Symulacja w Badaniach i Rozwoju
Vol. 6, No. 3/2015

Anna BOROWSKA
Politechnika Białostocka, Wydział Informatyki
ul. Wiejska 45A, 15-351 Białystok
E-mail: a.borowska@pb.edu.pl

Properties of reducible polynomials

1 Introduction

This paper describes properties of binary polynomials over the finite field F_2. Presented facts were obtained by tests. If p(x) of degree n is reducible, then the set of all polynomials of degree <n contains several groups with respect to multiplication modulo p(x). Properties of these groups are described in Section 3. In Section 4 is presented a polynomial factorization algorithm. In Section 2 we give some well-known facts concerning finite fields GF(p) and GF(2^n), which are essential to understand the presented algorithm. In Section 5 we provide an example of contemporary use of irreducible polynomials in cryptography. They are used, for example, in cryptosystems based on elliptic curves.

2 Finite fields GF(p) and GF(2^n)

Elliptic curves (used in contemporary cryptography) are defined by cubic equations over a given finite field. We give essential facts to define them over the field GF(2^n). The set Z_n ($n \in Z_+$) of integers {0, 1, …, n-1}, together with the arithmetic operations (of addition and multiplication) modulo n, is a commutative ring with a multiplicative identity. Any integer $b \in Z_n$ has a multiplicative inverse if and only if b is relatively prime to n. Thus, in the set Z_p ($p \in P$, where P denotes the set of prime numbers) for each nonzero element there exists a multiplicative inverse. The set Z_p together with the arithmetic operations modulo p is a finite field. We denote it by GF(p) or F_p ($p \in P$). However, for a finite field $F=<F, \oplus, \otimes; 0, 1>$ we denote the multiplicative group by $F^*=<F-\{0\}, \otimes; 1>$.

The field GF(2^n) (also denoted F_{2^n}) consists of 2^n polynomials of degree <n over the field F_2. We identify elements of GF(2^n) with n-bit binary strings of the form $(a_{n-1}, ...,a_1,a_0)$, $a_i \in \{0,1\}$, $i \in \{0,1,...,n-1\}$. A generator of $F^*_{2^n}$ is an element g such that powers g^k (k=0, …, 2^n-2) determine all the elements of $F^*_{2^n}$.

Arithmetic operations on elements of GF(2^n) follow the ordinary rules of polynomial arithmetic in which coefficients belong to F_2. Additionally, multiplication results are reduced modulo some irreducible polynomial p(x) of degree n. That is, we divide them by p(x) and keep the remainder. For a polynomial f(x), the remainder is defined as r(x) = f(x)mod p(x) (cf. [10]).

$$(a_{n-1},...,a_1,a_0)+(b_{n-1},...,b_1,b_0)=(a_{n-1}+b_{n-1},\ldots,a_1+b_1,a_0+b_0),$$

$$(a_{n-1},...,a_1,a_0)*(b_{n-1},...,b_1,b_0)=(r_{n-1},...,r_1,r_0),$$

where $(r_{n-1},...,r_1,r_0)$ represents the remainder after division the product of two polynomials $a(x)=a_{n-1}x^{n-1}+ ...+a_1x+a_0$ and $b(x)=b_{n-1}x^{n-1}+...+b_1x+b_0$ by p(x).

It is convenient when we can use the polynomial f(x)=x (binary g=(00..10)) to the construction of the field GF(2^n). Multiplication by x can be realized as 1-bit left shift followed by the reduction modulo p(x) (if necessary). But for some irreducible polynomials $p(x) \in F_2[x]$, f(x)=x is not a generator of the cyclic group $F_{2^n}^*$ (e.g. $p(x)=x^n+x^{n-1}+...+x+1$ (for n>2) (cf. [6]), (1001001), (1010111), (1110101)).

Addition (and subtraction) of polynomials (in GF(2^n)) corresponds to a bitwise XOR operation. For multiplication of two elements in GF(2^n) we use the equality $g^r * g^t = g^{(r+t) \bmod (2^n-1)}$. Exponentiation is carried out as follows. Let $a=g^r$, then $a^t = g^{(r*t) \bmod (2^n-1)}$. Division (in GF($2^n$)) by a polynomial $a(x) \neq 0$ can be realized as multiplication by the multiplicative inverse of a(x), i.e. $b/a = b*a^{-1}$, where $a^{-1} = a^{2^n-2}$ (cf. [6]). If all the elements of GF(2^n) are stored on the indexed list A, we can find a multiplicative inverse of A[i] in the position $A[2^n-1-i]$. The above-mentioned facts are taken from publications [6, 3, 9, 7] and [10].

