Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 49(63), Fascicola 2, 2004

Analysis of Simple Inversable Functions Defined On Galois Fields for Cryptography Use

Luminita Scripcariu[1], Petrut Duma[2]

Abstract – The private character of the information transmitted on a communication channel or network could be ensured using some cryptography codes. Different encryption techniques performances depend on the processing time and the complexity of the encoding algorithm. We propose a new and advantageous method for symbol permutation, using inversable algebraic function defined on Galois Fields (GF), which minimizes the necessary memory capacity of the encoding algorithm, ensures a great diversity of data and is harder to attack.

Keywords: encryption, permutation, Galois Field

I. INTRODUCTION

Secure communications imply encryption methods use. Different encryption algorithms (DES – Data Encryption Standard, 3DES, MD – Message Digest etc) are known but new cryptography principles are searched for higher diversity and efficiency of the communication system [1].

The coded sequence is deduced as:

$$\bar{c} = E_k(\bar{a}) \qquad (1)$$

We denote:

$\bar{a}$ - the data sequence;

$\bar{c}$ - the coded sequence;

E_k - the encryption function.

Symmetric encryption systems (Fig.1) use secret keys and could be implemented software or hardware as media-access cards (MAC). Data are extracted from the received sequence with the same encryption function:

$$\bar{a} = E_k(\bar{c}) = E_k(E_k(\bar{a})) \qquad (2)$$

Public keys, defined as 'one-way' functions, are applied for asymmetric encryption (Fig.2) and the decoder applies the inversed encryption function:

$$\bar{a} = E_k^{-1}(\bar{c}) = E_k^{-1}(E_k(\bar{a})). \qquad (3)$$

The encryption keys are specified in large tables, which request high-capacity of memory.

The symmetric encryption code could use different methods based on substitution and/or permutation of the symbols. The combined methods have better performances. The coding-rate is about 1:1, so the transmission rate is not affected.

The permutation order is hardly to inverse and to store for long sequences. For example, DES, used on Inetrnet by the SSH (Secure Shell) protocol, is a powerful encryption algorithm (ANSI X3.92), with a 64 bits secret key, which permutes an input word of 64 symbols.

Symbol or character permutation is a frequently used cryptography method, hardly to detect if the sequence length is high [2]. For example, a 20-symbols sequence could be permuted in 2.4 $*10^{18}$ different ways.

Large tables are used for permutation of long sequences and high memory capacity is required. The access time to the memory and the processing time of the encryption algorithm are increased when a longer permutation length is used.

Therefore an algebraic method for permutation could reduce the encryption time and the coding complexity.

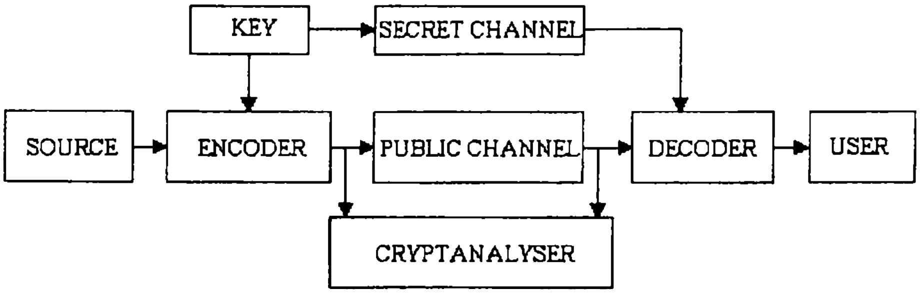

Fig 1 Secret-Key Cryptosystem

[1] Facultatea de Electronică şi Telecomunicaţii Iasi, Romania, Departamentul Telecomunicaţii Bd Copou Nr. 11, Iasi, E-mail lscripca@etc.tuiasi.ro

[2] ***, E-mail pduma@etc.tuiasi.ro

The performances of the encryption algorithm will be improved using algebraic functions to generate the permutation order.

