ADVANCES IN CYBER-PHYSICAL SYSTEMS

Vol. 2, No. 2, 2017

GALOIS FIELDS ELEMENTS PROCESSING UNITS FOR CRYPTOGRAPHIC DATA PROTECTION IN CYBER-PHYSICAL SYSTEMS

Valerii Hlukhov, Andrii Kostyk, Ivan Zholubak, Mohammed Rahma

Lviv Polytechnic National University, 12, S. Bandera str., Lviv, 79013, Ukraine
Authors e-mail*: glukhov@polynet.lviv.ua*

Submitted on 01.12.2017

Abstract: **Currently, elliptic curves are the mathematical basis for digital signature processing. Elliptic curve points processing is based on the performance of operations in Galois field GF(2m) in normal or polynomial bases. Characteristics of multipliers for these bases are different. In this paper, the time complexity of software multipliers for binary Galois fields GF(2m) and fields GF(dn) was investigated. Fields with approximately the same number of elements were investigated. Elements of these fields were represented in a polynomial basis. It is established that the Galois field GF(3т) provides the greatest time complexity of software multiplication, and the prime Galois field GF(P) has the least time complexity. It is also shown that the use of polynomial basis allows, in contrast to the normal basis, to realize larger part of multiplier on FPGA chip.**

Index Terms: **Structural complexity, time complexity, Galois fields, extended fields, field degree, field order, normal basis, polynomial basis, multiplier.**

I. INTRODUCTION

Currently, elliptic curves are the mathematical basis for digital signature processing [1]. In this case, the processing of the points of the elliptic curve is based on the operations in the fields of Galois $GF(2^m)$, $m \leq 1000$ [15], the field elements can be represented in polynomial and normal bases. Hardware implementation of multipliers for such tasks and fields requires high costs of equipment. In [2] it is shown that the hardware multiplication in polynomial and normal bases requires roughly identical hardware and time costs, the program multiplication in a polynomial basis is executed by 1-2 orders of magnitude faster. But the disadvantage of a polynomial basis is the dependence of Galois fields inverse elements computing time on the value of operands [2]. Multipliers can be parallel (including, based on the Guild cells [3]), sequential and parallel-sequential – sectional. For a normal basis, the hardware complexity of serial multipliers allows to implement them on modern FPGAs. But with large values of field order and number of sections it is impossible to implement sectional and parallel multipliers because of their high structural complexity [4], Methods and results of evaluating the structural complexity of a successive multiplier are given in [5], of multi-sectional multipliers – in [6], An estimation based on the use of hardware and software model is presented in papers [7], [8], in [9] it is shown that the structural complexity of the main element of the multiplier for the normal basis of the Galois field GF (2^m), the multiplication matrix, lies within the range $(1/2–3/4)m^2$. The development of methods for assessing structural complexity allowed to develop methods for its reducing [10].

One of the possible problem solution is the transition to the use of Galois fields with a base n greater than 2, first of all, with the base 3 [11]. After changing the fields, the time characteristics of the multiplier can be changed too. In [12] multipliers for extended Galois fields $GF(d^n)$ with bases d greater than 2 and with approximately the same number of elements $d^n \approx 2^m$ are estimated from this point of view. The polynomial basis for representing the Galois field elements and the multiplier with a matrix structure based on modified Guild cells is selected for analysis [12]. It is shown that the time complexity of the multiplier for the field $GF(3^n)$ for the FPGA with the 6-input LUTs is approximately 1.5 times less than the time complexity of the multiplier for the Galois field $GF(2^m)$. In [13] it is shown that the hardware complexity of the triple fields in the polynomial basis has an advantage over binary ones.

