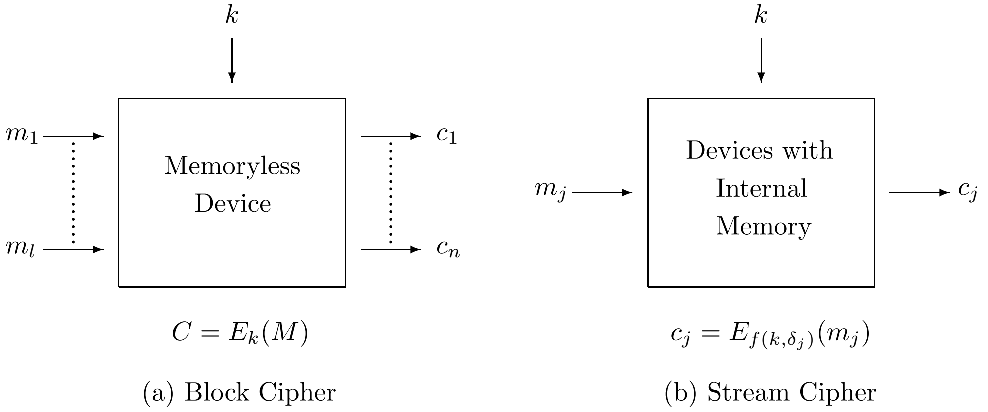

Figure 1.1: The basic two enciphering principles.

Therefore, stream ciphers will be a hot topic and attract lots of research attention in the coming years. And we believe that in the near future, there will be some standardized stream cipher that is widely used just as AES for block ciphers.

2. Basic Principles

Generally speaking, symmetric cryptosystems are divided into two types: block ciphers and stream ciphers. Block ciphers operate an enciphering transformation on each "message block" independently, for example every 64-bit string in DES. In contrast, stream ciphers encipher each character of the message with a time-varying function to control its internal state. The most obvious distinction between block and stream ciphers is "memory", described by Figure 1.1 [32, Chapter 2].

A block cipher breaks the plaintext $M = (m_1, m_2, \ldots, m_l)$ into a number of message blocks with the same length and transforms them to the ciphertext $C = (c_1, c_2, \ldots, c_n)$ via an encryption function controlled by a secret key k. The encryption function is memoryless, which means that the current ciphertext block depends only on the current input message block and k. In fact, the encryption functions are *permutations* [3, Theorem 3.6.2]. On the other hand, the stream cipher transforms each character of the plaintext m_j to the ciphertext c_j with an encryption function having several internal memories, which implies that the current jth state is decided by several previous states denoted by δ_j and the key k. Since each enciphering step of the stream cipher is controlled by time-varying parameters, even two identical plaintext characters do not have the same ciphertext characters in general. However, the encryption function of a block cipher may map two identical

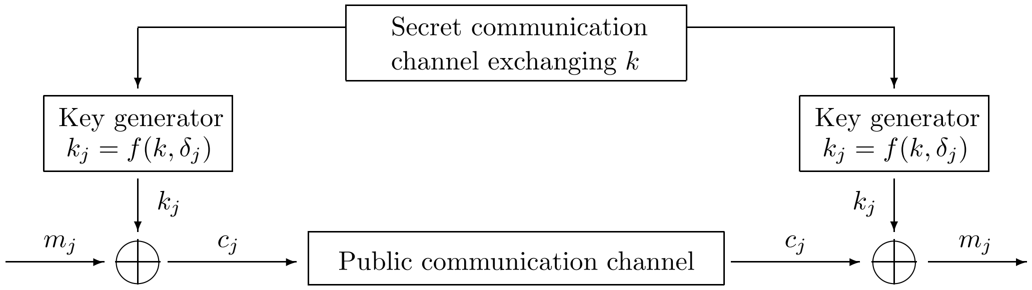
Secret communication channel exchanging k
Key generator $k_j = f(k, \delta_j)$
Key generator $k_j = f(k, \delta_j)$
k_j
k_j
m_j
c_j
Public communication channel
c_j
m_j

Figure 1.2: The basic working principle of stream cipher.

plaintext characters to the same objects. Therefore, the hackers may break some ciphertext by comparing it with some ciphertext whose plaintext is already known. They also could obtain some information about the plaintext by injecting, deleting, or replaying some ciphertext. To avoid possible attacks against block ciphers caused by the above defect, the block ciphers are usually implemented by encryption functions with additional memories, such as *electronic codebook mode, cipherblock chaining mode, cipher feedback mode, output feedback mode*, etc. For details, please refer to [3, Section 3.8].

The encryption function $E_{f(k,\delta_j)}$ used in stream ciphers is usually realized by a simple operation: addition. For the convenience of implementation by hardware, the addition is usually embedded in a field whose characteristic is 2. So one can easily see that the encryption and decryption of stream ciphers are symmetric. The former works this way: the cryptosystem generates a key stream by $f(k, \delta_j)$ for $j = 1, 2, \ldots$. Here k is the secret key and δ_j represents several previous states deciding the current jth state of the key stream. Then each ciphertext character is given by $c_j = m_j \oplus f(k, \delta_j)$. The procedure of decryption is almost the same and the plaintext is obtained by $m_j = c_j \oplus f(k, \delta_j)$. To get a clearer picture of how it works, please see the Figure 1.2.

From the working procedures of stream ciphers, one can see that anyone having the key stream could break the ciphertext easily because of the simplicity and symmetry of its encryption and decryption functions. Therefore, for stream ciphers, the security of the ciphertext relies on the secrecy of the key stream. So there are two basic requirements on its security. The first one is that: from previous states of the key stream, one cannot compute the next states of the key stream in a short time for practical purposes. Any tools or methods used to predict the following states of the key stream are no better than random guessing. Why do we require this property? This is because it is easy to obtain the old states of a key stream after they were used. Now suppose a key stream was used to exchange intelligence in a local war. As the ciphertexts were transmitted in the public communication channel, anyone would have them without any difficulty if he/she wanted. Also some powerful men, or even ordinary people would get its corresponding plaintext after a relatively short time, for example the secret military intelligence was released right after the war. So these people could have the old states of the key stream just by adding the ciphertext and plaintext. If from the previous states of the key stream, the hackers could obtain some information about the key generator to guess the next states of the key correctly with a high probability, then the stream ciphers obviously would be no longer secure to exchange secret information in the future. That is why the first requirement is imposed. The second requirement is that: the secret communication channel must be perfectly secret, which means that only the two parties exchanging the secret information have the secret key. Obviously, this is a universal requirement for all symmetric cryptosystems. Cryptographers are usually more concerned about the first requirement. They try their best to design fast and secure key stream generators, so that breaking the ciphertext by usual cryptanalysis techniques is difficult and time consuming, especially for practical purposes. Most successful breakings of the cipher, as history has shown, are due to the violation of the second requirement caused by humans leaking secrets and bad administrative procedures, especially in the key management domain [40, Chapter 2].

3. Synchronous and Self-Synchronous Stream Ciphers

The different ways to produce key streams specify two kinds of stream ciphers: synchronous stream ciphers and self-synchronous stream ciphers. Differences between them come from the parameters determining the current state of the key stream.

For synchronous stream ciphers, the current state is dependent on several previous key stream states, but independent of the previous ciphertext characters. This implies that the key stream generation is independent of the ciphertext transformation and can be carried out separately both at the sender's and the receiver's ends. So when the Party A enciphers a message using a synchronous stream cipher and sends the ciphertext to the receiver, Party B, it must build a synchronization to guarantee the successful communication. More exactly, when Party A sends the ciphertext character $c_j = m_j \oplus k_j$ to Party B, Party B must generate the corresponding key stream k_j to obtain m_j at the same time. If some error happens during the transmission of c_j, say it is lost or changed, Party A and Party B must again set up their synchronization for communication.

It is easy to imagine that the rebuilding of the synchronization involves quite complex procedures. Whenever the synchronization is lost, the receiver, Party B, has two options to deal with it. One is to search all the possible previous states of the key stream and try to figure out the state at which the synchronization was lost, then compute the key stream from that state. The other one is to contact the sender, Party A, then require the sender to resend the ciphertext from some state where they could synchronize again. So no matter which method is chosen, the resynchronization requires either lots of searching and computation or a number of additional communication data.

However, this major disadvantage of synchronous stream ciphers is simultaneously a defence against almost all active attacks on symmetric cryptosystems [32, Section 2.3]. Active attacks such as *injection, deletion, replay of ciphertext*, must lead to the loss of synchronization. Therefore, the receiver and then the sender will be notified instantly that there might be an active attack from a third party. If some hackers who could wiretap the public communication channel change some characters of the ciphertext to simulate the transmission errors caused by the communication channel, then either infrequent substitutions will be corrected by the coding systems (nowadays, almost all the data transmission hardware adopts the error-correcting coding systems), or too many substitutions will be notified by the sender and receiver by the failure of transmissions, which is caused by the number of errors exceeding the system's tolerance. Therefore, these possible active attacks on symmetric cryptosystems all fail when they are applied to synchronous stream ciphers. Hence, we see a tradeoff between the security and the difficulty of synchronization. For the sake of top security, it is worth the senders' and the receivers' great efforts to build the synchronization between them.

As for the self-synchronous stream ciphers, the current state of the key stream is decided by several previous characters of the ciphertext. Say the number of deciding characters being n, and the key being k. The most common mode of self-synchronous stream ciphers is *cipher feedback mode.* So with a key stream generating function f, each character of key stream is given by: $k_j = f(k, c_{j-1}, c_{j-2}, \ldots, c_{j-n})$. See Figure 1.3 to understand how it works.

If a character of the ciphertext is lost or changed during the transmission, the error propagates will forward to n characters in the key stream. Until another n correct ciphertext characters are received, the sender and the receiver could re-establish the synchronization. Compared to synchronous stream ciphers, the self-synchronous stream ciphers can be only slightly immune to active attacks [32, Section 2.3], such as injection, deletion and replay

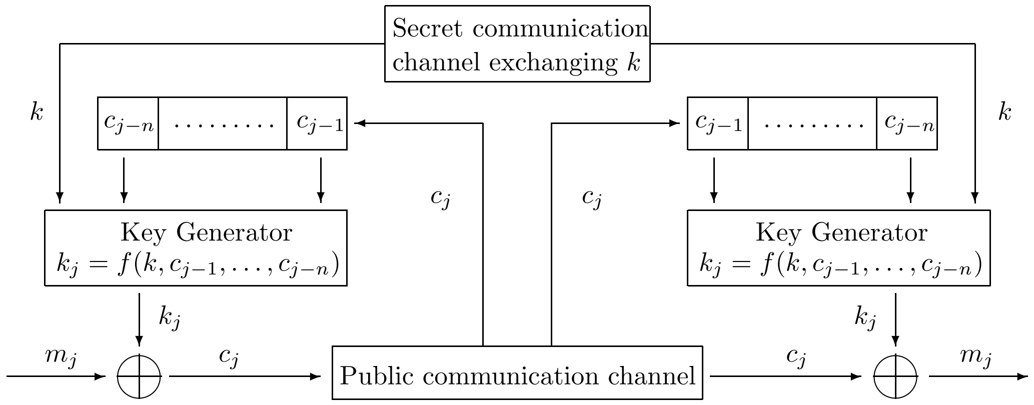
Secret communication channel exchanging k
k
c_{j-n}
....
c_{j-1}
c_j
c_j
c_{j-1}
....
c_{j-n}
k
Key Generator $k_j = f(k, c_{j-1}, \ldots, c_{j-n})$
Key Generator $k_j = f(k, c_{j-1}, \ldots, c_{j-n})$
k_j
k_j
m_j
c_j
Public communication channel
c_j
m_j

Figure 1.3: The self-synchronous stream ciphers.

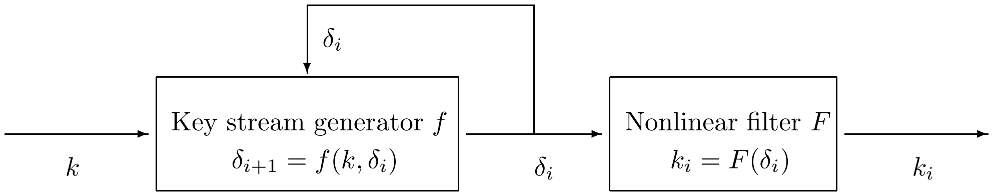
δ_i
k
Key stream generator f $\delta_{i+1} = f(k, \delta_i)$
δ_i
Nonlinear filter F $k_i = F(\delta_i)$
k_i

Figure 1.4: The working principle of the key stream generator.

of the ciphertext, therefore their security level is somewhat lower. And since their key streams depend on the plaintext, more exactly on the ciphertext, there is a big limit on the analyzability of self-synchronous stream ciphers.

4. The Key Stream Generator

As one can see from the above discussion, the key stream generator is the core part in a stream cipher cryptosystem. So the understanding of key stream generators' working principles is quite important. Generally, the key stream generator consists of one or several finite state machine/s, for example, *linear feedback shift register*, and a nonlinear filter. After initialization, the finite state machine f, controlled by the secret key k, is input a current state δ_i and maps it to the next state $\delta_{i+1} = f(k, \delta_i)$. The nonlinear filter F maps each state to an element $k_i = F(\delta_i)$ in its embedded field and outputs it as a character of the key stream. See Figure 1.4.

Since the outputs of every finite state machine are ultimately periodic (it will be proved in Chapter 3), any key stream generators could be implemented by Linear Feedback Shift Registers (LFSRs) for the convenience of hardware. To achieve a high level of security, a good nonlinear filter F is needed because LFSRs are linear devices and could be analyzed easily. Therefore the design of good nonlinear filters has to satisfy several crucial requirements [32, Section 2.2] for the top security. Call the sequences generated by the finite state machine the periodic driving sequences. Now we list these requirements:

1. F transfers the statistical properties of the periodic driving sequences to the key stream.

2. F maximizes the period of the key stream compared to the periods of the driving sequences.

3. F maximizes the linear complexity of the key stream.

4. F does not leak, which means that it is immune to modularizing attack.

5. F is easy to implement and can be computed fast.

6. F should easily be controlled by the key k.

Under known plaintext attacks, the security of stream ciphers relies on the key stream. So the basic idea to design the keystream generator is making the key stream unpredictable. The above requirements are necessary to guarantee the unpredictability. To get a more concrete picture of the key generators and nonlinear filters, one could refer to the last section in this chapter for some real examples and Chapter 2 for principles.

1.2 Theoretical Security of One-Time Pad Ciphers

The popularity of stream ciphers in military and diplomatic organizations is due to the *perfect security* of one-time pad ciphers. In fact, the core criterion of a secure key stream, the unpredictability, is the main characteristic of one-time pad cipher's key stream. After understanding the mathematical fundamentals of the theoretical security of Vernam one-time pad ciphers, one could clearly realize the design principles of stream ciphers.

1. Perfect Security

Let $\mathcal{M}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}$ be the finite plaintext space, the finite ciphertext space, the finite key space, the finite family of the encryption functions and the finite family of the decryption functions, respectively. The encryption and decryption functions with key k are denoted by E_k and D_k, respectively. Suppose M and K represent real-valued random variables on $\mathcal{M}$ and $\mathcal{K}$, respectively, and let $P_M : \mathcal{M} \to [0,1]$, $P_C : \mathcal{C} \to [0,1]$, $P_K : \mathcal{K} \to [0,1]$ be the probability maps. For the simplicity of notations, use P_m, P_c and P_k to represent the probability of the plaintext $M = m$ for $m \in \mathcal{M}$, the ciphertext $C = c$ for $c \in \mathcal{C}$ and the probability of the key $K = k$ in the key space $k \in \mathcal{K}$, respectively. Use $H(X)$ to denote the entropy of the random variable X in the space $\mathcal{X}$.

Definition 1.1 A symmetric cryptosystem $(\mathcal{M}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is called *perfectly secure* (or has *perfect security*) if $H(M|C) = H(M)$ for every probability distribution P_M.

Therefore from the definition, in a perfectly secure cryptosystem, knowing the ciphertext distribution does not help to lower the uncertainty of the plaintext, i.e, observing the ciphertext via a public communication channel does not help the hackers to derive any information on the plaintext. This would be the ideal situation for the sender and receiver since the only information hackers could get, the ciphertext, would not leak any information about the perfectly secure cryptosystem. So the next task is to find some perfectly secure cryptosystem.

Since we suppose $\mathcal{M}$ and $\mathcal{C}$ are both finite spaces, we can assume that $P_{m_i} > 0$ and $P_{c_j} > 0$ for all $m_i \in \mathcal{M},\ c_j \in \mathcal{C}$. By the definition of entropy, $H(M) = \sum_{i=1}^{p} P_{m_i} \log_2(\frac{1}{P_{m_i}})$. Therefore for some c_j, we have $H(M|c_j) = \sum_{i=1}^{q} P_{m_i|c_j} \log_2(\frac{1}{P_{m_i|c_j}})$. Then, the *conditional entropy* $H(M|C)$ is defined to be the weighted average of the conditional uncertainty of M

given that $C = c_j$:

$$H(M|C) = \sum_{j=1}^{q} P_{c_j} H(M|c_j) = \sum_{j=1}^{q}\sum_{i=1}^{p} P_{c_j} P_{m_i|c_j} \log_2(\frac{1}{P_{m_i|c_j}}) = \sum_{j=1}^{q}\sum_{i=1}^{p} P_{(m_i,c_j)} \log_2(\frac{1}{P_{m_i|c_j}}),$$

where $P_{(m_i,c_j)}$ represents the probability of m_i being enciphered to c_j. Since the *joint entropy* is given by $H(M,C) = \sum_{i=1}^{p}\sum_{j=1}^{q} P_{(m_i,c_j)} \log_2(\frac{1}{P_{(m_i,c_j)}})$ and $P_{(m_i,c_j)} = P_{c_j} P_{m_i|c_j}$, we have:

$$H(M,C) = \sum_{i=1}^{p} P_{m_i} \log_2(\frac{1}{P_{m_i}}) + \sum_{i=1}^{p}\sum_{j=1}^{q} P_{(m_i,c_j)} \log_2(\frac{1}{P_{c_j|m_i}}) = H(M) + H(C|M).$$

By the same argument, one could easily get $H(M,C) = H(C) + H(M|C)$. Now, define the *system mutual information* $I(M;C) = H(M) - H(M|C)$. Then $I(M;C) = H(M) + H(C) - H(M,C)$. Therefore, we have:

$$\begin{aligned} I(M;C) &= \sum_{i=1}^{p} P_{m_i} \log_2(\frac{1}{P_{m_i}}) + \sum_{j=1}^{q} P_{c_j} \log_2(\frac{1}{P_{c_j}}) - \sum_{i=1}^{p}\sum_{j=1}^{q} P_{(m_i,c_j)} \log_2(\frac{1}{P_{(m_i,c_j)}}) \\ &= \sum_{i=1}^{p}\sum_{j=1}^{q} P_{(m_i,c_j)} [\log_2 P_{(m_i,c_j)} - \log_2 P_{m_i} - \log_2 P_{c_j}] \\ &= \sum_{i=1}^{p}\sum_{j=1}^{q} P_{(m_i,c_j)} \log_2[\frac{P_{(m_i,c_j)}}{P_{m_i} P_{c_j}}]. \end{aligned}$$

By the *Gibbs inequality*, we have the $I(M;C) \geq 0$, and the equality holds *only* when $P_{(m_i,c_j)} = P_{m_i} P_{c_j}$ for all i,j. This means that the cryptosystem is perfectly secure *if and only if* M and C are independent. So the probabilistic independence between the plaintext and ciphertext is the core of perfect security.

2. The Perfect Security of the Vernam One-Time Pad Cipher

The famous cryptosystem *Vernam one-time pad cipher* was invented and patented in 1917 by Gilbert Vernam. But its perfect security was not proved until 1949 by Shannon in [36]. Say the finite meaningful message space $\mathcal{M}$ consists of strings with length n and each character is embedded in the binary field $\mathbb{F}_2$, then $\mathcal{M}$ is a subset of $\mathbb{F}_2^n$. So is the key space $\mathcal{K}$ and the ciphertext space $\mathcal{C}$. The protocol of the Vernam one-time pad cipher is:

Input: A plaintext $m_i = (x_1^i, x_2^i, \ldots, x_n^i)$; a key $k_t = (k_1^t, k_2^t, \ldots, k_n^t)$, whose bits are independent and uniformly distributed.
Output: The corresponding ciphertext $c_j = (x_1^i \oplus k_1^t, x_2^i \oplus k_2^t, \ldots, x_n^i \oplus k_n^t)$.

Since each character of the key stream is independent and uniformly distributed, then $P_{k_t} = \dfrac{1}{2^n}$ and $\mathcal{K} = \mathbb{F}_2^n$. Then for any $m_i \in \mathcal{M}$, $m_i \oplus \mathcal{K} \doteq \{m_i \oplus k$, for all $k \in \mathcal{K}\}$ must be equal $\mathcal{K}$, which means the ciphertext space $\mathcal{C}$ is equal to $\mathbb{F}_2^n$. So for any m_i, c_j, there is exactly one key k_t such that $c_j = m_i \oplus k_t$. Moreover, $\mathcal{M}, \mathcal{K}, \mathcal{C}$ are all finite, so $P_{m_i} > 0, P_{c_j} > 0, P_{k_t} > 0$ for any $m_i \in \mathcal{M}, c_j \in \mathcal{C}, k_t \in \mathcal{K}$.

Firstly $P_{(m_i,c_j)} = P_{m_i} \times \frac{1}{2^n}$ because the probability of the key $k = m_i \oplus c_j$ is $\frac{1}{2^n}$ and choosing this key is independent of the plaintext content. Although here we represent the key k by c_j, it is choosing the key first and then the ciphertext is decided. Next, $P_{c_j} = \sum_{e=1}^{N} P_{m_e} \times P_{k_e}$, where $m_e \oplus k_e = c_j$. Since we already know the existence and the uniqueness of the key k such that $c = m \oplus k$ for any $m \in \mathcal{M}, c \in \mathcal{C}$, m_e must go through all elements in $\mathcal{M}$. Since $P_{k_e} = \frac{1}{2^n}$, we have that $P_{c_j} = \frac{1}{2^n}(\sum\limits_{e} P_{m_e}) = \frac{1}{2^n}$. Therefore, $P_{(m_i,c_j)} = P_{m_i}P_{c_j}$ for all i, j, which means $H(M) = H(M|C)$. Hence we have proved that the Vernam one-time pad cipher is perfectly secure.

The perfect security of the Vernam one-time pad cipher relies on the key stream, a sequence of independent and uniformly distributed bits. So simulating a "random looking" sequence, whose bits are uniformly and independently distributed, is the main task of stream cipher key generator design. But it is quite a hard mission. Firstly, there is no definite mathematical definition of true randomness. The best situation is that we could define *pseudorandomness* to serve some special purpose, which we will discuss in Chapter 3. Secondly, any concrete algorithm must be deterministic, but the deterministic procedures cannot produce truly independent outputs since any current procedure must rely on previous procedures. Therefore, until now, no one could design a device to produce a truly random sequence. Moreover, the keystream should be at least as long as the plaintext and each key should be used only once (this is because repeatedly using a key k will render the non-first-time encryption insecure under the known plaintext attacks and replay attacks). Then the exchanging of the private key becomes difficult due to its big data volume and costs a lot because of its one-time use. If the sender and the receiver have an ideal secret channel to exchange the keys of the Vernam one-time pad cipher, then would it not be a better idea to deliver the secret messages directly via this channel? So we say the perfect security of the Vernam one-time pad cipher is theoretical. But this ideal model informs cryptographers that they should try to simulate random sequences to achieve perfect security. Now we are more aware of the criteria of the practical design of stream ciphers:

- The key, controlling the key stream generation, should be relatively short and easy to transmit via a secret channel with low cost.
- The generated key stream should look random from the distribution and complexity viewpoints.
- The algorithm to generate the key stream should be computed fast and implemented easily.

To summarize, the perfect security of the Vernam one-time pad cipher makes the stream ciphers using the same encryption method popular because if the key stream is "truly random", which means all the bits distribute uniformly and independently, then the stream ciphers must be perfectly secure. However, the intrinsic drawbacks make the Vernam one-time pad cipher unpractical. So these operational disadvantages led to the development of the stream ciphers, especially synchronous stream ciphers. In the next section, we will briefly introduce some good key generators of stream ciphers published in the open literature.

1.3 Practical Realizations of the Key Stream in Stream Ciphers

In this section, we divide these key stream generators into two parts. The first kind of key stream generators is given by the combinations of Linear Feedback Shift Registers and some special nonlinear filters. LFSRs are easy to implement by hardware and fast to compute. So they are widely used in the design of the key stream generators. Here we give some concrete examples. Denote the ith LFSR and its jth state by $< L_i, M_i(x) >$ and $\mathbf{x_j}$, respectively.

1. Knapsack Generator
In fact, this is a real implementation of the first kind of the nonlinear filters which will be described in the following chapter.

Input: An LFSR $< L, M(x) >$ with the initial state $\mathbf{x_0} = (x_1^0, x_2^0, \ldots, x_L^0)$; modulus Q; L knapsack weights $w_1, w_2, \ldots, w_L$ of size n bits each.
Algorithm: For $i = 1, 2, \ldots$, do

```
    1. Compute the $ith$ state of the LFSR $\mathbf{x_i}$.
    2. Compute the knapsack sum $S_i = \sum_{k=1}^{L} x_k^i w_k \bmod Q$.
    3. Extract some bits of $S_i$ to be $Z_i$.
```

Output: the sequence Z_i, for $i = 1, 2, \ldots$.

The security of the nonlinear filter, the knapsack sum, lies in the hardness of the knapsack problem. It is already known as an NP-complete problem [12]. For details on the analysis of this generator, one can refer to [31].

2. Threshold Generator
Actually, this is a published proposal of a key stream generator, being an implementation of the second type of the nonlinear filters which will be discussed later.

Input: n LFSRs $< L_i, M_i(x) >$ with initial states $\mathbf{{}_iI^0} = ({}_iI_1^0, {}_iI_2^0, \ldots, {}_iI_{L_i}^0)$.
Algorithm: For $j = 1, 2, \ldots$, do

```
    1. For $i = 1, 2, \ldots, n$, compute the $jth$ state of each LFSR $< L_i, M_i(x) >$ and extract ${}_iI_1^j$ for each $i$.
```

2. Compute the integer sum of the current output bits $s_j = \sum_{i=1}^{n} {}_iI_1^j$.

3. $z_j = 1$ if $s_j > \frac{n}{2}$; $z_j = 0$ otherwise.

Output: the sequence z_j, for $j = 1, 2, \ldots$.

This generator can have large linear complexity while it still maintains good statistical properties [2]. Its algorithm is also very simple and fast. And $\mathbf{z} = \{z_j\}$ will be balanced when n is odd [40, Chapter 2]. But there is positive correlation between $\mathbf{z}$ and the n LFSRs $< L_i, M_i(x) >$, so it is cryptographically weak [40, Chapter 2].

3. Multiplexer Generator

Multiplexer generator, although it still belongs to the second kind of nonlinear filters, represents one kind of design ideas of key stream generators: technology-driven protocols. Some algorithms are developed according to the availability of some hardware so they could be implemented easily and conducted fast.