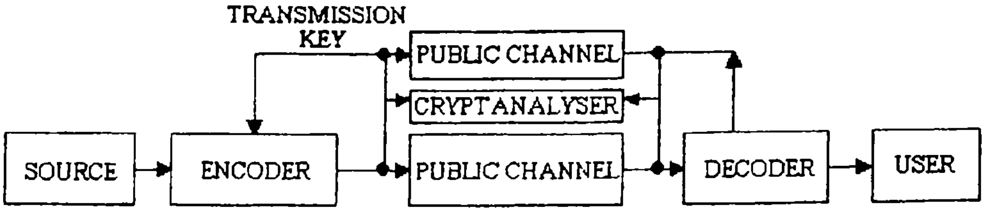

Fig 2 Public-Key Cryptosystem

II. GALOIS FIELDS

The cryptographic methods could be easier implemented using the Galois Field (GF) theory [3].

A GF(2^m) has 2^m elements where m is the length of the binary sequence associated with a symbol of the field.

Internal addition and multiplication operations are defined on the GF.

Let us denote by $\bar{a}$ an element of the GF. It could be written in an equivalent mode as a binary sequence:

$$\bar{a} = \overline{a_{m-1}a_{m-2}..a_0} \qquad (4)$$

or as a polynomial:

$$\bar{a} \leftrightarrow a(x) = a_0 + a_1 x + ... + a_{m-1}x^{m-1} \qquad (5)$$

Addition of two symbols is made modulo-2 bit-by-bit.

The null element (0) does not change the result of an addition.

The opposite element is the element itself.

Multiplication of two elements is defined based on the polynomials product of the two elements and an irreducible m-degree polynomial p(x):

$$c = a \cdot b \Leftrightarrow c(x) = a(x)b(x) \bmod[p(x)] \qquad (6)$$

The unit element (1) does not change the result of a multiplication.

If the product of two elements is equal to one, than they are named inversed elements:

$$a \cdot b = 1 \Rightarrow a^{-1} = b,\ b^{-1} = a. \qquad (7)$$

The substraction and the division are defined based on the opposite and the inversed elements.

$$a - b = a + b \qquad (8)$$

$$a / b = a \cdot b^{-1} \qquad (9)$$

Polynomial functions defined on GF(2^m) could be used as permutation transform for different encryption algorithms:

$$E_{\bar{k}}(\bar{a}) = \sum_{i=0}^{M-1} k_i \bar{a}^i,\ M = 2^m - 1 \qquad (10)$$

The sequence of coefficients from the GF, $\bar{k} = [k_0\ k_1 ... k_m]$, represents the encryption key. Some coefficients combinations do not generate all the symbols of the GF and therefore the decoding process becomes catastrophic. These sequences could not be used as encryption keys because there is no inverse function in these cases. It is necessary to find out which combination of coefficients ensures the permutation of the GF symbols. The original sequence is considered the reference, so the identity function could not be considered an encryption transform.

Other sequences of coefficients make the same permutation and the chances of the cryptanalyzer to find the key are increased. These combinations are called 'weak keys'. Only those keys which uniquely generate a permutation of symbols could be used as 'strong keys'. These keys are classified as symmetric or asymmetric keys.

On GFs, a large number of simple and inversable polynomial functions could be used for encryption:

$$E_{\bar{k}}(\bar{a}) = \bar{c} = k_0 + k_1 \bar{a}^{k_2},\ k_1 \neq 0, k_2 \neq 0, \qquad (11)$$

$$k_2 \neq 2^m - 1,\ (k_0, k_1, k_2) \neq (0,1,1)$$

These functions have only three coefficients which compose the transmission key of an encryption system.

The inversed functions, defined on the same GF, are:

$$E_{\bar{k}}^{-1}(\bar{c}) = \left[k_1^{-1}(\bar{c} + k_0)\right]^q \qquad (12)$$

The integer exponent q is the inverse key component which verifies that:

$$(\bar{a}^{k_2})^q = \bar{a}. \qquad (13)$$

and

$$(k_2 \cdot q) \bmod (2^m - 1) = 1 \qquad (14)$$

The existence of q for any value of k_2 is quaranteed only if m is a prime number and all the GF's elements have the maximum order equal to 2^m-1. In fact, m and 2^m-1 should be simultaneously prime numbers to ensure the maximum number of simple encryption functions defined on a GF(2^m). We deduce some optimum values of m: 3, 5, 7, 13, 17, 19, and 31.