3 Properties of reducible polynomials

This section contains observations and conclusions (concerning the properties of reducible polynomials) based on test results. The set of all binary sequences (except for zero) of length n is denoted by S^n. The sequence of n binary coefficients $(a_{n-1},\ldots,a_1,a_0)$ representing a polynomial f(x) is denoted by f. Additionally (to shorten the notation) a polynomial f(x) will be denoted as f.

The multiplicative group $F_{2^n}^*$ of the field F_{2^n} is cyclic, i.e. there exists $g \in F_{2^n}^*$ such that each element of $F_{2^n}^*$ is an exponent of g ($F_{2^n}^* = \{g^j: 0 \leq j < 2^n-1\}$) (cf. [6]). If 2^n-1 is a Mersenne prime and p(x) of degree n is irreducible, then each element of $F_{2^n}^*$ (except for the identity) is a generator of the group $F_{2^n}^*$ (cf. [9]). If 2^n-1 is not a Mersenne prime, generating a periodic sequence can be applied to check whether f(x)=x is a generator.

The properties of reducible polynomials depending on its factors are described below.

(a) Let a polynomial p of degree n be the product of different irreducible polynomials $p_1, p_2, ..., p_m$ of degrees $n_1, n_2, ..., n_m$ ($n=n_1+n_2+...+n_m$), respectively. Then we can distinguish in the set S^n $2\left\{{m \atop 2}\right\}+1$ [(1)] groups with respect to multiplication modulo p (denoted $*_p$). Let $w=\{w_1, w_2\}$ denote any partition of the set $\{p_1, p_2, ..., p_m\}$ into 2 blocks. Let the set X_{w1} consist of the elements of S^n that are divisible by polynomials of the set w_2 and simultaneously indivisible by polynomials of the set w_1. The set X_{w1} is a group with respect to multiplication modulo p(x) (i.e. the following 4 conditions are fulfilled: closure under the operation $*_p$, associativity of the operation $*_p$, there is an identity in X_{w1}, for each element a in X_{w1} there is an inverse of a in X_{w1}). We denote this group by $(X_{w1},*_p)$. Similarly, the group $X_{w2}=(X_{w2},*_p)$ consists of the elements of S^n that

(1) $S(n,k) = \left\{{n \atop 2}\right\} = 2^{n-1}-1$, n>0, S(n,k) is the notation for Stirling numbers of the second kind.

are divisible by polynomials of the set w_1 and simultaneously indivisible by polynomials of the set w_2. There are $\binom{m}{2}$ of such pairs. Neither X_{w1} nor X_{w2} has to be cyclic. If the identity of the group X_{w1} is $(a_{n-1},...,a_1,a)$, then the identity of the group X_{w2} is $(a_{n-1},...,a_1,b)$, where $a\equiv(1+b) \bmod 2$. If, for instance, $w_1=\{p_1, p_3\}$, whereas $w_2=\{p_2, p_4\}$, then the identity of the group X_{w1} is $e=(p_2*p_4)^k \bmod p$, where k is the order of the largest subgroup of the group X_{w1}. The group $A=(A,*_p)$ consists of the remaining elements of the set S^n. Its identity is e=(00..01). The group A does not have to be cyclic. In order to show that p is a reducible polynomial, it is sufficient to find an identity of any group X_w in the set S^n, i.e. an element g that satisfies the following property

(W1) $g = g^2 \bmod p$ and $g\neq(00..01)$

Elements g that satisfy the property (W1) are denoted as =N.

