B-it **Winter School 2008**

Lecture Notes

Stream Ciphers

Lecturer: L. El Aimani

Contents

Chapter 1

Algebraic tools

This chapter provides a short survey on algebraic tools that will be employed throughtout the document. Many of the concepts treated here can be found in any undergraduate book of algebra.
The chapter is structured as follows: section 1 gives basic material in Number theory. Section 2 is a collection of facts about the elementary structures in abstract algebra which are: groups,rings and fields and finally polynomial rings.

1.1 Number theory

1.1.1 The integers

The set of integers $\{..., -3, -2, -1, 0, 1, 2, 3, ...\}$ is denoted by the symbol $\mathbb{Z}$.

Definition 1. *Let a and b be integers. Then a divides b (equivalentely: a is a divisor of b, or a is a factor of b) if there exists an integer c such that $b = ac$. If a divides b, then this is denoted by $a \mid b$.*

Example 1. $-3 \mid 18$, $173 \mid 0$,...

Fact 1.

1. *$a \mid a$,*
2. *if $a \mid b$ and $b \mid c$, then $a \mid c$,*
3. *if $a \mid b$ and $a \mid c$ then $a \mid bx + cy$ $\forall x, y \in \mathbb{Z}$,*
4. *if $a \mid b$ and $b \mid a$ then $a = \pm b$*

Definition 2. *(Division algorithm for integers) If a and b are integers with $b \geq 1$, then ordinary long division of a and b yields two integers : q (the quotient) and r (the remainder) such that*

$$a = qb + r \text{ where } 0 \leq r < b$$

Moreover, q and r are unique. The remainder of the division is denoted $a \bmod b$ and the quotient is denoted $a \operatorname{div} b$.

Example 2. $a = 73, b = 17 \Rightarrow q = 4, r = 5$

Definition 3. *An integer c is a common divisor of a and b if $c \mid a$ and $c \mid b$.*

Definition 4. *A non-negative integer d is the greatest common divisor of integers a and b denoted $d = gcd(a, b)$ if:*

1. *d is a common divisor of a and b and*
2. *whenever $c \mid a$ and $c \mid b$, then $c \mid d$*

Equivalently, $gcd(a, b)$ is the largest positive integer that divides both a and b with the exception that $gcd(0, 0) = 0$.

Example 3. *The common divisors of* 12 *and* 18 *are* $\{\pm 1, \pm 2, \pm 3, \pm 6\}$ *and* $gcd(18, 12) = 6$

Definition 5. *A non negative integer m is the least common multiple of integers a and b , denoted $m = lcm(a, b)$ if:*

1. *$a \mid m$ and $b \mid m$ and*
2. *whenever $a \mid c$ and $b \mid c$, then $m \mid c$*

Equivalently, $lcm(a, b)$ is the smallest non-negative integer divisble by both a and b.

Example 4. $lcm(12, 18) = 36$

Definition 6. *Two integers a and b are said to be relatively prime or co-prime if $gcd(a, b) = 1$.*

Definition 7. *An integer $p \geq 2$ is said to be prime if its only positive divisors are* 1 *and p. Otherwise, p is called composite.*

Fact 2. *(Fundamental theorem of arithmetic) Every integer $n \geq 2$ has a factorization as a product of prime powers:*

$$n = p_1^{e_1} ... p_k^{e_k}$$

where p_i are distinct primes and e_i are positive integers. Furthermore, the factorization is unique.

Fact 3. *if $a = p_1^{e_1} p_2^{e_2} ... p_k^{e_k}$ and $p_1^{f_1} p_2^{f_2} ... p_k^{e_k}$ then:*

$$gcd(a, b) = p_1^{min(e_1, f_1)} p_2^{min(e_2, f_2)} ... p_k^{min(e_k, f_k)}$$

$$lcm(a, b) = p_1^{max(e_1, f_1)} p_2^{max(e_2, f_2)} ... p_k^{max(e_k, f_k)}$$

Remark 1. *The above definition implies that* $a.b = gcd(a,b).lcm(a,b)$

Definition 8. *For* $n \geq 1$*, let* $\phi(n)$ *denote the number of integers in the interval* $[1,n]$ *which are relatively prime to* n*. The function* $\phi(n)$ *is called the* Euler Phi function *(or the* Euler totient function*)*

Fact 4. *(Properties of Euler phi function)*

1. *If* p *is prime and* e *is a positive integer, then* $\phi(p^e) = p^e - p^{(e-1)}$.
2. *The Euler phi function is* multiplicative. *That is, if* $gcd(m,n) = 1$ *then* $\phi(mn) = \phi(m).\phi(n)$.
3. *If* $n = p_1^{e_1} p_2^{e_2} ... p_k^{e_k}$ *is the prime factorization of* n*, then:*

$$\phi(n) = n(1 - \frac{1}{p_1})(1 - \frac{1}{p_2})...(1 - \frac{1}{p_k})$$

1.1.2 Algorithms in $\mathbb{Z}$

Let a and b be non negative integers of size $|a|$ and $|b|$ respectively. We will consider that $|a|, |b| \leq n$, in other terms, their binary representation needs at most n bits. The number of bit operations (or the complexity) of the four basic integer operations of addition, substraction, multiplication and division using the classical algorithms is summarized in the following table.

Operation	**Bit complexity**
Addition $a + b$	$O(max(\|a\|, \|b\|)) = O(n)$
Substraction $a - b$	$O(max(\|a\|, \|b\|)) = O(n)$
Multiplication $a.b$	$O(\|a\|.\|b\|) = O(n^2)$
Division $a = qb + r$	$O(\|q\|.\|b\|) = O(n^2)$

The greatest common divisor of two integers a and b can be computed via Fact 3. However, computing a gcd by first obtaining the prime factorization of the given numbers does not result in an efficient algorithm, as the problem of factoring integers appears to be difficult. The Euclidean algorithm is an efficient algorithm for computing the greatest common divisor of two integers that does not require the factorization of the integers. It is based on the following fact:

Fact 5. *If* a *and* b *are positive integers with* $a \geq b$*, then* $gcd(a,b) = gcd(b, a \bmod b)$

Proof. Let $a = qb + r$ be the euclidean division of a by b, then $r \equiv a \bmod b$. We need to prove two things:

1. $gcd(a,b) \mid gcd(b, a \bmod b)$: We have $gcd(a,b) \mid a$ and $gcd(a,b) \mid b$, then $gcd(a,b) \mid bq$ and therefore $gcd(a,b) \mid a - qb = r$, by definition of the gcd, this results in $gcd(a,b) \mid gcd(a, r \equiv a \bmod b)$.

2. $gcd(b, a \bmod b) \mid gcd(a, b)$: We have $gcd(b, r \equiv a \bmod b) \mid b$ and $gcd(b, r) \mid r$, then $gcd(b, r) \mid bq$ and therefore $gcd(b, r) \mid bq + r = a$, finally by definition of the gcd, $gcd(b, r) \mid gcd(a, b)$.

Thereby, $gcd(a, b) = \pm gcd(b, a \bmod b)$. As the gcds are, by definition, non-negative integers, we conclude then that $gcd(a, b) = gcd(b, a \bmod b)$. □

The above fact leads to the following algorithm for computing the gcd of two integers:

Algorithm 1. ***Euclidean Algorithm for computing the greatest common divisor of two integers***
INPUT: two integers a and b with $a \geq b$.
OUTPUT: the greatest common divisor of a and b.