Input: Two LFSRs $< L_1, M_1(x) >, < L_2, M_2(x) >$ with initial states $\mathbf{x_0}, \mathbf{y_0}$; a positive integer n; a control vector $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1})$ such that $1 \leq v_0 < v_1 < \cdots < v_{n-1} \leq L_1$.

Algorithm: For $i = 1, 2, \ldots$, do

1. Compute the *ith* state of each LFSR $< L_1, M_1(x) >, < L_2, M_2(x) >$, $\mathbf{x_i} = (x_1^i, x_2^i, \ldots, x_{L_1}^i)$, and $\mathbf{y_i} = (y_1^i, y_2^i, \ldots, y_{L_2}^i)$.

2. Compute the integer $a_i = \sum_{k=0}^{n-1} 2^k x_{v_k}^i$.

3. Extract $z_i = y_{\theta(a_i)}^i$, where θ is an invertible mapping from $\{0, 1, \ldots, 2^n - 1\}$ to $\{1, 2, \ldots, L_2\}$.

Output: the sequence z_i, for $i = 1, 2, \ldots$.

This generator is due to the invention of the *multiplexer circuit.* Detailed analysis can be found in Jennings [13], [14]. Besides multiplexer generator, *Pless generator* is also an example which was developed after the availability of the hardware, *J-K flip-flop circuits.*

The second kind of key stream generators does not employ LFSRs. Their design ideas originate from number theory and well-known hard problems, such as *discrete logarithm, quadratic residue*, etc.. The security of most generators is built on computational complexity. So ideas of these generators are classified under the complexity-theoretic approach [40, Chapter 2]. Let us introduce a concrete generator as an example.

4. Quadratic Residue Generator

As is well known, given $y \in \mathbb{Z}$, it is in general hard to find $x \in \mathbb{Z}$ such that the congruence $y = x^2 \mod N$ is satisfied. If there exists such x, then y is called a *quadratic residue* mod N. Denote by QR_N the set of all quadratic residues mod N.

```
Input: A modulus $N$ of length $n$; $x_1 \in QR_N$ which is chosen randomly.
Algorithm: For $i = 1, 2, \ldots$, do
    1. $z_i = x_i \mod 2$, $z_i \in \{0, 1\}$
    2. $x_{i+1} = x_i^2 \mod N$
Output: the sequence $z_i$, for $i = 1, 2, \ldots$.
```

The security of the quadratic residue generator relies on the difficulty of solving quadratic congruences. In [40, Chapter 2], under the *quadratic residuosity assumption*, the generating key stream $\{z_i\}$ is proved to be unpredictable, which means the probability of any *predictor* to predict each key bit successfully is less than $\frac{1}{2}$, and indistinguishable by all polynomial-time statistical tests. So the quadratic residue generator is called *perfect*.

Chapter 2

Linear Recurring Sequences and Linear Complexity

In the 1950s, Linear Feedback Shift Registers were introduced into stream cipher application. In the following decades, since they are easy to implement by hardware and fast to process, LFSRs were often recommended to be the pseudorandom sequences generators. And what is more important is that because of the adoption of LFSRs in stream ciphers, cryptographers and mathematicians could use rigorous mathematical theory to analyze their security. Also because LFSRs are linear devices, the linear complexity is a vital concept to determine the security levels of stream ciphers (for other complexity measurements such as higher-order complexity, 2-adic complexity measures and complexity measures based on pattern counting, one could refer to [27]). And the linear complexity profile is also a good tool to measure the randomness of generated sequences which is discussed in Chapter 3. Therefore, in this chapter, we will give the detailed theoretical analysis of Linear Recurring Sequences and their linear complexities.

2.1 Introduction and Mathematical Fundamentals

In this section, we will introduce some concepts and algebraic tools which will be employed later. There are lots of excellent references on the finite field algebraic structures. One may refer to [18] for basic background and exciting results. Firstly, some important definitions should be given. From now on, let $\mathbb{F}_q$ be the finite field of order q, where q is an arbitrary prime power.

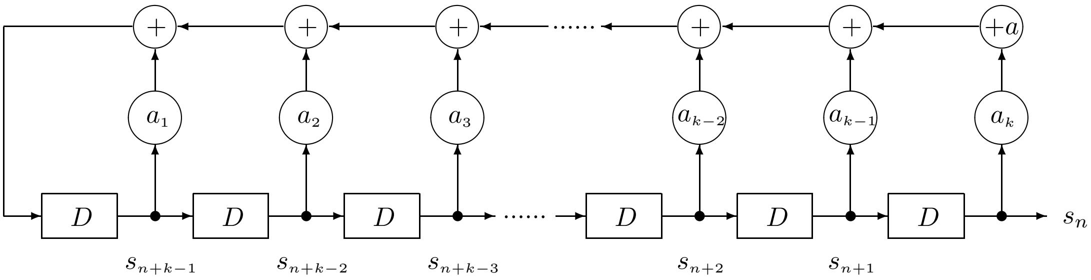

Figure 2.1: The General Form of Feedback Shift Register.

Definition 2.1: Let k be a positive integer, and $a_i \in \mathbb{F}_q$ for $1 \leq i \leq k+1$. Then call a sequence $\tilde{\mathbf{s}} = (s_1, s_2, \ldots, s_i, \ldots)$ of elements in $\mathbb{F}_q$ satisfying

$$s_{n+k} = a_1 s_{n+k-1} + a_2 s_{n+k-2} + \cdots + a_k s_n + a_{k+1} \qquad \text{for } n = 1, 2, \ldots \tag{2.1}$$

a k**th-order linear recurring sequence** in $\mathbb{F}_q$. The terms $s_1, s_2, \ldots, s_k$ are called **initial values**. If $a_{k+1} = 0$, we call (2.1) **homogeneous** and $\tilde{\mathbf{s}}$ a **homogeneous linear recurring sequence**. Otherwise, (2.1) and $\tilde{\mathbf{s}}$ are called **inhomogeneous** and **inhomogeneous linear recurring sequence**, respectively.

Linear recurring sequences usually are generated by a *feedback shift register*. It is a kind of electronic switching circuit consisting of four basic types. The first one is an *adder* with two inputs $a, b \in \mathbb{F}_q$ and one output $a \bigoplus b \in \mathbb{F}_q$; the second one is a *constant k multiplier* with an input $a \in \mathbb{F}_q$ and an output $ka \in \mathbb{F}_q$; the third one is a *constant k adder* with an input a and an output $a \bigoplus k \in \mathbb{F}_q$; the last one is a *delay element D* with an input and an output controlled by a time parameter such that the output is one time unit later than the input. A general form of a feedback shift register for (2.1) is given by Figure 2.1 ($a = a_{k+1}$).

Definition 2.2: Call $\tilde{\mathbf{s}} = (s_1, s_2, \ldots, s_i, \ldots)$ **ultimately periodic** if there are positive integers n_0 and T such that $s_{n+T} = s_n$ for all $n \geq n_0$. The number T is called a **period** of $\tilde{\mathbf{s}}$. If $n_0 = 1$, then $\tilde{\mathbf{s}}$ is called **periodic**. The smallest T is called the **minimal period**.

Let us consider the homogeneous linear recurring sequences in $\mathbb{F}_q$ satisfying the linear recurrence relation

$$s_{n+k} = a_1 s_{n+k-1} + a_2 s_{n+k-2} + \cdots + a_k s_n \qquad \text{for } n = 1, 2, \ldots, \tag{2.2}$$

where $a_i \in \mathbb{F}_q$ for $1 \le i \le k$. Now we introduce some crucial definitions in the analysis of homogeneous linear recurring sequences.

Definition 2.3: The **characteristic polynomial** $f_{\tilde{\mathbf{s}}}(x)$ of the linear recurring sequence $\tilde{\mathbf{s}} = (s_1, s_2, \ldots, s_i, \ldots)$ satisfying (2.2) is defined by

$$f_{\tilde{\mathbf{s}}}(x) = x^k - a_1 x^{k-1} - a_2 x^{k-2} - \cdots - a_{k-1}x - a_k \quad \in \ \mathbb{F}_q[x]. \tag{2.3}$$

And its **reciprocal characteristic polynomial** $f^*_{\tilde{\mathbf{s}}}(x)$ is given by

$$f^*_{\tilde{\mathbf{s}}}(x) = x^k f(\frac{1}{x}) = 1 - a_1 x^1 - a_2 x^2 - \cdots - a_{k-1}x^{k-1} - a_k x^k \quad \in \ \mathbb{F}_q[x]. \tag{2.4}$$

Set $s_{\tilde{\mathbf{s}}}(x) = s_1 x^{T-1} + s_2 x^{T-2} + \cdots + s_{T-1}x + s_T$ and $a_0 = -1$. Then a very important relationship between a homogeneous linear recurring sequence with period T and its characteristic polynomial is given by the following identity [18, Section 6.2]:

$$f_{\tilde{\mathbf{s}}}(x) s_{\tilde{\mathbf{s}}}(x) = (1 - x^T) h_{\tilde{\mathbf{s}}}(x), \tag{2.5}$$

where $h_{\tilde{\mathbf{s}}}(x)$ is given by

$$h_{\tilde{\mathbf{s}}}(x) = \sum_{j=0}^{k-1} \sum_{i=1}^{k-j} a_{k-i-j} s_i x^j \in \mathbb{F}_q[x]. \tag{2.6}$$

There is also another very interesting connection between s_i and the roots of the characteristic polynomial of $\tilde{\mathbf{s}}$, when it is irreducible over $\mathbb{F}_q$. Suppose $\tilde{\mathbf{s}}$ is a kth order homogeneous linear recurring sequence in $\mathbb{F}_q$ and $f_{\tilde{\mathbf{s}}}(x)$ is irreducible. Let α be a root of $f_{\tilde{\mathbf{s}}}(x)$ in the extension field $\mathbb{F}_{q^k}$, then there exists a unique $\theta \in \mathbb{F}_{q^k}$ such that

$$s_i = Tr_{\mathbb{F}_{q^k}/\mathbb{F}_q}(\theta\alpha^i) = \theta\alpha + (\theta\alpha^i)^q + (\theta\alpha^i)^{q^2} + \cdots + (\theta\alpha^i)^{q^{k-1}} \text{ for } i = 1, 2, \ldots. \tag{2.7}$$

Definition 2.4: A nonzero homogeneous linear recurring sequence over $\mathbb{F}_q$ whose characteristic polynomial is a *primitive polynomial* over $\mathbb{F}_q$ is called a **maximal period sequence**.

An immediate conclusion following Definition 2.4 is that a kth order maximal period sequence is periodic and its minimal period achieves the largest possible value $q^k - 1$ [18, Section 6.2].

Definition 2.5: Let $\tilde{\mathbf{s}} = (s_1, s_2, \ldots, s_i, \ldots)$ be a homogeneous linear recurring sequence in $\mathbb{F}_q$ and L be the set of all linear recurrence relations which can generate $\tilde{\mathbf{s}}$. Then the characteristic polynomial of the minimal linear recurrence relation which has the smallest order in L is called the **minimal polynomial** of $\tilde{\mathbf{s}}$. And the smallest order, which is the same as the degree of the minimal polynomial of $\tilde{\mathbf{s}}$, is called the **linear complexity** of $\tilde{\mathbf{s}}$.

Definition 2.6: Let $\tilde{\mathbf{s}} = (s_1, s_2, \ldots, s_i, \ldots)$ over $\mathbb{F}_q$. Then its **generating function** is defined to be a formal power series with an indeterminate x given by

$$G_{\tilde{\mathbf{s}}}(x) = s_1 + s_2 x + s_3 x^2 + \cdots + s_n x^{n-1} + \cdots = \sum_{i=1}^{\infty} s_i x^{i-1} \in \mathbb{F}_q[[x]]. \tag{2.8}$$

For a finite sequence $\tilde{\mathbf{s}} = (s_1, s_2, \ldots, s_n)$ over $\mathbb{F}_q$, we assign zeros to be the terms after n.

The algebraic computations of these formal power series involve addition and multiplication. Suppose $B(x) = \sum\limits_{i=1}^{\infty} b_i x^{i-1}$ and $C(x) = \sum\limits_{j=1}^{\infty} c_j x^{j-1}$, then their sum is defined by

$$B(x) + C(x) = \sum_{n=1}^{\infty} (b_n + c_n) x^{n-1}$$

and their multiplication is given by

$$B(x)C(x) = \sum_{n=1}^{\infty} d_n x^{n-1}, \qquad \text{where } d_n = \sum_{k=1}^{n} b_k c_{n+1-k} \text{ for } \ n = 1, 2, \ldots.$$

Based on the above definitions, we have $[B(x)C(x)]D(x) = B(x)[C(x)D(x)]$ and $B(x)[C(x) + D(x)] = B(x)C(x) + B(x)D(x)$. Moreover, the formal power series $B(x) = \sum\limits_{i=1}^{\infty} b_i x^{i-1}$ has a multiplicative inverse if and only if $b_1 \neq 0$ [18, Section 6.3]. Now we are ready to introduce some strong tools to analyze the linear recurring sequences.

Theorem 2.1: Let $\tilde{\mathbf{s}} = (s_1, s_2, \ldots, s_i, \ldots)$ be a kth order homogeneous linear recurring sequence in $\mathbb{F}_q$, whose linear recurrence relation is given by (2.2). Suppose $f^*_{\tilde{\mathbf{s}}}(x) \in \mathbb{F}_q[x]$ is the reciprocal characteristic polynomial of $\tilde{\mathbf{s}}$, and let $G_{\tilde{\mathbf{s}}}(x) \in \mathbb{F}_q[[x]]$ be its generating function. Set $a_0 = -1$. Then there is a $g_{\tilde{\mathbf{s}}}(x) \in \mathbb{F}_q[x]$ such that

$$G_{\tilde{\mathbf{s}}}(x) = \frac{g_{\tilde{\mathbf{s}}}(x)}{f^*_{\tilde{\mathbf{s}}}(x)}, \qquad \text{where } g_{\tilde{\mathbf{s}}}(x) = -\sum_{j=0}^{k-1} (\sum_{i=1}^{j+1} a_{j+1-i} s_i) x^j. \tag{2.9}$$

Conversely, if $g(x)$ is any polynomial over $\mathbb{F}_q$ with $\deg(g(x)) < k$ and if $f^*(x)$ is equal to (2.4), then the formal power series $G(x) \in \mathbb{F}_q[[x]]$ defined by $G(x) = \dfrac{g(x)}{f^*(x)}$ is the generating function of a kth order homogeneous linear recurring sequence in $\mathbb{F}_q$ whose linear recurrence relation is given by (2.2).

Proof: Firstly let us consider the first part of this theorem. We have

$$\begin{aligned} f^*_{\tilde{\mathbf{s}}}(x)G_{\tilde{\mathbf{s}}}(x) &= -\Big(\sum_{n=0}^{k} a_n x^n\Big)\Big(\sum_{n=1}^{\infty} s_n x^{n-1}\Big) \\ &= -\sum_{j=0}^{k-1}\Big(\sum_{i=1}^{j+1} a_{j+1-i}s_i\Big)x^j - \sum_{j=k}^{\infty}\Big(\sum_{i=1}^{k+1} a_{k+1-i}s_{j-k+i}\Big)x^j \\ &= g_{\tilde{\mathbf{s}}}(x) - \sum_{j=k}^{\infty}\Big(\sum_{i=1}^{k+1} a_{k+1-i}s_{j-k+i}\Big)x^j. \end{aligned} \tag{2.10}$$

From (2.2), we have $s_{j+1} = \sum_{i=1}^{k} a_i s_{j+1-i}$, which implies $\sum_{i=1}^{k+1} a_{k+1-i}s_{j-k+i} = 0$. Therefore, given $f^*(0) = 1$, the identity (2.9) holds.

Then, consider the second part of this theorem. Since $G(x) = \dfrac{g(x)}{f^*(x)}$, we have $g(x) = f^*(x)G_{\tilde{\mathbf{s}}}(x)$. Therefore, the fact $\deg(g(x)) < k$ forces $\sum_{i=1}^{k+1} a_{k+1-i}s_{j-k+i} = 0$, which implies $s_{j+1} = \sum_{i=1}^{k} a_i s_{j+1-i}$ for all $j \geq k$. Hence, $\tilde{\mathbf{s}} = (s_1, s_2, \ldots, s_i, \ldots)$ whose generating polynomial is $G(x)$ satisfies the linear recurrence relation (2.2). □

Remark: Here we have a relation between $g_{\tilde{\mathbf{s}}}(x)$ and $h_{\tilde{\mathbf{s}}}(x)$ defined in (2.6), which is given by $x^{k-1}g_{\tilde{\mathbf{s}}}(\frac{1}{x}) = -h_{\tilde{\mathbf{s}}}(x)$.

Theorem 2.2: Let $\tilde{\mathbf{s}} = (s_1, s_2, \ldots, s_i, \ldots)$ be a homogeneous linear recurring sequence over $\mathbb{F}_q$. Then there exists a unique polynomial $m_{\tilde{\mathbf{s}}}(x)$ such that it is the minimal polynomial of $\tilde{\mathbf{s}}$ and a monic polynomial $f(x) \in \mathbb{F}_q[x]$ with positive degree is a characteristic polynomial of $\tilde{\mathbf{s}}$ if and only if $m_{\tilde{\mathbf{s}}}(x) \mid f(x)$.

Proof: Since every homogeneous linear recurring sequence over $\mathbb{F}_q$ is ultimately periodic, we can suppose the period of $\tilde{\mathbf{s}}$ is T. Now let $f_{\tilde{\mathbf{s}}}(x)$ be the characteristic polynomial of $\tilde{\mathbf{s}}$,

$h_{\tilde{\mathbf{s}}}(x)$ be given by (2.6) and $g_{\tilde{\mathbf{s}}}(x)$ be defined as in (2.9). Suppose $d(x) = \gcd(f_{\tilde{\mathbf{s}}}(x), h_{\tilde{\mathbf{s}}}(x))$ to be monic, then define $m_{\tilde{\mathbf{s}}}(x) = \dfrac{f_{\tilde{\mathbf{s}}}(x)}{d(x)}$. Obviously, $m_{\tilde{\mathbf{s}}}(x)$ is monic.

Now let $f(x) \in \mathbb{F}_q[x]$ be an arbitrary characteristic polynomial of $\tilde{\mathbf{s}}$. And let $h(x)$ and $g(x)$ be its corresponding polynomials defined by (2.6) and (2.9) given the linear recurrence relation of $\tilde{\mathbf{s}}$ defined by $f(x)$. Therefore according to Theorem 2.1, we have

$$G_{\tilde{\mathbf{s}}}(x) = \frac{g_{\tilde{\mathbf{s}}}(x)}{f^*_{\tilde{\mathbf{s}}}(x)} = \frac{g(x)}{f^*(x)}.$$

Hence $g(x)f^*_{\tilde{\mathbf{s}}}(x) = g_{\tilde{\mathbf{s}}}(x)f^*(x)$. By the remark after Theorem 2.1,

$$h(x)f_{\tilde{\mathbf{s}}}(x) = -x^{\deg[f(x)]-1}g(\frac{1}{x})x^{\deg[f_{\tilde{\mathbf{s}}}(x)]}f^*_{\tilde{\mathbf{s}}}(\frac{1}{x}) = -x^{\deg[f_{\tilde{\mathbf{s}}}(x)-1]}g_{\tilde{\mathbf{s}}}(\frac{1}{x})x^{\deg[f(x)]}f^*(\frac{1}{x}) = h_{\tilde{\mathbf{s}}}(x)f(x).$$

Now divide $d(x)$ on both sides of $h(x)f_{\tilde{\mathbf{s}}}(x) = h_{\tilde{\mathbf{s}}}(x)f(x)$, then we have $h(x)m_{\tilde{\mathbf{s}}}(x) = \dfrac{h_{\tilde{\mathbf{s}}}(x)}{d(x)}f(x)$, which forces $m_{\tilde{\mathbf{s}}}(x) \mid f(x)$ by the definition of $d(x)$.

Now suppose $f(x) \in \mathbb{F}_q$ is a monic polynomial and $f(x) = m_{\tilde{\mathbf{s}}}(x)r(x)$. Then $f^*(x) = m^*_{\tilde{\mathbf{s}}}(x)r^*(x)$. Let $R(x) = \dfrac{h_{\tilde{\mathbf{s}}}(x)}{d(x)}$. Hence, by $h_{\tilde{\mathbf{s}}}(x)m_{\tilde{\mathbf{s}}}(x) = R(x)f_{\tilde{\mathbf{s}}}(x)$, we have

$$g_{\tilde{\mathbf{s}}}(x)m^*_{\tilde{\mathbf{s}}}(x) = -x^{\deg[f_{\tilde{\mathbf{s}}}(x)]-1}h_{\tilde{\mathbf{s}}}(\frac{1}{x})x^{\deg[m_{\tilde{\mathbf{s}}}(x)]}m_{\tilde{\mathbf{s}}}(\frac{1}{x}) = -x^{\deg[m_{\tilde{\mathbf{s}}}(x)]-1}R(\frac{1}{x})x^{\deg[f_{\tilde{\mathbf{s}}}(x)]}f_{\tilde{\mathbf{s}}}(\frac{1}{x}).$$

Since $\deg[R(x)] < \deg[m_{\tilde{\mathbf{s}}}(x)]$, $t(x) = -x^{\deg[m_{\tilde{\mathbf{s}}}(x)]-1}R(\frac{1}{x})$ is a polynomial in $\mathbb{F}_q[x]$. By $g_{\tilde{\mathbf{s}}}(x)m^*_{\tilde{\mathbf{s}}}(x) = t(x)f^*_{\tilde{\mathbf{s}}}(x)$,

$$G_{\tilde{\mathbf{s}}}(x) = \frac{g_{\tilde{\mathbf{s}}}(x)}{f^*_{\tilde{\mathbf{s}}}(x)} = \frac{t(x)}{m^*_{\tilde{\mathbf{s}}}(x)} = \frac{t(x)r^*(x)}{f^*(x)}.$$

Because $\deg[t(x)r^*(x)] = \deg[t(x)] + \deg[r^*(x)] < \deg[m_{\tilde{\mathbf{s}}}(x)] + \deg[r(x)] = \deg[f(x)]$, from Theorem 2.1, we conclude that $f(x)$ is a characteristic polynomial of $\tilde{\mathbf{s}}$.

Suppose $M_{\tilde{\mathbf{s}}}(x)$ is the minimal polynomial of $\tilde{\mathbf{s}}$. We have $m_{\tilde{\mathbf{s}}}(x) \mid M_{\tilde{\mathbf{s}}}(x)$. But according to the definition of the minimal polynomial, we get $\deg[M_{\tilde{\mathbf{s}}}(x)] \leq \deg[m_{\tilde{\mathbf{s}}}(x)]$, then we have $m_{\tilde{\mathbf{s}}}(x) = M_{\tilde{\mathbf{s}}}(x)$ is the minimal polynomial. Suppose there is another minimal polynomial $m(x)$ of $\tilde{\mathbf{s}}$. Then given both are monic, $m_{\tilde{\mathbf{s}}}(x) \mid m(x)$ and $m(x) \mid m_{\tilde{\mathbf{s}}}(x)$ imply the uniqueness. □

Definition 2.7 Let $\tilde{\mathbf{s}} = (s_1, s_2, \ldots, s_n)$ be a finite sequence over $\mathbb{F}_q$. Denote the linear complexity of the first i terms $(s_1, s_2, \ldots, s_i)$ by $L(\tilde{\mathbf{s}}^i)$. Then the **linear complexity profile** of $\tilde{\mathbf{s}}$ is defined to be the sequence $(L(\tilde{\mathbf{s}}^1), L(\tilde{\mathbf{s}}^2), \ldots, L(\tilde{\mathbf{s}}^n))$. For an infinite sequence $\tilde{\mathbf{s}} = (s_1, s_2, \ldots, s_i, \ldots)$, its linear complexity profile is defined to be the corresponding infinite sequence $(L(\tilde{\mathbf{s}}^1), L(\tilde{\mathbf{s}}^2), \ldots, L(\tilde{\mathbf{s}}^i), \ldots)$.

According to the discussion of Chapter 3, the linear complexity and the linear complexity profile of a sequence $\tilde{\mathbf{s}}$ are two vital characteristic parameters to measure its security when it is used as the key stream. Therefore, we will introduce two methods to compute the linear complexity profile of $\tilde{\mathbf{s}}$ next. The first one is called Berlekamp-Massey algorithm invented by J. L. Massey [20] in 1969, which is based on the iterative algorithm first introduced by Berlekamp for decoding BCH codes.

Now let $\tilde{\mathbf{s}} = (s_1, s_2, \ldots, s_i, \ldots)$ over $\mathbb{F}_q$, $\tilde{\mathbf{s}}^n = (s_1, s_2, \ldots, s_n)$ and the linear complexity of $\tilde{\mathbf{s}}^n$ be denoted by $L(\tilde{\mathbf{s}}^n)$. Suppose $f^*_{\tilde{\mathbf{s}}^n}(x) = 1 - a_1^n x - a_2^n x^2 - \cdots - a^n_{L(\tilde{\mathbf{s}}^n)} x^{L(\tilde{\mathbf{s}}^n)} \in \mathbb{F}_q[x]$ to be the reciprocal characteristic polynomial of $\tilde{\mathbf{s}}^n$. The basic idea of the Berlekamp-Massey algorithm is that: if for some m such that $s_{n+m-1} = \sum\limits_{i=1}^{L(\tilde{\mathbf{s}}^n)} a_i^n s_{n+m-1-i}$ but $s_{n+m} \neq \sum\limits_{i=1}^{L(\tilde{\mathbf{s}}^n)} a_i^n s_{n+m-i}$, then we have

$$L(\tilde{\mathbf{s}}^{n+m}) = \max(L(\tilde{\mathbf{s}}^n), n+m-L(\tilde{\mathbf{s}}^n)), \tag{2.11}$$

and the new reciprocal characteristic polynomial of $\tilde{\mathbf{s}}^{n+m}$ is given by

$$f^*_{\tilde{\mathbf{s}}^{n+m}}(x) = f^*_{\tilde{\mathbf{s}}^n}(x) - \frac{s_{n+m} - (\sum\limits_{i=1}^{L(\tilde{\mathbf{s}}^n)} a_i^n s_{n+m-i})}{s_t - (\sum\limits_{i=1}^{t-L(\tilde{\mathbf{s}}^n)} a_i^t s_{t-i})} x^{n+m-t} f^*_{\tilde{\mathbf{s}}^{t-1}}(x), \tag{2.12}$$

where t is the positive integer such that $L(\tilde{\mathbf{s}}^{t-1}) < L(\tilde{\mathbf{s}}^t) = L(\tilde{\mathbf{s}}^n)$.