Global comparison of the structural complexity of multipliers for expanded Galois fields with representation of their elements in polynomial and normal bases was not performed. The first attempt was to compare parallel multipliers that simultaneously form all product digits for the Galois binary fields $GF(2^m)$ [17]. The paper shows the advantages of a polynomial basis in front of a normal one. But identified advantages were not illustrated by the possibilities of implementing in FPGAs of specific multipliers for normal and polynomial bases. Also, for the polynomial basis, the best field was not defined, in which, unlike the hardware implementation of multiplication, its program implementation has the greatest time complexity. This is important for additional protection against cracking devices that use multipliers.

The purpose of the work is to study the time complexity of program implementations of multipliers for Galois $GF(2^m)$ binary fields and $GF(d^n)$ fields with

approximately equal number of elements in a field and with representation of these elements in a polynomial basis that is necessary to determine the field in which general purpose computers spend the most time for calculations. Also, the goal is to evaluate the possibility of implementing identical multipliers into identical FPGAs for polynomial and normal bases.

II. SECTIONAL MULTIPLIER FOR NORMAL BASIS

The serial Massey-Omura multiplier for multiplication in the normal basis of the elements of the field GF(2^m) (Fig. 1) consists of two operands shift registers RGA and RGB and the multiplication matrix *M*. Sectional multiplier contains several matrices (e.g., M0, ..., M15 in Fig. 2) pipeline and output register file for results accumulation.

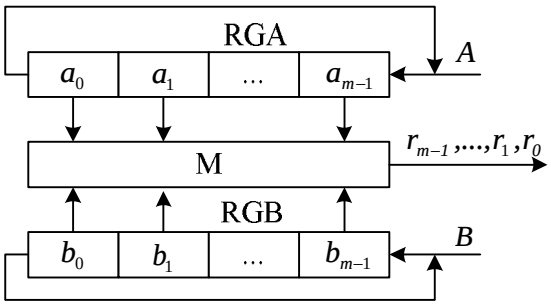

Fig. 1. Massey-Omura multiplier

The bit r_0 of the product R is calculated as $r_0=AMB^T$ (for example,

$$r_0 = a_2b_0 \oplus (a_2 \oplus a_3)b_1 \oplus (a_0 \oplus a_1)b_2 \oplus (a_1 \oplus a_3)b_3$$

according to the calculation circuit Fig. 3).

Each subsequent product bit is calculated after operands one bit rotation.

The structural complexity of the multiplication matrix can be estimated by analyzing its implementation in an imaginary FPGA, each logical element of which (squares in Fig. 4, which corresponds to the scheme of calculation on Fig. 3) can realize the arbitrary function of two variables.

It is possible to estimate the structural complexity of the multiplier topology by the total length L of the joints inside the square domain Sqr in Fig. 4 (in [6] it is shown that the Conv convolution unit makes insignificant contribution to the structural complexity of multiplication matrix): the length of the horizontal connection g_i in the i-th row is $g_i = x_i+1$, where x_i is the column number of the most valid "1" in the i-th line, the vertical length of the connection in the j-th column is equal to $v_i=m+d_i+1$, where d_i is the difference between the number of rows with "1" in the j-th column.

The final expression:

$$L = \sum_{i=0}^{m-1}(g_i + v_i) \approx (1/2\ldots3/4)\ m^2.$$

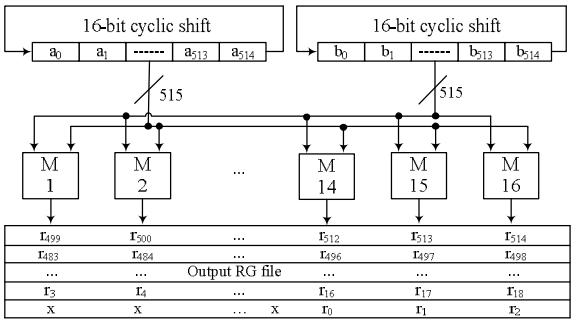

Fig. 2. Sectional multiplier

$$r_0 = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Fig. 3. Computation of the product and the calculation scheme