1. *while $b \neq 0$ do the following:*
   (a) *Set $r \leftarrow a \bmod b$*
   (b) *Set $a \leftarrow b$*
   (c) *Set $b \leftarrow r$*
2. *return (a)*

We remark that the euclidean algorithm has a complexity $O(|b|.|b|.|q|)$ $(a = bq + r)$. If we consider that the given numbers have size less than n, then the running time of the euclidean algorithm is $O(n^3)$. However, we can notice that the bs are decreasing,this leads to a running time of $= O(n^2)$.

Example 5. *(Euclidean algorithm) The following are the division steps of algorithm 1 for computing $gcd(4864, 3458) = 38$:*

$$4864 = 1.3458 + 1406$$

$$3458 = 2.1406 + 646$$

$$1406 = 2.646 + 114$$

$$646 = 5.114 + 76$$

$$114 = 1.76 + 38$$

$$76 = 2.38 + 0$$

The euclidean algorithm can be extended so that it does not only yield the greatest common divisor of two integers a and b, but also integers x and y satisfying $ax+by = gcd(a, b)$. We first notice that the Euclidean algorithm calculates a sequence defined by a two terms recurrence:

$$a_0 = a, a_1 = b$$

$$a_{n-1} = q_n a_n + a_{n+1}$$

Where $q_n = \lfloor(\frac{a_{n-1}}{a_n})\rfloor$.
In other terms:

$$a_{n+1} = -q_n a_n + a_{n-1} \tag{1.1}$$

This leads to another version of the Euclidean algorithm which is as follows:

Algorithm 2. ***Euclidean Algorithm (2)***
INPUT: two integers a and b with $a \geq b$.
OUTPUT: the greatest common divisor of a and b.

1. *Set $a_0 \leftarrow a$, $a_1 \leftarrow b$, $i \leftarrow 1$.*
2. *While $a_i \neq 0$ do the following:*
   (a) *Set $a_{i+1} \leftarrow a_{i-1} \bmod a_i$*
   (b) *Set $i \leftarrow i + 1$*
3. *return (a_{i-1})*

Now, if we put:

$$a_n = ax_n + by_n$$

The sequences x_n and y_n satisfy the same equation 1.1.($x_{n+1} = -q_n x_n + x_{n-1}$) and ($y_{n+1} = -q_n y_n + y_{n-1}$), where $q_n = \lfloor(\frac{a_{n-1}}{a_n})\rfloor$. This leads to the following algorithm:

Algorithm 3. ***Extended Euclidean Algorithm***
INPUT: two integers a and b with $a \geq b$.
OUTPUT: $gcd(a, b)$ and two integers x and y, such that $a.x+b.y = gcd(a, b)$.

1. *Set:*
   (a) $a_0 \leftarrow a$, $a_1 \leftarrow b$.
   (b) $x_0 \leftarrow 1$, $x_1 \leftarrow 0$.
   (c) $y_0 \leftarrow 0$, $y_1 \leftarrow 1$.
   (d) $i \leftarrow 1$
2. *While $a_i \neq 0$ Set:*
   (a) $q_i = \lfloor(\frac{a_{i-1}}{a_i})\rfloor$
   (b) $a_{i+1} \leftarrow a_{i-1} - q_i a_i$
   (c) $x_{i+1} \leftarrow x_{i-1} - q_i x_i$
   (d) $y_{i+1} \leftarrow y_{i-1} - q_i y_i$

(e) $i \leftarrow i + 1$

3. *return* $(a_{i-1}, x_{i-1}, y_{i-1})$

Example 6. 1. $a_0 = 4864$, $a_1 = 3458$, $x_0 = 1$, $x_1 = 0$, $y_0 = 0$,$y_1 = 1$, $i = 1$.

2. *the loop:*

   (a) *step 1:* $q_1 = 1$, $a_2 = 1406$, $x_2 = 1$, $y_2 = -1$, $i = 2$ *(*$1406 = (1).4864 + (-1).3458$*)*

   (b) *step 2:* $q_2 = 2$, $a_3 = 646$, $x_3 = -2$, $y_3 = 3$, $i = 3$ *(*$646 = (-2).4864 + (3).3458$*)*

   (c) *step 3:* $q_3 = 2$, $a_4 = 114$, $x_4 = 5$, $y_4 = -7$, $i = 4$ *(*$114 = (5).4864 + (-7).3458$*)*

   (d) *step 4:* $q_4 = 5$, $a_5 = 76$, $x_5 = -27$, $y_5 = 38$, $i = 5$ *(*$76 = (-27).4864 + (38).3458$*)*

   (e) *step 5:* $q_5 = 1$, $a_6 = 38$, $x_6 = 32$, $y_6 = -45$, $i = 6$ *(*$38 = (32).4864 + (-45).3458$*)*

   (f) *step 6:* $q_6 = 2$, $a_7 = 0$, $x_7 = -91$, $y_7 = 128$, $i = 7$ *(*$0 = (-91).4864 + (128).3458$*)*

3. *return* $(a_6 = 38, x_6 = 32, y_6 = -45)$

1.1.3 The integers modulo n

Let n be a positive integer.

Definition 9. *If a and b are integers, then a is said to be* congruent to b modulo n*, written $a \equiv \pmod{n}$, if n divides $(a-b)$. The integer n is called the* modulus *of the congruence.*

Example 7. 1. $24 \equiv 9 \pmod{5}$ *since* $24 - 9 = 3.5$

2. $-11 \equiv 17 \pmod{7}$ *since* $-11 - 17 = -4.7$

Fact 6. ***(properties of congruences)*** *For all $a, a_1, b, b_1, c \in \mathbb{Z}$, the following are true:*

1. $a \equiv b \pmod{n}$ *if and only if a and b leave the same remainder when divided by n.*

2. *(reflexivity)* $a \equiv a \pmod{n}$

3. *(symmetry) if* $a \equiv b \pmod{n}$ *then* $b \equiv a \pmod{n}$

4. *(transitivity) if* $a \equiv b \pmod{n}$ *and* $b \equiv c \pmod{n}$ *then* $a \equiv c \pmod{n}$

5. *if* $a \equiv a_1 \pmod{n}$ *and* $b \equiv b_1 \pmod{n}$*, then* $a + b \equiv a_1 + b_1 \pmod{n}$ *and* $ab \equiv a_1 b_1 \pmod{n}$.

The *equivalence class* of an integer a, denoted $cl(a)$, is the set of all integers congruent to a modulo n. From properties $2, 3$ and 4, it can be seen that for a fixed n, the relation of *congruence modulo* n **partitions** $\mathbb{Z}$ into equivalence classes. In fact, for all $i \in \mathbb{Z}$, $a \in cl(a)$ (reflexivity) thus $\mathbb{Z} = \bigcup_{i \in \mathbb{Z}} cl(i)$. From the other hand, if $x \in cl(a) \cap cl(b)$, then $a \in cl(b)$ (symmetry + transitivity), it follows by the same properties that $cl(a) = cl(b)$, which proves that the equivalence classes are disjoint.
Now if $a = qn + r$, where $0 \leq r < n$, then $a \equiv r \pmod{n}$. Hence, each integer a is congruent modulo n to a unique integer between 0 and $n - 1$. Thus a and r are in the same equivalence class, and so r may simply be used to represent this equivalence class.