Berlekamp-Massey Algorithm:
Input: $\tilde{\mathbf{s}}^n = (s_1, s_2, \ldots, s_n)$.

```
1. Initialization: $f^*(x) := 1, B(x) := 1, r := 1, L := 0, b := 1, i := 1$.
2. For i=1:n, do
```

```
    • Compute $d = s_i - \sum_{j=1}^{L} a_j s_{i-j}$.
      Here if $L = 0$, let $\sum_{j=1}^{L} a_j s_{i-j} = 0$; If $L = 1, i = 1$, let $\sum_{j=1}^{L} a_j s_{i-j} = 0$.
    • If $d = 0$, then let $r := r + 1$ and $i := i + 1$.
    • If $d \neq 0$ and $2L \leq i$, then let
        $T(x) := f^*(x)$;
        $f^*(x) := f^*(x) - db^{-1}x^r B(x)$;
        $L := i + 1 - L; B(x) := T(x)$;
        $b := d, r := 1$;
        $i := i + 1$;
  3. Return $L$.
Output: $L(\tilde{\mathbf{s}}^n) = L$.
```

The second method to determine the linear complexity of $\tilde{\mathbf{s}} = (s_1, s_2, \ldots, s_i, \ldots)$ is given by the *continued fraction expansion* of $\widetilde{G}_{\tilde{\mathbf{s}}}(x) = \sum\limits_{i=1}^{\infty} s_i \left(\frac{1}{x}\right)^i$. This method was first introduced by Harald Niederreiter in [26]. We summarize it here. Firstly, let $D = \mathbb{F}_q[[\frac{1}{x}]]$. Then for every $g \in D$, there is a unique continued fraction expansion of g given by

$$g = A_0 + 1/(A_1 + 1/(A_2 + \cdots)) := [A_0, A_1, A_2, \ldots], \tag{2.13}$$

where $A_i \in \mathbb{F}_q[x]$ for all $i \geq 0$ and $\deg(A_i) \geq 1$ for all $i \geq 1$. For $S = \sum\limits_{i=r}^{\infty} s_i x^{-i} \in \mathbb{F}_q[[\frac{1}{x}]]$, define its polynomial part by

$$Pol(S) = \sum_{i=r}^{0} s_i x^{-i}.$$

Now suppose $\tilde{\mathbf{s}} = (s_1, s_2, \ldots, s_i, \ldots)$ is in $\mathbb{F}_q$, $\tilde{\mathbf{s}}^n = (s_1, s_2, \ldots, s_n)$ and $\widetilde{G_{\tilde{\mathbf{s}}}} = \sum\limits_{i=1}^{\infty} s_i \left(\frac{1}{x}\right)^i$. Set $A_0 = Pol(\widetilde{G_{\tilde{\mathbf{s}}}})$, $B_0 = \widetilde{G_{\tilde{\mathbf{s}}}} - Pol(\widetilde{G_{\tilde{\mathbf{s}}}})$, $P_{-1} = 1, P_0 = A_0, Q_{-1} = 0, Q_0 = 1$. Now define A_j, Bj, P_j, Q_j recursively by

$$\begin{array}{lll} A_{j+1} = Pol(B_j^{-1}), & B_{j+1} = B_j^{-1} - Pol(B_j^{-1}) & \text{for } j \geq 0, \\ P_j = A_j P_{j-1} + P_{j-2}, & Q_j = A_j Q_{j-1} + Q_{j-2} & \text{for } j \geq 1. \end{array}$$

Then, the linear complexity $L(\tilde{\mathbf{s}}^n)$ is given by $L(\tilde{\mathbf{s}}^n) = \deg(Q_j)$, where $j \geq 0$ is uniquely determined by

$$\deg(Q_{j-1}) + \deg(Q_j) \leq n < \deg(Q_j) + \deg(Q_{j+1}).$$

One can refer to [26] for the rigorous proof.

Besides these two algorithms, we will consider another way to find the linear complexity of periodic sequences. This is very important for the discussion in Chapter 4.

Theorem 2.3: Let $\tilde{\mathbf{s}} = (s_1, s_2, \ldots, s_i, \ldots)$ be a periodic sequence over $\mathbb{F}_q$ with the minimal period T. Let $g_{\tilde{\mathbf{s}}}(x) = s_1 + s_2 x + \cdots + s_T x^{T-1}$ and $D(x) = \gcd(g_{\tilde{\mathbf{s}}}(x), 1 - x^T)$. Then its linear complexity $L(\tilde{\mathbf{s}})$ is given by $L(\tilde{\mathbf{s}}) = \deg\Big(\dfrac{1-x^T}{D(x)}\Big)$.

Proof: Let $G_{\tilde{\mathbf{s}}}(x)$ be the generating function of $\tilde{\mathbf{s}}$. Since $s_{n+T} = s_n$ for all $n \geq 1$ and the constant term of $1 - x^T$ is nonzero, we have

$$G_{\tilde{\mathbf{s}}}(x) = \frac{g_{\tilde{\mathbf{s}}}(x)}{1 - x^T}.$$

From the proof of Theorem 2.2, we know that the minimal polynomial $m_{\tilde{\mathbf{s}}}(x)$ of $\tilde{\mathbf{s}}$ is given by $\dfrac{f_{\tilde{\mathbf{s}}}(x)}{\gcd(f_{\tilde{\mathbf{s}}}(x), h_{\tilde{\mathbf{s}}}(x))}$, where $h_{\tilde{\mathbf{s}}}(x)$ is defined as in (2.6). Since $\tilde{\mathbf{s}}$ is periodic with the minimal period T, its characteristic polynomial is given by $f_{\tilde{\mathbf{s}}}(x) = x^T - 1$. After computation, we have $h_{\tilde{\mathbf{s}}}(x) = -x^{T-1} g_{\tilde{\mathbf{s}}}(\frac{1}{x})$ and $g_{\tilde{\mathbf{s}}}(x) = -x^{T-1} h_{\tilde{\mathbf{s}}}(\frac{1}{x})$. Suppose $x^T - 1 = d(x)a(x)$ and $h_{\tilde{\mathbf{s}}}(x) = d(x)b(x)$, where $d(x) = \gcd(x^T - 1, h_{\tilde{\mathbf{s}}}(x))$. Then by the remark after Theorem 2.1, we have $g_{\tilde{\mathbf{s}}}(x) = -x^{T-1} d(\frac{1}{x}) b(\frac{1}{x})$. Obviously, we have $m_{\tilde{\mathbf{s}}}(x) = a(x)$, $a(0) \neq 0$ and $L(\tilde{\mathbf{s}}) = T - \deg(d(x))$. Now consider

$$\begin{aligned}
{}_{\tilde{\mathbf{s}}}m(x) &\doteq \frac{1 - x^T}{\gcd(1 - x^T, g_{\tilde{\mathbf{s}}}(x))} = \frac{-d(x)a(x)}{\gcd(-d(x)a(x), -x^{T-1} d(\frac{1}{x}) b(\frac{1}{x}))} \\
&= \frac{x^T d(\frac{1}{x}) a(\frac{1}{x})}{\gcd\Big(x^T d(\frac{1}{x}) a(\frac{1}{x}), -x^{T-1} d(\frac{1}{x}) b(\frac{1}{x})\Big)} \\
&= \frac{x^{\deg(a(x))} a(\frac{1}{x})}{\gcd\Big(x^{\deg(a(x))} a(\frac{1}{x}), -x^{\deg(a(x))-1} b(\frac{1}{x})\Big)}.
\end{aligned}$$

For any irreducible polynomial $p(x) \in \mathbb{F}_q[x]$ with degree $m \geq 2$, we have

$$p(x) = \prod_{i=0}^{m-1} (x - \theta^{q^i}), \qquad \text{where } p(\theta) = 0.$$

Then,

$$p^*(x) = x^m p(\frac{1}{x}) = \prod_{i=0}^{m-1}(-\theta^{q^i})\prod_{j=0}^{m-1}(x-(\frac{1}{\theta})^{q^j}).$$

Since $\gcd(q^m-1, q^m-2) = 1$, then $\frac{1}{\theta} = \theta^{q^m-2}$ is still a defining element of $\mathbb{F}_{q^m}$, which implies $p^*(x)$ is also irreducible. If $m = 1$, the conclusion holds obviously. Now, factoring $a(x), b(x)$ to canonical forms, we conclude that $\gcd(a^*(x), b^*(x)) = 1$. Because $a^*(0) \neq 1$ and $\deg(b(x)) < \deg(a(x))$, ${}_{\tilde{\mathbf{s}}}m(x)$ must be equal to $a^*(x) = x^{\deg(a(x))}a(\frac{1}{x})$. Therefore $\deg({}_{\tilde{\mathbf{s}}}m(x)) = \deg(a^*(x)) = \deg(a(x)) = T - \deg(d(x)) = L(\tilde{\mathbf{s}})$. □

Now according to Theorem 2.3, we can just find the minimal polynomial of a periodic sequence $\tilde{\mathbf{s}}$ directly from $1 - x^T$ and the corresponding polynomial of its first minimal period terms by computing their greatest common divisor.

Proposition 2.1: Let $\tilde{\mathbf{s}}$ be a periodic sequence over $\mathbb{F}_q$ with period T and $G_{\tilde{\mathbf{s}}}(x)$ be its generating function. Suppose $m_{\tilde{\mathbf{s}}}(x)$ is the minimal polynomial of $\tilde{\mathbf{s}}$ and $g_{\tilde{\mathbf{s}}}(x) = s_1 + s_2x + \cdots + s_Tx^{T-1}$. Then $m_{\tilde{\mathbf{s}}}(x) = E^*(x)$, where $E(x) = \frac{1-x^T}{\gcd(g_{\tilde{\mathbf{s}}}(x), 1-x^T)}$.

Proof: Directly from the proof of Theorem 2.3. □

Remark: Notice that if $a(x), b(x) \in \mathbb{F}_q[x]$ such that $a(0)b(0) \neq 0$, then $(a^*)^*(x) = a(x)$ and so does $b(x)$. Therefore, $a(x) \mid b(x)$ if and only if $a^*(x) \mid b^*(x)$. Consider the minimal polynomial of periodic sequences. By Proposition 2.1, they do not contain factors x^i. So according to the above discussion, any properties of the minimal polynomial can be transferred to its reciprocal polynomial (this means we can also define $E(x)$ in Proposition 2.1 to be the minimal polynomial by replacing the Definition 2.5, if our discussion is confined to periodic sequences). This conclusion is important to the remaining content in this chapter and Chapter 4. In the following sections, sometimes we will derive the degrees of the minimal polynomials of periodic sequences directly from (2.9).

For the linear complexity of special sequences of cryptologic interests, a lot of research has been done by various authors. In Caballero-Gil [4], [5], Garcia-Villalba and F*ú*ster-Sabater [11] and Tan [42], they have investigated the linear complexity of typical hardware keystream generators. Konyagin *et al.* [16] and Meidl and Winterhof [25] considered the linear complexity of the discrete logarithm function. For further information, one can refer to the books of Shparlinski [38], [39].

2.2 Two Types of Nonlinear Filters

Generally speaking, there are two kinds of non-linear filters [32]. The first one is conducting nonlinear operations on several states of one periodic sequence whose minimal polynomial is irreducible. The other one is conducting nonlinear operations on the same state of several periodic sequences. In this section, we will discuss these two types of nonlinear operations on sequences over $\mathbb{F}_2$.

1. Nonlinear Operations on One Maximal Period Sequence

From the definition of the maximal period sequence, if its minimal polynomial is primitive with degree k, then its minimal period is $2^k - 1$. Although it has good statistical properties, it is highly predictable in its first period. This is because its linear complexity k is very small compared to its minimal period $2^k - 1$. Thus we need to employ some nonlinear operations on k states of the sequence such that the linear complexity of the sequence $\tilde{\mathbf{z}} = (z_1, z_2, \ldots, z_i, \ldots)$ filtered by the nonlinear device F, where each z_j is given by $z_j = F(s_j, s_{j+1}, \ldots, s_{j+k-1})$, will exhibit large linear complexity. See Figure 2.2.

Suppose $\tilde{\mathbf{s}} = (s_1, s_2, \ldots, s_i, \ldots)$ is an arbitrary maximal period sequence with the primitive minimal polynomial $m_{\tilde{\mathbf{s}}}(x)$ whose degree is L. And if we define $\sigma_j = (s_j, s_{j+1}, \ldots, s_{j+L-1})$, it is already known that all σ_j are distinct and nonzero for $j = 1, 2, \ldots, 2^L - 1$. Therefore, the nonlinear filter F, a mapping from $\mathbb{F}_2^L$ to $\mathbb{F}_2$, whose input is a vector with dimension L, is uniquely determined by the output $\tilde{\mathbf{z}} = (z_1, z_2, \ldots, z_i, \ldots)$, where $z_j = F(\sigma_j)$. This property does not hold if $m_{\tilde{\mathbf{s}}}(x)$ is just irreducible but not primitive. From the discussion in Chapter 3, there must be some sequences $\tilde{\mathbf{z}}$ having large linear complexity equal or almost equal to $2^L - 1$. Given that we can choose each term of $\tilde{\mathbf{z}}$ freely, then, there must exist some nonlinear filter F such that the filtered sequences of arbitrary maximal period sequences with period $2^L - 1$ have their linear complexities equal or almost equal to $2^L - 1$.

In fact, since the underlying field is $\mathbb{F}_2$, the function F is a sum of some products since there is a unique canonical form, **algebraic normal form** for the **boolean function** F. Therefore F can be represented by

$$\begin{aligned} F(x_1, x_2, \ldots, x_L) = \ & a_0 + a_1x_1 + a_2x_2 + \cdots + a_Lx_L \\ & + a_{1,2}x_1x_2 + a_{1,3}x_1x_3 + \cdots + a_{L-1,L}x_{L-1}x_L \\ & + \cdots \\ & + a_{1,2,\ldots,L}x_1x_2\cdots x_L. \end{aligned}$$

To obtain F such that $F : \tilde{\mathbf{s}} \mapsto \tilde{\mathbf{z}}$, we have three ways. Now let us briefly introduce them. The first one is that representing F, i.e $\tilde{\mathbf{z}}$, by the linear combination of the basis

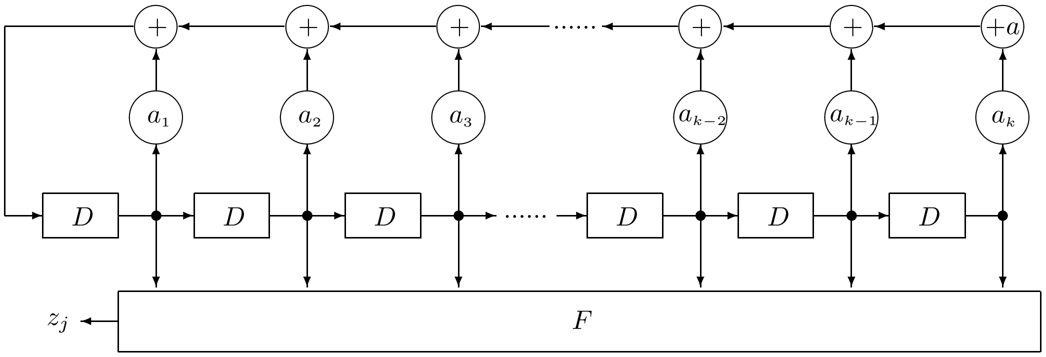

Figure 2.2: The First Kind of Nonlinear Filter.

of the linear space $\mathbb{F}_2^{2^L-1}$. Denote $\delta_i(\tilde{\mathbf{s}}^{2^L-1}) = (s_{1+i}, s_{2+i}, \ldots, s_{2^L-1}, s_1, s_2, \ldots, s_i)$ and $\tilde{\mathbf{s}}\tilde{\mathbf{z}} = (s_1z_1, s_2z_2, \ldots)$. When $\tilde{\mathbf{s}}^{2^L-1} \neq (0, 0, \ldots, 0)$, the following $2^L - 1$ sequences

$$\begin{aligned}
&\delta_0(\tilde{\mathbf{s}}^{2^L-1}), \delta_1(\tilde{\mathbf{s}}^{2^L-1}), \ldots, \delta_{L-1}(\tilde{\mathbf{s}}^{2^L-1}),\\
&\delta_0(\tilde{\mathbf{s}}^{2^L-1})\delta_1(\tilde{\mathbf{s}}^{2^L-1}), \ldots, \delta_0(\tilde{\mathbf{s}}^{2^L-1})\delta_{L-1}(\tilde{\mathbf{s}}^{2^L-1}), \ldots, \delta_{L-2}(\tilde{\mathbf{s}}^{2^L-1})\delta_{L-1}(\tilde{\mathbf{s}}^{2^L-1}),\\
&\cdots\cdots\\
&\delta_0(\tilde{\mathbf{s}}^{2^L-1})\delta_1(\tilde{\mathbf{s}}^{2^L-1}) \cdots \delta_{L-2}(\tilde{\mathbf{s}}^{2^L-1})\delta_{L-1}(\tilde{\mathbf{s}}^{2^L-1}),
\end{aligned}$$

form a basis of $\mathbb{F}_2^{2^L-1}$. If we suppose the output of F with zero input is zero, we can set the constant term in F being 0. Now denote the matrix of this basis by S. Therefore F can be represented by

$$\begin{aligned}
F(x_1, x_2, \ldots, x_L) = \ & a_1x_1 + a_2x_2 + \cdots + a_Lx_L\\
&+a_{1,2}x_1x_2 + a_{1,3}x_1x_3 + \cdots + a_{L-1,L}x_{L-1}x_L\\
&+\cdots\\
&+a_{1,2,\ldots,L}x_1x_2 \cdots x_L,
\end{aligned}$$

where x_i corresponds to $\delta_i(\tilde{\mathbf{s}}^{2^L-1})$ above. Then the coefficients vector $\mathbf{a}$ of the variables in F can be obtained by $\mathbf{a} = S^{-1}(\tilde{\mathbf{z}}^{2^L-1})^T$. Choosing this basis is very convenient since we could directly connect some states of the LFSR generating $\tilde{\mathbf{s}}$ by the logic AND and XOR functions to implement F.

The second method is the same as the first one but employs a different basis, the *natural basis* or *impulse response sequences.*

The third one is a little complex. Let $G_{\tilde{\mathbf{z}}}(x)$ be the generating polynomial for $\tilde{\mathbf{z}}$ and

$g_{\tilde{\mathbf{z}}}(x)$ be the corresponding polynomial of $\tilde{\mathbf{z}}^{2^L-1}$. Then we have

$$G_{\tilde{\mathbf{z}}}(x) = \frac{g_{\tilde{\mathbf{z}}}(x)}{1+x^{2^L-1}} = \sum_i \frac{c_i(x)}{p_i(x)} \qquad \deg(c_i(x)) < \deg(p_i(x)),$$

where $p_i(x)$ is irreducible. For each LFSR whose reciprocal minimal polynomial is $p_i(x)$, we can (by using natural basis) find the coefficients of its corresponding polynomial $c_i(x)$, which are actually the initial states for the LFSR. For the $c_i(x) \neq 0$, which means the corresponding $p_i(x)$ contributes to the linear complexity of $\tilde{\mathbf{z}}$, we can determine a nonlinear function $f_{i,j}$ for this LFSR such that the filtered sequence $\tilde{\mathbf{z}}_{i,j}$ of $\tilde{\mathbf{s}}$ by $f_{i,j}$ can simulate the behavior of this LFSR initiated by the corresponding natural base $e_{i,j}$, whose length and jth term are equal to $\deg(p_i(x))$ and 1, respectively. More exactly, denote the coefficients of $c_i(x)$ by $\mathbf{c}_i$ (start from zero degree, $\deg(c_1(x)) = 0$). Let $\mathbf{C} = (\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_l)$. And extend the basis from the natural basis for each LFSR according to their linear recurrence relation determined by $p_i(x)$ such that each basis has length $2^L - 1$. Denote the matrix of this basis by D. Now the coefficients of F, $\mathbf{a}$ can be expressed by $S^{-1}DC^T$. One could refer to [32, Section 5.1] for details and an example using the three methods introduced above.

However, in the above methods, we are facing some difficulties in application since we have to solve a large-scale linear equation system with dimension $2^L - 1$. We want $2^L - 1$ to be large enough to achieve good security. But it renders computing F impossible.

The strategy to solve the computational difficulty for F is constructing F directly and estimating the lower bound of the linear complexity of $\tilde{\mathbf{z}}$ after having been filtered by F.

Lemma 2.1: Suppose $\tilde{\mathbf{s}} = (s_1, s_2, \ldots, s_i, \ldots)$ is a maximal period sequence over $\mathbb{F}_2$ with its minimal polynomial $m_{\tilde{\mathbf{s}}}(x)$ having degree L. Let ${}_{t_i}\tilde{\mathbf{s}} = \delta_{t_i}(\tilde{\mathbf{s}})$ for $i = 1, 2, \ldots, k$. Denote $\alpha \in \mathbb{F}_{2^L}$ a root of $m_{\tilde{\mathbf{s}}}(x)$, the product $\prod_{i=1}^{k} ({}_{t_i}\tilde{\mathbf{s}})$ by $\tilde{\mathbf{z}}$ and the Hamming weight of the radix-2 form of N by $w_2(N)$. Then α^n, where $w_2(n) = k$, is a root of the minimal polynomial $m_{\tilde{\mathbf{z}}}(x)$ of $\tilde{\mathbf{z}}$ if

$$D_n = \begin{vmatrix} \alpha^{t_1 2^{e_1}} & \alpha^{t_2 2^{e_1}} & \cdots & \alpha^{t_k 2^{e_1}} \\ \alpha^{t_1 2^{e_2}} & \alpha^{t_2 2^{e_2}} & \cdots & \alpha^{t_k 2^{e_2}} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{t_1 2^{e_k}} & \alpha^{t_2 2^{e_k}} & \cdots & \alpha^{t_k 2^{e_k}} \end{vmatrix} \neq 0,$$

where $n = 2^{e_1} + 2^{e_2} + \cdots + 2^{e_k}$ and $0 \leq e_1 < e_2 < \cdots < e_k < L$.

Proof: For the convenience of expression, we use i as internal index variable, which is only effective for one step, and j as the global index variable, which is effective in the whole proof. From (2.7), without loss of generality, we can assume $s_j = Tr(\alpha^j) = Tr_{\mathbb{F}_{q^L}/\mathbb{F}_q}(\alpha^j)$, so

$$\begin{aligned} z_j &= \prod_{i=1}^{k}({}_{t_i}s_j) = \prod_{i}^{k} Tr(\alpha^{t_i}\alpha^j) \qquad (2.14)\\ &= \prod_{i=1}^{k}(\alpha^{t_i}\alpha^j + \alpha^{2t_i}\alpha^{2j} + \cdots + \alpha^{2^{L-1}t_i}\alpha^{2^{L-1}j}). \end{aligned}$$

Collect the coefficients of the $\alpha^{n_i j}$, then

$$z_j = \sum_i E_{n_i}\alpha^{n_i j}, \qquad (2.15)$$

where $n_i = 2^{e_{n_i,1}} + 2^{e_{n_i,2}} + \cdots + 2^{e_{n_i,w_2(n_i)}}$ and $0 \le e_{n_i,1} < e_{n_i,2} < \cdots < e_{n_i,w_2(n_i)} < L$. Because E_{n_i} is only determined by n_i and $\{t_1, t_2, \ldots, t_k\}$, so it is independent of j. And when $w_2(n_i) = k$, we have $E_{n_i} = D_{n_i}$ given that there is no difference between the positive and minus signs in the determinant over $\mathbb{F}_2$.

Since we are working in $\mathbb{F}_2$, we can suppose the minimal polynomial for $\tilde{\mathbf{z}}$ is

$$m_{\tilde{\mathbf{z}}}(x) = x^{d_N} + x^{d_{(N-1)}} + \cdots + x^{d_1},$$

where $d_N > d_{(N-1)} > \cdots > d_1 \ge 0$. Therefore, we have

$$z_{d_N+j} = z_{d_{(N-1)}+j} + z_{d_{(N-2)}+j} + \cdots + z_{d_1+j}. \qquad (2.16)$$

By (2.14) and (2.16),

$$\prod_{i=1}^{k} Tr(\alpha^{t_i}\alpha^{d_N+j}) + \prod_{i=1}^{k} Tr(\alpha^{t_i}\alpha^{d_{(N-1)}+j}) + \prod_{i=1}^{k} Tr(\alpha^{t_i}\alpha^{d_{(N-2)}+j}) + \cdots + \prod_{i=1}^{k} Tr(\alpha^{t_i}\alpha^{d_1+j}) = 0.$$

Given (2.15) and E_{n_i} is independent of j,

$$\sum_i E_{n_i}(\alpha^{n_i(d_N+j)} + \alpha^{n_i(d_{(N-1)}+j)} + \cdots + \alpha^{n_i(d_1+j)}) = \sum_i E_{n_i} m_{\tilde{\mathbf{z}}}(\alpha^{n_i})\alpha^{n_i j} = 0, \qquad (2.17)$$

for all $j \ge 1$. Therefore, by constructing a Vandermonde determinant, we conclude that

$$E_{n_i} m_{\tilde{\mathbf{z}}}(\alpha^{n_i}) = 0. \qquad (2.18)$$

Hence, $E_{n_i} \neq 0$ implies α^{n_i} is a root of the minimal polynomial $m_{\tilde{\mathbf{z}}}(x)$ of $\tilde{\mathbf{z}}$, which contributes to the linear complexity of $\tilde{\mathbf{z}}$. When $w_2(n_i) = k$, we have a very explicit expression D_{n_i} for E_{n_i}, then the conclusion follows. □

Lemma 2.2: Suppose $\tilde{\mathbf{s}} = (s_1, s_2, \ldots, s_i, \ldots)$ is a maximal period sequence over $\mathbb{F}_2$ with its minimal polynomial $m_{\tilde{\mathbf{s}}}(x)$ having degree L. Let ${}_{(t+i\phi)}\tilde{\mathbf{s}} = \delta_{(t+i\phi)}(\tilde{\mathbf{s}})$ for $i = 0, 1, \ldots, k-1$. Denote the product $\prod_{i=0}^{k-1} [{}_{(t+i\phi)}\tilde{\mathbf{s}}]$ by $\tilde{\mathbf{z}}$. If $\gcd(2^L - 1, \phi) = 1$, then $L(\tilde{\mathbf{z}}) \geq \begin{pmatrix} L \\ k \end{pmatrix}$.

Proof: Suppose α is a root of $m_{\tilde{\mathbf{s}}}(x)$ in $\mathbb{F}_{2^L}$ and $n = 2^{e_1} + 2^{e_2} + \cdots + 2^{e_k}$ where $0 \leq e_1 < e_2 < \cdots < e_k \leq L$. Then,

$$D_n = \begin{vmatrix} \alpha^{t2^{e_1}} & \alpha^{(t+\phi)2^{e_1}} & \cdots & \alpha^{(t+(k-1)\phi)2^{e_1}} \\ \alpha^{t2^{e_2}} & \alpha^{(t+\phi)2^{e_2}} & \cdots & \alpha^{(t+(k-1)\phi)2^{e_2}} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{t2^{e_k}} & \alpha^{(t+\phi)2^{e_k}} & \cdots & \alpha^{(t+(k-1)\phi)2^{e_k}} \end{vmatrix} = \prod_{i=1}^{k} \alpha^{t2^{e_i}} \prod_{x=2}^{k} \prod_{y=1}^{x-1} (\alpha^{\phi 2^{e_x}} - \alpha^{\phi 2^{e_y}}).$$

Therefore, $D_n \neq 0$ by considering $\gcd(2^L - 1, \phi) = 1$. According to Lemma 2.1, α^n is a root of the minimal polynomial $m_{\tilde{\mathbf{z}}}(x)$ of $\tilde{\mathbf{z}}$. Because we have $\binom{L}{k}$ values for n such that $w_2(n) = k$, $m_{\tilde{\mathbf{z}}}(x)$ must have at least $\binom{L}{k}$ roots. Hence, $L(\tilde{\mathbf{z}}) = \deg(m_{\tilde{\mathbf{z}}}(x)) \geq \begin{pmatrix} L \\ k \end{pmatrix}$. □

Theorem 2.4: Let $\tilde{\mathbf{s}} = (s_1, s_2, \ldots, s_i, \ldots)$ be a maximal period sequence over $\mathbb{F}_2$ with its minimal polynomial $m_{\tilde{\mathbf{s}}}(x)$ having degree L. Let ${}_{(t+i\phi)}\tilde{\mathbf{s}} = \delta_{(t+i\phi)}(\tilde{\mathbf{s}})$ for $i = 0, 1, \ldots, k-1$. Denote the linear combination of products $\sum_{x=0}^{N-1} \prod_{y=0}^{k-1} c_x[{}_{(t_x+y\phi)}\tilde{\mathbf{s}}]$ by $\tilde{\mathbf{z}}$ where N is a positive integer, $t_0 < t_1 < \cdots < t_{N-1}$ and not all c_j are zero. If $\gcd(2^L - 1, \phi) = 1$, then $L(\tilde{\mathbf{z}}) \geq \begin{pmatrix} L \\ k \end{pmatrix} - t_{N-1}$.