Example 1. Let a polynomial p of degree n be the product of different irreducible polynomials p_1, p_2, p_3, p_4 of degrees n_1, n_2, n_3, n_4 ($n=n_1+n_2+n_3+n_4$), respectively. 15 groups with respect to the operation $*_p$ can be identified in the set S^n. Let $B_1=\{a\in S^n: p_2p_3p_4|a \wedge \neg p_1|a\}$ (B_1 consists of $2^{n_1}-1$ elements). Then $B_1=(B_1,*_p)$ is a group. The identity of the group is $e=(p_2p_3p_4)^k \bmod p$, where $k=2^{n_1}-1$. This group is cyclic. The group is isomorphic with the multiplicative group $F^*_{2^{n_1}}$ of the field $F_{2^{n_1}}$, where (in proven cases) the function α_1: $B_1\rightarrow F^*_{2^{n_1}}$ (α_1 is the reduction modulo p_1) is an isomorphism. It is a well-known fact that irrespectively of what irreducible polynomial p of degree n we choose, we will obtain (with the accuracy of the isomorphism) the same field (cf. [6]). Let $B_2=\{a\in S^n: p_1p_3p_4|a \wedge \neg p_2|a\}$. Then $B_2=(B_2,*_p)$ is a group. Groups $B_2=(B_2,*_p)$, $B_3=(B_3,*_p)$ and $B_4=(B_4,*_p)$ satisfy similar properties.

Let $C_{12}=\{a\in S^n: p_3p_4|a \wedge \neg p_1|a \wedge \neg p_2|a\}$. Then $C_{12}=(C_{12},*_p)$ is a group. The identity of the group is $e=(p_3p_4)^k \bmod p$, where k is the order of the largest subgroup of the group C_{12}. This group does not have to be cyclic. If the sequence $(a_{n-1},...,a_1,a)$ is the identity of the group C_{12}, then the identity of the group C_{34} is the sequence $(a_{n-1},...,a_1,b)$, where $a\equiv(1+b) \bmod 2$. Let $C_{13}=\{a\in S^n: p_2p_4|a \wedge \neg p_1|a \wedge \neg p_3|a\}$. Then $C_{13}=(C_{13},*_p)$ is a group. The groups $C_{13}=(C_{13},*_p)$, $C_{14}=(C_{14},*_p)$, $C_{23}=(C_{23},*_p)$, $C_{24}=(C_{24},*_p)$, $C_{34}=(C_{34},*_p)$ satisfy similar properties.

Let $D_{123}=\{a\in S^n: p_4|a \wedge \neg p_1|a \wedge \neg p_2|a \wedge \neg p_3|a\}$. Then $D_{123}=(D_{123},*_p)$ is a group. The identity of the group is $e=(p_4)^k \bmod p$, where k is the order of the largest subgroup C_{123}. This group does not have to be cyclic. Let $D_{124}=\{a\in S^n: p_3|a \wedge \neg p_1|a \wedge \neg p_2|a \wedge \neg p_4|a\}$. Then $D_{124}=(D_{124},*_p)$ is a group. Groups $D_{124}=(D_{124},*_p)$, $D_{134}=(D_{134},*_p)$, $D_{234}=(D_{234},*_p)$ satisfy similar properties. If the sequence $(a_{n-1},...,a_1,a)$ is the identity of the group B_1, then the identity of the group D_{234} is the sequence $(a_{n-1},...,a_1,b)$, where $a\equiv(1+b) \bmod 2$. The group $A=(A,*_p)$ consists of the remaining elements of the set S^n, i.e. $A=S^n-\{B_1\cup B_2\cup B_3\cup B_4\cup C_{12}\cup C_{13}\cup C_{14}\cup C_{23}\cup C_{24}\cup C_{34}\cup D_{123}\cup D_{124}\cup D_{134}\cup D_{234}\}$.
The group A does not have to be cyclic. Its identity is (00..01).

Example 2. Let us consider a reducible polynomial p represented by the sequence (110001). The polynomial p is the product of irreducible polynomials p_1 and p_2 represented by $p_1=(111)$, $p_2=(1011)$, respectively. Elements of the set S^5 are given below. Each element of S^5 was raised to the k-th power (for k=2, 3, …, 2^5-2) modulo p.

We distinguished three groups $A=(A,*_p)$, $B_1=(B_1,*_p)$, $B_2=(B_2,*_p)$ (The elements are written in hexadecimal).