Definition 10. *The* integers modulo n*, denoted* $\mathbb{Z}_n$*, is the set of (equivalence classes of) integers* $\{0, 1, ...n - 1\}$*. Addition, substraction and multiplication in* $\mathbb{Z}_n$ *are performed modulo* n*.*

Example 8. $\mathbb{Z}_2 = \{0, 1\}$*. In* $\mathbb{Z}_2$*,* $1 + 1 = 0$*, since* $1 + 1 = 2 = 0 \pmod{2}$*. Similarly* $1.1 = 1$ *in* $\mathbb{Z}_2$*. It is obvious that the relation* congruent modulo 2 ***partitions*** $\mathbb{Z}$ *into two dsjoints sets, the set of integers congruent to* 1 *modulo* 2 *, and the set of integers congruent to* 0 *modulo* 2*, in other terms, it partions* $\mathbb{Z}$ *into the set of odd integers and the set of even integers.*

Definition 11. *Let* $a \in \mathbb{Z}_n$*. The multiplicative inverse of* $a \pmod{n}$ *is an integer* $x \in \mathbb{Z}_n$ *such that* $ax = 1 \pmod{n}$*. If such inverse exists, it is unique and is denoted* a^{-1}*.*

Fact 7. *a is inversible if and only if* $gcd(a, n) = 1$*. Moreover, this inverse can be efficientely computed using the Extended Euclidean Algorithm.*

Proof. Extended Euclidean Algorithm. □

Finally, we end this section about integers with a theorem of great importance: The Chinese Remainder Theorem CRT.

Theorem 1. ***(Chinese remainder theorem, CRT)*** *If the integers* $n_1, n_2, .., n_k$ *are pairwise relatively prime, then the system of simultaneous congruences :*

$$x \equiv a_1 \pmod{n_1}$$
$$x \equiv a_2 \pmod{n_2}$$
$$.$$
$$.$$
$$.$$
$$x \equiv a_k \pmod{n_k}$$

has a unique solution modulo $n = n_1 n_2 ... n_k$*.*

1.2 Abstract Algebra

1.2.1 Groups

Definition 12. *A group $(G, *)$ consists of a set G with a binary operation satisfying:*

1. *The group operation is* associative. *That is $a * (b * c) = (a * b) * c$ for all $a, b, c \in G$.*
2. *There is an element $1 \in G$, called the* identity element, *such that $a * 1 = 1 * a = a$ for all $a \in G$.*
3. *For each $a \in G$ there exits an element $a^{-1} \in G$,* inverse *of a, such that $a * a^{-1} = 1$*
   A group G is abelian *(or* commutative*) if furthermore,*
4. *$a * b = b * a$ for all $a, b \in G$.*

Example 9. 1. *$(\mathbb{Z}, +)$, $(\mathbb{Z}_n, +)$ (where n is a positive integer), $(\mathbb{R}^*, .)$,...*

2. *The multiplicative group of $\mathbb{Z}_n$:*

$$\mathbb{Z}_n^* = \{a \in \mathbb{Z}_n / gcd(a, n) = 1\}$$

The group operation is of course the multiplication modulo n, wich is associative, the identity element is 1 and every element has an inverse due to fact 7.

Definition 13. *A group G is called a* finite group *if the set G is finite. The number of elements in a finite group is called its* order.

Example 10. 1. *$(\mathbb{Z}_n, +)$ is a finite group of order n.*

2. *$(\mathbb{Z}_n^*, .)$ is a finite group of order $\phi(n)$.*

Definition 14. *A non-empty subset H of a group G is a* subgroup *of G if H is itself a group with respect to the operation of G.*

Example 11. *$(\mathbb{Z}, +)$ is a subgroup of $(\mathbb{Q}, +)$ which is itself a subgroup of $(\mathbb{R}, +)$*

Definition 15. *A group G is cyclic if there is an element $\alpha \in G$ such that for each $b \in G$, there is an integer i with $b = \alpha^i$. Such element α is called a* generator *of G.*

Fact 8. *If G is a group and $a \in G$, then the set of all powers of a forms a cyclic subgroup of G, called the subgroup* generated by a, *and denoted by $< a >$*

Definition 16. *Let G be a group and $a \in G$. The* order *of a is defined to be the least positive integer t such that $a^t = 1$, provided that such integer exists. If such a t does not exist, then the order of a is defined to be ∞*

Fact 9. *Let G be a group and $a \in G$. The order of the group generated by a, $< a >$ is exactely the order of the element a.*

Fact 10. ***(Lagrange's theorem)*** *If G is a finite group and H is a subgroup of G, then $ord(H) \mid ord(G)$. Hence, if $a \in G$, the order of a divides $ord(G)$.*

Fact 11. *Let $(G, .)$ be a finite group. Then $\forall a \in G, a^{ord(G)} = 1$, where 1 is the identity element in G.*

Before ending this paragraph about groups, it is important to exhibit an important algorithm for exponentiation in multiplicative groups. Recall that in the Diffie-Hellman key exchange, we stated that, in a multiplicative group, exponentiation is easy, but we did not give an algorithmic solution to it. Actually, if we perform the exponentiation by successive multiplications, it results in a very inefficient algorithm (whose running time depends exponentially on the exponent). However, if we use the following remark:

$$g^{\sum_{i=0}^{k} t_i 2^i} = \prod_{i=0}^{k} g^{t_i 2^i} = (g^{2^0})^{t_0} (g^{2^1})^{t_1} ... (g^{2^k})^{t_k}$$

we will have an efficient algorithm running in a polynomial time.

Algorithm 4. ***(Square and multiply algorithm)***
INPUT: $g \in (G, .)$, where G is a multiplicative group whose identity element is denoted 1, and $t = \sum_{i=0}^{k} t_i 2^i$, an integer ($t \leq n$) given by its binary representation.
OUTPUT: g^t

1. *Set $b \leftarrow 1$, if $t = 0$ then return (b)*
2. *Set $B \leftarrow g$*
3. *If $t_0 = 1$ then set $b \leftarrow g$*
4. *For i from 1 to k do the following:*
   (a) *Set $B \leftarrow B^2$*
   (b) *If $t_i = 1$ then set $b \leftarrow b.B$*
5. *return (b)*

The above algorith has complexity $O(k.mult_G)$, where $mult_G$ is the complexity of multiplying two elements in the group G.