Proof: Denote $Tr_{\mathbb{F}_{q^L}/\mathbb{F}_q}(\alpha^j)$ again by $Tr(\alpha^j)$. Suppose α is a root of $m_{\tilde{\mathbf{s}}}(x)$ in $\mathbb{F}_{2^L}$. Similarly as in (2.15),

$$z_j = \sum_{x=0}^{N-1} \prod_{y=0}^{k-1} c_x[{}_{(t_x+y\phi)}s_j] = \sum_{x=0}^{N-1} \prod_{y=0}^{k-1} c_x Tr(\alpha^{t_x+y\phi}\alpha^j) \tag{2.19}$$

$$= \sum_{x=0}^{N-1} \sum_{i} c_x E_{n_i}(t_x, y)\alpha^{n_i j}. \tag{2.20}$$

Here $E_{n_i}(t_x, y)$ is independent of j.

Still as in Lemma 2.1, we suppose the minimal polynomial for $\tilde{\mathbf{z}}$ is

$$m_{\tilde{\mathbf{z}}}(x) = x^{d_m} + x^{d_{(m-1)}} + \cdots + x^{d_1},$$

where $d_m > d_{(m-1)} > \cdots > d_1 \geq 0$. Therefore, we have

$$z_{d_m+j} = z_{d_{(m-1)}+j} + z_{d_{(m-2)}+j} + \cdots + z_{d_1+j}.$$

By replacing each term with (2.19) and simplification,

$$\sum_{x=0}^{N-1}\prod_{y=0}^{k-1} c_x Tr(\alpha^{x+y\phi}\alpha^{d_m+j}) + \sum_{x=0}^{N-1}\prod_{y=0}^{k-1} c_x Tr(\alpha^{x+y\phi}\alpha^{d_{(m-1)}+j}) + \cdots + \sum_{x=0}^{N-1}\prod_{y=0}^{k-1} c_x Tr(\alpha^{x+y\phi}\alpha^{d_1+j}) = 0,$$

$$\sum_{v=1}^{m}(\sum_{x=0}^{N-1}\sum_{i} c_x E_{n_i}(t_x, y)\alpha^{n_i(d_v+j)}) = 0,$$

$$\sum_{i}[\sum_{x=0}^{N-1} c_x E_{n_i}(t_x, y)] m_{\tilde{\mathbf{z}}}(\alpha^{n_i})\alpha^{n_i j} = 0. \tag{2.21}$$

Since (2.21) holds for all j, then by constructing a Vandermonde determinant, we conclude that

$$m_{\tilde{\mathbf{z}}}(\alpha^{n_i})[\sum_{x=0}^{N-1} c_x E_{n_i}(t_x, y)] = 0. \tag{2.22}$$

Now we consider n_i such that $n_i = 2^{e_1} + 2^{e_2} + \cdots + 2^{e_k}$ where $0 \leq e_1 < e_2 < \cdots < e_k \leq L$. Then

$$E_{n_i}(t_x, y) = \prod_{i=1}^{k} \alpha^{t_x 2^{e_i}} \prod_{p=2}^{k}\prod_{q=1}^{p-1}(\alpha^{\phi 2^{e_p}} - \alpha^{\phi 2^{e_q}}).$$

Therefore,

$$\sum_{x=0}^{N-1} c_x E_{n_i}(t_x, y) = [\prod_{p=2}^{k}\prod_{q=1}^{p-1}(\alpha^{\phi 2^{e_p}} - \alpha^{\phi 2^{e_q}})] \sum_{x=0}^{N-1} c_x \alpha^{t_x(\sum_{i=1}^{k} 2^{e_i})} = A\sum_{x=0}^{N-1} c_x \alpha^{t_x n_i},$$

where $A = \prod_{p=2}^{k}\prod_{q=1}^{p-1}(\alpha^{\phi 2^{e_p}} - \alpha^{\phi 2^{e_q}}) \neq 0$ since $\gcd(2^L - 1, \phi) = 1$. If α^{n_i} is not a root of

$$B(x) = c_{(N-1)}x^{t_{(N-1)}} + c_{(N-2)}x^{t_{(N-2)}} + \cdots + c_0 x^{t_0},$$

then from(2.21) $m_{\tilde{\mathbf{z}}}(\alpha^{n_i}) = 0$, which implies α^{n_i} contributes to the linear complexity of $\tilde{\mathbf{z}}$. Since there are at most t_{N-1} roots for $B(x)$ and $\binom{L}{k}$ choices for such n_i, then we conclude that

$$L(\tilde{\mathbf{z}}) \geq \binom{L}{k} - t_{N-1}.$$

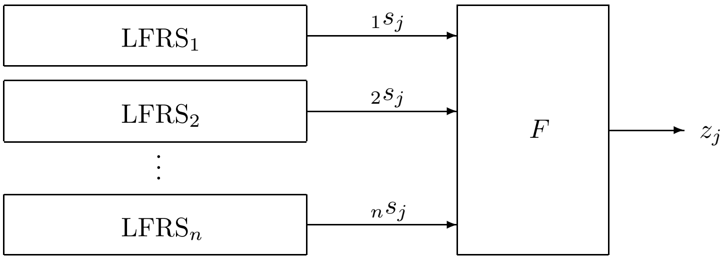

Figure 2.3: The Second Kind of Nonlinear Filter.

This completes the proof. □

An immediate consequence of Theorem 2.4 is that $L(\tilde{\mathbf{z}}) \geq \binom{L}{k} - N + 1$ by setting $t_i = i$ for $0 \leq i \leq N-1$. From the proof of Theorem 2.4, we can see that the obtained lower bound for the linear complexity of the generated sequence is only due to n_i in (2.20) with $w_2(n_i) = k$. (In fact we neglect some roots because of the difficulty in the expression of their coefficients in (2.21).) Then, the lower bound of the linear complexity of $\tilde{\mathbf{z}}$ will not decrease if we add some extra terms into (2.19), which can be expressed by summing $p < k$ products of $s_{i_1+j}, s_{i_2+j}, \ldots, s_{i_p+j}$ where $i_q \in \{t_x + y\phi\}$ for $1 \leq q \leq p, 0 \leq x \leq N-1$, and $0 \leq y \leq k-1$. In conclusion, to choose a nonlinear filter described in Figure 2.2 for a maximal period sequence with linear complexity L, we only need to select some appropriate parameters k, ϕ, c_x, t_m and d_m, such that

$$\tilde{\mathbf{z}} = \sum_{x=0}^{N-1} \prod_{y=0}^{k-1} c_x [{}_{(x+y\phi)}\tilde{\mathbf{s}}] + \sum_{m=0}^{N-1} \prod_{n=0}^{p-1} d_m [{}_{(t_m+n\phi)}\tilde{\mathbf{s}}],$$

where $p < k$. Then $L(\tilde{\mathbf{z}}) \geq \binom{L}{k} - N + 1$.

2. Nonlinear Operations on Several Maximal Period Sequences

The second kind of nonlinear filter consists of a nonlinear mapping device F and several maximal period sequence generating devices. Instead of using several different states of a maximal period sequence as the input of F in the first kind of nonlinear filter, here we use the same state of several different maximal period sequences ${}_i\tilde{\mathbf{s}}$ generated by LFSR$_i$ as the input. See Figure 2.3.

To obtain the main results, we need some more algebraic tools.

Lemma 2.3: Suppose $\alpha \in \mathbb{F}_{q^m}$ and $\beta \in \mathbb{F}_{q^n}$, where $\gcd(m, n) = 1$, then

$$Tr_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\alpha)Tr_{\mathbb{F}_{q^n}/\mathbb{F}_q}(\beta) = Tr_{\mathbb{F}_{q^{mn}}/\mathbb{F}_q}(\alpha\beta). \tag{2.23}$$

Proof: Since $m \mid mn$ and $n \mid mn$, $\mathbb{F}_{q^m}$ and $\mathbb{F}_{q^m}$ are both subfields of $\mathbb{F}_{q^{mn}}$. Therefore we have

$$Tr_{\mathbb{F}_{q^{mn}}/\mathbb{F}_q}(\alpha\beta) = \sum_{i=0}^{mn-1} (\alpha\beta)^{q^i} = \sum_{i=0}^{mn-1} (\alpha)^{q^i}(\beta)^{q^i}. \tag{2.24}$$

Consider $\alpha \in \mathbb{F}_{q^m}$ and $\beta \in \mathbb{F}_{q^n}$, then $\alpha^{q^i} = \alpha^{q^{i \bmod m}}$ and $\beta^{q^i} = \beta^{q^{i \bmod n}}$. Because $\gcd(m, n) = 1$, by the Chinese remainder theorem, there is a one-to-one correspondence between i and $(i \bmod m, i \bmod n)$. Therefore, replace i with $(i \bmod m, i \bmod n)$ in (2.24) and reorder the terms, then we have

$$Tr_{\mathbb{F}_{q^{mn}}/\mathbb{F}_q}(\alpha\beta) = \sum_{x=0}^{m-1}\sum_{y=0}^{n-1} \alpha^{q^x}\beta^{q^y} = \sum_{x=0}^{m-1} \alpha^{q^x} \sum_{y=0}^{n-1} \beta^{q^y} = Tr_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\alpha)Tr_{\mathbb{F}_{q^n}/\mathbb{F}_q}(\beta). \tag{2.25}$$

□

Definition 2.8: Call $\alpha \in \mathbb{F}_{q^m}^*$ a **quintessential element** of $\mathbb{F}_{q^m}/\mathbb{F}_q$ (or briefly of $\mathbb{F}_{q^m}$) if $\{\alpha, \alpha^q, \alpha^{q^2}, \ldots, \alpha^{q^{m-1}}\}$ are all distinct.

An important fact is that a nonzero root of an irreducible polynomial $f(x)$ over $\mathbb{F}_q$ is a quintessential element of $\mathbb{F}_{q^{\deg(f(x))}}$.

Lemma 2.4: Suppose $\gcd(m, n) = 1$. Let $\alpha \in \mathbb{F}_{q^m}$ be a quintessential element of $\mathbb{F}_{q^m}$ and $\beta \in \mathbb{F}_{q^n}$ be a quintessential element of $\mathbb{F}_{q^n}$. If $\gcd(q-1, m) = 1$ and $\gcd(q-1, n) = 1$, then $\alpha\beta$ is a quintessential element of $\mathbb{F}_{q^{mn}}$.

Proof: Firstly consider $q = 2$. Since $m \mid mn$ and $n \mid mn$, $\mathbb{F}_{2^m}$ and $\mathbb{F}_{2^n}$ are both subfields of $\mathbb{F}_{2^{mn}}$, which implies $\alpha\beta \in \mathbb{F}_{2^{mn}}$. Suppose $(\alpha\beta)^{2^x} = (\alpha\beta)^{2^y}$ for some $0 \le x < y \le mn - 1$, then $\alpha^{2^x - 2^y} = \beta^{2^y - 2^x}$. Because $\gcd(m, n) = 1$, $\mathbb{F}_{2^m} \cap \mathbb{F}_{2^n} = \mathbb{F}_2$. Then, we must have $\alpha^{(2^x - 2^y)} = c$ and $\beta^{(2^y - 2^x)} = c$ for some $c \in \mathbb{F}_2$. Then $\alpha^{2^{x \bmod m}} = c\alpha^{2^{y \bmod m}}$. Obviously, c must be 1. Otherwise we have $\alpha = 0$. So $c = 1$ forces $\alpha^{(2^x - 2^y)} = 1$, which implies $x \equiv y \bmod m$. Similarly, we can conclude $x \equiv y \bmod n$ by considering β. Since $0 \le x, y \le mn - 1$, we must have $x = y$. Contradiction to $x < y$.

Finally, consider $q > 2$. Here we claim that for any two elements in $\{\alpha, \alpha^q, \alpha^{q^2}, \ldots, \alpha^{q^{m-1}}\}$, one cannot be a scalar multiple over $\mathbb{F}_q$ of the other one. Suppose not, say we have $\alpha^{q^x} = c\alpha^{q^y}$ where $c \in \mathbb{F}_q^*$ and $x < y$. Since $c^q = c$, we can raise the identity $\alpha^{q^x} = c\alpha^{q^y}$ to the qth power for $m-1$ times. Then we have m identities. Since $\alpha^{q^{x+i}}$ and $\alpha^{q^{y+i}}$ for $0 \leq i \leq m-1$ run through all elements in $\{\alpha, \alpha^q, \alpha^{q^2}, \ldots, \alpha^{q^{m-1}}\}$, then after multiplying the m identities, we must have $c^m = 1$. Therefore, the order ord_c of c in $\mathbb{F}_q^*$, the smallest positive integer such that $c^{ord_c} = 1$, must divide $q-1$ and m. However, $\gcd(q-1, m) = 1$. Hence $c = 1$ which is a contradiction to the definition of a quintessential element.

Now suppose $(\alpha\beta)^{q^x} = (\alpha\beta)^{q^y}$ for some $0 \leq x < y \leq mn-1$, then $\alpha^{q^x - q^y} = \beta^{q^y - q^x}$. Because $\gcd(m, n) = 1$, $\mathbb{F}_{q^m} \cap \mathbb{F}_{q^n} = \mathbb{F}_q$. Then, we must have $\alpha^{(q^x - q^y)} = c$ and $\beta^{(q^y - q^x)} = c$ for some $c \in \mathbb{F}_q^*$. Then $\alpha^{q^{x \bmod m}} = c\alpha^{q^{y \bmod m}}$. By our above claim, c must be 1. So $c = 1$ forces $\alpha^{(q^x - q^y)} = 1$, which implies $x \equiv y \bmod m$. Similarly, we can conclude $x \equiv y \bmod n$ by considering β. Since $0 \leq x, y \leq mn-1$, we must have $x = y$. Contradiction to $x < y$. □

Lemma 2.5: Let ${}_a\tilde{\mathbf{s}}$ and ${}_b\tilde{\mathbf{s}}$ be two periodic sequences with the irreducible minimal polynomials $m_{{}_a\tilde{\mathbf{s}}}(x)$ and $m_{{}_b\tilde{\mathbf{s}}}(x)$ over $\mathbb{F}_q$. Suppose $\deg(m_{{}_a\tilde{\mathbf{s}}}(x)) = m$ and $\deg(m_{{}_b\tilde{\mathbf{s}}}(x)) = n$ with $q-1, m, n$ being pairwise coprime. Then the minimal polynomial $m_{\tilde{\mathbf{z}}}(x)$ of the product sequence $\tilde{\mathbf{z}} = ({}_a\tilde{\mathbf{s}})({}_b\tilde{\mathbf{s}})$ is irreducible in $\mathbb{F}_q$ with degree mn.

Proof: From (2.7), we can assume ${}_as_j = Tr_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\theta_a\alpha^j)$ and ${}_bs_j = Tr_{\mathbb{F}_{q^n}/\mathbb{F}_q}(\theta_b\beta^j)$, where $m_{{}_a\tilde{\mathbf{s}}}(\alpha) = 0$, $m_{{}_b\tilde{\mathbf{s}}}(\beta) = 0$ and $\theta_a \in \mathbb{F}_{q^m}$ $\theta_b \in \mathbb{F}_{q^m}$ are nonzero elements. Therefore, α is a quintessential element of $\mathbb{F}_{q^m}$ and β is a quintessential element of $\mathbb{F}_{q^n}$ given the irreducibility. Now

$$z_j = ({}_as_j)({}_bs_j) = Tr_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\theta_a\alpha^j)Tr_{\mathbb{F}_{q^n}/\mathbb{F}_q}(\theta_b\beta^j). \tag{2.26}$$

By Lemma 2.3, $z_j = Tr_{\mathbb{F}_{q^{mn}}/\mathbb{F}_q}(\theta(\alpha\beta)^j)$ where $\theta = \theta_a\theta_b$. Let

$$m_{\tilde{\mathbf{z}}}(x) = x^{d_m} - a_{m-1}x^{d_{(m-1)}} - \cdots - a_1x^{d_1},$$

where $d_m > d_{(m-1)} > \cdots > d_1 \geq 0$. Therefore, we have

$$z_{d_m+j} = a_{m-1}z_{d_{(m-1)}+j} + a_{m-2}z_{d_{(m-2)}+j} + \cdots + a_1z_{d_1+j}.$$

Next for the simplicity of expression, we denote $Tr_{\mathbb{F}_{q^{mn}}/\mathbb{F}_q}(x)$ by $Tr(x)$ without confusion. So we have

$$Tr(\theta(\alpha\beta)^{d_m+j}) - a_{m-1}Tr(\theta(\alpha\beta)^{d_{m-1}+j}) - \cdots - a_1Tr(\theta(\alpha\beta)^{d_1+j}) = 0. \tag{2.27}$$

Expand each term in (2.27) and simplify it, then for $j \geq 1$ we have

$$\sum_{i=0}^{mn-1} \theta^{q^i} m_{\tilde{\mathbf{z}}}((\alpha\beta)^{q^i})(\alpha\beta)^{q^i j} = 0. \tag{2.28}$$

By Lemma 2.4, $\alpha\beta$ is a quintessential element in $\mathbb{F}_{q^{mn}}$. Therefore, by constructing a Vandermonde determinant, we conclude that

$$\theta^{q^i} m_{\tilde{\mathbf{z}}}((\alpha\beta)^{q^i}) = 0 \tag{2.29}$$

for $0 \leq i \leq mn-1$. Since $\theta \neq 0$, $(\alpha\beta)^{q^i}$ must be a root of $m_{\tilde{\mathbf{z}}}(x)$, which implies the minimal polynomial of $\alpha\beta$ is a factor of $m_{\tilde{\mathbf{z}}}(x)$. However, the minimal polynomial of $\alpha\beta$ is a characteristic polynomial after direct verification. Hence, $m_{\tilde{\mathbf{z}}}(x)$ is the same as the minimal polynomial of $\alpha\beta$, so it is irreducible over $\mathbb{F}_q$ and has degree mn. □

Theorem 2.5: Let ${}_a\tilde{\mathbf{s}}$ and ${}_b\tilde{\mathbf{s}}$ be two periodic sequences with the minimal polynomials $m_{{}_a\tilde{\mathbf{s}}}(x)$ and $m_{{}_b\tilde{\mathbf{s}}}(x)$ over $\mathbb{F}_q$. Suppose $\deg(m_{{}_a\tilde{\mathbf{s}}}(x)) = m$, the roots of $m_{{}_a\tilde{\mathbf{s}}}(x)$ are simple and lie in $\mathbb{F}_{q^M} \setminus \mathbb{F}_q$, and none of its roots is a scalar multiple over $\mathbb{F}_q$ of its any other root. Also suppose $\deg(m_{{}_b\tilde{\mathbf{s}}}(x)) = n$, the roots of $m_{{}_b\tilde{\mathbf{s}}}(x)$ are simple and lie in $\mathbb{F}_{q^N} \setminus \mathbb{F}_q$ and none of its roots is a scalar multiple over $\mathbb{F}_q$ of its any other root. Let M, N, and $q-1$ be pairwise coprime. Then the minimal polynomial $m_{\tilde{\mathbf{z}}}(x)$ of the product sequence $\tilde{\mathbf{z}} = ({}_a\tilde{\mathbf{s}})({}_b\tilde{\mathbf{s}})$ has mn simple roots in $\mathbb{F}_{q^{MN}} \setminus \{\mathbb{F}_{q^M} \bigcup \mathbb{F}_{q^N}\}$.

Proof: Let $G_{{}_a\tilde{\mathbf{s}}}(x)$ and $G_{{}_b\tilde{\mathbf{s}}}(x)$ be the generating functions of ${}_a\tilde{\mathbf{s}}$, ${}_b\tilde{\mathbf{s}}$, respectively. Then by (2.9) we have

$$\begin{aligned} G_{{}_a\tilde{\mathbf{s}}}(x) &= \frac{g_{{}_a\tilde{\mathbf{s}}}(x)}{m^*_{{}_a\tilde{\mathbf{s}}}(x)} &&= \sum_i \frac{{}_aQ_i(x)}{{}_aP_i(x)}, \text{ where } {}_aP_i(x) \text{ are all irreducible factors of } m_{{}_a\tilde{\mathbf{s}}}(x), \\ G_{{}_b\tilde{\mathbf{s}}}(x) &= \frac{g_{{}_b\tilde{\mathbf{s}}}(x)}{m^*_{{}_b\tilde{\mathbf{s}}}(x)} &&= \sum_j \frac{{}_bQ_j(x)}{{}_bP_j(x)}, \text{ where } {}_bP_i(x) \text{ are all irreducible factors of } m_{{}_b\tilde{\mathbf{s}}}(x). \end{aligned}$$

Obviously, $\deg({}_aQ_i(x)) < \deg({}_aP_i(x))$ and $\deg({}_bQ_j(x)) < \deg({}_bP_j(x))$. If we denote ${}_am_i = \deg({}_aP_i(x))$ and ${}_bm_j = \deg({}_bP_j(x))$, then by the field theory, ${}_am_i \mid M$ and ${}_bm_j \mid N$ since all the roots of $m_{{}_a\tilde{\mathbf{s}}}(x)$ are in $\mathbb{F}_{q^M}$ and all the roots of $m_{{}_b\tilde{\mathbf{s}}}(x)$ are in $\mathbb{F}_{q^N}$. Let ${}_a^i\tilde{\mathbf{s}}$ denote the sequence corresponding to $\frac{{}_aQ_i(x)}{{}_aP_i(x)}$, ${}_b^j\tilde{\mathbf{s}}$ denote the sequence corresponding to $\frac{{}_bQ_j(x)}{{}_bP_j(x)}$ and ${}^{ij}\tilde{\mathbf{z}}$ be $({}_a^i\tilde{\mathbf{s}})({}_b^j\tilde{\mathbf{s}})$. Therefore,

$$\tilde{\mathbf{z}} = ({}_a\tilde{\mathbf{s}})({}_b\tilde{\mathbf{s}}) = [\sum_i ({}_a^i\tilde{\mathbf{s}})][\sum_j ({}_b^j\tilde{\mathbf{s}})] = \sum_i \sum_j ({}_a^i\tilde{\mathbf{s}})({}_b^j\tilde{\mathbf{s}}) = \sum_i \sum_j ({}^{ij}\tilde{\mathbf{z}}). \tag{2.30}$$

Since $M, N, q-1$ are pairwise coprime, we have ${}_am_i, {}_bm_j, q-1$ are pairwise coprime for all i, j. Now according to Lemma 2.5, we conclude that the minimal polynomial $m_{{}^{ij}\tilde{\mathbf{z}}}(x)$ of ${}^{ij}\tilde{\mathbf{z}}$ is irreducible and has degree $({}_am_i)({}_bm_j)$. Hence we can represent the nth term of ${}^{ij}\tilde{\mathbf{z}}$ by

$$ {}^{ij}z_n = Tr_{\mathbb{F}_{q^{({}_am_i)({}_bm_j)}}/\mathbb{F}_q}(A_iB_j(\alpha_i\beta_j)^n), \tag{2.31}$$

where $A_i \in \mathbb{F}_{q^{({}_am_i)}}, B_j \in \mathbb{F}_{q^{({}_bm_j)}}$, α_i is a root of ${}_aP_i(x)$ and β_j is a root of ${}_bP_j(x)$.

Given all $\alpha_i \in \mathbb{F}_{q^M} \setminus \mathbb{F}_q, \beta_j \in \mathbb{F}_{q^N} \setminus \mathbb{F}_q$, we claim that the minimal polynomial over $\mathbb{F}_q$ of $\alpha_i\beta_j$ and the minimal polynomial over $\mathbb{F}_q$ of $\alpha_{i'}\beta_{j'}$ are different if $i \neq i'$ and $j \neq j'$. Suppose not, then given that the roots of irreducible polynomials are conjugate, we have $\alpha_i\beta_j = (\alpha_{i'}\beta_{j'})^{q^c}$ for some positive integer c. Then $\alpha_i\alpha_{i'}^{-q^c} = \beta_{j'}^{q^c}\beta_j^{-1}$. Since $\mathbb{F}_{q^M} \cap \mathbb{F}_{q^N} = \mathbb{F}_q$, we conclude that $\alpha_i\alpha_{i'}^{-q^c}, \beta_{j'}^{q^c}\beta_j^{-1} \in \mathbb{F}_q$, which implies $\alpha_i = d\alpha_{i'}^{q^c}$ and $\beta_{j'}^{q^c} = d\beta_j$ for some $d \in \mathbb{F}_q$. Contradiction. Hence we conclude that the degree of the minimal polynomial of $\tilde{\mathbf{z}}$ is $\sum\limits_{i,j}({}_am_i)({}_bm_j) = \sum\limits_{i}({}_am_i)n = mn$. Moreover, $\alpha_i\beta_j$ obviously lies in $\mathbb{F}_{q^{MN}}$. If it is in $\mathbb{F}_{q^M}$, we have β_j in $\mathbb{F}_{q^M}$, contradiction. Similarly for $\mathbb{F}_{q^N}$. So we conclude that $\alpha_i\beta_j \in \mathbb{F}_{q^{MN}} \setminus \{\mathbb{F}_{q^M} \bigcup \mathbb{F}_{q^N}\}$. □

Theorem 2.6: Let ${}_i\tilde{\mathbf{s}}$ for $i = 1, 2, \ldots, N$ be periodic sequences over $\mathbb{F}_q$, whose corresponding minimal polynomials are $m_{{}_i\tilde{\mathbf{s}}}(x)$ with degree $m_{{}_i\tilde{\mathbf{s}}}$. Suppose all $m_{{}_i\tilde{\mathbf{s}}}(x)$ have only simple roots ${}_i\alpha_j \in \mathbb{F}_{q^{m_{{}_i\tilde{\mathbf{s}}}}} \setminus \mathbb{F}_q$ where $j = 1, 2, \ldots, m_{{}_i\tilde{\mathbf{s}}}$ and none of them is a scalar multiple over $\mathbb{F}_q$ of ${}_{i'}\alpha_{j'}$, which is a root of $m_{{}_{i'}\tilde{\mathbf{s}}}(x)$ for all $1 \leq i' \leq N, 1 \leq j' \leq m_{{}_{i'}\tilde{\mathbf{s}}}$. If $\gcd(m_{{}_i\tilde{\mathbf{s}}}, m_{{}_j\tilde{\mathbf{s}}}) = 1$ for all $i \neq j$, then $\tilde{\mathbf{z}} = \prod\limits_{i=1}^{N}({}_i\tilde{\mathbf{s}})$ has the minimal polynomial $m_{\tilde{\mathbf{z}}}(x)$ of degree $\prod\limits_{i=1}^{N} m_{{}_i\tilde{\mathbf{s}}}$ whose roots are all simple and lie in $\mathbb{F}_{q^m} \setminus \bigcup\limits_{j} \mathbb{F}_{q^j}$ where $m = \prod\limits_{i=1}^{N} m_{{}_i\tilde{\mathbf{s}}}$ and j runs through all $(N-1)$th-order products of $m_{{}_i\tilde{\mathbf{s}}}$.