A={01,02,04,08,10,11,13,17,1F,0F,1E,0D,1A,05,0A,14,19,03,06,0C,18},
$e=(01)_{16}=(00001)_2$,
B_1={1D,0B,16}, $e=(1D)_{16}=(11101)_2$,
B_2={1C,07,15,09,0E,1B,12}, $e=(1C)_{16}=(11100)_2$,

```
00010(02): 01 02 04 08 10 11 13 17 1F 0F 1E 0D 1A 05 0A 14 19 03 06 0C 18   i=21
11000(18): 01 18 0C 06 03 19 14 0A 05 1A 0D 1E 0F 1F 17 13 11 10 08 04 02
00011(03): 01 03 05 0F 11 02 06 0A 1E 13 04 0C 14 0D 17 08 18 19 1A 1F 10
10000(10): 01 10 1F 1A 19 18 08 17 0D 14 0C 04 13 1E 0A 06 02 11 0F 05 03
00100(04): 01 04 10 13 1F 1E 1A 0A 19 06 18 02 08 11 17 0F 0D 05 14 03 0C
01100(0C): 01 0C 03 14 05 0D 0F 17 11 08 02 18 06 19 0A 1A 1E 1F 13 10 04
00101(05): 01 05 11 06 1E 04 14 17 18 1A 10 03 0F 02 0A 13 0C 0D 08 19 1F
11111(1F): 01 1F 19 08 0D 0C 13 0A 02 0F 03 10 1A 18 17 14 04 1E 06 11 05
01101(0D): 01 0D 02 1A 04 05 08 0A 10 14 11 19 13 03 17 06 1F 0C 0F 18 1E
11110(1E): 01 1E 18 0F 0C 1F 06 17 03 13 19 11 14 10 0A 08 05 04 1A 02 0D
10001(11): 01 11 1E 14 18 10 0F 0A 0C 08 1F 05 06 04 17 1A 03 02 13 0D 19
11001(19): 01 19 0D 13 02 03 1A 17 04 06 05 1F 08 0C 0A 0F 10 18 14 1E 11
00110(06): 01 06 14 1A 0F 13 08           i=7
01000(08): 01 08 13 0F 1A 14 06
01111(0F): 01 0F 06 13 14 08 1A
11010(1A): 01 1A 08 14 13 06 0F
10011(13): 01 13 1A 06 08 0F 14
10100(14): 01 14 0F 08 06 1A 13
01010(0A): 01 0A 17                       i=3
10111(17): 01 17 0A
00001(01): 01                             i=1
//--------
00111(07): 1C 07 15 09 0E 1B 12           i=7
10010(12): 1C 12 1B 0E 09 15 07
01001(09): 1C 09 12 15 1B 07 0E
01110(0E): 1C 0E 07 1B 15 12 09
10101(15): 1C 15 0E 12 07 09 1B
11011(1B): 1C 1B 09 07 12 0E 15
11100(1C): 1C                             i=1
//--------
01011(0B): 1D 0B 16                       i=3
10110(16): 1D 16 0B
11101(1D): 1D                             i=1
```

The group B_2 is isomorphic with $F^*_{2^3}$. Hence (as 2^3-1 is a Mersenne prime) each of its elements (except for the identity (1C)) is of the order 7 and may be its generator. The group B_1 is isomorphic with $F^*_{2^2}$. All its elements (except for the identity (1D)) are of the order 3. α_1: $B_1 \rightarrow F^*_{2^2}$ and α_2: $B_2 \rightarrow F^*_{2^3}$, where α_1 is the reduction modulo (111) and α_2 is the reduction modulo (1011), are isomorphisms. In this case, the group A is cyclic. It consists of 12 elements of the order 21, 6 elements of the order 7, 2 elements of the order 3, and 1 element of the order 1. In order to verify whether the polynomial represented by the sequence (110001) is reducible, it is sufficient to determine that $(1C)=(1C)^2 \bmod p$, $((01)\neq(1C))$ or $(1D)=(1D)^2 \bmod p$, $((1D)\neq(01))$. In case of the polynomial p, finding a maximum of 21 consecutive powers of any element (except for the identity) is sufficient to confirm that the polynomial is reducible, because 2^5-1 is a Mersenne prime and (for any $g \in S^n$) we obtain $g^i=ie$ faster than for $i=2^5$-1, where ie satisfies the property (W1) or $ie=(00001)_2$.

(b) For certain $i \geq 2$ let a polynomial p of degree n be of the form $p=(p_1)^i$, where p_1 of degree n_1 is irreducible. Then, we can partition the set S^n into two subsets: the set A, which together with the operation $*_p$ forms the group $(A, *_p)$ (not necessarily cyclic) and the set of elements g that satisfy the following property

(W0) $g^k \bmod p=(00..0)$ for certain $1<k\leq 2^n-2$.