1.2.2 Rings and Fields

Definition 17. *A ring $(R, +, .)$ consists of a set R with two binary operations denoted $+$ (addition) and $.$ (multiplication) satisfying the following axioms:*

1. *$(R, +)$ is an abelian group with identity denoted 0.*
2. *The operation $.$ is associative. That is $a.(b.c) = (a.b).c = a.b.c$ for all $a, b, c \in R$*
3. *There is a multiplicative identity denoted 1, with $1 \neq 0$, such that $1.a = a.1 = a$ for all $a \in R$*
4. *The operation $.$ is* distributive *over $+$. That is, $a.(b+c) = (a.b)+(a.c)$ and $(b + c).a = (b.a) + (c.a)$ for all $a, b, c \in R$*

The ring is a *commutative ring* if $a.b = b.a$ for all $a, b \in R$

Example 12.
1. *The set of integers $\mathbb{Z}$ with the usual operations of addition and multiplication is a commutative ring.*
2. *the set $\mathbb{Z}_n$ with addition and multiplication performed modulo n is a commutative ring.*

Definition 18. *A set I is called an ideal of a ring $(R, +, .)$ if:*

1. *I is a subset of R*
2. *$\forall a, b \in I \ a + b \in I$*
3. *$\forall a \in I, r \in R \ a.r \in I$*

Example 13.
1. *The set $\{0\}$ is an ideal of $(\mathbb{Z}, +, .)$*
2. *The set of even numbers, denoted $2\mathbb{Z}$, is an ideal of $(\mathbb{Z}, +, .)$*
3. *In general, the set of multiples of any integer a, denoted $a\mathbb{Z}$, is an ideal of $(\mathbb{Z}, +, .)$.*

Fact 12. *The ideals of $(\mathbb{Z}, +, .)$ are $a\mathbb{Z}$ where a is an integer.*

Proof.
1. $a\mathbb{Z}$ is clearly an ideal of $(\mathbb{Z}, +, .)$, for all $a \in \mathbb{Z}$.
2. Let I be an ideal of $(\mathbb{Z}, +, .)$ and let $a = min(I \cap \mathbb{N}^*)$. If $x \in I$, then $x = qa + r$, where $0 \leq r < a$. $a \in I$, then $-qa \in I$, hence $r = x - qa \in I$. Since $a = min(I \cap \mathbb{N}^*)$, then $r = 0$ and therefore $x = qa$, in other terms $x \in a\mathbb{Z}$. Finally $I \subset a\mathbb{Z}$. The other inclusion is obvious by definition of an ideal.

□

This motivates the following definition:

Definition 19. *A ring is called principal if all its ideals can be generated by a unique element.*

In this way, we call $\mathbb{Z}$ a principal ring.

Definition 20. *A field F is a commutative ring in which all non-zero elements have multiplicative inverses or equivalentely, F^*, which denotes the set F without the zero element, is a group for the multiplication.*

Fact 13. *For a prime integer p, $\mathbb{Z}_p$ is a field.*

1.2.3 Polynomial rings

Definition 21. *Let p be a prime integer. $\mathbb{Z}_p[x]$ denotes the set of polynomials whose coefficients are in $\mathbb{Z}_p$.*

In the rest of this paragraph, $f(x)$ will denote a polynomial of $\mathbb{Z}_p[x]$ of degree $n \geq 1$.

Fact 14. *$(\mathbb{Z}_p[x], +, .)$ is a commutative ring for the usual addition and multiplication of polynomials*

Definition 22. *Let $f(x)$, $g(x)$ and $h(x)$ be elements of $\mathbb{Z}_p[x]$, where $deg(f) = n \geq 1$.*

1. *$f(x)$ is said to divide $g(x)$ (and we denote $f(x) \mid g(x)$) if there exists a polynomial $q(x) \in \mathbb{Z}_p[x]$ such that:*

$$g(x) = q(x)f(x)$$

2. *$g(x)$ is said to be congruent to $h(x)$ modulo $f(x)$, and we denote:*

$$g(x) \equiv h(x) \pmod{f(x)}$$

if

$$f(x) \mid (g(x) - h(x))$$

Now, let's consider the ring of polynomials modulo $f(x)$ denoted $\mathbb{Z}_p[x]/f(x)$. The construction is the same as the construction of $\mathbb{Z}_n$ from $\mathbb{Z}$.
If $deg(f) = n$ and if we divide $g(x)$ by $f(x)$, we get $q(x)$ and $r(x)$ such that

$$g(x) = q(x)f(x) + r(x)$$

where $deg(r(x)) < n$.
We get then to the important result: every polynomial in $\mathbb{Z}_p[x]$ is congruent (modulo $f(x)$) to a unique polynomial in $\mathbb{Z}_p[x]$ of degree at most $n - 1$.

Definition 23. *$\mathbb{Z}_p[x]/f(x)$ is the set of polynomials of $\mathbb{Z}_p[x]$ of degree at most $n-1$*

Fact 15. *$(\mathbb{Z}_p[x]/f(x), +, .)$, where $+$ and $.$ denote respectively addition and multiplication of polynomials in $\mathbb{Z}_p[x]$ followed by a reduction modulo $f(x)$, is a ring*

Definition 24. *Let $f(x)$ be a polynomial in $\mathbb{Z}_p[x]$. $f(x)$ is said to be irreducible if there exists no polynomials $f_1(x)$ and $f_2(x)$ in $\mathbb{Z}_p[x]$ such that :*

$$f(x) = f_1(x)f_2(x)$$

with $def(f_1) > 0$ and $deg(f_2) > 0$

Fact 16. *$\mathbb{Z}_p[x]/f(x)$, where $f(x)$ is irreducible, is a field and the order of $(\mathbb{Z}_p[x]/f(x))^*$ (seen as a group whose operation is the multiplication of polynomials in $\mathbb{Z}_p[x]$, followed by a reduction modulo $f(x)$)is $p^n - 1$*

Proof.
- Extended Euclidean Algorithm for polynomials.
- The set of polynomials, whose coefficients are in $\mathbb{Z}_p$, of degree at most $n-1$ contains p^n elements, if we discard the zero element, then there are exactly $p^n - 1$ elements.

□

Before going further in this section, it is convinient to note the symmetry between arithmetic on integers and arithmetic on polynomials. We sum up the basic similarities in the following table:

Integers	**Polynomials**
$\mathbb{Z}$	$\mathbb{Z}_p[x]$, where p is prime
$\mathbb{Z}_n$	$\mathbb{Z}_p[x]/f(x)$
prime modulus n	irreducible modulus $f(x)$
ideals are $a\mathbb{Z}$	ideals are $b(x)\mathbb{Z}_p[x]$

Example 14. *In this example, we construct the field of $8 = 2^3$ elements. We can do it by looking for an irreducible polynomial of degree 3 in $\mathbb{Z}_2[x]$, for example $f(x) = x^3 + x + 1$. The elements of $\mathbb{Z}_2[x]$ are: $0, 1, x, x+1, x^2, x^2+1, x^2+x, x^2+x+1$.*

- *addition:* $x^2 + (x^2 + x + 1) = 2.x^2 + x + 1 = x + 1 \pmod{(x^3 + x + 1)}$. *(the coefficients of the polynomials are in $\{0, 1\}$)*
- *multiplication:* $(x^2 + 1)(x^2 + x + 1) = x^4 + x^3 + x + 1 = (x + 1)(x^3 + x + 1) + x^2 + x = x^2 + x \pmod{(x^3 + x + 1)}$.
- *the order of $(\mathbb{Z}_2[x]/f(x))^*$ is $7 = 2^3 - 1$*

Definition 25. *$f(x)$ is said to be a primitive polynomial if x is a generator of the multiplicative group $(\mathbb{Z}_p[x]/f(x))^*$*

Fact 17. *In the group $(\mathbb{Z}_p[x]/f(x))^*$, $ord(x) \mid (p^n - 1)$, with equality if $f(x)$ is a primitive polynomial.*

Proof. Lagrange theorem plus the above definition □

In the above example, the polynomial $f(x) = x^3 + x + 1$ is a primitive polynomial, one can check easily that all the elements of $(\mathbb{Z}_2[x]/f(x))^*$ can be obtained by successive powers of x.
Finally, there is one more concept needed in the next chapter which is the order of a polynomial.