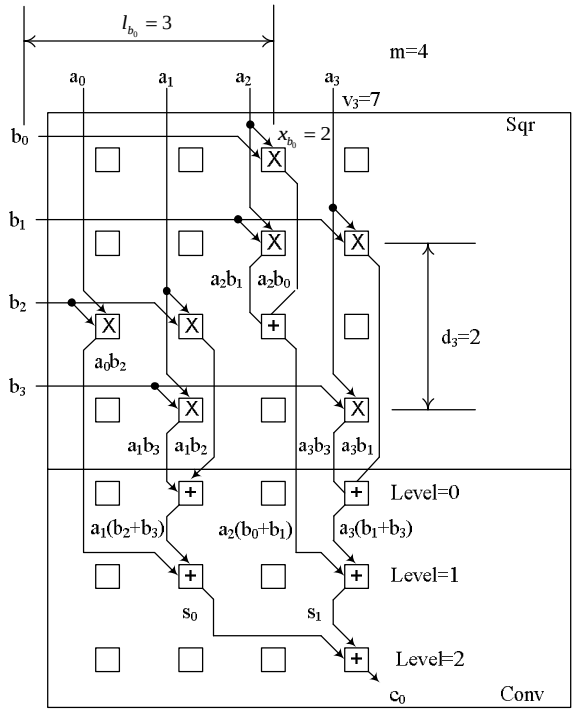

Fig. 4. Topology of multiplication matrix imaginary FPGA

Sectional multiplier (Fig. 5) is formed from serial multipliers (sections), the number of sections n can be from 1 (serial multiplier) to m (parallel multiplier), all sections are of the same size and differ in the cyclic displacement of adders and multipliers along the vertical and horizontal sides in a square region (Fig. 4), which is equivalent to the operands rotation in the calculation of each next bit of the product. For simplicity, we assume that the sections are placed on a crystal in the form of a square matrix with a maximum size for the parallel multiplier $V = q * q$ elements, $q = \lceil \sqrt{m} \rceil$.

For large m (m is directed to 1000) structural complexity of parallel multiplier is equal to $C \approx (k+1)m^3$, $k = ½ \ldots ¾$.

The structural complexity of a parallel multiplier for the normal basis of the Galois binary fields GF(2^m) can be estimated as O(m^3).

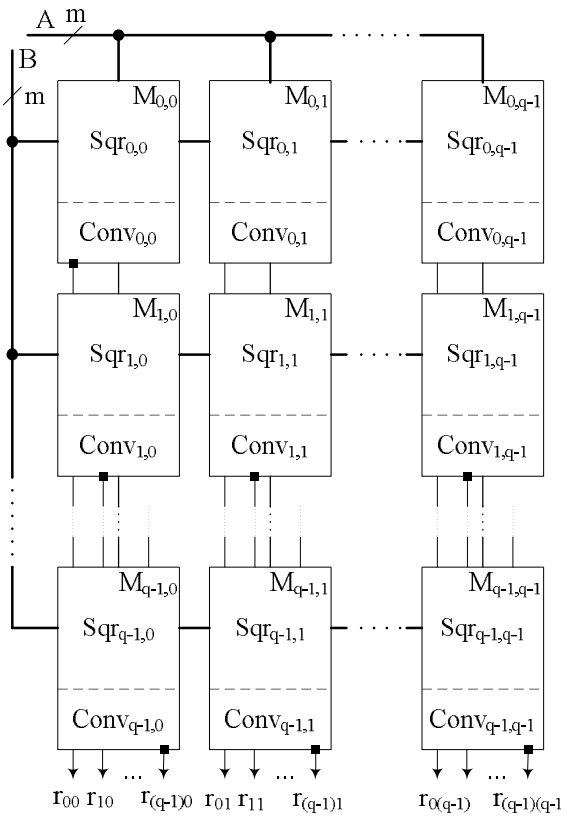

Fig. 5. Imaginary FPGA chip topology of a multi-section multiplier

III. PARALLEL MULTIPLIER FOR A POLYNOMIAL BASIS.

Galois field GF(d^m) multiplier (Fig. 6) is used the modified Guild cells, the detailed scheme of which is shown in Fig. 7. The drawings are marked: p_i – elements of a polynomial that forms a field, $p = \lceil log_2 d \rceil$ – the number of bits in the record of the number d (for the binary Galois fields $d = 2$, $p = 1$).