Elements g that satisfy the property (W0) are denoted as =Z. The identity of the group A is e=(00..01). In order to show that p is reducible, it is sufficient to find any element g=Z in the set S^n. In the proposed algorithm, the smallest element g=Z whose square is 0 is searched (i.e. element $g^2 \bmod p=(00..0)$).

(c) Let a polynomial p of degree n be of the form $p = p_1 \ldots p_t p_{t+1}^{s_1} \ldots p_m^{s_{m-t}}$, where polynomials p_i (i=1, ..., m) are irreducible and pairwise different. Then the set S^n contains elements that satisfy the property (W0) and elements that satisfy the property (W1), among other things. In order to show that the polynomial p is reducible, it is sufficient to find an element that satisfies the property (W0) or the property (W1) in the set S^n.

4 Polynomial factorization algorithm

The algorithm presented below makes use of the properties described in Section 3. The operation used most extensively is squaring polynomials with binary coefficients. This operation was implemented with a minimal cost (see example 3). The presented algorithm uses a database (possibly empty) of previously detected irreducible polynomials. Finding an irreducible polynomial that is a factor of a polynomial p in the database significantly accelerates the process of factorization.

Example 3. In order to square a polynomial p (in ordinary polynomial arithmetic), it is sufficient to insert 0 between each of its two digits, i.e. for p=(11011011101), p^2=(10100010100010101010001).

The algorithm `FReduc()` factors a polynomial p into irreducible polynomials. For this purpose it uses two types of elements g=Z and g=N. These elements are searched in a certain subset of the set S^n among elements ending with the digit 1. The search range (subset) can be changed. In the paper, all elements =Z and =N are within the range. The algorithm finds the first element and stops searching. An element g=Z enables the factorization of a polynomial p into two polynomials, b and c (p=b*c), where b is a polynomial with single factors that appear in p an odd number of times, whereas the polynomial c consists of the remaining factors of p (see example 4(a)). The polynomial b is put on the list `TR` (reducible polynomials with single factors). For c, an element g=Z is searched, which is then put on the auxiliary list `Th`. An element g=N enables the factorization of a polynomial p into two polynomials, gcd and pr2 (p=gcd*pr2), where gcd=GCD(p,g). "GCD" is an abbreviation of the *greatest common divisor*. The polynomial pr2 consists of the remaining factors of the polynomial p. The form of a polynomial g=N ensures that GCD(gcd,pr2)=1. Both polynomials gcd and pr2 are put on the auxiliary list `Th` (see example 4(b)). Irreducible polynomials are put on the list `TN`. Elements from the list `Th` are analysed in a similar manner. These polynomials have lower degrees than p. Therefore, they require fewer number of calculations. Only the elements g=N are necessary for the factorization of elements from the list `TR`.

The algorithm determines that a polynomial p is irreducible when it is not in the database and neither an element g=Z nor g=N is present in the range set for p. Example 4(c) shows the factorization of a polynomial p. A<< means "algorithm searches", A>> means "algorithm determines".

Example 4.

(a) Let $p=11^5*111^4*1011*1101^6$. Then $g=Z=11^3*111^2*1011*1101^3$.
It can easily be seen that $g^2 \bmod p=0$. In ordinary polynomial arithmetic:
$b = g^2/p = 11*1*1011*1$, $c = p/b = 11^4*111^4*1101^6$ and $b*c = p$.

(b) Let $p=11^5*111^4*1011*1101^6$ and
g=N=1011*111011*1011011*10000110101*1101100001,
Then gcd=1011, $pr2=11^5*111^4*1101^6$,

(c) The factorization of a polynomial p according to the algorithm `FReduc()`.

```
FMultiple(1101110010111010110100010111000l101) // p=11^5*111^4*1011*1101^6
 A<< g=Z=10100001110010010001                   // g=Z=11^3*111^2*1011*1101^3
 A>> b=11101(→TR) A>> c=10101000001000001010001010010001
 //  b=11*1011         c=11^4*111^4*1101^6
 A<< g=Z=1110010011011101(→Th)                  // g=Z=11^2*111^2*1101^3
FMultiple(11100100110111101)                    // p=11^2*111^2*1101^3
 A<< g=Z=1011011001                             // g=Z=11*111*1101^2
 A>> b=1101(→TN) A>> c=1010000010001            // c=11^2*111^2*1101^2
 A<< g=Z=1100101(→Th)                           // g=Z=11*111*1101
FMultiple(1001)                                 // p=11*111
 A<< g=N=111
FRedSing(111) pr1=1001 A>> gcd=111(→TN) A>> pr2=11(→TN) // pr1=11*111
FSingle(1011) p=1011(→TN) Th:
TR: , TN: 1101 111 11 1011
```