Definition 26. *Let $f(x) \in \mathbb{Z}_p[x]$ be a polynomial such that $f(0) \neq 0$. We define the order of f, e, as the least positive integer such that $f(x) \mid x^e - 1$. Equivalentely, such that $x^e \equiv 1 \pmod{f(x)}$.*

Remark 2. *In case the polynomial $f(x)$ is irreducible, the order of $f(x)$ and the order of the polynomial x in the multiplicative group $(\mathbb{Z}_p[x]/f(x))^*$ coincide.*

Chapter 2

Linearly recurrent sequences

Sequences whose terms depend in a simple manner on their predecessors are of great importance for a variety of applications. Such sequences are easy to generate by recursive procedures, which is certainly advatageous from the computational viewpoint. In this chapter, we are particularly interested in the case where the terms depend linearly on a fixed number of predecessors, resulting in a so-called linearly recurrent sequences. Such sequences are used in many applications, for instance in stream ciphers as we will see in later chapters.
In section 1, we define what a linearly recurrent sequence is and discuss some of its basic periodicity properties. Section 2 introduces the concept of the minimal polynomial of a linearly recurrent sequence, further periodicity properties are also treated in this way. Section 3 is of both theoretical and practical interest, in fact, it shows how to implement the generation of linearly recurrent sequences on special switching circuits called *Linear Feedback Shift Registers* or *LFSRs*, and establishes then the interface between mathematics and electrical engineering. The last section is dedicated to the summary of the already stated results.
Last but not least, the content of this chapter is freely inspired from the book of von zur Gathen and Gerhard [GG03] as well as the book of Lidl and Niederreiter [LN86].

2.1 Introduction

2.1.1 Basic definitions

Let F be a field. Then $F^{\mathbb{N}}$ is the (infinite-dimentional) field of infinite sequences $(s_i)_{i\in\mathbb{N}}$, with all $s_i \in F$.

Definition 27. *A sequence $a = (s_i)_{i\in\mathbb{N}} \in F^{\mathbb{N}}$ is **linearly recurrent (over***

F) *if there exist* $L \in \mathbb{N}$ *and* $c_0, ..., c_L \in F$ *with* $c_L \neq 0$ [1] *such that*

$$\sum_{0 \leq j \leq L} c_j s_{n+j} = c_L s_{n+L} + ... + c_1 s_{n+1} + c_0 s_n = 0 \tag{2.1}$$

for all $n \in \mathbb{N}$.

1. *L is called the order of the sequence.*
2. *The row vector* $(s_n, s_{n+1}, ..., s_{n+L-1})$ *is referred to as the* nth state vector *of the linear recurring sequence.In particualr, the vector* $(s_0, s_{0+1}, ..., s_{0+L-1})$ *denotes the* initial state vector.[2]
3. *The polynomial* $c = \sum_{0 \leq j \leq L} c_j x^j \in F[x]$ *of degree L, which depends only on the coefficients of the linear recurrence relation, is called a* ***characteristic*** *(or annihilating)* ***polynomial*** *of a.*

Example 15. 1. *$F = \mathbb{R}$, $s_n = 0$ for all $n \in \mathbb{N}$. This sequence is linearly recurrent of order 0, and any nonzero polynomial f is a characteristic polynomial.*

2. *$F = \mathbb{Q}$, $s_0 = 0$, $s_1 = 1$, $s_{n+2} = s_{n+1} + s_i$ for all $n \geq 0$. Then $s = (s_n)_{n \geq 0}$ is the* ***Fibonacci sequence****,which is linearly recurrent of order 2 having $(0, 1)$ as an initial state vector and $c = x^2 - x - 1$ is a characteristic polynomial of s.*

2.1.2 Periodicity properties

So far, we gave a general definition on what a lineraly recurrent sequence is. Now, and in the rest of this document, we will consider a special category of linearly recurrent sequences that is *linearly recurrent sequences over finite fields*[3]. Such sequences have the characteristic feature that is, after a possible irregular behavior in the beginning, the sequences are periodic. This paragraph will mention some basic properties about the periodicity of such sequences. Before anouncing these properties, we establish first the necessary terminology for periodic or ultimately periodic sequences.

Definition 28. *Let S be an arbitrary nonempty set, and let $s_0, s_1, ...$ be a sequence of elements of S. If there exist integers $T > 0$ and $n_0 \geq 0$ such that $s_{n+T} = s_n$ for all $n \geq n_0$. Then the sequence is said to be* ultimately periodic *and T is called* a period *of the sequence. The smallest period is called* least period *of the sequence.*

[1] In the rest of the document, we will consider that this condition is satisfied automatically if the given linearly recurrent sequence is of order L

[2] this definition will be justified in section 5

[3] All linearly recurrent sequence that will be considered in the rest of this document are assumed to take values in $\mathbb{F}_q$,(The field of q elements) unless an explicit definition of the field is given

Lemma 1. *Every period of an ultimately periodic sequence is divisible by the least period.*

Proof. Let T be an arbitrary period of the ultimately periodic sequence $s_0, s_1, ...$ and let T_1 be its least period, so we have $s_{n+T} = s_n$ for all $n \geq n_0$ and $s_{n+T_1} = s_n$ for all $n \geq n_1$. We could write $T = qT_1 + r$ where $0 < r < T_1$. Then for all $n \geq max(n_0, n_1)$ we get:

$$s_n = s_{n+T} = s_{n+qT_1+r} = s_{n+(q-1)T_1+r} = ... = s_{n+r}$$

and so r is a period of the sequence, which contradicts with the definition of the least period. □

Definition 29. *An ultimately periodic sequence $s_0, s_1, ...$ with least period T is called* periodic *if $s_{n+r} = s_n$ holds for all $n = 0, 1, ...$*

The follwing condition, which is sometimes found in the literature, is equivalent to the definition of a periodic sequence.

Lemma 2. *The sequence $s_0, s_1, ...$ is periodic if and only if there exists an integer $T > 0$ such that $s_{n+T} = s_n$ for all $n = 0, 1, ...$*

Proof. The necessity of the condition is obvious. Conversely, if the condition is satisfied, then the sequence is ultimately periodic and has a least period T_1. Therefore, with a suitable n_0 we have $s_{n+T_1} = s_n$ for all $n \geq n_0$. Now let n be an arbitrary integer, and m an integer such that $m \equiv n \bmod T$. Then, $s_{n+T_1} = s_{m+T_1} = s_m = s_n$, which shows that the sequence is periodic in the sense of Definition29. □

Remark 3. *If $s_0, s_1, ...$ is ultimately periodic with least periodic T, then the least nonnegative integer n_0 such that $s_{n+T} = s_n$ for all $n \geq n_0$ is called the* preperiod*. The sequence is periodic precisely if the preperiod is* 0*.*

Now, we come back to linearly recurrent sequences and give the basic results concerning the periodic behavior of such sequences.