Proof: Let ${}_1\tilde{\mathbf{z}} = ({}_1\tilde{\mathbf{s}})({}_2\tilde{\mathbf{s}})$. Since none of the roots of $m_{{}_i\tilde{\mathbf{s}}}(x)$ is a scalar multiple over $\mathbb{F}_q$ of ${}_{i'}\alpha_{j'}$, which is a root of $m_{{}_{i'}\tilde{\mathbf{s}}}(x)$ for all $1 \leq i' \leq N, 1 \leq j' \leq m_{{}_{i'}\tilde{\mathbf{s}}}$, then from the proof of Lemma 2.4, we can neglect the requirement $\gcd(m_{{}_i\tilde{\mathbf{s}}}, q-1) = 1$. Hence, by Theorem 2.5, we conclude that ${}_1\tilde{\mathbf{z}}$ has the minimal polynomial $m_{{}_1\tilde{\mathbf{z}}}(x)$ of degree $m_{{}_1\tilde{\mathbf{s}}}m_{{}_2\tilde{\mathbf{s}}}$ whose roots are all simple and lie in $\mathbb{F}_{q^{m_{{}_1\tilde{\mathbf{s}}}m_{{}_2\tilde{\mathbf{s}}}}} \setminus \{\mathbb{F}_{q^{m_{{}_1\tilde{\mathbf{s}}}}} \bigcup \mathbb{F}_{q^{m_{{}_2\tilde{\mathbf{s}}}}}\}$. Then consider ${}_k\tilde{\mathbf{z}} = ({}_{k-1}\tilde{\mathbf{z}})({}_{k+1}\tilde{\mathbf{s}})$ for $2 \leq k \leq N-1$ and use Theorem 2.5 repeatedly, the conclusion follows. □

Theorem 2.7: Suppose the nonlinear function F over $\mathbb{F}_q$ is given by

$$\begin{aligned} F(x_1, x_2, \ldots, x_n) = \quad & a_0 + \sum_i a_i x_i + \sum_{i,j} a_{i,j} x_i x_j + \cdots \\ & + a_{1,2,\ldots,n} x_1 x_2 \cdots x_n, \end{aligned}$$

where all the coefficients are in $\mathbb{F}_q$. Let ${}_i\tilde{\mathbf{s}}$ for $i = 1, 2, \ldots, n$ be sequences over $\mathbb{F}_q$, whose corresponding minimal polynomials are $m_{{}_i\tilde{\mathbf{s}}}(x)$ with degree $m_{{}_i\tilde{\mathbf{s}}}$. Suppose all $m_{{}_i\tilde{\mathbf{s}}}(x)$ have only simple roots ${}_i\alpha_j \in \mathbb{F}_{q^{m_{{}_i\tilde{\mathbf{s}}}}} \setminus \mathbb{F}_q$ where $j = 1, 2, \ldots, m_{{}_i\tilde{\mathbf{s}}}$ and none of them is a scalar multiple over $\mathbb{F}_q$ of ${}_{i'}\alpha_{j'}$, which is the root of $m_{{}_{i'}\tilde{\mathbf{s}}}(x)$ for all $1 \leq i' \leq n, 1 \leq j' \leq m_{{}_{i'}\tilde{\mathbf{s}}}$. If $\gcd(m_{{}_i\tilde{\mathbf{s}}}, m_{{}_j\tilde{\mathbf{s}}}) = 1$ for all $i \neq j$, then

$$\tilde{\mathbf{z}} = F({}_1\tilde{\mathbf{s}},\ {}_2\tilde{\mathbf{s}}, \ldots,\ {}_n\tilde{\mathbf{s}}) \tag{2.32}$$

has the minimal polynomial $m_{\tilde{\mathbf{z}}}(x)$ of degree

$$M = \bar{F}(m_{{}_1\tilde{\mathbf{s}}}, m_{{}_2\tilde{\mathbf{s}}}, \ldots, m_{{}_n\tilde{\mathbf{s}}}), \tag{2.33}$$

where $\bar{F}$ is defined as F, but the coefficients for each term are 1 if the corresponding coefficients of F are nonzero and zero otherwise.

Proof: For each term in (2.32), by Theorem 2.5, we know that the minimal polynomial of $\prod_j ({}_{i_j}\tilde{\mathbf{s}})$ has degree $\prod_j m_{{}_{i_j}\tilde{\mathbf{s}}}$. The nonzero scalar multiplication over $\mathbb{F}_q$ of a sequence does not change its linear recurrence relations, which implies it does not change its linear complexity. Since $\gcd(m_{{}_i\tilde{\mathbf{s}}}, m_{{}_j\tilde{\mathbf{s}}}) = 1$ for all $i \neq j$, no two distinct terms in (2.32) after expansion have the same minimal polynomial. Thus, the sum of all the terms in (2.32), $\tilde{\mathbf{z}}$ must have its minimal polynomial being the product of all the minimal polynomials of each terms in (2.32). Consequently, $\deg(m_{\tilde{\mathbf{z}}}(x))$ must be the sum of all the degrees of the minimal polynomials of nonzero terms in (2.32), which is expressed by (2.33). □

2.3 The BAA Attacks on the Two Nonlinear Filters

The best affine approximation (BAA) attack was first introduced by Rueppel for the analysis of the S-boxes of the Data Encryption Standard in about 1986. And in 1988, C. Ding, G. Xiao and W. Shan developed the BAA method to analyze stream ciphers with some algebraic techniques and error-correcting techniques in [9]. Under the assumption that we have known the nonlinear filter structure already, we will introduce it in this section. Here we denote $xy = \sum\limits_{i=1}^{n} x_i y_i$ for $x, y \in \mathbb{F}_2^n$ and let $\oplus$ be the addition over $\mathbb{F}_2$.

Definition 2.9: Let $f(x)$ be a function from $\mathbb{F}_2^n$ to $\mathbb{F}_2$. Then call $wx \oplus l$ the **best affine approximation** of $f(x)$ for $w, l \in \mathbb{F}_2^n$, if the sum over the real number field

$$\sum_{x \in \mathbb{F}_2^n} f(x) \oplus wx \oplus l \tag{2.34}$$

achieves its minimal value.

Definition 2.10: For $x, y \in \mathbb{F}_2^n$, the **Walsh function** Q is defined to be $Q(x, y) = (-1)^{xy}$.

Definition 2.11: For any Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$, define its **first kind of Walsh transformation** S_f as

$$S_f(w) = \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} f(x) Q(w, x), \tag{2.35}$$

and its **second kind of Walsh transformation** $S_{(f)}$ as

$$S_{(f)}(w) = \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} Q(w, x)(-1)^{f(x)}. \tag{2.36}$$

The relation between f and its two Walsh transformations is given by

$$f(x) = \sum_{w \in \mathbb{F}_2^n} S_f(w) Q(w, x) = \frac{1}{2} - \frac{1}{2} \sum_{w \in \mathbb{F}_2^n} S_{(f)}(w) Q(w, x), \tag{2.37}$$

and

$$S_{(f)}(w) = \begin{cases} -2S_f(w), & w \neq 0, \\ 1 - 2S_f(w), & w = 0. \end{cases} \tag{2.38}$$

Theorem 2.8: Let $P_f(wx \oplus l)$ denote the probability of $f(x) = wx \oplus l$ for $w, l \in \mathbb{F}_2^n$. Then, $P_f(wx) = \frac{1}{2} + \frac{1}{2}S_{(f)}(w)$ and $P_f(wx) = \frac{1}{2} - S_f(w)$ if $w \neq 0$ while $P_f(wx) = 1 - S_f(w)$ if $w = 0$.

Proof: By the definition of the second Walsh transform, we have

$$\begin{aligned} S_{(f)}(w) &= \frac{1}{2^n}[\#\{x|f(x) = wx\} - \#\{x|f(x) \neq wx\}] \\ &= \frac{1}{2^n}[\#\{x|f(x) \neq wx\} - 2^n + \#\{x|f(x) = wx\}] \\ &= \frac{1}{2^n}[2\#\{x|f(x) = wx\} - 2^n] \\ &= 2P_f(wx) - 1 \\ &= \frac{1}{2^n}[2^n - 2\#\{x|f(x) = wx \oplus 1\}] \\ &= 1 - 2P_f(wx \oplus 1). \end{aligned} \tag{2.39}$$

Now from the above formula and (2.38), the conclusion follows. □

Theorem 2.9: Let $M = \max\{|S_{(f)}(w)| \mid w \in \mathbb{F}_2^n\}$ and $|S_{(f)}(w_0)| = M$. If $S_{(f)}(w) \geq 0$, then w_0x is the best affine approximation of $f(x)$ and the probability of agreement is given by $P_f(wx) = \frac{1}{2} + \frac{1}{2}M$; else, where $S_{(f)}(w) < 0$, $w_0x \oplus 1$ is the BAA and $P_f(wx \oplus 1) = \frac{1}{2} + \frac{1}{2}M$.

Proof: From (2.39), we have $P_f(wx) = \frac{1}{2} + \frac{1}{2}S_{(f)}(w)$ and $P_f(wx \oplus 1) = \frac{1}{2} - \frac{1}{2}S_{(f)}(w)$, then the conclusion is obvious. □

Under the assumption of knowing the nonlinear filter, we could find $|S_{(f)}(w_0)| = \max\{|S_{(f)}(w)| \mid w \in \mathbb{F}_2^n\}$ by computation before the cipher analysis. Then if $|S_{(f)}(w_0)|$ is big enough to guarantee a high agreement probability between F and w_0x, then we can use w_0x to replace the nonlinear filter F, which will decrease the linear complexity of the filtered key stream dramatically. This is the basic idea of the BAA attack. Let us consider a nonlinear filter of the first kind pictured in Figure 2.2 for example. Suppose the sequence generated by LFSR has linear complexity $2L$ and the nonlinear filter F is given by

$$F(x_1, x_2, \ldots, x_{2L}) = \sum_{i=1}^{L} x_i + \prod_{j=L+1}^{2L} x_j. \tag{2.40}$$

According to the results in Section 2.2, we know the filtered sequence has the linear complexity greater than $\binom{2L}{L} - L$ if we fill each variable in F as in Theorem 2.4. But after computing the second kind of Walsh transform of F, we have $M = \max\{|S_{(F)}(w)| \mid w \in \mathbb{F}_2^n\} = 1 - 2^{1-L}$ with $w_0 x = x_1 \oplus x_2 \oplus \cdots \oplus x_L$ and $P_F(w_0 x) = \frac{1}{2} + \frac{1}{2} S_{(F)}(w_0) = 1 - 2^{-L}$. Obviously, when $L \geq 10$, $P_F(w_0 x) \geq 0.999$. And after replacing the nonlinear filter F by $w_0 x$, the linear complexity of the generating key stream is less than or equal to the linear complexity of the sequence obtained by the LFSR before filtering, which decreases from $\binom{2L}{L} - L$ to $2L$ with very high agreement probability greater than 0.999 if $L \geq 10$. Therefore, if we know $2L$ bits of the key stream, we could predict the following key stream bits with high correct probability, which means the BAA attack to this nonlinear filter F defined by (2.40) is very successful.

By a similar method, we conduct the BAA attack on the second kind of nonlinear filter depicted in Figure 2.3. Suppose the linear complexity of LFSR$_i$ in Figure 2.3 is m_i, and the nonlinear filter F is given by (2.40) too. Then still after some computation, we have found $w_0 x = x_1 \oplus x_2 \oplus \cdots \oplus x_L$ and $P_F(w_0 x) = \frac{1}{2} + \frac{1}{2} S_{(F)}(w_0) = 1 - 2^{-L}$. Denote $\tilde{\mathbf{z}} = \sum_{i=1}^{L} ({}_i\tilde{\mathbf{s}})$ over $\mathbb{F}_2$. So we have $G_{{}_i\tilde{\mathbf{s}}}(x) = \frac{g_{{}_i\tilde{\mathbf{s}}}(x)}{m^*_{{}_i\tilde{\mathbf{s}}}(x)}$, which implies $G_{\sum_j ({}_j\tilde{\mathbf{s}})}(x) = \sum_j \frac{g_{{}_j\tilde{\mathbf{s}}}(x)}{m^*_{{}_j\tilde{\mathbf{s}}}(x)} = \frac{g_{\tilde{\mathbf{z}}}(x)}{\prod_j m_{{}_j\tilde{\mathbf{s}}}(x)}$. Let $\bar{F}$ be defined as in Theorem 2.7. Since the reciprocal polynomial of the minimal polynomial of $\tilde{\mathbf{z}}$ is a factor of $\prod_j m_{{}_j\tilde{\mathbf{s}}}(x)$ and $\tilde{\mathbf{z}}$ is periodic, $L(\tilde{\mathbf{z}}) \leq \sum_{i=1}^{L} m_i$, which will be generally much smaller than $\bar{F}(m_1, m_2, \ldots, m_L)$. Hence, with $2\sum_{i=1}^{L} L_i$ consecutive bits of the key stream, we can predict all the bits with the correct probability almost being 1. Therefore, the BAA attack can successfully break the key stream generated by the $2L$ LFSRs and filtered by the F defined in (2.40).

Chapter 3

Random Sequences and Their Linear Complexity Profiles

From the proof of perfect security of the Vernam one-time pad cipher in Section 1.2, we see that just a random sequence being the key stream over $\mathbb{F}_q$ is sufficient to guarantee *perfect security.* So the task cryptographers face is to construct random sequences to be the key streams. But how to describe the randomness using mathematical language, or equivalently, how to measure unpredictability by mathematical tools? Answers to these questions are the core contents in this chapter. After we justify the method to measure the randomness by *linear complexity profiles* of sequences, we will explore the probabilistic properties of random sequences.

3.1 Randomness of Sequences

In this section, we will discuss the basic fundamentals of a reasonable tool to measure randomness. Firstly, let us consider the outputs of finite state machines. Suppose M is an n-state machine and it is embedded in the finite field $\mathbb{F}$ with k elements. Each time M maps its internal states to an output. Since it could only have k^n different n-tuples $\{s_1^i, s_2^i, \ldots, s_n^i\}$ for $i = 1, 2, \ldots, k^n$ as its internal state, after finitely many steps, which is equal to or less than k^n, the internal states must repeat. Therefore, its outputs must be ultimately periodic with the ultimate period being equal to or less than k^n. If M is a linear device, then the ultimate period of the outputs is a divisor of $(k^n - 1)$. So we have:

Theorem 3.1 The outputs of any finite state machine are ultimately periodic.

Proof: By the above discussion. □

From Theorem 3.1, we know that any key stream coming from the practical key generators in our real world must be ultimately periodic. Say a key stream has the period T. Then the key stream could be generated by the *Linear Feedback Shift Register* $s_{i+T} = s_i$ for $i = 1, 2, \ldots$. Therefore, all the key streams could be implemented by LFSRs. Also the "ultimately periodic" property of the finite state machine outputs forces that we cannot generate an infinite random looking sequence, hence the best thing we can expect for the key stream is that the first period of the sequence looks random. So we are confined to the sequences with only finitely many terms.

1. Non-Regularity

From our intuition, a random sequence must highly lack any regularity, so each state of the sequence is difficult to predict, or equivalently, unpredictable. Difficulty of prediction means that the probability of successful guessing is very small. Since most of our key generators are embedded in $\mathbb{F}_2$, we could say that for each bit of the key stream, the chances of correct prediction should be equal to or less than $\frac{1}{2}$. Then as the finite sequences are concerned, the longer they are, the more difficult to predict. But one should notice that, just the small probability of the sequence being chosen from a very huge candidate space is necessary but not sufficient to mean the unpredictability at all. For example, let us look at the following four sequences over $\mathbb{F}_2$ whose lengths are all 40:

$$\begin{aligned}
\mathbf{s}_1 &= (0000000000000000000000000000000000000000)\\
\mathbf{s}_2 &= (0110011001100110011001100110011001100110)\\
\mathbf{s}_3 &= (1001000000000011000000000101001100000001)\\
\mathbf{s}_4 &= (0101110001101001101010110101101110010110)
\end{aligned}$$

No one would dispute that the four sequences are with the equal probability 2^{-40} if we choose randomly from the space $\mathbb{F}_2^{40}$ and the probability is really small enough to be viewed as 0. However, the four sequences are quite different. For $\mathbf{s}_1$, it is all zeros. So we could describe it by *all bits are 0*. If someone sees the first ten or fifteen bits of $\mathbf{s}_1$, this person could guess easily that the following bits are all 0. As for $\mathbf{s}_2$, we could get it by *repeating 0110 for 10 times*. Some persons could easily and correctly guess all the remaining bits after observing three or more 0110. Both of $\mathbf{s}_1$ and $\mathbf{s}_2$ have so regular patterns that they are easily predicted. But for $\mathbf{s}_3$ and $\mathbf{s}_4$, there are no no obvious regular patterns.

Let us explore more. No regular patterns implies the sequence needs much data to be described exactly. If there is some regular pattern in the sequence, we could compress the data to represent it. Still from our intuition, the more regular the sequence is, the less data is needed to describe it. Refer to the above example again:

Sequence	Representation	Data Volume /*characters*
$\mathbf{s}_1$	40 0s.	6
$\mathbf{s}_2$	10 0110s.	9
$\mathbf{s}_3$	1001, 10 0s, 11, 9 0s,1010011, 7 0s, 1.	39
$\mathbf{s}_4$	0101110001101001101010110101101110010110.	41

One can see that the highly regular sequences $\mathbf{s}_1$ and $\mathbf{s}_2$ could be represented just by a few characters with the data volume much less than their original lengths. So they are not good candidates for our "random sequences". Conversely, $\mathbf{s}_3$ and $\mathbf{s}_4$ are almost without regularity, which means there is some difficulty to predict each bit, and they are represented with nearly the same data volumes as their lengths. So we conclude that random sequences should be represented by almost the same amount of data as expressing them directly. This empirical analysis inspires us to measure the randomness of the sequence by the data volume or the size of its representation. Obviously, the description method with the minimal data volume interests us most:

Use abstract programs to denote the different representation ways for a sequence. Say all the programs producing the sequence $\mathbf{s}$ *are* $\{P_i\}$ *for* $i = 1, 2, \ldots$*, then, the randomness of* $\mathbf{s}$ *could be measured by the size of* P_s*, the smallest one in* $\{P_i\}$*.*

Actually this approach leads to the formal concept of **Kolmogorov complexity**. In 1964 and 1965, R. Solomonov in [41] and A. Kolmogorov in [15] had used the "patternlessness" of a finite sequence, which is the length of the shortest Turing machine program generating it, to measure the randomness of this finite sequence. By the above method, we could say the bigger the size of P_s is, the more random $\mathbf{s}$ should be. But one thing which should be noticed is that for sequences with small lengths it is difficult to measure their randomness. Why? Because their lengths are already small, the data volume of different representations may not be more efficient to express them than by showing each bit directly. Look at the extreme case where the sequence only has one bit. So is $\{0\}$ random or $\{1\}$ random? Maybe both. Maybe none.

2. Uniform Distribution

Another intuition of the random sequences is related to their distribution properties. Since each bit of the sequence is independent and uniformly distributed, we could expect the occurring frequency of each k-tuple for $k = 1, 2, \ldots$ is almost equal and the bigger k is, the smaller the frequency of each k-tuple. And the theoretical value for the frequency of a k-tuple $(b_1, b_2, \ldots, b_k)$ in a binary sequence with length n should be:

$$P_k = \frac{\#\{\mathbf{v}_i = (s_i, s_{i+1}, \ldots, s_{i+k-1}) \mid \mathbf{v}_i = (b_1, b_2, \ldots, b_k)\}}{n - k + 1} \approx (\frac{1}{2})^k.$$

Let $k = 1$ for instance, then the number of 1's and the number of 0's should be almost the same in a random sequence. Now look again at the four sequences $\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \mathbf{s}_4$ above. Obviously $\mathbf{s}_1$ is not uniformly distributed since there is no 1 in it. $\mathbf{s}_3$ is lacking regularity, but from the distribution viewpoint, the frequency $\frac{9}{40}$ of 1's is much smaller than $\frac{1}{2}$. Therefore we do not consider $\mathbf{s}_3$ to be a good candidate for random sequences. If a hacker had observed, say, the first 20 bits of $\mathbf{s}_3$, then she/he calculates the frequency of 1 and 0. After finding the frequency of 0 is nearly 3 times the frequency of 1, then she/he could guess most of the next 20 bits are 0's, which is true. After decoding the plaintext obtained by the hacker using guessing but with high correct probability for each bit, she/he may successfully extract the information by using the language redundancy. So $\mathbf{s}_3$ being a key stream is not secure. As for $\mathbf{s}_4$, it looks uniformly distributed, in addition to the independence of each of its bits (which means it is without any regularity). Therefore, we expect that $\mathbf{s}_4$ could be a good key stream. In fact, $\mathbf{s}_4$ is produced by the author tossing a fair Singapore one dollar coin.

However, a sequence with uniform distribution does not necessarily lack regularity. Recall the maximal period sequences. If their linear complexities are L, then their periods are $2^L - 1$. And in [18, Chapter 6], the authors have shown that the maximal period sequences pass the probabilistic tests in one of their minimal periods. These so-called pseudo-noise sequences with $2^L - 1$ bits are highly predictable if L consecutive terms are observed, although they have good distribution properties.

After the above discussion from our intuition, we could postulate some requirements on a random looking finite sequence to satisfy our aims for constructing a good key stream, although we even do not know whether there exist any finite truly random sequences.

Call these sequences *Pseudo-Random Sequences.* Requirements for a good pseudo-random sequence $\mathbf{S}$ should include:

- $\mathbf{S}$ should have no regularity. This also means that the data volume of any representation for $\mathbf{S}$ must be incompressible compared to the original length of $\mathbf{S}$.
- $\mathbf{S}$ should have uniform distribution. Any k-tuples should have equally occurring probability for $k = 1, 2, \ldots$ and the probability decreases while k increases.
- The minimal period of $\mathbf{S}$ should be long enough to display some uncertainty since randomness is meaningless for short sequences.

3. Measuring Randomness by Linear Complexity Profile

In fact, a good and practical representation method of a finite sequence, which is used to measure the randomness, was developed in the 1970s. As mentioned before, in 1964 and 1965, R. Solomonov in [41] and A. Kolmogorov in [15] used the length of the shortest Turing machine program generating it to measure the randomness of a finite sequence. P. Martin-Löf further developed this approach in 1966 in [19]. Finally in [17], A. Lempel and J. Ziv justified using the shortest length of LFSR models to measure the unpredictability of the sequences in 1976. Referring to the discussion of "non-regularity", each P_i is replaced by an LFSR with an initial state, so the linear complexity of $\mathbf{s}$ is a measurement of its randomness. This is very appealing postulation because each key stream generated by some finite state machine can be produced by an LFSR and there is also a very efficient Berlekamp-Massey algorithm to compute its linear complexity profile. One could refer to the papers of the above authors and [6, Chapters 3, 5, 6] for rigorous mathematical proofs from logic and algorithm perspectives.

Recall the definition of the linear complexity profile in Chapter 2. Now let us look at the following sequence obtained from $\mathbf{s}_4$ above by the author tossing a fair Singapore coin with length 31 and expand it by repeating the first 31 bits:

$$\tilde{\mathbf{s}} = (0101110001101001101010110101101)^{\infty}.$$

Then use the Berlekamp-Massey LFSR Synthesis Algorithm (the program by Maple 9.01 is in the Appendix) to find the linear complexity profile of $\tilde{\mathbf{s}}$ for its first 62 bits:

$$\begin{aligned}\mathbf{L}(\tilde{\mathbf{s}}^{62}) = \ & [0,2,2,2,3,3,3,3,6,6,6,6,7,7,8,8,8,8,11,11,11,11,12,12,13,13,\\ & 14,14,15,15,16,16,17,17,17,19,19,19,19,21,21,21,21,23,23,23,\\ & 23,23,23,27,27,27,27,27,27,29,29,29,29,31,31,31]\end{aligned}$$

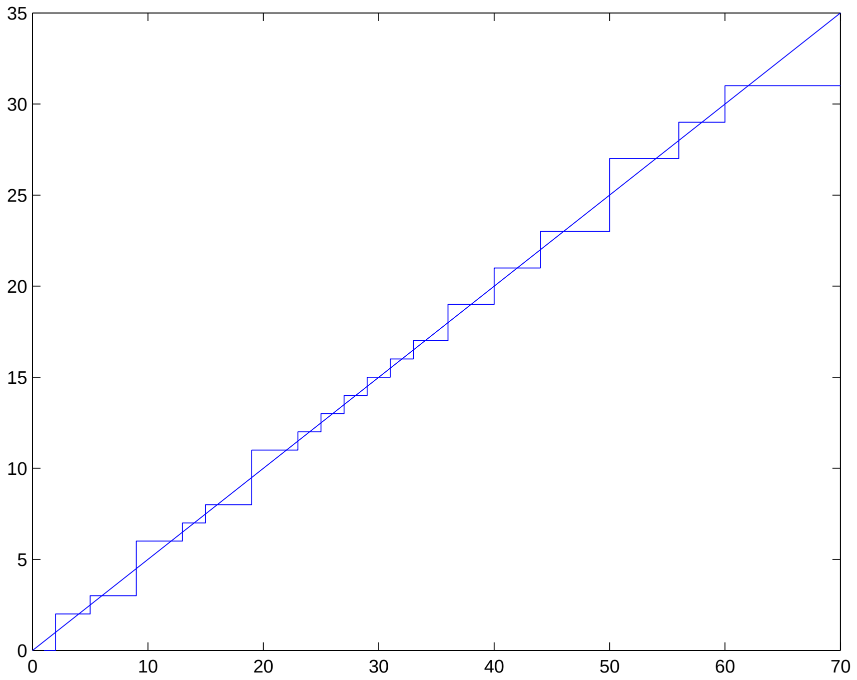

Figure 3.1: The Linear Complexity Profile of $\tilde{\mathbf{s}}^{70}$.

Obviously, the minimal period of $\tilde{\mathbf{s}}$ is 31. Plot the first two periods and 8 more bits as well as their corresponding linear complexities in Figure 3.1.