An implementation of the factorization algorithm (in Cpp language) is presented below.

```
(01) void Polyn::FReduc(String p, int n){String RE="TN: ";
(02) TR=new TStringList(); TN=new TStringList(); Th=new TStringList();
(03) p=FDBase(p,n); Th->Add(p);
(04) while(Th->Count){
(05)   p=Th->operator[](0); Th->Delete(0); FMultiple(p); FDelete();}
(06) while(TR->Count){
(07)   p=TR->operator[](0); TR->Delete(0); FSingle(p);   FDelete();}
(08) for(int i=0; i<TN->Count; i++)RE+=TN->operator[](i)+" ";
(09) delete TR; delete TN; delete Th;
(10) out<<RE;}
//--------
(11) void Polyn::FMultiple(String p){
(12) String g, g2, g2p, gend, b, c; Boolean bl=false;
(13) g="1"+toL("0",(p.Length()-1)/2); gend="1"+toL("0",p.Length()-1);
(14) while(g!=gend&&!bl){
(15)   g=nextBin(g); g2=PMult2(g); g2p=toGFM2(g2,p);
(16)   if(PEq(g,g2p)){
(17)     bl=true; Th->Insert(0,p); FRedSing(g); break;}
(18)   else
(19)     if(PEq(g2p,zero)){
(20)       bl=true;
(21)       if(PInDBase(g)){TN->Add(g); break;}
(22)       b=PDiv(g2,p); if(PInDBase(b))TN->Add(b); else if(b!="1")TR->Add(b);
(23)       c=PDiv(p,b);  if(c=="1")break;
(24)       g="1"+toL("0",(c.Length()-1)/2); gend="1"+toL("0",c.Length()-1);
(25)       while(g!=gend){
(26)         g=nextBin(g); g2p=toGFM2(PMult2(g),c);
```

```
(27)         if(PEq(g2p,zero)){
(28)           if(PInDBase(g))TN->Add(g); else Th->Add(g);
(29)           break;}}}}
(30) if(!bl)TN->Add(p);}
//--------
(31) void Polyn::FRedSing(String g){
(32) String pr1, pr2, gcd;
(33) pr1=Th->operator[](0); Th->Delete(0);
(34) gcd=GCD(g,pr1); pr2=PDiv(pr1,gcd);
(35) if(PInDBase(gcd))TN->Add(gcd); else if(gcd!="1")Th->Add(gcd);
(36) if(PInDBase(pr2))TN->Add(pr2); else if(pr2!="1")Th->Add(pr2);}
//--------
(37) void Polyn::FSingle(String p){
(38) String g, g2, gend; Boolean bl=false;
(39) g="1"+toL("0",(p.Length()-1)/2); gend="1"+toL("0",p.Length()-1);
(40) if(PInDBase(p))TN->Add(p);
(41) else{
(42)   Th->Insert(0,p);
(43)   while(g!=gend&&Th->Count){
(44)     g=nextBin(g); g2=toGFM2(PMult2(g),p);
(45)     if(PEq(g,g2)){bl=true; FRedSing(g);}}
(46)   if(!bl){TN->Add(p); Th->Delete(0);}}
(47) for(int i=0; i<Th->Count; i++){
(48)   TN->Add(Th->operator[](0)); Th->Delete(0);}}
```