Theorem 2. *Let $\mathbb{F}_q$ be the finite field of q elements and k a positive integer. Then every Lth-order linearly recurrent sequence in $\mathbb{F}_q$ is ultimately periodic with least period satisfying $T \leq q^L - 1$*

Proof. We note that there are exactly q^{L-1} nonzero[4] distinct L-tuples of elements of $\mathbb{F}_q$. Therefore by considering the state vectors s_n, $0 \leq n \leq q^L$, of a given Lth-order linear recurring sequence in $\mathbb{F}_q$, it follows that $s_i = s_j$ for some i and j with $0 \leq i < j < q^L$. Using the linear recurrence relation and the induction, we arrive at $s_{n+j-i} = s_n$ for all $n \geq i$, which shows that the linear recurring sequence itself is ultimately periodic with least period $T \leq j - i \leq q^L - 1$. □

[4]We discard the zero tuple because the result is the zero sequence which is obviously periodic of least period 1

An important sufficient condition for the periodicity of a linearly recurrent sequence is provided by the following result.

Theorem 3. *If $s_0, s_1, ...$ is a linearly recurrent sequence in a finite field satisfying the linear recurrence relation 2.1 and if the coefficient c_0 is non zero, then the sequence $s_0, s_1, ...$ is periodic*

Proof. According to Theorem 2, the given sequence is ultimately periodic. Let T be its least period and n_0 its preperiod, then $s_{n+T} = s_n$ for all $n \geq n_0$. Suppose we had $n_0 > 0$. from the linear recurrence relation with $n = n_0 + T - 1$ and the fact that $c_0 \neq 0$, we obtain:

$$\begin{aligned} s_{n_0-1+T} &= -c_0^{-1}(c_L s_{n_0+L-1+T} + c_{L-1}s_{n_0+L-2+T} + ... + a_1 s_{n_0+T}) \\ &= -c_0^{-1}(c_L s_{n_0+L-1} - c_{L-1}s_{n_0+L-1} + ... + a_1 s_{n_0}) \end{aligned}$$

We find the same expression for s_{n_0-1}, it follows then that $s_{n_0-1+T} = s_{n_0-1}$. This is a contradiction with the definition of the preperiod. □

2.2 The minimal polynomial

So far, we saw that linearly recurrent sequences are periodic or ultimately periodic. In case they are periodic of period T, the polynomial $x^T - 1$ can be viewed as a characteristic polynomial of the given sequence. This motivates the following questions, what are the characteristic polynomials of a given linearly recurrent sequence? what is the relationship between these characteristic polynomials? how can the choice of the characteristic polynomial impact the predictability of a given sequence?
This section tries to answer such questions by defining a special characteristic polynomial called *the minimal polynomial* which is of a crucial importance in determining the predictability of the given linearly recurrent sequence.

2.2.1 Multiplication by polynomials

At this stage, it is convinient to define the multiplication of sequences in order to be able to approach linearly recurrent sequences from an ideal theory viewpoint.
Let F be a field. Then $F^{\mathbb{N}}$ is the (infinite-dimentional) field of infnite sequences $(s_n)_{n\in\mathbb{N}}$, with all $s_n \in F$.

Definition 30. *Let $f = \sum_{0\leq j\leq L} f_j x^j$ be a polynomial in $F[x]$ of degree L and $s = (s_n)_{n\in\mathbb{N}}$ a sequence in $F^{\mathbb{N}}$. The multiplication of the sequence s by the polynomial f is set as follows:*

$$f \bullet s = (\textstyle\sum_{0\leq j\leq L} f_j s_{n+j})_{n\in\mathbb{N}} \in F^{\mathbb{N}}$$

The constants $f \in F$ act on sequences in the usual way, and the indeterminate x acts as a shift operator:

$$x \bullet s = (s_{n+1})_{n\in\mathbb{N}}.$$

Example 16. *1. $s = (0)_{n\in\mathbb{N}} = \boldsymbol{0}$ is the zero sequence. Then $f \bullet s = \boldsymbol{0}$ is again the zero sequence*[5] *for all $f \in F[x]$.*

2. f is the zero polynomial. Then $f \bullet s = \boldsymbol{0}$ for all $s \in F^{\mathbb{N}}$.

3. $f = 1$. Then $f \bullet s = s$ for all $s \in F^{\mathbb{N}}$.

4. s is the Fibonacci sequence and $f = x^2 - x - 1$. Then $f \bullet s = (-s_n - s_{n+1} - s_{n+2})_{n\in\mathbb{N}} = (0)_{n\in\mathbb{N}}$.

We notice that the operation $\bullet$ has the following properties:

$$f \bullet (s + t) = f \bullet s + f \bullet t \tag{2.2}$$

$$f \bullet \mathbf{0} = \mathbf{0} \tag{2.3}$$

$$(f + g) \bullet s = f \bullet s + g \bullet s \tag{2.4}$$

$$(fg) \bullet s = f \bullet (g \bullet s) = g \bullet (f \bullet s) \tag{2.5}$$

$$0 \bullet s = \mathbf{0} \tag{2.6}$$

$$1 \bullet s = s \tag{2.7}$$

for all $f, g \in F[x]$ and $s, t \in F^{\mathbb{N}}$. The proof follows from the simple application of the definition.
Now, let's come back to the examples mentionned above. The last example leads to a reformulation of the property of being a characteristic polynomial in terms of the operation $\bullet$: $c \in F[x]$ is a characteristic polynomial of $s \in F^{\mathbb{N}}$ if and only if $c \bullet s = \mathbf{0}$. Using the above properties, we can state that the set of all characteristic polynomials of a sequence $s \in F^{\mathbb{N}}$, together with the zero polynomial, is an ideal in $F[x]$: if f, g are both characteristic polynomials or zero, then so is $f + g$, and if $r \in F[x]$ is arbitrary, then rc is either zero or a characteristic polynomial, by (2.3),(2.4) and (2.5). This ideal is called the **annihilator** of s and denoted by $Ann(s)$. Since $F[x]$ is a principal ring, then either $Ann(s) = \{0\}$ or there is a unique monic polynomial $m \in Ann(s)$ of least degree such that $< m >= \{rm : r \in F[x]\} = Ann(s)$. This polynomial is called the **minimal polynomial** of s and divides any other characteristic polynomial of s. We denote it by m_s. If s is not linearly recurrent, then $Ann(s) = \{0\}$, and we set $m_s = 0$. The degree of m_s is called the **recursion order** of s. To sum up, we have the following equivalences for $f \in F[x]$ and $s \in F^{\mathbb{N}}$:

$$c = 0 \text{ or } c \text{ is a characteristic polynomial of } s \Leftrightarrow c \bullet s = \mathbf{0}$$

$$\Leftrightarrow c \in Ann(a) \Leftrightarrow m_s | c$$

[5] the zero sequence will be denoted in this document by $\mathbf{0}$

$$s \in F^{\mathbb{N}} \text{ is linearly recurrent} \Leftrightarrow \exists c \in F[x]\backslash\{0\} : c \bullet s = \mathbf{0}$$

$$\Leftrightarrow Ann(s) \neq \{0\} \Leftrightarrow m_a \neq 0$$

Example 17. *1. Any polynomial annihilates the zero sequence, by (2.3). Thus* $Ann(\boldsymbol{0}) = F[x]$ *and* $m_{\boldsymbol{0}} = 1$.