For example, on GF(8) these couples (k_2, q) are:

(2 – 4), (3 – 5), (4 -2), (5 – 3) and (6 – 6).

Other Galois fields, such as GF(16), GF(64), GF(256), do not allow any combination of coefficients for simple polynomial inversable functions because the order of some elements is less then 2^m-1. But there are some values of the coefficients which produce inversable functions and these GFs could be used with few constraints.

III. ENCRYPTION ALGORITHMS

The polynomial inversable functions defined on GFs could be used for symbol permutation.

For optimum m, the number of simple polynomial functions defined on GF(2^m) is equal to the number of the generated permutations (except the identity one) and it is given by:

$$M = 2^m \cdot (2^m - 1) \cdot (2^m - 2) - 1 \tag{15}$$

The coefficients of the 3-coefficients functions could be randomly generated to change the permutation order in a fast way.

The function could be applied directly on the data symbols to permute the bits of a symbol or indirectly, on the position index of each data symbol from a block of 2^m elements, resulting a permutation of symbols.

We call the direct method the **Value Encryption Algorithm** (VEA).

The indirect method is called the **Position Encryption Algorithm** (PEA).

Both algorithms could be applied simultaneously on the data with different encryption functions, defined on different GFs. The last case represents the **Combined Value-Position Encryption Algorithm** (CVPEA) which is robust against the differential attacks.

The direct method could be applied in a fast way with different encryption functions for short sequences of symbols.

VEA has no constraints but PEA is constrained to be applied on a sequence of exactly 2^m elements.

The coefficients of the encryption key could be fast and randomly generated to ensure great value diversity.

For high GFs dimensions the efficiency of the algorithms is increased but the processing time of the algorithm does not become very high because only arithmetical operations defined on GFs are used.

Example:

Let us consider the binary data sequence:

A = [1 0 1 0 1 0 1 1 1 1 0 0 0 0 1 0 0 0 1 1 0 0 1 0]

If the GF(8) is chosen for VEA, the binary data stream is transformed into a sequence of symbols expressed on three bits:

B = [5 2 7 4 1 0 6 2]

A simple inversable function is applied. for the "value encryption" of data:

$$\bar{c} = E_1(\bar{a}) = 1 + 5\bar{a}^2$$

After value encryption, it results:

C1 = [7 3 5 2 4 1 0 3]

For position encryption let us use other functions defined on GF(4):

$$E_2(\bar{a}) = 3\bar{a}^2, \quad E_3(\bar{a}) = 2\bar{a}^2 + 2$$

First function permutes the reference sequence of 4 symbols (0 1 2 3) into (0 3 2 1).

The second function permutes the reference sequence into (2 0 3 1).

Each block of four symbols will be permuted according to a different function. The final symbol sequence is:

C2 = [7 2 5 3 0 4 3 1].

The transmitted bits stream is:

C = [1 1 1 0 1 0 1 0 1 0 1 1 0 0 0 1 0 0 0 1 1 0 0 1].

In this case, if the encryption functions coefficients are not changed, the CVPEA permutes 24 bits.

In a similar way, longer permutation length could be obtained.

If VEA uses a GF with 2^v elements and the PEA uses another GF with 2^p symbols, then the permutation length of the CVPEA is:

$$L = v \cdot 2^p \ (bits) \tag{16}$$

For longer permutation length, the GF dimension of the PEA has to be increased first because the dimension of the GF used for VEA affects harder the encryption algorithm complexity then those used for PEA.

For a higher diversity of the coded sequence, both VEA and PEA must use larger GFs.

The transmission key contains the GFs dimensions and the coefficients of the encryption functions or the parameters of the coefficients generator.