The `FReduc()` method of the `Polyn` class factors a polynomial p into irreducible factors. Class `Polyn` contains three lists of strings: `TN` (the list of irreducible factors of a polynomial p), `TR` (the list of reducible polynomials with single factors) and `Th` (the auxiliary list). The `FDBase()` method puts on `TN` divisors of a polynomial p which are present in the database and returns a polynomial indivisible by any of the irreducible polynomials (of degree ≤n) in the database. The `FMultiple()` method finds a polynomial g=Z or g=N for a polynomial p and factors p into two polynomials as described in the comment to the example 4. The range of search for an element =Z or =N is set in the line (13). The `toL()` method returns a sequence of characters given by the first parameter repeated n times (n is the second parameter). The `nextBin()` method determines the next element of the set S^n ending with the digit 1. The `PMult2()` method squares a polynomial g. The `toGFM2()` method reduces a polynomial g defined by the first parameter modulo a polynomial p defined by the second parameter. The method is optimized, i.e. instead of reducing g modulo p, the reduction of a suffix (of the length n) of a polynomial g modulo a certain imprint is performed. The same imprint is used for the reduction of all polynomials with the same prefix. The `PEq()` method checks if two binary sequences have the same values. The `FRedSing()` method (lines 31-36) factors a polynomial pr1 (taken from the beginning of the list `Th`) into two polynomials with the use of a polynomial g=N defined by the parameter (see comment to the example 4). The `PInDBase()` method checks if there exists a polynomial defined by the parameter in the database of irreducible polynomials. The `PDiv()` method returns the quotient of polynomials (without the remainder). The `FMultiple()` method is called until a polynomial p is factored into a certain number of polynomials formed from single irreducible factors. The `FSingle()` method factors a polynomial p that is the product of single irreducible polynomials. First it checks if p is present in the database and if it is not, the method searches next elements g=N and factors p into irreducible polynomials (as described

in the comment to example 4). When the condition in the line (43) is false, the list `Th` consists of irreducible polynomials only. Then the elements of `Th` are appended to `TN`. The `FDelete()` method deletes these elements from the list `TN` that appear on it multiple times and leaves these polynomials on lists `TR` and `Th` that are no longer divisible by any of the polynomials from `TN`.

Example 5. Below, 6 consecutive polynomials (ending in the digit 1) of degree 43 are factored into irreducible polynomials.

```
P1=1100110100101001001000100101000101001010101101=
=111*1011*1010111*1110011*111011001100101*1000111110001,
p2=1100110100101001001000100101000101001010101111=
=11^3*10110111001*110101101111*11110011010001111011,
p3=1100110100101001001000100101000101001010110001=
=11^5*111*10011*100101*110110001*1000110001111010101 1,
p4=1100110100101001001000100101000101001010110011=
=1010100101*1111110001110101110110001010110111 1,
p5=1100110100101001001000100101000101001010110101=
=101001*101010101011*11001010100000001001000000 11
p6=1100110100101001001000100101000101001010110111=
=11*111011*111010111*1110010100101*111000111010111001
```

5 Elliptic curves. Applications of irreducible polynomials

Definition 1. (cf. [5]) *Elliptic curve* over the field $GF(2^n)$ is the set of solutions $(x,y)\in GF(2^n)\times GF(2^n)$ of the equation

$$y^2+xy = x^3+ax^2+b \qquad (b\neq 0) \text{ or} \qquad (1)$$
$$y^2+cy = x^3+ax+b \qquad (c\neq 0) \qquad (2)$$

(where $a, b, c\in GF(2^n)$) together with the unique point O, called the *zero point*.

Let the set $E(GP(2^n))$ be the elliptic curve described by the equation (1). The set together with the operation + specified below is a finite (abelian) group. All arithmetic operations are performed in $GF(2^n)$.

Let $P=(x_P,y_P)$, $Q=(x_Q, y_Q)$ be points on the elliptic curve $E(GP(2^n))$.

- $P + O = P$
- $P + (x_P, x_P+y_P) = O \qquad (-P = (x_P, x_P+y_P))$
- $P + P = R = (x_R, y_R)$, where

  $x_R=\lambda^2+\lambda+a$, $y_R= x_P^2 +(\lambda+1)x_R$, $\lambda = x_P + \frac{y_P}{x_P}$
- If $P\neq Q$ i $P\neq -Q$, then $P+Q = R = (x_R, y_R)$, where

  $x_R=\lambda^2+\lambda+x_P+x_Q+a$, $y_R=\lambda(x_P+x_R)+x_R+y_P$, $\lambda = \frac{y_Q + y_P}{x_Q + x_P}$

Example 6. As an example of using irreducible polynomials, the elliptic curve $E(GP(2^{11}))$ was defined. The curve is described by the equation

$$y^2+xy = x^3+g^4x^2+1 \qquad (3)$$

over the field GF(2^{11}). GF(2^{11}) was constructed with the use of the irreducible polynomial represented by p = (101100111111). The curve has 2116 points. They are shown on the diagram. The polynomial f(x)=x was the generator.