One can see that the graph of the linear complexity profile of $\tilde{\mathbf{s}}$ is quite close to the line $L(\tilde{\mathbf{s}}^n) = \dfrac{n}{2}$ and $L(\tilde{\mathbf{s}}^n)$ stays at 31, its minimal period, after $n = 61$. Some properties showing in this figure are not by chance. In fact, the linear complexity profile of unpredictable sequences over $\mathbb{F}_2$ should be close to the $\dfrac{n}{2}$-line, at least for the first period. On one hand, since the pseudo-random sequence are incompressible, the minimal data volume needed to represent this sequence should be almost the same as its length. So does using LFSR. On the other hand, for an LFSR with length L, to decide all its outputs, we need L coefficients to determine the LFSR characteristic polynomial (notice that it is a monic polynomial), and L values for the initial state. So $2L$ values are needed to determine the outputs. Consider both, then $2L \approx |\tilde{\mathbf{s}}| = n$. So the graph should display the property that $L(\tilde{\mathbf{s}}^n) \approx \dfrac{n}{2}$ while $n \leq T$. While $T < n \leq 2T$, this property should still keep to some extent, since after repeating several bits with length less than the minimal period, it is hard to estimate the minimal period just by observing the first n bits where $n < 2T - n_0$ (here n_0 is a small positive integer dependent on T). Also because the period of $\tilde{\mathbf{s}}$ is $T = 31$, it could be generated by $s_{i+T} = s_i$ for all i. Then, $L(\tilde{\mathbf{s}}) \leq T = 31$. So 31 is the biggest possible ultimate value for $L(\tilde{\mathbf{s}}^n)$. Therefore, generally, the linear complexity profile of random sequences should increase approximately as the $\frac{n}{2}$-line, which means its linear complexity profile should be close to $\{\frac{n}{2}\}$ for $n = 1, 2, \ldots 2T$, and achieve the minimal period T after $2T$ bits. A remark should be stated here. High linear complexity itself does not mean unpredictable. For example, look at the sequence

$$\tilde{\mathbf{s}}_c = (0000000000000000000000000000001).$$

Although the linear complexity of $\tilde{\mathbf{s}}_c$ being 31 is the same as $\tilde{\mathbf{s}}$ pictured in Figure 3.1, its linear complexity increases sharply at the $31st$ bit from 0 to 31, which does not grow close to the $\frac{n}{2}$-line at all. And this sequence does not have uniform distribution either. So many zeros and their regular distributions make this sequence highly predictable. Actually, in the next chapter, one will see that the 1-error linear complexity of $\tilde{\mathbf{s}}_c$ is zero, which renders $\tilde{\mathbf{s}}_c$ very insecure as a key stream.

3.2 Probabilistic Properties of Random Sequences

Now we will explore the probabilistic properties of sequences over some finite field in order to find the expected linear complexity, the variation, and the expected increasing ratio of the linear complexity. In this way, we could estimate the number of good candidate sequences for key streams and find average behaviors of sequences randomly chosen from all the candidates. At last, periodic sequences will be considered since most key streams in practical use are periodic.

1. The Number of Sequences over $\mathbb{F}_q$ with Linear Complexity c
Firstly, we must determine the number $N_n(c)$ of all sequences over $\mathbb{F}_q$ whose linear complexities are c and lengths are n since this enumeration is fundamental for all discussions of the probabilistic properties. To determine $N_n(c)$, there are at least four methods known up to now. The first one is based on a recursion method. One could refer to [32, Chapter 4] for a special case and expand it to the general case without difficulties. Another method relies on the continued fraction expansion of the *generating function*, a Laurent series. A detailed discussion could be found in [28, Chapter 7]. The third one is offered by W. Meidl recently based on the relationship between linear complexity profiles and lattice profiles in [21]. Now by introducing two concepts, *jump point* and *balance point*, we use a fourth approach based on the Berlekamp-Massey algorithm to compute $N_n(c)$.

Definition 3.1: Let $\tilde{\mathbf{s}} = (s_1, s_2, \ldots, s_i, \ldots)$ be a sequence over a finite field $\mathbb{F}_q$, then $k \geq 1$ is called a **jump point** if $L(\tilde{\mathbf{s}}^{k-1}) < L(\tilde{\mathbf{s}}^k)$.

Definition 3.2: Call the number $2k \geq 2$ a **balance point** if $L(\tilde{\mathbf{s}}^{2k}) = k$.

Lemma 3.1: The number of jump points is equal to the number of balance points in a sequence $\tilde{\mathbf{s}}^n = (s_1, s_2, \ldots, s_n)$ over $\mathbb{F}_q$, where $n \geq 2L(\tilde{\mathbf{s}}^n)$.

Proof: Note that sequences in our discussion begin from s_1, so s_0 is not defined. But if we define $s_0 = 0$ and $L(\tilde{\mathbf{s}}^0)$=0, then the lemma must be true because between two balance points, there must be a jump point (refer to Figure 3.2) and any jump point at i must satisfy $L(\tilde{\mathbf{s}}^i) > \frac{i}{2}$ according to Berlekamp-Massey algorithm. □

Lemma 3.2: Let $\tilde{\mathbf{s}} = (s_1, s_2, \ldots, s_i, \ldots)$ be a sequence over a finite field $\mathbb{F}_q$. Suppose

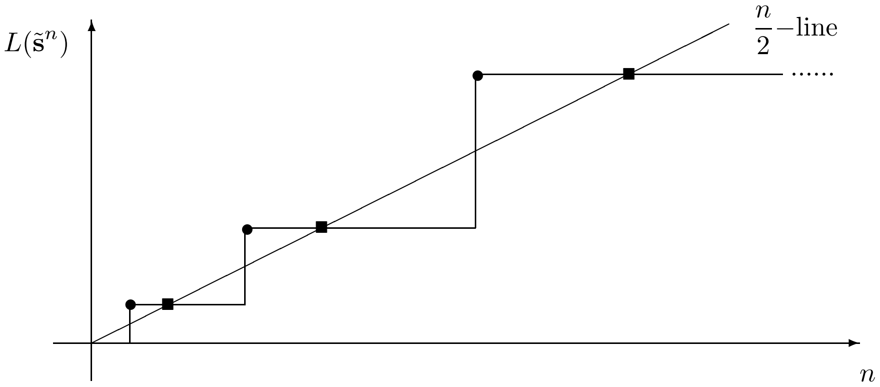

Figure 3.2: Jump points (●) and balance points (■).

the balance points are $\{i_1, i_2, \ldots, i_t\}$, all being even numbers by definition, then the jump points less than i_t are at:

$$\{\frac{i_1}{2}, \ \frac{i_1+i_2}{2}, \ \ldots, \ \frac{i_{t-1}+i_t}{2}\}.$$

Proof: There is one and only one jump point s_x between $s_{i_j}, s_{i_{j+1}}$. Also, according to the Berlekamp-Massey algorithm, $L(\tilde{\mathbf{s}}^x) = x - L(\tilde{\mathbf{s}}^{x-1})$. So we have $\frac{i_{j+1}}{2} = x - \frac{i_j}{2}$, then $x = \frac{i_j + i_{j+1}}{2}$. □

Remark: Lemma 3.2 means that if two adjacent balance points are given, then the jump point between them is their middle term.

Now based on the above lemma, we conclude that a linear complexity profile is uniquely determined by the set of balance points. So we are ready to determine the number of different linear complexity profiles, with the last linear complexity term being c.

Theorem 3.2: Suppose all the sequences discussed have n terms. Then the number of different possible linear complexity profiles for the linear complexity c is 2^{c-1} if $0 < c \leq \frac{n}{2}$, and is 2^{n-c} if $c > \frac{n}{2}$.

Proof: Firstly, suppose $0 < c \leq \frac{n}{2}$, then according to Lemma 3.2, the linear complexity profile corresponds to the set of balance points. So we just need to choose some even numbers from $\{2, 4, 6, \ldots, 2(c-1)\}$ to be the balance points in order to construct some linear

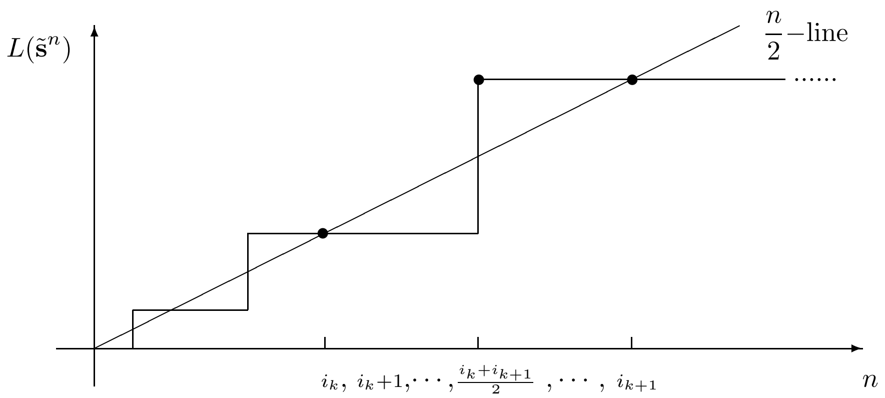
$L(\tilde{\mathbf{s}}^n)$
$\frac{n}{2}$−line
......
i_k, i_k+1,···,$\frac{i_k+i_{k+1}}{2}$,··· , i_{k+1}
n

Figure 3.3: Obtaining the jump point from its two adjacent balance points I.

complexity profile. Since this set has $(c-1)$ elements, the total number N of choices is the number of subsets of this set, which is $N = 2^{c-1}$. Secondly if $c > \frac{n}{2}$, the last balance point s_x must satisfy $\frac{x+2c}{2} \leq n$ since the last jump point is in front of s_n. So the set to choose the balance points' subscripts changes to $\{2, 4, \ldots, 2(n-c)\}$. Now the total number N of choices is the number of subsets of this set, which is $N = 2^{n-c}$. □

After the enumeration of different linear complexity profiles for the linear complexity c, we proceed by exploring the number of candidate sequences for each linear complexity profile. Firstly, let c be a linear complexity, satisfying $c \leq \frac{n}{2}$, and k be the number of balance points, satisfying $k \leq c-1$. Start from s_1 to s_n. If $(i, L(\tilde{\mathbf{s}}^i))$ is under the $\frac{n}{2}$-line, then the value of s_i is decided by the previous elements $\{s_1, s_2, \ldots, s_{i-1}\}$. If i is a jump point, then it can be any element of $\mathbb{F}_q$ except a special value (which is decided by the previous terms). If $(i, L(\tilde{\mathbf{s}}^i))$ is above or on the $\frac{n}{2}$-line, then the choice for it is totally free. So let $B = \{i \mid i \text{ is a jump point}\}$, $D = \{i \mid L(\tilde{\mathbf{s}}^i) \geq \frac{n}{2}\}$. Then for this kind of linear complexity profile, there are $(q-1)^{|B|}q^{|D|-|B|}$ candidate sequences over $\mathbb{F}_q$.

From Lemma 3.1, we have $|B| = k$. Now let us determine $|D| - |B|$. Suppose i_k and i_{k+1} are two adjacent balance points and consider $i_k+1, i_k+2, \ldots, i_{k+1}$. Refer to Figure 3.3 above. Clearly, $L(\tilde{\mathbf{s}}^j) < \frac{j}{2}$ when $j = i_k+1, i_k+2, \ldots, \frac{i_k+i_{k+1}}{2} - 1$. Therefore the number of graph points $(m, L(\tilde{\mathbf{s}}^m))$ above and on the $\frac{n}{2}$-line is $\frac{i_{k+1}-i_k}{2} + 1$ in the interval $(i_k, i_{k+1}]$.

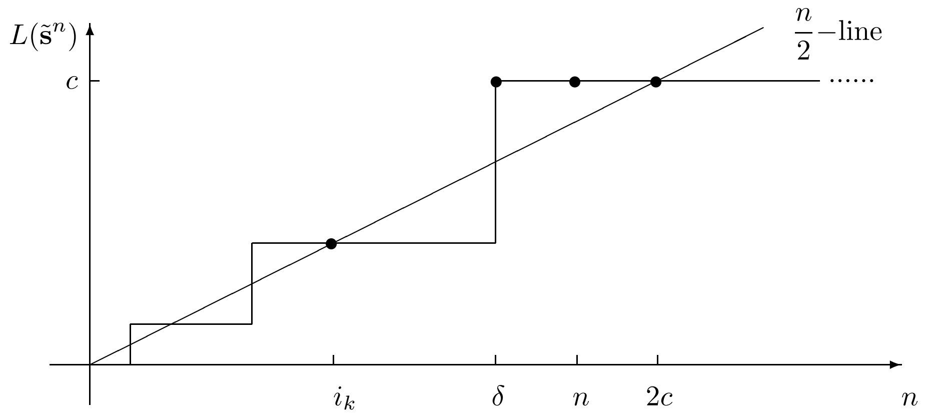

Figure 3.4: Obtaining the jump point from its two adjacent balance points II.

This means that the number of points belonging to D is $\frac{i_{k+1}-i_k}{2}+1$. If we combine all these intervals between balance points, we can get $|D| = \frac{2c}{2}+k$. Then $|D|-|B| = c$, which implies $(q-1)^{|B|}q^{|D|-|B|} = (q-1)^k q^c$.

Consider the other situation of $c > \frac{n}{2}$, and suppose i_k and δ are the last balance point and jump point, respectively. See Figure 3.4.

In $\{1, 2, \ldots, i_k\}$, there are $\frac{i_k}{2}+k-1$ elements lying in D by the above argument since $i_k = 2 \times \frac{i_k}{2}$. Now in the last interval $(i_k, n]$, there are $n-\delta+1$ elements on or above the $\frac{n}{2}$-line. However, we have $\delta = \frac{2c+i_k}{2}$. So the total number in D is $n-c$. Therefore, in this situation the number of candidate sequences for the linear complexity profile is given by $(q-1)^{|B|}q^{|D|-|B|} = (q-1)^k q^{n-c}$. We summarize the two results:

Theorem 3.3: Suppose a linear complexity profile is $\{A_1, A_2, \ldots, A_n = c\}$ and there are k different values in $\{A_1, A_2, \ldots, A_n\}$, where $0 \le A_1 \le A_2 \le \cdots \le A_n = c$ and A_i is an integer for all $1 \le i \le n$. Then the number of candidate sequences over $\mathbb{F}_q$ for this linear complexity profile, whose lengths are n, is:

$$N = \begin{cases} (q-1)^k q^c & \text{if } \ 0 < c \le \frac{n}{2}, \\ (q-1)^k q^{n-c} & \text{if } \ c > \frac{n}{2}. \end{cases}$$

Proof: By the above discussion. □

Based on Theorem 3.3, we can determine the exact number $N_n(c)$ of sequences over $\mathbb{F}_q$, with length n, whose linear complexities are c. The method is that: if $0 < c \leq \dfrac{n}{2}$, choose i, where $0 \leq i \leq c$, balance points from $\{2, 4, 6, \ldots, 2(c-1)\}$. If $c \geq \dfrac{n}{2}$, choose i, s.t. $0 \leq i \leq (n-c)$, balance points from $\{2, 4, 6, \ldots, 2(n-c)\}$ except 0. So for both situations, there are $i+1$ jump points. Then compute the number of candidate sequences having this special linear complexity profile. In the end, we can get the number $N_n(c)$ just by summing all the numbers of the sequences for a special linear complexity profile.

Theorem 3.4: The number $N_n(c)$ of sequences $\tilde{\mathbf{s}} = (s_1, s_2, \ldots, s_i, \ldots)$ over $\mathbb{F}_q$, of length n, whose linear complexities are exactly c, is:

$$N_n(c) = \begin{cases} (q-1)q^{2c-1} & \text{if } \ 0 < c \leq \frac{n}{2}, \\ (q-1)q^{2n-2c} & \text{if } \ c > \frac{n}{2}. \end{cases}$$

Proof: Sum up all the candidate sequences for each linear complexity profile with the final value, i.e. the linear complexity of each sequence, being c:
If $0 < c \leq \frac{n}{2}$,

$$N_n(c) = \sum_{i=0}^{c-1} (q-1)^{i+1} q^c \binom{c-1}{i} = (q-1)q^c(1+q-1)^{c-1} = (q-1)q^{2c-1}.$$

If $c \geq \frac{n}{2}$,

$$N_n(c) = \sum_{i=0}^{n-c} (q-1)^{i+1} q^{n-c} \binom{n-c}{i} = (q-1)q^{n-c}(1+q-1)^{n-c} = (q-1)q^{2n-2c}.$$

□

This is a method different from the other three we described at the beginning of this section. We can obtain more information on the structure of the sequences with a given linear complexity. For example, recall the phenomenon in the last section that the linear complexity profiles of random sequences are close to the $\dfrac{n}{2}$-line. Actually, this implies that in a random sequence, there are many balance points because the more the balance points, the closer the linear complexity graph is to the $\dfrac{n}{2}$-line. By Theorem 3.3, a large value for the number of balance points means a large number of the candidate sequences when

$q \neq 2$. Therefore, we could expect that a sequence chosen randomly will exhibit a close relationship between its linear complexity profile and the $\frac{n}{2}$-line.

2. The Expected Linear Complexity

Next we will use the result of Theorem 3.4 to obtain the expected linear complexity $E[L(\widetilde{\mathbf{S}}^n)]$ of the finite sequences of variables $\widetilde{\mathbf{S}}^n = (S_1, S_2, \ldots, S_n)$ with length n over $\mathbb{F}_q$. We view all $\{S_i\}$ for $i = 1, 2, \ldots, n$ to be n independent random variables. Let $P(X = x_i)$ be the probability of $X = x_i$. By definition,

$$E[L(\widetilde{\mathbf{S}}^n)] = \sum_{i=1}^{q^n} L(\tilde{\mathbf{s}}_i^n) P(\widetilde{\mathbf{S}}^n = \tilde{\mathbf{s}}_i^n), \qquad \text{where } \tilde{\mathbf{s}}_i^n \in \mathbb{F}_q^n. \tag{3.1}$$

Since the S_i are independent and uniformly distributed random variables, then each sequence $\tilde{\mathbf{s}}_i$ must be chosen with equal probability. So we have $P(\widetilde{\mathbf{S}}^n = \tilde{\mathbf{s}}_i^n) = \frac{1}{q^n}$. Then, (3.1) can be simplified to

$$E[L(\widetilde{\mathbf{S}}^n)] = \frac{1}{q^n} \sum_{i=1}^{q^n} L(\tilde{\mathbf{s}}_i^n).$$

Now divide the q^n sequences into $n+1$ groups (one group is with the linear complexity being 0). For each group, all the sequences lying in it have the same linear complexity. So we have:

$$E[L(\widetilde{\mathbf{S}}^n)] = \frac{1}{q^n} \sum_{c=1}^{n} c \times N_n(c) = \frac{1}{q^n} (\sum_{c=1}^{\lfloor \frac{n}{2} \rfloor} c(q-1)q^{2c-1} + \sum_{c=\lceil \frac{n+1}{2} \rceil}^{n} c(q-1)q^{2n-2c}). \tag{3.2}$$

Compute each sum term in the above formula (3.2) by introducing a formal variable: we have $\sum_{i=1}^{k} iq^{2i-1} = \frac{(q^2k - k - 1)q^{2k+1} + q}{(1-q^2)^2}$, so we get that:

$$\sum_{c=1}^{\lfloor \frac{n}{2} \rfloor} c(q-1)q^{2c-1} = \begin{cases} (q-1)\dfrac{(q^2-1)nq^{n+1} - 2q^{n+1} + 2q}{2(1-q^2)^2} & \text{if } n \text{ is even,} \\ \\ (q-1)\dfrac{(q^2-1)nq^n - (q^2+1)q^n + 2q}{2(1-q^2)^2} & \text{if } n \text{ is odd.} \end{cases} \tag{3.3}$$

And we obtain $\sum\limits_{i=1}^{k} iq^{2n-2i} = q^{2n-2k}(\frac{q^{2k+2}-(k+1)q^2+k}{(1-q^2)^2})$, so

$$\sum_{c=\lceil\frac{n+1}{2}\rceil}^{n} c(q-1)q^{2n-2c} = \begin{cases} (q-1)\dfrac{(q^2-1)nq^n+2q^{n+2}-2(n+1)q^2+2n}{2(1-q^2)^2} & \text{for even } n, \\ \\ (q-1)\dfrac{(n+1)q^{n+3}-(n-1)q^{n+1}-2(n+1)q^2+2n}{2(1-q^2)^2} & \text{for odd } n. \end{cases} \tag{3.4}$$

Now we are ready to conclude the exact value of $E[L(\widetilde{\mathbf{S}}^n)]$:

Theorem 3.5: The expected linear complexity $E[L(\widetilde{\mathbf{S}}^n)]$ of the variable sequence $\widetilde{\mathbf{S}}^n = (S_1, S_2, \dots, S_n)$ over $\mathbb{F}_q$, where all the random variables $\{S_1, S_2, \dots, S_n\}$ are independent and uniformly distributed, is:

$$E[L(\widetilde{\mathbf{S}}^n)] = \begin{cases} \dfrac{n}{2} + \dfrac{q}{(q+1)^2} - \dfrac{1}{q^n}\Big(\dfrac{n}{1+q} + \dfrac{q}{(1+q)^2}\Big) & \text{if } n \text{ is even}, \\ \\ \dfrac{n}{2} + \dfrac{(q^2+1)}{2(1+q)^2} - \dfrac{1}{q^n}\Big(\dfrac{n}{1+q} + \dfrac{q}{(1+q)^2}\Big) & \text{if } n \text{ is odd}. \end{cases} \tag{3.5}$$

Proof: Sum the results of (3.3) and (3.4). □

Recall that an LFSR with linear complexity L needs $2L$ data to specify a sequence and the random sequence needs almost the same amount of data as its length to represent it. So to obtain an ideal candidate sequences space that by randomly choosing would return a sequence with the expected linear complexity being half of its length, we at least need to guarantee $\lim\limits_{n\to\infty} \frac{E(L(\widetilde{\mathbf{S}}^n))}{n} = \frac{1}{2}$. Since $\frac{q}{(q+1)^2}$ is decreasing when $q > 1$, $\frac{q}{(q+1)^2} \leq \frac{2}{9}$ and $\lim\limits_{q\to\infty} \frac{q}{(q+1)^2} = 0$. Also $\frac{(q^2+1)}{2(1+q)^2} \leq \frac{1}{2}$ although it is increasing when $q > 1$. Therefore, Theorem 3.5 tells us that the expected linear complexity over any finite field will have the value quite close to $\frac{n}{2}$ just with a slight difference less than $\frac{1}{2}$. As a result, our status quo has naturally achieved the ideal situation that a randomly chosen sequence from $\mathbb{F}_q^n$ will exhibit randomness with respect to having a linear complexity close to $\frac{n}{2}$.

3. The Variance of the Linear Complexity

Now, let us look at the variance of the linear complexity of all the sequences with length

n over $\mathbb{F}_q$ since it is the second important parameter characterizing randomly chosen sequences. By definition, the variance $V[L(\widetilde{\mathbf{S}}^n)]$ is given by:

$$V[L(\widetilde{\mathbf{S}}^n)] = E([L(\widetilde{\mathbf{S}}^n) - E[L(\widetilde{\mathbf{S}}^n)]]^2) = E([L(\widetilde{\mathbf{S}}^n)]^2) - (E[L(\widetilde{\mathbf{S}}^n)])^2. \tag{3.6}$$

Because we already have $E[L(\widetilde{\mathbf{S}}^n)]$, the rest of our task is to compute

$$\begin{aligned} E([L(\widetilde{\mathbf{S}}^n)]^2) = & \sum_{i=1}^{q^n}[L(\tilde{\mathbf{s}}_i^n)]^2 P(\widetilde{\mathbf{S}}^n = \tilde{\mathbf{s}}_i^n) = \frac{1}{q^n}\sum_{i=1}^{q^n}[L(\tilde{\mathbf{s}}_i^n)]^2 \\ = & \frac{1}{q^n}\sum_{c=1}^{n} c^2 \times N_n(c) = \frac{1}{q^n}[\sum_{c=1}^{\lfloor \frac{n}{2} \rfloor} c^2(q-1)q^{2c-1} + \sum_{c=\lceil \frac{n+1}{2} \rceil}^{n} c^2(q-1)q^{2n-2c}]. \end{aligned}$$

Following the same procedures by introducing a formal variable but with much more complicated computation, we have

$$\begin{aligned} \sum_{i=1}^{k} i^2 q^{2i-1} = & \frac{q^{2k+1}(q^2-1)^2k^2 - 2q^{2k+1}(q^2-1)k + (q^{2k}-1)(q^3+q)}{(q^2-1)^3}, \\ \sum_{i=1}^{k} i^2 q^{2n-2i} = & \frac{(q^{2k+2}+q^{2k}-q^2-1)q^{2n-2k+2} - q^{2n-2k}(q^2-1)^2k^2 - 2q^{2n-2k+2}(q^2-1)k}{(q^2-1)^3}. \end{aligned}$$

Now we are ready to obtain the main result for this part:

Theorem 3.6: The variance $V[L(\widetilde{\mathbf{S}}^n)]$ of the linear complexity of the variable sequence $\widetilde{\mathbf{S}}^n = (S_1, S_2, \ldots, S_n)$ over $\mathbb{F}_q$, where all the random variables $\{S_1, S_2, \ldots, S_n\}$ are independent and uniformly distributed, is:

$$V[L(\widetilde{\mathbf{S}}^n)] = \frac{q^5+q^4+4q^3+q^2+q}{(q-1)^2(q+1)^4} + \frac{1}{q^n}O(n) + \frac{1}{q^{2n}}O(n^2). \tag{3.7}$$

Proof: By the above two summation formulas and many computations with quite complicated simplifying steps, we have that when n is even,

$$E([L(\widetilde{\mathbf{S}}^n)]^2) = [\frac{1}{4} - \frac{1}{(q+1)q^n}]n^2 + [\frac{q}{(q+1)^2} - \frac{2q^2}{(q-1)(q+1)^2q^n}]n + [\frac{q(q^2+1)}{(q^2-1)^2} - \frac{q(q^2+1)}{(q^2-1)^2q^n}],$$

and while n is odd,

$$E([L(\widetilde{\mathbf{S}}^n)]^2) = [\frac{1}{4} - \frac{1}{(q+1)q^n}]n^2 + [\frac{q^2+1}{2(q+1)^2} - \frac{2q^2}{(q-1)(q+1)^2q^n}]n + [\frac{q^4+6q^2+1}{4(q^2-1)^2} - \frac{q(q^2+1)}{(q^2-1)^2q^n}].$$

By (3.3) and (3.4), we obtain:

$$V(L(\widetilde{\mathbf{S}}^n)) = \begin{cases} \dfrac{q^5+q^4+4q^3+q^2+q}{(q-1)^2(q+1)^4} - \dfrac{1}{q^n}H_1(q,n) - \dfrac{1}{q^{2n}}H_2(q,n) & \text{if } n \text{ is even },\\ \\ \dfrac{q^5+q^4+4q^3+q^2+q}{(q-1)^2(q+1)^4} - \dfrac{1}{q^n}H_3(q,n) - \dfrac{1}{q^{2n}}H_4(q,n) & \text{if } n \text{ is odd.} \end{cases} \tag{3.8}$$

And the exact value of each $H_i(q,n)$ for $i = 1,2,3,4$ is:

$$H_1(q,n) = \frac{q(q^2+3)n}{(q-1)(q+1)^3} + \frac{q(q^4+6q^2+1)}{(q-1)^2(q+1)^4},$$

$$H_2(q,n) = H_4(q,n) = \frac{n^2}{(q+1)^2} + \frac{2qn}{(q+1)^3} + \frac{q^2}{(q+1)^4},$$

$$H_3(q,n) = \frac{(3q^2+1)n}{(q-1)(q+1)^4} + \frac{4q^2(q^2+1)}{(q-1)^2(q+1)^4}.$$

Since $\lim\limits_{n\to\infty} \dfrac{n^2}{q^n} = 0$ when $q > 1$, we have $\lim\limits_{n\to\infty} \dfrac{H_i(q,n)}{q^n} = 0$ for any fixed $q \geq 2$. Therefore, the assertion of (3.7) is established. □

On one hand, $\dfrac{n}{q^n}$ goes to zero very fast when n is increasing given $q \geq 2$. Say $n = 10$, then the difference between $V(L(\widetilde{\mathbf{S}}^n))$ and $\dfrac{q^5+q^4+4q^3+q^2+q}{(q-1)^2(q+1)^4}$ is less than 0.01 given that the coefficients of the highest degree n terms in $H_i(q,n)$ are smaller than 1. On the other hand, $v(q) = \dfrac{q^5+q^4+4q^3+q^2+q}{(q-1)^2(q+1)^4}$ is strictly decreasing while $q \geq 2$ and $v(2) \approx 1, v(3) \leq 0.5$. Therefore, $V(L(\widetilde{\mathbf{S}}^n))$ is really small with the value less than $\dfrac{86}{81}$. On the whole, we conclude that the linear complexity of $\tilde{\mathbf{s}}^n$ chosen randomly from $\mathbb{F}_q^n$ should be very close to the expected value which is almost $\dfrac{n}{2}$ in most cases. To get a numerical concept of this conclusion, we employ *Chebyshev's inequality*. Therefore, we have

$$P\{|L(\widetilde{\mathbf{S}}^n) - E[L(\widetilde{\mathbf{S}}^n)]| \geq r\} \leq \frac{V(L(\widetilde{\mathbf{S}}^n))}{r^2}. \tag{3.9}$$

If we let $r = 3$, (3.9) implies that about 90% of the sequences over any finite field with length n will have their linear complexities in the range $\dfrac{n}{2} \pm 3$.