Fast and random generation of the key components ensures a large diversity of the encrypted sequence.

A pseudorandom sequence generator could be used by the VEA for faster permutation of the composing bits of each symbol. In this case, a high dimension GF should be used.

IV. NUMERICAL RESULTS

Different GFs are analyzed to establish the number of permutations obtained with the simple polynomial functions.

Small dimensions of GFs are sufficient if combinations of GFs are used to generate high-length permutations with CVPE which is very efficient, very fast and hard to attack with an acceptable computational complexity.

For example, if both VEA and PEA use functions defined on GF(16) then the minimum permutation length of CVPEA is about 64 bits, but if we change randomly the coefficients of the encryption functions, then longer binary sequences are permuted.

For higher GFs dimensions, longer sequences permutation is made but the computational complexity and the processing time are both increased.

A. *GF(4)*

This is a small algebraic field with 2-bits elements so it is not efficient for value encryption but it could be used by the PEA. Position permutation is made on 4-symbols vectors.

There are 4!-1 = 23 possible permutations without the identity one (0 1 2 3) (Table 1).

All these permutations could be generated using simple polynomial functions with the maximum degree equal to 2:

$$E_{\bar{k}}(\bar{a}) = \bar{c} = k_0 + k_1 \bar{a}^{k_2}; k_1 \neq 0;$$

$$k_2 \in \{1; 2\},\ (k_0, k_1, k_2) \neq (0,1,1)$$

All these functions could be inversed:

$$E_{\bar{k}}^{-1}(\bar{c}) = \left[k_1^{-1}(\bar{c} + k_0)\right]^q = q_0 + q_1 \bar{c}^{q_2}$$

The inverse functions are simple polynomial functions with another set of coefficients.

On GF(4) we use two couples (k_2, q): (1, 1) and (2, 2).

Table 1.

Encryption Key			Permutation	Inverse Permutation	Key Type
k_0	k_1	k_2			
0	1	1	(0,1,2,3)	(0,1,2,3)	Identity
1	1	1	(1,0,3,2)	(1,0,3,2)	S
2	1	1	(2,3,0,1)	(2,3,0,1)	S
3	1	1	(3,2,1,0)	(3,2,1,0)	S
0	2	1	(0,2,3,1)	(0,3,1,2)	A
1	2	1	(1,3,2,0)	(3,0,2,1)	A
2	2	1	(2,0,1,3)	(1,2,0,3)	A
3	2	1	(3,1,0,2)	(2,1,3,0)	A
0	3	1	(0,3,1,2)	(0,2,3,1)	A
1	3	1	(1,2,0,3)	(2,0,1,3)	A
2	3	1	(2,1,3,0)	(3,1,0,2)	A
3	3	1	(3,0,2,1)	(1,3,2,0)	A
0	1	2	(0,1,3,2)	(0,1,3,2)	S
1	1	2	(1,0,2,3)	(1,0,2,3)	S
2	1	2	(2,3,1,0)	(3,2,0,1)	A
3	1	2	(3,2,0,1)	(2,3,1,0)	A
0	2	2	(0,2,1,3)	(0,2,1,3)	S
1	2	2	(1,3,0,2)	(2,0,3,1)	A
2	2	2	(2,0,3,1)	(1,3,0,2)	A
3	2	2	(3,1,2,0)	(3,1,2,0)	S
0	3	2	(0,3,2,1)	(0,3,2,1)	S
1	3	2	(1,2,3,0)	(3,0,1,2)	A
2	3	2	(2,1,0,3)	(2,1,0,3)	S
3	3	2	(3,0,1,2)	(1,2,3,0)	A

The encryption key type is specified:

S – symmetric;
A – asymmetric.

There are 9 symmetric keys different from the identity and 14 asymmetric keys.

For the asymmetric encryption system, it is easy to deduce the inverse polynomial functions coefficients from Table 1. There are 7 couples of direct and inverse keys:

(0,2,3,1) - (0,3,1,2);
(1,2,0,3) - (2,0,1,3);
(1,2,3,0) - (3,0,1,2);
(1,3,0,2) - (2,0,3,1);
(1,3,2,0) - (3,0,2,1);
(2,1,3,0) - (3,1,0,2);
(2,3,1,0) - (3,2,0,1).