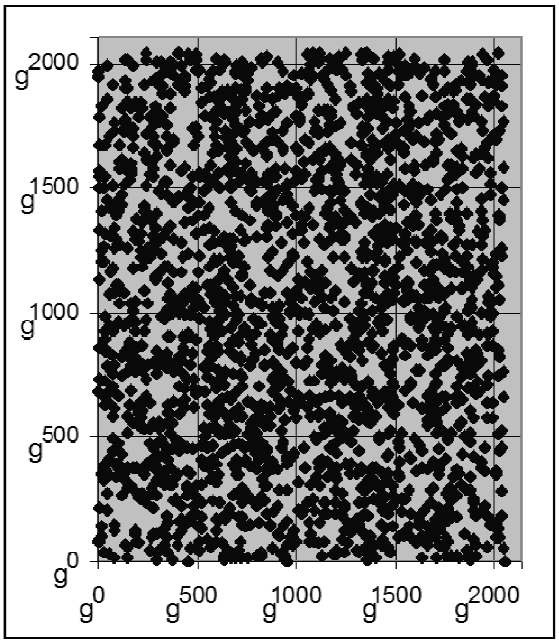

Fig. 1. The elliptic curve E(GF(2^{11}))

Rys. 1. Punkty krzywej eliptycznej

References

1. Białynicki-Birula A.: *Algebra*. PWN, Warsaw 1971
2. Blake J., Seroussi G., Smart N.: *Elliptic Curves in Cryptography*. Cambridge University Press, New York 2001
3. Browkin J.: *Field Theory*. PWN, Warsaw 1977
4. Bryński M.: *Elements of the Galois Theory*. Alfa Publishing House, Warsaw 1985
5. Chmielowiec A.: *Efficient Methods for Generating Secure Parameters of Public-Key Algorithms*. Doctoral dissertation, Warsaw 2012
6. Gawinecki J., Szmidt J.: *Applications of Finite Fields and Elliptic Curves in Cryptography*. WAT, Warsaw 1999
7. Jurkiewicz M., Gawinecki J., Bora P., Kijko T.: *Applications of Elliptic Curves for Designing Secure Algorithms and Cryptographic Protocols*. Cryptology and Cybersecurity, WAT, Warsaw 2014
8. Koblitz N.: *Algebraic Aspects of Cryptography*. WNT, Warsaw 2000
9. Pieprzyk J., Hardjono T., Seberry J.: *Fundamentals of Computer Security*. Springer-Verlag Berlin Heidelberg, Helion, 2003

10. Stallings W.: *Cryptography and Network Security: Principles and Practices*. Helion, Gliwice 2011
11. Stinson D. R.: *Cryptography. Theory and Practice*. WNT, Warsaw 2005

Summary

We consider polynomials p(x) over the 2-element field F_2. If p(x) of degree n is irreducible, then a set of polynomials of degree <n together with operations (of addition and multiplication) modulo p(x) forms the finite field GF(2^n). If p(x) of degree n is reducible, then the set of all polynomials of degree <n contains several groups with respect to multiplication modulo p(x). Properties of these groups are described in Section 3. In Section 4 is presented a polynomial factorization algorithm. Irreducible polynomials are widely used (for instance in cryptography) due to the possibility of an efficient representation of all the elements from GF(2^n) on a fixed number of bits.

Keywords: irreducible polynomials, factorization, cryptography, elliptic curves

Własności wielomianów redukowalnych

Streszczenie

Analizowano wielomiany z jedną zmienną nad ciałem skończonym F_2. Jeśli wielomian p(x) stopnia n jest nierozkładalny, to zbiór wielomianów stopnia <n wraz z operacjami (dodawania i mnożenia) modulo p(x) tworzy ciało skończone GF(2^n). Jeżeli p(x) stopnia n jest rozkładalny, w zbiorze wielomianów stopnia <n można wyróżnić kilka podzbiorów, które wraz z działaniem $*_p$ (mnożenie modulo p(x)) tworzą grupy. Własności tych grup (oparte na wykonanych testach) opisano w sekcji 3. W sekcji 4 zaproponowano algorytm faktoryzacji wielomianów. Wydajność zapisywania elementów GF(2^n) na ustalonej liczbie bitów zachęca do wykorzystywania wielomianów nierozkładalnych na przykład w kryptografii.

Słowa kluczowe: wielomian nieredukowalny, faktoryzacja, kryptografia, krzywe eliptyczne

Acknowledgement:

The research presented in this paper was founded by the BST S/WI/1/2014.