2. The minimal polynomial of the Fibonacci sequence is $m_s = x^2 - x - 1$. *This is because the polynomial is irreducible over* $\mathbb{Q}$ *(its roots* $(1 \pm \sqrt{5})/2$ *are irrational), and hence no proper divisor of* m_s *annihilates* s.

The above definition of the minimal polynomial gives rise to a precise description of the least period in the special case where the minimal polynomial is irreducible.

Theorem 4. *Let* $s = (s_i)_{i\mathbb{N}}$ *be a linearly recurrent sequence over* $\mathbb{F}_q$ *and* $c(x)$ *its minimal polynomial of degree* L*:*

1. *The least period of the sequence* s *is uniquely determined by the order of* $c(x)$.
2. *The least period of* s *divides* $q^L - 1$ *if* $c(x)$ *is irreducile*
3. *The least period of* s *equals* $q^L - 1$ *if* $c(x)$ *is a primitive polynomial.*

Proof. 1. Let T be the least period of the sequence s. Then, $x^T - 1$ is a characteristic polynomial of s,then, by definition of the minimal polynomial, $c(x)$ devides $x^T - 1$, the rest follows from the definition of the period and the order of a polynomial.

2. If $c(x)$ is irreducible, then $F = \mathbb{F}_q[x]/c(x)$ is a field of q^L elements. The order of $c(x)$ is, by definition of the order of a polynomial and of an element in a group, exactely the order of x in the multiplicative group F^*, by Lagrange theorem , this order divides the cardinal of F^* that is $q^L - 1$.

3. if $c(x)$ is a primitive polynomial, then x is a generator of F^* and its order is equal to $q^L - 1$.

□

Theorem 5. *The minimal polynomial of a linearly recurrent sequence can be efficiently computed using the Extended Euclidean Algorithm, provided we have* $2n$ *initial values of the given sequence, where* n *is a bound on the recursion order. It uses* $O(n^2)$ *operations in the field* $\mathbb{F}_q$.

Proof. It is not examinable, but if interested see the following paragraph. □

2.2.2 Computation of the minimal polynomial

In this paragraph, we indicate how to compute the minimal polynomial of a given sequence $s = (s_i)_{i\in\mathbb{N}} \in F^{\mathbb{N}}$, provided that we know an upper bound $n \in \mathbb{N}$ on the recursion order. We define first the concept of the reversal of a polynomial.

Definition 31. *Let $f = f_d x^d + ... + f_0 \in F[x]$ be a polynomial of degree d. The reversal of f is defined as follows :*

$$rev(f) = x^d f(x^{-1}) = f_0 x^d + f_1 x^{d-1} + ... + f_d \in F[x]$$

Lemma 3. *Let $s = (s_i)_{i\in\mathbb{N}} \in F^{\mathbb{N}}$ be linearly recurrent, $h = \sum_{i\in\mathbb{N}} s_i x^i \in F[x]$, the formal power series whose coefficients are the coefficients of the sequence s, $c \in F[x]$ of degree d and $r = rev(c)$ its reversal.*

1. *The following are equivalent:*
   - *c is a characteristic polynomial of s,*
   - *$r.h$ is a polynomial of degree less than d,*
   - *$h = g/r$ for some $g \in F[x]$ with degree less than d.*
2. *if c is the minimal polynomial of s, then $d = max\{1 + deg\, g, deg\, r\}$ and $gcd(g, r) = 1$*

Proof. 1. $r.h = (\sum_{j=0}^{d} c_{d-j}x^j)(\sum_{i\in\mathbb{N}} s_i x^i) = \sum_{i=0}^{d-j-1} \sum_{j=0}^{d} s_i c_{d-j} x^{i+j} + \sum_{i=d-j}^{\infty} \sum_{j=0}^{d} s_i c_{d-j} x^{i+j}$

$\sum_{i=d-j}^{\infty} \sum_{j=0}^{d} s_i c_{d-j} x^{i+j} = \sum_{i=j}^{\infty} \sum_{j=0}^{d} s_i c_j x^{i+d-j} = \sum_{i=0}^{\infty} \sum_{j=0}^{d} s_{i+j} c_j x^{i+d} = x^d \sum_{i=0}^{\infty} x^i \sum_{j=0}^{d} s_{i+j} c_j$

2. (a) We note that $deg\, r \leq d$, with equality if and only if x does not devide f, and hence $d \geq max\{1 + deg\, g, deg\, r\}$. Now let $c = m_a$, and s.t $d > max\{1 + deg\, g, deg\, r\}$, then $x \mid c$, $r = rev(c/x)$, and c/x is a characteristic polynomial of degree $d-1$. This contradicts the minimality of c. Thus $d = max\{1 + deg\, g, deg\, r\}$
   (b) Let $u = gcd(g, r)$. Then $c^* = c/rev(u)$ is a polynomial of degree $d - deg\, u$, $r/u = rev(c^*)$, and $(r/u)h = (g/u)$ is a polynomial of degree less than $d - deg\, u$. Hence c^* is a characteristic polynomial of s. By the minimality of c $deg\, u = 0$.

□

If $L \in \mathbb{N}$ is an upper bound on the recursion order of s, then we can compute m_s by solving the Padé approximation problem :

$$h \equiv \frac{u}{v} \bmod x^{2n},\ x \text{ not devide } v,\ deg\, u < n, deg\, v \leq n,\ gcd(u, v) = 1 \quad (2.8)$$

- By Lemma 3, $(u, v) = (g, r)$ is a solution to equation 2.8.
- This solution is unique and can be computed with the Extended Euclidean Algorithm.

This leads to the following algorithm:

Algorithm 5. Minimal polynomial for $s \in F^{\mathbb{N}}$
Input: An upper bound L on the recursion order and the first $2L$ entries s_0,...$s_{2L-1} \in F$ of a linearly recurrent sequence $s \in F^{\mathbb{N}}$.
Output: The minimal polynomial $m_s \in F[x]$ of s.