We are not interested to store the inverse function coefficients because the inverse algorithm depends only on k_0, k_1 and q equal to k_2.

A decimal identifier of each permutation could be used as the encryption key.

B. *GF(8)*

This field has eight 3-bits symbols of order 7 and it could be used by VEA and PEA.

There are 8!-1 = 40 319 possible permutations without the identity one (0 1 2 3 4 5 6 7) but not all these permutations are generated using simple polynomial functions defined on GF(8).

There are 335 simple 3-coefficients functions:

$$E_{\bar{k}}(\bar{a}) = \bar{c} = k_0 + k_1 \bar{a}^{k_2}, k_1 \neq 0, k_2 \neq 0,$$

$$k_2 \neq 7,\ (k_0, k_1, k_2) \neq (0,1,1)$$

On GF(8) the couples (k_2, q) are: (1, 1), (2, 4), (3, 5), (4, 2), (5, 3) and (6, 6).

Other polynomial inversable functions defined on GF(8) have the following expression:

$$E_{\bar{k}}(\bar{a}) = k_0 + k_1 \bar{a} + k_2 \bar{a}^2 + k_3 \bar{a}^3, k_1 \cdot k_2 \neq 0, k_3 = k_1^6 k_2^2$$

These functions generate different permutations then those obtained with the simple functions but the inverse function is difficult to deduce.

C. *GF(16)*

This field has sixteen 4-bits symbols and it could be used by VEA, PEA or CVPEA.

Only some elements of this field have the maximum order 15.

There are 8 couples (k_2, q) which can be used: (1, 1), (2, 8), (4, 4), (7, 13), (8, 2), (11, 11), (13, 7) and (14, 14).

There are $16! - 1 \cong 21 \cdot 10^{12}$ possible permutations of 16 symbols.

There are 1920 simple polynomial inversable functions defined on GF(16).

The experimental analysis of these functions showed that only the linear and the square functions generate unique permutations. For higher exponents, the permutations are repeated. So we can use 479 functions for permutation on GF(16):

$$E_{\bar{k}}(\bar{a}) = \bar{c} = k_0 + k_1 \bar{a}^{k_2}, k_1 \neq 0,$$
$$k_2 \in \{1, 2\}, (k_0, k_1, k_2) \neq (0, 1, 1)$$

D. *GF(32)*

This field has all 5-bits symbols of order 31 and it is optimum to define permutation functions for CVPEA. There are $32! - 1 \cong 2.6 \cdot 10^{35}$ possible permutations of 32 symbols except the identity one.

29759 permutations are generated using simple polynomial functions defined on GF(32):

$$E_{\bar{k}}(\bar{a}) = k_0 + k_1 \bar{a}^{k_2}, k_1 \neq 0,$$
$$k_2 \neq 0, k_2 \neq 31, (k_0, k_1, k_2) \neq (0, 1, 1)$$

V. CONCLUSIONS

Encryption is the base of any secure communication system. Simple polynomial inversable functions defined on Galois Fields are proposed for symbol permutation. Efficient and fast cryptography algorithms are introduced: Value Encryption Algorithm (VEA), Position Encryption Algorithm (PEA) and Combined Value-Position Encryption Algorithm (CVPEA). Different Galois Fields and the 3-coefficients polynomial functions are analyzed.

REFERENCES

[1] R.E Blahut, *Digital Transmission of Information*, Addison-Wesley Publishing Co., 1990.

[2] A.Menezes, *Handbook of Applied Cryptography*, CRC Press, Inc., 1997.

[3] Scripcariu L., Duma P, "About Some Cryptography Functions Defined on Galois Fields", *Buletinul Institutului Politehnic Iasi, Romania, Sect. III 'Electrotehnica, Energetica, Electronica'*, Tom L(LIV), Fasc.1-2, 2004, pp.65-70.