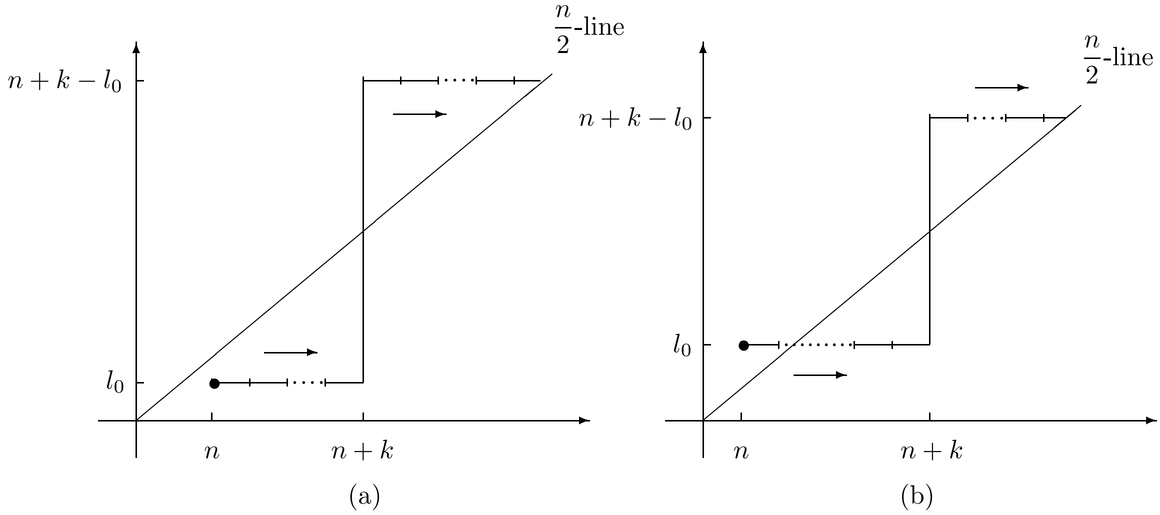

Figure 3.5: One step of random walk: (a) $L(\tilde{\mathbf{s}}^n) = l_0 \leq \frac{n}{2}$, (b) $L(\tilde{\mathbf{s}}^n) = l_0 > \frac{n}{2}$

4. The Random Walk of Linear Complexity

Suppose $\tilde{\mathbf{s}} = (s_1, s_2, \ldots, s_i, \ldots)$ and $L(\tilde{\mathbf{s}}^n) = l_0$. By the Berlekamp-Massey algorithm, the linear complexity of $\tilde{\mathbf{s}}^{n+k}$ will change to $n+k-l_0$ from l_0 for some k if s_{n+k} does not satisfy the linear recurrence relation decided by the previous terms. Therefore, it is quite natural to consider the expected value for k, which is the average length of the steps of the "random walk" under or above the $\frac{n}{2}$-line. See Figure 3.5. This characterization of random sequences over $\mathbb{F}_2$ was first investigated by Rueppel in [32, Chapter 4]. Now we extend his discussions on binary sequences to the sequences over $\mathbb{F}_q$ without any difficulty.

Denote the infinite sequences over $\mathbb{F}_q$ by the variable sequence $\widetilde{\mathbf{S}} = (S_1, S_2, \ldots, S_i, \ldots)$. And here the S_i are uniformly and independently distributed random variables over $\mathbb{F}_q$. Now suppose the first n terms are given by $S_i = s_i \in \mathbb{F}_q$ for $i = 1, 2, \ldots, n$. Firstly, consider $L(\tilde{\mathbf{s}}^n) = l_0 \leq \frac{n}{2}$ described in Figure 3.5 (a). Then since each S_j for $j \geq n+1$ must hold one and only one exact value in $\mathbb{F}_q$ to keep the line straight, then the probability of the first jump point $n+k$ happening when $k = i_0$ is $\frac{1}{q^{i_0-1}} \times (1 - \frac{1}{q})$. Therefore, given $q > 1$ implying the convergence of the following infinite sum, the expected value for the random variable k is

$$E(k) = \sum_{i=1}^{\infty} i \times \frac{1}{q^{i-1}} \times \frac{q-1}{q} = \frac{q}{q-1}, \text{ where } l_0 \leq \frac{n}{2}. \qquad (3.10)$$

Secondly, consider $L(\tilde{\mathbf{s}}^n) = l_0 > \frac{n}{2}$ by looking at Figure 3.5 (b). Then there must be no jump point while $k \leq 2l_0 - n$ by the Berlekamp-Massay algorithm. Then we just need to concern about the variable terms after S_{2l_0} by neglecting the values of $\{S_{n+1}, S_{n+2}, \ldots, S_{2l_0-1}\}$. By (3.10) the expected value of the random variable $k' = k - (2l_0 - n)$, which represents the step length starting from $2l_0$, is $\frac{q}{q-1}$. Therefore, we have

$$E(k) = (2l_0 - n) + \frac{q}{q-1}, \text{ where } l_0 > \frac{n}{2}. \tag{3.11}$$

Based on the above discussion, it is easy to conclude that the expected value for the random variable w representing the length between two balance points is $\frac{2q}{q-1}$. A little more computation will return the variance of w:

$$V(w) = E(w^2) - [E(w)]^2 = \sum_{i=1}^{\infty} \frac{(q-1)i^2}{q^i} - (\frac{q}{q-1})^2 = \frac{10q-3}{(q-1)^2}. \tag{3.12}$$

Since $f(q) = \frac{10q-3}{(q-1)^2}$ is decreasing when $q \geq 2$, and $f(2) = 17$, $f(q) = \frac{10q-3}{(q-1)^2}$ has to be less than or equal to 17. And when $q \geq 13$, the variance is already less than 1. Now summarize the above discussions to:

Theorem 3.7: Let $\widetilde{\mathbf{S}} = (S_1, S_2, \ldots, S_i, \ldots)$ be an infinite sequences of independent and uniformly distributed random variables over $\mathbb{F}_q$. If $S_i = s_i \in \mathbb{F}_q$ for $i = 1, 2, \ldots n$, and $L(\tilde{\mathbf{s}}^n) = l_0$ where $\tilde{\mathbf{s}}^n = (s_1, s_2, \ldots, s_n)$, then the expected length k, such that $n + k$ is the first appearing jump point after n, is

$$E(k) = \begin{cases} \frac{q}{q-1}, & \text{if } l_0 \leq \frac{n}{2}, \\ (2l_0 - n) + \frac{q}{q-1}, & \text{if } l_0 > \frac{n}{2}. \end{cases} \tag{3.13}$$

And the expected length between two balance points is $\frac{2q}{q-1}$ with the variance $\frac{10q-3}{(q-1)^2}$.

Proof: By (3.10), (3.11), and (3.12). □

From Theorem 3.7, we could expect that the graph of the linear complexity profile of a random sequence would look like "an irregular staircase" with most step lengths being the expected value $\frac{2q}{q-1}$ and most stair heights being $\frac{q}{q-1}$ especially when $q \geq 13$. Again refer to Figure 3.1. Although all terms are over $\mathbb{F}_2$ and only finitely many terms

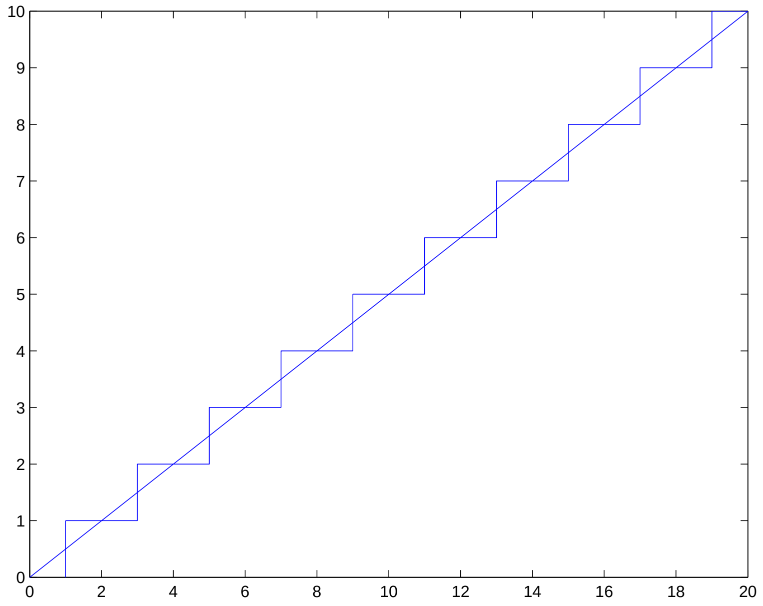

Figure 3.6: The Linear Complexity Profile of ${}^{p}\tilde{\mathbf{s}}^{20}$

are involved, it really shows "a typical irregular staircase" with step length 4 and height 2. One should pay attention to the word "irregular". Randomness necessarily needs the non-regularity as we discussed in Section 1. So any regular characteristics should be excluded from a true random sequence. Although $V(w)$ is not very big especially when $q \geq 13$ as shown in Theorem 3.7, one cannot expect a random sequence to have a regular linear complexity graph, which means all or almost all the "stairs" have the length $\dfrac{2q}{q-1}$ and the height $\dfrac{q}{q-1}$. A very famous example is given by the binary sequence ${}^{p}\tilde{\mathbf{s}} = (s_1, s_2, \ldots, s_i, \ldots)$ where,

$$s_i = \begin{cases} 1, & \text{if } i = 2^j - 1, \text{ for } j = 1, 2, \ldots, \\ 0, & \text{otherwise .} \end{cases}$$

This sequence can be represented in a very simple way and it is almost a zero sequence. Therefore, it cannot be a random sequence from our viewpoints because each bit in this sequence is highly predictable. However, in [32, Chapter 4], the author has shown that $L({}^{p}\tilde{\mathbf{s}}^{n}) = \lfloor \dfrac{n+1}{2} \rfloor$ for $n = 1, 2, \ldots$. So the step length and the step height in the graph of its linear complexity are 4 and 2 respectively, which are both exactly the expected values (see Figure 3.6), which implies that the linear complexity graph of ${}^{p}\tilde{\mathbf{s}}$ is very regular.

5. Some Discussions on Periodic Sequences

In most cases, the key generators will generate periodic sequences since they are implemented by linear devices such as LFSRs. Therefore the expected linear complexity of periodic sequences should be considered. In this part, a heuristic argument is provided and related research results are listed but without rigorous mathematical proofs.

Let $\tilde{\mathbf{s}}^T = (s_1, s_2, \ldots, s_T)$ be the first T terms of the infinite periodic sequence $\tilde{\mathbf{s}} = (s_1, s_2, \ldots, s_i, \ldots)$ over $\mathbb{F}_q$, where T is its minimal period and each s_i for $i = 1, 2, \ldots, T$ is uniformly and independently chosen from $\mathbb{F}_q$. Based on the linear recurrence relation $s_{i+T} = s_i$, the expected linear complexity of $\tilde{\mathbf{s}} = (\tilde{\mathbf{s}}^T, \tilde{\mathbf{s}}^T, \cdots, \tilde{\mathbf{s}}^T)$ must be equal to or less than T. Let $s(x)$, $s^*(x)$ be the generating polynomial for $\tilde{\mathbf{s}}$ and the corresponding polynomial of $\tilde{\mathbf{s}}^T$ respectively, then we have

$$s(x) = \frac{s^*(x)}{1 - x^T} = \frac{s_1 + s_2 x + \ldots + s_T x^{T-1}}{1 - x^T} = \sum_{i=1}^{n} \sum_{j=1}^{m_i} \frac{c_{i,j}(x)}{[P_i(x)]^j}, \tag{3.14}$$

where $P_i(x)$ for $i = 1, 2, \ldots, n$ are the irreducible factors over $\mathbb{F}_q$ with the multiplicities m_i of $x^T - 1$ and $deg[c_{i,j}(x)] < deg[P_i(x)]$ for all i, j. For fixed T, it is well known that there is a bijective mapping between $s^*(x)$ and the partial fraction expansion $\sum_{i=1}^{n} \sum_{j=1}^{m_i} \frac{c_{i,j}(x)}{[P_i(x)]^j}$. Therefore, all the coefficients of $c_{i,j}(x)$ are chosen independently and uniformly from $\mathbb{F}_q$ means so are all the coefficients of $s^*(x)$. This proposition implies that all s_i for $i = 1, 2, \ldots, T$ are selected uniformly and independently from $\mathbb{F}_q$ just as random variables. Therefore, computing the expected linear complexity of the periodic sequence $\tilde{\mathbf{s}}$ could be done by computing the expected degree of the minimal polynomial of $\tilde{\mathbf{s}}$ given that all the coefficients of $c_{i,j}(x)$ for all i, j are chosen independently and uniformly from $\mathbb{F}_q$.

However, obtaining the expected degree of the minimal polynomial directly is generally not easy for a symbol parameter T because it is difficulty to find all the factors over $\mathbb{F}_q$ of $x^T - 1$ and their corresponding multiplicities are highly dependent on T. Firstly, let us consider two extreme cases for the general conclusion. For the first one, let $q = 2$ and $T = q^p - 1 = 2^p - 1$ where p is a prime. In [32, Chapter 4], it was shown that

$$E[L(\tilde{\mathbf{s}})] \geq e^{-\frac{1}{p}}(2^p - \frac{3}{2}). \tag{3.15}$$

Therefore, from (3.15) we conclude that $E[L(\tilde{\mathbf{s}})] \approx 2^p - 1 - \frac{1}{2} = T - \frac{1}{2} \approx T$ as $p \to \infty$, which implies that the expected linear complexity of the periodic sequence $\tilde{\mathbf{s}}$ is close to T, its minimal period, when T is big enough. The other extreme case is given by $T = 2^n$. Still in [32, Chapter 4], the author proved that the infinite sequence $\widetilde{\mathbf{S}}$ generated by repeating $(S_1, S_2, \ldots, S_T)$, a sequence of T independent and uniformly distributed binary variables, has the expected linear complexity

$$E[L(\tilde{\mathbf{s}})] = 2^n - 1 + \frac{1}{2^{2^n}} \approx T. \tag{3.16}$$

Hence, from the above two extreme but heuristic cases, we conclude that the expected linear complexities of periodic sequences should be close to their minimal periods in general cases. This conjecture was first given by Rueppel in [32, Chapter 4] in 1986, but it was not proved in the following 16 years. In 2002, Meidl and Niederreiter proved it by employing the generalized discrete Fourier transform in [24].

Suppose $w \geq 1$ and $j \geq 0$ are both integers and $\gcd(q, w) = 1$, then the *cyclotomic coset* $C_j \mod w$ with respect to powers of q is defined as

$$C_j = \{0 \leq k \leq w - 1 : k \equiv jq^r \mod w \text{ for some } r \geq 0\}.$$

Now let $T = p^v w$, where p is the characteristic of $\mathbb{F}_q$, $v \geq 0$ and $\gcd(p, w) = 1$. By employing the different cyclotomic cosets $\mod w$, $\{B_i\}$ for $i = 1, 2, \ldots, h$, Meidl and Niederreiter [24] proved that

$$E[L(\tilde{\mathbf{s}})] = T - \sum_{j=1}^{h} \frac{|B_j|(1 - q^{-|B_j|p^v})}{q^{|B_j|} - 1}. \tag{3.17}$$

Then, for any T, by (3.17) we can deduce $E[L(\tilde{\mathbf{s}})] > T - \frac{w}{q-1}$. Especially, when $\gcd(T, q) = 1$, we have $E[L(\tilde{\mathbf{s}})] \geq (1 - \frac{1}{q})T$. It is likely and easy to see $E[L(\tilde{\mathbf{s}})]$ and T are quite close if q is big or w is small. For some small q like $q = 2$, in [24], it was also shown that $E[L(\tilde{\mathbf{s}})] \geq T - \frac{w+2}{3}$ when $w > 0$ and $E[L(\tilde{\mathbf{s}})] \geq \frac{3T-1}{4}$ when $w = 0$. To sum up, the expected linear complexities of periodic sequences are actually close to their minimal period.

Therefore, by recalling the discussions in the last two sections, we could expect that periodic sequences will have almost the same performance as or be indistinguishable from random sequences with respect to the linear complexity profiles in their first two periods.

On the whole, from all the discussions in this chapter, we could expect that a sequence over $\mathbb{F}_q$ generated by a key generator should have these properties to simulate random sequences with respect to the linear complexity:

- Its linear complexity profile graph should be close to the $\frac{n}{2}$-line in its first two periods.
- Its linear complexity graph should consist of irregular staircases with average height $\frac{q}{q-1}$ and average length $\frac{2q}{q-1}$ in its first two periods.
- Its linear complexity should be close to its minimal period.

Chapter 4

The k-Error Linear Complexity

As we mentioned, all the practical stream ciphers are implemented by linear devices, especially by Linear Feedback Shift Registers for hardware reasons. Since the Berlekamp-Massey algorithm is very efficient to compute the linear complexity of any sequences over $\mathbb{F}_q$, a *cryptographically strong* key stream should necessarily have a large linear complexity so that from some previous states of the key stream, even if their amount is relatively huge, it is impossible to decide the structure of the key generator for obtaining the key stream.

However, only large linear complexity itself does not guarantee the cryptographic strength. Suppose the linear complexity of a key stream decrease dramatically after changing k terms, where k is relatively small compared to the key stream length. Then it is easy to decide the altered key stream or so called *k-error key stream* by the Berlekamp-Massey algorithm. Although the plaintext obtained by using the k-error key stream is not exactly the original plaintext, one could recover all or most of the errors in the plaintext by *the information redundancy* because the number of errors in the plaintext is relatively small compared to the length of the plaintext given k being small. Therefore, another important parameter to measure the security of the key stream is the so-called *k-error linear complexity.* A cryptographically strong key stream must not only have big linear complexity but also have large k-error linear complexity for relatively small k. In this chapter, we will investigate the latter characteristic of the sequences.

4.1 Bounds for the k-Error Linear Complexity

Generally speaking, it is hard to decide the exact value of the k-error linear complexity given a sequence over any finite field $\mathbb{F}_q$. However, by employing some tools in number theory, we could estimate the k-error linear complexities of periodic sequences. Obviously, we are more concerned about the lower bound. So in this section, we will develop some tools to obtain lower bounds for the k-error linear complexity. Firstly, let us give the exact definition of the k-error linear complexity.

Definition 4.1: Let $\tilde{\mathbf{s}}_i^n = (s_1^i, s_2^i, \ldots, s_n^i)$ and $\tilde{\mathbf{s}}_j^n = (s_1^j, s_2^j, \ldots, s_n^j)$ be two sequences over $\mathbb{F}_q$. Then the **Hamming distance** d between $\tilde{\mathbf{s}}_i^n$ and $\tilde{\mathbf{s}}_j^n$ is defined to be

$$d(\tilde{\mathbf{s}}_i^n, \tilde{\mathbf{s}}_j^n) = \sum_{k=1}^{n} t_k, \quad \text{where } t_k = 0 \text{ if } s_k^i = s_k^j, \text{ and } t_k = 1 \text{ otherwise.}$$

Definition 4.2: Let $\tilde{\mathbf{s}}^n = (s_1, s_2, \ldots, s_n)$ be a sequence over $\mathbb{F}_q$, k be an integer such that $0 \leq k \leq n$ and $\tilde{\mathbf{s}}_e^n$ be an error sequence of $\tilde{\mathbf{s}}^n$ with length n over $\mathbb{F}_q$. Then the **k-error linear complexity** $L_k(\tilde{\mathbf{s}}^n)$ of $\tilde{\mathbf{s}}^n$ is defined to be

$$L_k(\tilde{\mathbf{s}}^n) = \min_{d(\tilde{\mathbf{s}}_e^n, \tilde{\mathbf{s}}^n) \leq k} L(\tilde{\mathbf{s}}_e^n).$$

Some notation should be clarified here. In Definition 4.2, we just define the k-error linear complexity for finite sequences. However, in this chapter, we are more concerned about the infinite periodic sequences with minimal period T. So we specify the notation $L_k(\tilde{\mathbf{s}})$ for $\tilde{\mathbf{s}} = (s_1, s_2, \ldots, s_i, \ldots)$, where $\tilde{\mathbf{s}}$ is an infinite periodic sequence with minimal period T, to represent the value of $\min\limits_{\tilde{\mathbf{s}}_e^T}\{L[(\tilde{\mathbf{s}}_e^T)^\infty] \mid d(\tilde{\mathbf{s}}_e^T, \tilde{\mathbf{s}}^T) \leq k\}$.

Before any discussion, we should offer an extreme example to show the dramatic difference between the linear complexity and the k-error linear complexity of a sequence. Let us consider

$$\tilde{\mathbf{s}}_c = (0000000000000000000000000000001).$$

By the Berlekamp-Massey algorithm, the linear complexity of $\tilde{\mathbf{s}}_c$ is 31, the same as its length. However, its 1-error linear complexity is obviously 0 since the Hamming distance between $\tilde{\mathbf{s}}_c$ and the zero sequence with length 31 is just 1.

To find the connection between periodic sequences over $\mathbb{F}_q$ and number theory, we need to introduce some number-theoretic tools first.

Definition 4.3: Let n be a positive integer and $B = \{m_i \mid \gcd(n, m_i) = 1 \text{ and } 1 \leq m_i \leq n\}$. Then the **Euler function** $\phi(n)$ is defined to be $|B|$, the cardinality of the finite set B.

Definition 4.4: Let φ, n be positive integers such that $\gcd(\varphi, n) = 1$. Then the **order of φ modulo** n, denoted by $ord_n(\varphi)$, is defined to be the smallest positive integer k such that $\varphi^k \equiv 1 \bmod n$.

Definition 4.5: Call φ **a primitive root modulo** n if $ord_n(\varphi) = \phi(n)$. For an element $\xi \in \mathbb{F}_q$, if n is the smallest positive number such that $\xi^n = 1$ over $\mathbb{F}_q$, then ξ is called an **nth primitive root unity over $\mathbb{F}_q$**.

Definition 4.6: Let n be a positive integer and p be the characteristic of $\mathbb{F}_q$, where $p \nmid n$. Let ξ be an nth primitive root of unity over $\mathbb{F}_q$. Then the **nth cyclotomic polynomial** over $\mathbb{F}_q$ is defined to be

$$Q_n(x) = \prod_{1 \leq s \leq n,\ \gcd(s,n)=1} (x - \xi^s).$$

From the definitions, it is not hard to obtain some basic properties. For the Euler function, it is actually *multiplicative*, which means that for any positive integers m, n, if $\gcd(m, n) = 1$, then $\phi(mn) = \phi(m)\phi(n)$. For the cyclotomic polynomial we have:

1. $\deg[Q_n(x)] = \phi(n)$.
2. $x^n - 1 = \prod_{d|n} Q_d(x)$.
3. $Q_n(x)$ is independent of the choice of the nth primitive root of unity.
4. Suppose $Q_n(x) = \sum_{i=0}^{\phi(n)} a_i x^i$, then $a_i \in \mathbb{F}_q$ for all i.
5. If $\gcd(q, n) = 1$, then $Q_n(x)$ is factored into $\dfrac{\phi(n)}{d}$ distinct monic irreducible polynomials in $\mathbb{F}_q[x]$ with the same degree d, where d is the order of q mod n, $ord_n(q)$.

Other interesting properties of Euler function, primitive root and cyclotomic polynomial with rigorous mathematical proofs can be found in [30, Chapter 6] and [18, Chapter 2].

Lemma 4.1: Let $n_1, n_2, \ldots, n_t$ be positive integers and $\gcd(n_i, n_j) = 1$ for all $1 \leq i < j \leq t$. Suppose $m \geq 1$ is an integer with $\gcd(n_k, m) = 1$ for all $1 \leq k \leq t$, then

$$ord_{n_1 n_2 \cdots n_t}(m) = \mathrm{lcm}[ord_{n_1}(m),\ ord_{n_2}(m),\ \ldots,\ ord_{n_t}(m)].$$

Proof: Let $n = n_1 n_2 \cdots n_t$. Since $\gcd(n_i, n_j) = 1$ for $i \neq j$, by the *Chinese remainder theorem*, we have $Z_n \cong Z_{n_1} \times Z_{n_2} \times \cdots \times Z_{n_t}$ and the isomorphism f is given by

$$f(x \bmod n) = (x \bmod n_1,\ x \bmod n_2,\ \ldots,\ x \bmod n_t).$$

Denote $\mathrm{lcm}[ord_{n_1}(m),\ ord_{n_2}(m),\ \ldots,\ ord_{n_t}(m)]$ by l. On one hand, given that $f(1)$ is the unit of $Z_{n_1} \times Z_{n_2} \times \cdots \times Z_{n_t}$, for any positive integer d such that $m^d \equiv 1 \bmod n$, we have $f(m^d \bmod n) = f(1 \bmod n) = (1, 1, \ldots, 1) = (m^d \bmod n_1,\ m^d \bmod n_2, \ldots,\ m^d \bmod n_t)$. This forces $m^d \equiv 1 \bmod n_i$ for all $1 \leq i \leq t$. Therefore $ord_{n_i}(m) \mid d$ for all i, which implies that l must divide d. Thus, we have

$$\mathrm{lcm}[ord_{n_1}(m),\ ord_{n_2}(m),\ \ldots,\ ord_{n_t}(m)] \mid ord_n(m). \tag{4.1}$$

On the other hand, $m^l \equiv 1 \bmod n_i$ for all $1 \leq i \leq t$. Therefore, $f(m^l \bmod n) = (1, 1, \ldots, 1)$, which forces $m^l \equiv 1 \bmod n$. So by the definition of order, we have

$$ord_n(m) \mid \mathrm{lcm}[ord_{n_1}(m),\ ord_{n_2}(m),\ \ldots,\ ord_{n_t}(m)]. \tag{4.2}$$

By (4.1) and (4.2), we conclude that $ord_n(m) = \mathrm{lcm}[ord_{n_1}(m),\ ord_{n_2}(m),\ \ldots,\ ord_{n_t}(m)]$.□

Lemma 4.2: Let p be a prime, then for any positive integers k and c with $\gcd(c, p) = 1$ we have $ord_{p^k}(c) \geq ord_p(c)$.