1. $h \leftarrow s_{2L-1}x^{2L-1} + ... + s_1 + 0$
   call *the Extended Euclidean Algorithm to compute $u,\ v \in F[x]$ s.t $v(0) = 1$ and equation 2.8 holds.*
2. $d \leftarrow max\{1 + deg\, u, deg\, v\}$
3. ***return*** $rev_d(v)$

Theorem 6. *Algorithm 5 correctly computes the minimal polynomial of a linearly recurrent sequence $(s_i)_{i\in\mathbb{N}}$ of recursion order at most L and uses $O(L^2)$ operations in F.*

Proof. Let $c \in F[x]$ be the minimal polynomial of s. The discussion above implies that $(g, r) = (u, v)$, where g, r are as in Lemma 3. Finally, If $c = rev_k(r)$ for some $k \in \mathbb{N}$, Lemma 3 implies that $k = d$ □

2.3 Implementation of linearly recurrent sequences

This section shows how the generation of linearly recurrent sequences over $\mathbb{F}_2$ can be implemented on a special electronic switching circuit called *linear feedback shift registers*or *LFSRs*

Definition 32. *A* linear feedback shift register (LFSR)*of length L consists of* L stages *numbered* $0, 1, ..., L-1$*, each capable of storing one bit and having one input and one output; and a clock which controls the movement of data. During each time unit, the following operations are performed:*

1. *the content of stage* 0 *is output and forms part of the* output sequence*,*
2. *the content of stage i is moved to stage $i-1$ for each i, $1 \leq i \leq L-1$*
3. *the new content of stage $L-1$ is the* feedback bit s_j

Such LFSR is depicted in figure 2.1. In this way, the new feedback bit is calculated as follows:

$$s_j = (c_1 s_{j-1} + c_2 s_{j-2} + ... + c_L s_{j-L}) \pmod 2 \text{ for } j \geq L$$

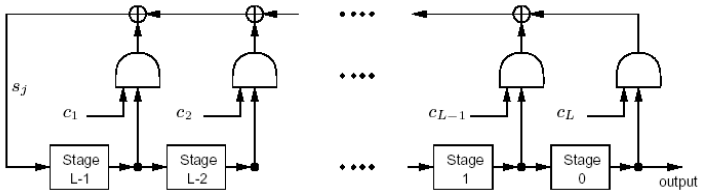

Figure 2.1: Initial state for a LFSR for the sequence s

The LFSR of Figure 2.1 is denoted $< L, c(x) >$, where $c(x) = 1 + c_1x + ... + c_Lx^L \in \mathbb{Z}_2[x]$ [6] is the *connection polynomial* which is obviousely the reversal (see Definition 31) of the characteristic polynomial $\tilde{c}$ of the sequence s ($\tilde{c}(x) = \tilde{c_L}x^L + \tilde{c_{L-1}}x^{L-1} + ... + \tilde{c_1}x + \tilde{c_0} = x^L + c_1x^{L-1} + c_2x^{L-2} ... + c_{L-1}x + c_L$). The initial content of stage i is $s_i \in \{0, 1\}$, for each i, $1 \leq i \leq L-1$, then $[s_{L-1}, ..., s_1, s_0]$ is called the initial state of the LFSR. If n is positive integer, then after n time units, the stage j will contain s_{j+n}. It is therefore natural to call the row vector $s_n = (s_n, s_{n+1}, ..., s_{n+L-1})$ the *nth state vector of the linearly recurrent sequence s (or of the linear feedback shift register).* We saw in the last section that the recursion order L of a linearly recurrent sequence is of crucial importance, in fact, it permits to recover the minimal polynomial of the sequence in polynomial time provided we have the first $2L$ enteries of the given sequence. This recursion order has another name in the electrical engineering world:

Definition 33. *The* linear complexity *of an infinite binary sequence s, denoted $L(s)$, is defined as follows:*

1. *if s is the zero sequence $s = 0, 0, ...,$ then $L(s) = 0$;*
2. *if no LFSR generates s, then $L(s) = \infty$;*
3. *otherwise, $L(s)$ is the length of the shortest LFSR that generates s.*

Similarly, the linear complexity *of a finite binary sequence s^n, denoted $L(s^n)$, is the length of the shortest LFSR that generates a sequence having s^n in its first n terms.*

Before moving to the next section, we summarize in this table the terminology used for linearly recurrent sequences or linear feedback shift registers:

[6] since the coefficients are in $\mathbb{Z}_2$, we have $-1 = 1$

L.R.S	LFSR
characteristic (annhilating) polynomial	reversal of the connection polynomial
minimal polynomial	reversal of the connection polynomial of the shortest LFS
minimal polynomial's order	period
recursion order	linear complexity

2.4 Summary

This chapter introduces the theory of linearly recurrent sequences. We basically recall that thought linearly recurrent sequences over finite fileds are efficiently generated by special switching circuits called linear feedback shift registers, they can be easily predicted via their linear complexity. Thereby, they can not be used directly in cryptographic applications. In practice, and in order to destroy the linear properties that linearly recurrent sequences have, we put a suitable non linear combiner on the output of several linearly recurrent sequences, the output of such combiner is still a linearly recurrent sequence but has large linear complexity and large period which allows it to be candidate for a stream cipher keystream generator.

Chapter 3

Non linear combinations of linearly recurrent sequences

As mentiond in the summary of last chapter, linearly recurrent sequences are widely used in keystream generators because they are well-suited for hardware implementation, produce sequences having large periods and good statistical properties, and they are readily analysed using algebraic techniques. Unfortunately, they are also easily predictible, as the following argument shows. Suppose that the output linearly recurrent sequence s has linear complexity L, or equivalently has a minimal polynomial of degree L. The minimal polynomial can be efficiently computed if we have $2L$ entries of the sequence. Once computing the minimal polynomial, we can use L entries to initialize the LFSR and then generate the remainder of the sequence. An adversary may obtain the required entries by mounting a known or chosen plaintext attack on the stream cipher: if the adversary knows the plaintext subsequence $m_1, m_2, ..., m_n$ corresponding to a ciphertext sequence $c_1, c_2, ..., c_n$, the corresponding keystream bits are obtained as $m_i \oplus c_i, 1 \leq i \leq n$.
We recall that a linearly recurrent sequence should never be used as a keystream generator. Nevertheless, linearly recurrent sequences are desirable because of their very low implementation costs. One solution to destroy their linear properties is the use of a suitable non linear combiner on the output of several LFSRs: The output sequence is still linearly recurrent but has a high linear complexity:

Fact 18. *Suppose that n maximum-length LFSRs* [1]*, whose linear complexities* $L_1, L_2, ..., L_n$ *are pairwise distinct and greater than* 2*, are combined by a nonlinear function* $f(x_1, x_2, .., x_n)$*. Then the linear complexity of the keystream is* $f(L_1, L_2, .., L_n)$*. (The expression* $f(L_1, L_2, .., L_n)$ *is evaluated over the integers rather than over* $\mathbb{Z}_2$ *)*

[1] a maximum-length LFSR is a linearly recurrent sequence whose minimal polynomial is primitive (the upper bound on the period is then achieved)

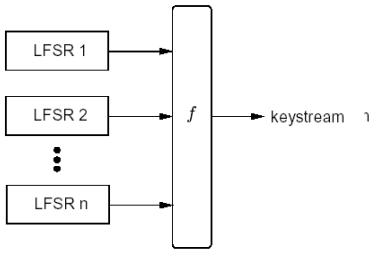

Figure 3.1: f is a nonlinear combining function

Example 18. *(**The Geffe generator**) The Geffe generator is defined by three maximum-length LFSRs whose lengths L_1, L_2, L_3 are pairwise relatively prime, with nonlinear combining function*

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

The key stream generated has period $(2^{L_1} - 1).(2^{L_2} - 1).(2^{L_3} - 1)$ and linear complexity $L = L_1L_2 + L_2L_3 + L_3$.

Bibliography

[GG03] Joachim von zur Gathen and Jrgen Gerhard. *Modern Computer Algebra, Second Edition.* Cambridge University Press, 2003.

[LN86] Rudolf Lidl and Harald Niederreiter. *Introduction to finite fields and their applications.* Cambridge University Press, 1986.