Proof: Let $d = ord_{p^k}(c)$. Then $c^d \equiv 1 \bmod p^k$. Hence, $c^d \equiv 1 \bmod p$. Still by the definition of order, we have that $ord_p(c) \mid d = ord_{p^k}(c)$, which implies $ord_{p^k}(c) \geq ord_p(c)$. □

Now based on the above two lemmas, we are ready to set up the connection between periodic sequences and number theory. This connection originated from [10] and was described in detail in [8, Chapter 3]. From now on, use $w(\tilde{\mathbf{s}}^n) = w[(s_1, s_2, \ldots, s_n)]$ to represent the *Hamming weight* of $\tilde{\mathbf{s}}^n$.

Theorem 4.1: Let $N = p_1^{e_1} p_2^{e_2} \cdots p_t^{e_t}$, where $p_1, p_2, \ldots, p_i$ are pairwise distinct primes and $e_i \geq 1$ for all i. Let q be a prime power such that $\gcd(q, N) = 1$. Suppose $k < \min\{w(\tilde{\mathbf{s}}^N), N - w(\tilde{\mathbf{s}}^N)\}$, then for each periodic sequence $\tilde{\mathbf{s}}$ over $\mathbb{F}_q$ with the minimal period N,

$$\begin{aligned} L(\tilde{\mathbf{s}}) \geq &\ \min\{ord_{p_1}(q),\ ord_{p_2}(q),\ \ldots,\ ord_{p_t}(q)\}, \\ L_k(\tilde{\mathbf{s}}) \geq &\ \min\{ord_{p_1}(q),\ ord_{p_2}(q),\ \ldots,\ ord_{p_t}(q)\}. \end{aligned}$$

Proof: Recall the properties we listed earlier for cyclotomic polynomials, and the fact that $\tilde{\mathbf{s}}$ is periodic with minimal period N. Next, we note that a *characteristic polynomial* for $\tilde{\mathbf{s}}$ is

$$x^N - 1 = \prod_{n_i | N} Q_{n_i}(x).$$

which is the same as its reciprocal polynomial. And $Q_{n_i}(x)$ is the product of $\dfrac{\phi(n_i)}{ord_{n_i}(q)}$ distinct monic irreducible polynomials over $\mathbb{F}_q$ with the same degree $ord_{n_i}(q)$.

Suppose $n_i \mid N$, then we have $n_i = p_{i_1}^{r_{i_1}} p_{i_2}^{r_{i_2}} \cdots p_{i_s}^{r_{i_s}}$, where $1 \leq r_{i_j} \leq e_{i_j}$ for $1 \leq j \leq s$. Here $1 \leq s \leq t$. According to Lemma 4.1, we have

$$ord_{n_i}(q) = \text{lcm}[ord_{p_{i_1}^{r_{i_1}}}(q),\ ord_{p_{i_2}^{r_{i_2}}}(q),\ \ldots,\ ord_{p_{i_s}^{r_{i_s}}}(q)].$$

Therefore, $ord_{n_i}(q) \geq \max[ord_{p_{i_1}^{r_{i_1}}}(q),\ ord_{p_{i_2}^{r_{i_2}}}(q),\ \ldots,\ ord_{p_{i_s}^{r_{i_s}}}(q)]$. By Lemma 4.2, however, we have

$$\begin{aligned} ord_{n_i}(q) &\geq \max[ord_{p_{i_1}^{r_{i_1}}}(q),\ ord_{p_{i_2}^{r_{i_2}}}(q),\ \ldots,\ ord_{p_{i_s}^{r_{i_s}}}(q)] \\ &\geq \max[ord_{p_{i_1}}(q),\ ord_{p_{i_2}}(q),\ \ldots,\ ord_{p_{i_s}}(q)] \\ &\geq \min[ord_{p_1}(q),\ ord_{p_2}(q),\ \ldots,\ ord_{p_t}(q)]. \end{aligned}$$

Hence the degree of the minimal polynomial for every periodic sequence over $\mathbb{F}_q$ with the minimal period N must be equal to or greater than $ord_{n_i}(q)$. Proposition 2.1 implies that

$$L(\tilde{\mathbf{s}}) \geq \min\{ord_{p_1}(q),\ ord_{p_2}(q),\ \ldots,\ ord_{p_t}(q)\}. \tag{4.3}$$

Now because $k < \min\{w(\tilde{\mathbf{s}}^N), N - w(\tilde{\mathbf{s}}^N)\}$, after changing the same k terms in every minimal period of $\tilde{\mathbf{s}}$, the error sequence must be non-constant and it is still periodic with the period N. Then its minimal polynomial cannot be $x-1$. Therefore, its linear complexity should be equal to the degree of the product of some factors of $x^N - 1$. Hence, we have that

$$L_k(\tilde{\mathbf{s}}) \geq \min\{ord_{p_1}(q),\ ord_{p_2}(q),\ \ldots,\ ord_{p_t}(q)\}. \tag{4.4}$$

□

Now, from Theorem 4.1, we see the basic connection between the lower bounds for the linear complexity and the k-error linear complexity of a sequence and number theory. In fact, the main idea of Theorem 4.1 inspires us to find some special N and q, such that all the nontrivial factors of $x^N - 1$ over $\mathbb{F}_q$, except $x - 1$, have large degrees.

4.2 Lower Bounds for the k-Error Linear Complexity with Special Period

To proceed for the lower bounds, we need some preparation on the existence of the primitive roots modulo n. Denote the group of units of $\mathbb{Z}/n\mathbb{Z}$ by $(\mathbb{Z}/n\mathbb{Z})^*$, where $n \geq 2$ is an integer. Then it is well known that $(\mathbb{Z}/n\mathbb{Z})^*$ is a *cyclic group* if and only if $n = 2, 4, p^m, 2p^m$ for some odd prime p. Therefore, we could find $g \in (\mathbb{Z}/n\mathbb{Z})^*$ such that it is a primitive root modulo n if $n = 2, 4, p^m, 2p^m$. In fact, this assertion originated from a number theory result first given by C. F. Gauss in the book *Disquisitiones Arithmeticae*. One could refer to lots of books on number theory, for example [34, Section 33], for details and proofs. Furthermore, to serve our purpose, we just cite a useful result on primitive roots modulo $n = p^m$ from number theory but without showing proofs.

Lemma 4.3: Let p, k be an odd prime and a positive integer, respectively. Say g is a primitive root modulo p. Then

- g is also a primitive root modulo p^2 if $g^{p-1} \not\equiv 1 \bmod p^2$.
- $g + p$ is a primitive root modulo p^2 if $g^{p-1} \equiv 1 \bmod p^2$.
- g is a primitive root modulo p^{k+1} if g is a primitive root modulo p^k, where $k \geq 2$.

Proof: Refer to [30, Theorem 8.8, Theorem 8.9]. □

Now we are ready to obtain some good bounds for the k-error linear complexity based on Theorem 4.1 and Lemma 4.3.

Theorem 4.2: Let p be an odd prime and $N = p^m$ where $m \geq 1$. For any nonconstant sequence $\tilde{\mathbf{s}} = (\tilde{\mathbf{s}}^N)^\infty$ whose period is N over $\mathbb{F}_q$, if q is a primitive root modulo p and $q^{p-1} \not\equiv 1 \bmod p^2$, then for any $k < \min\{w(\tilde{\mathbf{s}}^N), N - w(\tilde{\mathbf{s}}^N)\}$ we have $L_k(\tilde{\mathbf{s}}) \geq p - 1$.

Proof: Firstly, from Lemma 4.3, we conclude that q is a primitive root modulo p^t for any $t \geq 2$. From the property (2) of the cyclotomic polynomials we listed, we have that in the finite field $\mathbb{F}_q$, the characteristic polynomial of $\tilde{\mathbf{s}}$ can be factored as

$$x^N - 1 = x^{p^m} - 1 = (x - 1) \prod_{i=1}^{m} Q_{p^i}(x). \tag{4.5}$$

By the property (5) of the cyclotomic polynomials, $Q_{p^i}(x)$ could be factored into $\dfrac{\phi(p^i)}{ord_{p^i}(q)}$ irreducible polynomials over $\mathbb{F}_q$. However, by Lemma 4.3, $ord_{p^i}(q) = \phi(p^i)$. That means that $Q_{p^i}(x)$ itself is an irreducible polynomial over $\mathbb{F}_q$. Since $\phi(p^i) = (p-1)p^{i-1}$, $deg[Q_{p^i}(x)] = (p-1)p^{i-1} \geq p-1$ for all $1 \leq i \leq m$. Still according to $k < \min\{w(\tilde{\mathbf{s}}^N), N - w(\tilde{\mathbf{s}}^N)\}$, the k-error sequence $\tilde{\mathbf{s}}_\mathbf{e}$ after changing the same k terms in its every period cannot be a constant sequence. So 1 and $x-1$ cannot be its minimal polynomial. Hence we conclude that $L_k(\tilde{\mathbf{s}}) \geq p-1$. □

It is known that for every prime p, there are $\phi[\phi(p)] = \phi(p-1)$ primitive elements in $(\mathbb{Z}/p\mathbb{Z})^*$. Now suppose $q + p\mathbb{Z} \in (\mathbb{Z}/p\mathbb{Z})^*$ is the generating element of this cyclic group under multiplication, and the representing element q satisfying $1 < q \leq p-1$ is equal to some power of some prime. Then the significance of Theorem 4.2 arises: if we have a huge prime p, then according to Theorem 4.2, we could obtain an ideal lower bound.

Proposition 4.1: Let p be an odd prime. For any nonconstant sequence $\tilde{\mathbf{s}} = (\tilde{\mathbf{s}}^p)^\infty$ whose period is p over $\mathbb{F}_q$, if q is a primitive root modulo p, then for any $k < \min\{w(\tilde{\mathbf{s}}^p), N - w(\tilde{\mathbf{s}}^p)\}$ we have $L_k(\tilde{\mathbf{s}}) \geq p-1$.

Proof: Let $m = 1$. Then the conclusion directly follows Theorem 4.2 and its proof. □

Proposition 4.1 tells us that for a periodic sequence with its period being a large prime p, if the cardinality of its underlying field is a primitive root modulo p, then the k-error linear complexity for $\tilde{\mathbf{s}} = (\tilde{\mathbf{s}}^p)^\infty$ has a lower bound just slightly less than its period. Since the difference is just 1, this lower bound could be viewed the same as its period. Still by the same argument in Theorem 4.2, we have the linear complexity of this sequence is p or $p-1$, which also could be viewed the same as its period. Therefore, $\tilde{\mathbf{s}} = (\tilde{\mathbf{s}}^p)^\infty$ is an ideal key stream with respect to the linear complexity and k-error linear complexity since both of them almost achieve the maximal value, the minimal period of the sequence.

Since the cardinality of any finite field must be a power of some prime, the remaining task is to find a primitive root q modulo p^m, such that $q = r^t$ where r is a prime.

Definition 4.7: Let g, n be positive integers such that $\gcd(g, n) = 1$. Call e the **negative order modulo** n of g if e is the smallest positive integer such that $g^e \equiv -1 \bmod n$ and denote it by $nord_n(g)$.

Lemma 4.4: For $n > 4$, g is a primitive root modulo n if and only if $\frac{\phi(n)}{2}$ is the negative order of $g \bmod n$.

Proof: ($\Longrightarrow$) Suppose g is a primitive root modulo n. Then by definition or Fermat's little theorem, $g^{\phi(n)} \equiv 1 \bmod n$. Recall the fact that there are primitive roots modulo n if and only if $n = 2, 4, p^e, 2p^e$. Therefore, $\phi(n)$ must be an even number. Then $g^{\phi(n)} - 1 = (g^{\frac{\phi(n)}{2}} + 1)(g^{\frac{\phi(n)}{2}} - 1) = In$, where I is an integer. Hence, $g^{\frac{\phi(n)}{2}}$ must be equivalent to -1 modulo n, otherwise we will have $g^{\frac{\phi(n)}{2}} \equiv 1 \bmod n$, which contradicts to g is a primitive root modulo n. This proves the existence of the negative order of $g \bmod n$ and informs us that $nord_n(g) \leq \frac{\phi(n)}{2}$. Obviously, $\phi(n) = ord_n(g) \mid 2nord_n(g)$. This forces $nord_n(g) \geq \frac{\phi(n)}{2}$, then $nord_n(g) = \frac{\phi(n)}{2}$.

($\Longleftarrow$) Consider $nord_n(g) = \frac{\phi(n)}{2}$. Then $ord_n(g) \mid \phi(n) = 2nord_n(g)$. Suppose $ord_n(g) < nord_n(g) = \frac{\phi(n)}{2}$, then let $e \equiv nord_n(g) \pmod{ord_n(g)} < nord_n(g)$. However, $g^e \equiv -1 \bmod n$. Contradicts to the definition of negative order. If $nord_n(g) \leq ord_n(g) < 2nord_n(g)$, $e = 2nord_n(g) - ord_n(g)$ is also less than $nord_n(g)$, which leads to another contradiction $g^e \equiv -1 \bmod n$. Therefore, we must have $ord_n(g) \geq 2nord_n(g) = \phi(n)$. Still by Fermat's little theorem, we conclude that $ord_n(g) = \phi(n)$. □

Lemma 4.5: If g is a primitive root modulo n, where n is an integer greater than 2, it must not be a *quadratic residue* modulo n.

Proof: Primitive root implies that $g^{\frac{\phi(n)}{2}} \equiv -1 \bmod n$. If g is a quadratic residue, by supposing $g \equiv g_0^2 \bmod n$, then $g^{\frac{\phi(n)}{2}} \equiv g_0^{\phi(n)} \equiv 1 \bmod n$, which is an immediate contradiction given $n > 2$. □

Therefore, from Lemma 4.5, if r^t is a primitive root modulo p^m, then t is necessarily an odd integer. Because the key streams embedded in the finite field with characteristic 2 are easy to be implemented by hardware, we are more concerned about $r = 2$ here. Next we will check whether $2^t = 2^{2u+1}$ for some non-negative integer u is a primitive root modulo p^m or not based on the above lemmas. Generally, it is not easy to decide the primitive root modulo p^m especially when p is huge. Therefore, we limit p to some special forms and m to be 1 for discussion. Introduce two properties of the *Legendre symbol* from [29, Chapter 3]. Suppose p is an odd prime and $\gcd(a, p) = \gcd(b, p) = 1$, then

$$\left(\frac{a^2 b}{p}\right) = \left(\frac{b}{p}\right), \quad \left(\frac{-1}{p}\right) = (-1)^{\frac{p-1}{2}}. \tag{4.6}$$

Therefore by (4.6), for 2^{2u+1} and the odd prime p, we have

$$\left(\frac{2^{2u+1}}{p}\right) = \left(\frac{2}{p}\right) = (-1)^{\frac{p^2-1}{8}} = \begin{cases} +1 & \text{if } p = 8a+1, \text{ or } p = 8a+7, \\ -1 & \text{if } p = 8a+3, \text{ or } p = 8a+5. \end{cases} \tag{4.7}$$

If the Legendre symbol of p modulo n is 1, then p is a quadratic residue. If the Legendre symbol of p is -1, then p is not a quadratic residue modulo n.

Theorem 4.3: If $p = 4p^* + 1$ where p^* is a prime and $2^{4u+2} < p - 1$, then 2^{2u+1} is a primitive root modulo p.

Proof: By Fermat's little theorem and given $\phi(p) = 4p^*$, after factoring, we have

$$(2^{2u+1})^{\phi(p)} - 1 \equiv [(2^{2u+1})^{\frac{\phi(p)}{2}} + 1][(2^{2u+1})^{p^*} + 1][(2^{2u+1})^{p^*} - 1] \equiv 0 \bmod p. \tag{4.8}$$

Firstly, by (4.6), -1 is a quadratic residue. Then since $p^* = 2a + 1$ is a prime, p must be in the form of $8a + 5$. Because

$$\left(\frac{[2^{2u+1}]^{p^*}}{p}\right) = \left(\frac{2^{2up^*+2a+1}}{p}\right) = \left(\frac{2}{p}\right) = -1,$$

we conclude that $(2^{2u+1})^{p^*}$ is not a quadratic residue, which implies $(2^{2u+1})^{p^*} \not\equiv \pm 1 \bmod p$. Therefore, we must have $(2^{2u+1})^{\frac{\phi(p)}{2}} \equiv -1 \bmod p$. Hence $nord_p(2^{2u+1}) \mid 2p^*$. Now consider $0 < 2^{4u+2} < p - 1$, so we have $(2^{2u+1})^2 \not\equiv -1 \bmod p$. Thus, by employing the fact that $(2^{2u+1})^{p^*} \not\equiv -1 \bmod p$, $nord_p(2^{2u+1})$ must be equal to $2p^* = \dfrac{\phi(p)}{2}$. Then by Lemma 4.4, 2^{2u+1} is a primitive root modulo p. □

Of course, there are lots of results on primitive roots by setting different conditions on the modulus n. One could refer to [8, Section 3.4] for different conclusions. Moreover, in [7] one could find discussions on the distribution properties of primitive roots over finite fields. Also one could refer to [37] for algorithms to find primitive roots. Now combine the results of Theorem 4.2 and Theorem 4.3, we could achieve our aims to construct the ideal key streams, for which both linear complexities and k-error linear complexities are almost the same as their minimal periods with the little difference being 1 at most.

Theorem 4.4: Let $N = p = 4p^* + 1$ be an odd prime and $q = 2^{2u+1}$, where p^* is also a prime and $q^2 < p - 1$. For any nonconstant periodic sequence $\tilde{\mathbf{s}} = (\tilde{\mathbf{s}}^N)^\infty$ with the period N (minimal) over $\mathbb{F}_q$ we have $L(\tilde{\mathbf{s}}) \geq N - 1$, and for any $k < \min\{w(\tilde{\mathbf{s}}^N), N - w(\tilde{\mathbf{s}}^N)\}$ we have $L_k(\tilde{\mathbf{s}}) \geq N - 1$.

Proof: From Theorem 4.2 and Theorem 4.3, we have $L_k(\tilde{\mathbf{s}}) \geq N-1$ for any $k < \min\{w(\tilde{\mathbf{s}}^N), p-w(\tilde{\mathbf{s}}^N)\}$. Then let $k = 0$, we have $L(\tilde{\mathbf{s}}) \geq N - 1$. □

However, up to now, we still do not have any good method to find such big primes $p = 4p^* + 1$ where p^* is also a prime. Whether there are infinitely many such primes is still one of the difficult open problems in number theory. Right now the only available method for constructing these primes is just searching a large prime table for a matched pair (p, p^*) such that $p = 4p^* + 1$.

For the probabilistic properties, one could refer to Meidl and Niederreiter [23], [24] for lower bounds on the expected value of the k-error linear complexity. A special case when the period N is a prime different from the characteristic of $\mathbb{F}_q$, was considered in [22].

Bibliography

[1] N. Abramson, *Information Theory and Coding*, McGraw-Hill, New York, 1963.

[2] J. O. Bruer, On pseudorandom sequences as crypto generators, *Proceeding Int. Zurich Seminar on Digital Commmunication*, Switzerland, 1984.

[3] J. A. Buchmann, *Introduction to Cryptography, Second Edition*, Springer, New York, 2004.

[4] P. Caballero-Gil, Regular cosets and upper bounds on the linear complexity of certain sequences, *Sequences and Their Applications* (C. Ding, T. Helleseth, and H. Niederreiter, eds.), pp. 161-170, Springer, London, 1999.

[5] P. Caballero-Gil, New upper bounds on the linear complexity, *Computational Mathematics Applications*, vol. 39, no. 3-4, pp. 31-38, 2000.

[6] C. Calude, *Information and Randomness, An Algorithmic Perspective*, Springer-Verlag, Berlin, 1994.

[7] C. Carlitz, Distribution of primitive roots in a finite field, *Quarterly Journal of Mathematics*, vol. 4, pp. 4-10, 1953.

[8] T. W. Cusick, C. Ding, A. Renvall, *Stream Ciphers and Number Theory*, Elsevier, Amsterdam, 1998

[9] C. Ding, G. Xiao, W. Shan, New measure indexes on the security of stream ciphers, *Proceedings of the Third Chinese National Workshop on Cryptology*, Xi'an, China, pp. 5-15, 1988.

[10] C. Ding, G. Xiao, W. Shan, *The Stability Theory of Stream Ciphers*, Springer-Verlag, Berlin, 1991.

[11] L. J. Garcia-Villalba and A. F*ú*ster-Sabater, On the linear complexity of the sequences generated by nonlinear filterings, *Information Processing Letters*, vol. 76, no. 1-2, pp. 67-73, 2000.

[12] M. R. Garey and D.S. Johnson, *Computers and Intractability*, W.H. Freeman, New York, 1979.

[13] S. M. Jennings, Multiplexed sequences: some properties of the minimum polynomial, *Proceedings Workshop on Cryptography*, Springer-Verlag, Berlin, LNCS vol. 149, pp. 189-206, 1983.

[14] S. M. Jennings, Autocorrelation function of the multiplexed sequences, *IEEE Proceedings*, vol. 131, no. 2, pp. 169-172, 1984.

[15] A. N. Kolmogorov, Three approaches to the quantitative definition of information, *Problems Information Transmission*, vol. 1, pp. 1-7, 1965.

[16] S. Konyagin, T. Lange, and I. E. Shparlinski, Linear complexity of the discrete logarithm, *Designs Codes Cryptography*, vol. 28, pp. 135-146, 2003.

[17] A. Lempel, J. Ziv, On the complexity of finite sequences, *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 75-81, 1976.

[18] R. Lidl, H. Niederreiter, *Introduction to Finite Fields and Their Applications*, Cambridge University Press, Cambridge, 1994.

[19] P. Martin-Löf, The definition of random sequences, *Information and Control*, vol. 9, pp. 602-619, 1966.

[20] J. L. Massey, Shift-register synthesis and BCH decoding, *IEEE Transactions on Information Theory*, vol. 15, no. 1, pp. 122-127, 1969.

[21] W. Meidl, Enumeration results on linear complexity profiles and lattice profiles, *Journal of Complexity*, vol. 22, pp. 275-286, 2006.

[22] W. Meidl and H. Niederreiter, Linear complexity, k-error linear complexity, and the discrete Fourier transform, *Journal of Complexity*, vol. 18, pp. 87-103, 2002.

[23] W. Meidl and H. Niederreiter, Counting functions and expected values for the k-error linear complexity, *Finite Fields Applications*, vol. 8, pp. 142-154, 2002.

[24] W. Meidl and H. Niederreiter, On the expected value of the linear complexity and the k-error linear complexity of periodic sequences, *IEEE Transactions on Information Theory*, vol. 48, pp. 2817-2825, 2002.

[25] W. Meidl and A. Winterhof, Lower bounds on the linear complexity of the discrete logarithm in finite fields, *IEEE Transactions on Information Theory*, vol. 47, pp. 2807-2811, 2001.

[26] H. Niederreiter, Sequences with almost perfect linear complexity profile, *Advances in Cryptology - EUROCRYPT' 87: Workshop on the Theory and Application of Cryptographic Techniques, Amsterdam, The Netherlands, April 1987. Proceedings*, Springer, Berlin, LNCS vol. 304, pp. 37-51, 1988.

[27] H. Niederreiter, Some computable complexity measures for binary sequences, *Sequences and Their Applications* (C. Ding, T. Helleseth, and H. Niederreiter, eds.), pp. 67-78, Springer, London, 1999.

[28] H. Niederreiter, C. P. Xing, *Rational Points on Curves over Finite Fields, Theory and Applications*, London Mathematical Society Lecture Note Series 285, Cambridge University Press, Cambridge, 2001.

[29] D. Redmond, *Number Theory, An Introduction*, Marcel Dekker, New York, 1996.

[30] K. H. Rosen, *Elementary Number Theory and Its Applications*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1984.

[31] R. A. Rueppel, Correlation immunity and the summation combiners, *Advances in Cryptology: Proceeding Cryptography 85*, Springer-Verlag, Berlin, pp. 260-272, 1986.

[32] R. A. Rueppel, *Analysis and Design of Stream Ciphers*, Springer-Verlag, Berlin, 1986.

[33] A. Shamir, Stream cipher: dead or alive, *Advances in Cryptology - ASIACRYPT 2004: 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea*, Springer, Berlin, p. 78, 2004.

[34] D. Shanks, *Solved and Unsolved Problems in Number Theory*, Chelsea Publishing Company, New York, 1985.

[35] C. E. Shannon, The mathematical theory of communication, *The Bell System Technical Journal*, vol. 27, pp. 379-423, 1948.

[36] C. E. Shannon, Communication theory of secrecy systems, *The Bell System Technical Journal*, vol. 28, pp. 656-715, 1949.

[37] V. Shoup, Searching for primitive roots in finite fields, *Mathematics of Computation*, vol. 58, pp. 369-380, 1992.

[38] I. E. Shparlinski, *Number Theoretic Methods in Cryptography: Complexity Lower Bounds*, Birkhäuser, Basel, 1999.

[39] I. E. Shparlinski, *Cryptographic Applications of Analytic Number Theory: Complexity Lower Bounds and Pseudorandomness*, Birkhäuser, Basel, 2003.

[40] G. J. Simmons, *Contemporary Cryptology, The Science of Information Integrity*, IEEE Press, Piscataway, 1992.

[41] R. J. Solomonov, A formal theory of inductive inference, Part I, *Information and Control*, vol. 7, pp. 1-22, 1964.

[42] C. H. Tan, Period and linear complexity of cascaded clock-controlled generators, *Sequences and Their Applications* (C. Ding, T. Helleseth, and H. Niederreiter, eds.), pp. 371-378, Springer, London, 1999.

Appendix: The Maple Program of the Berlekamp-Massey Algorithm

```
% Input s is the sequence;
% N is its length;
% P is the number of elements in the underlying field;
% x is a formal symbol.
% The returning value of function ‘BM’ is the linear complexity of s.
% The file is saved as ‘BMA’ in the working space of Maple.

BM := proc(s, N, P, x)
local C, B, T, L, k, i, n, d, b, safemod;
safemod := (exp, P) - > ‘if’(P=0, exp, exp mod P);
B := 1; C := 1; L := 0;
k := 1; b := 1;
for n from 0 to N - 1 do
    d := s[n + 1];
    for i from 1 to L do
        d := safemod(d + coeff(C, x^i) * s[n - i + 1], P);
    od;
    if d=0 then k := k + 1 fi;
    if (d <> 0 and 2 * L > n) then
        C := safemod(expand(C - d * x^k * B/b), P);
        k := k + 1;
    fi;
    if (d <> 0 and 2 * L <= n) then
        T := C;
        $C$ := safemod(expand($C - d * x^k * B/b$), $P$);
        $B := T$;
        $L := n + 1 - L$;
        $k := 1$;
        $b := d$;
    fi;
od;
return $C$;
save BM, BMA;
```