IV. COMPARISON OF HARDWARE AND STRUCTURAL COMPLEXITIES OF MULTIPLIERS FOR POLYNOMIAL AND NORMAL BASES

A comparison of the hardware complexity of multipliers for polynomial and normal bases is made in [2], where it was shown (Table 1) that the hardware multiplication in the polynomial and normal bases requires roughly identical hardware and time costs. The program multiplication in a polynomial basis is executed by 1–2 orders of magnitude faster.

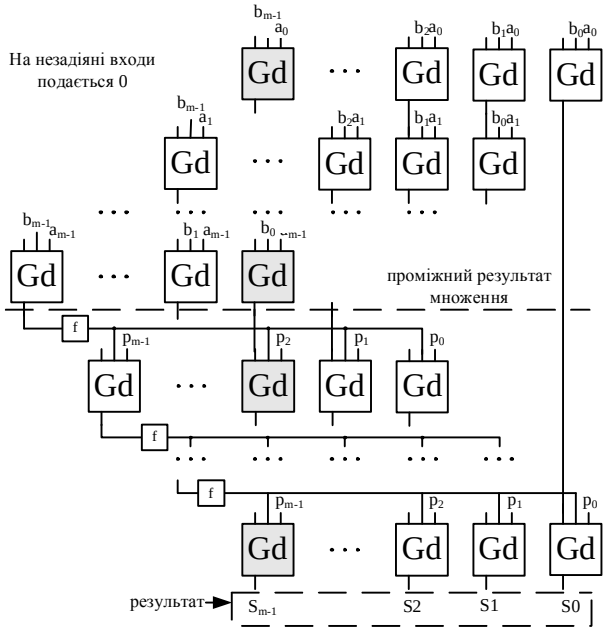

Fig. 6. Multiplier for field elements GF(d^m) using modified Guild cells

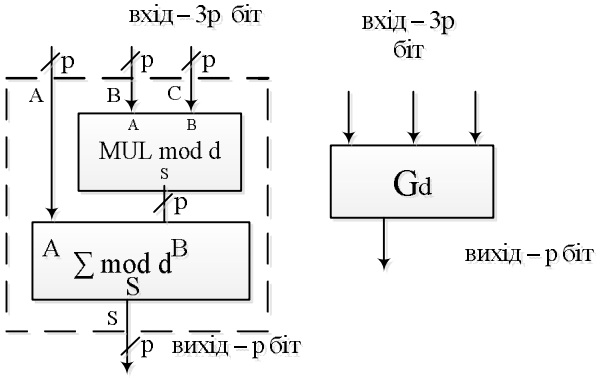

Fig. 7. Modified Guild's Field for the Galois Field GF(d^m)

Table 1

Comparison of multipliers for the field GF(2^{173})

Basis	Number of work cycles	Hardware cost, slices	Hardware cost, LUT	Maximum clock frequency, MHz	A comprehensive index, LUT / MHz
Polynomial	m=173	275	526	146	3,6
Normal	m=173	383	577	169	3,4

The structural complexity of a parallel multiplier for the normal basis of the Galois binary fields GF(2^m) can be estimated as O(m^3) [17].

The structural complexity of a parallel multiplier for the polynomial basis of the Galois binary fields GF (2m) can be estimated as O(m^2) [17].

The results of comparing of the structural complexity of multipliers for polynomial and normal bases are shown in Fig. 8 and Fig. 9.

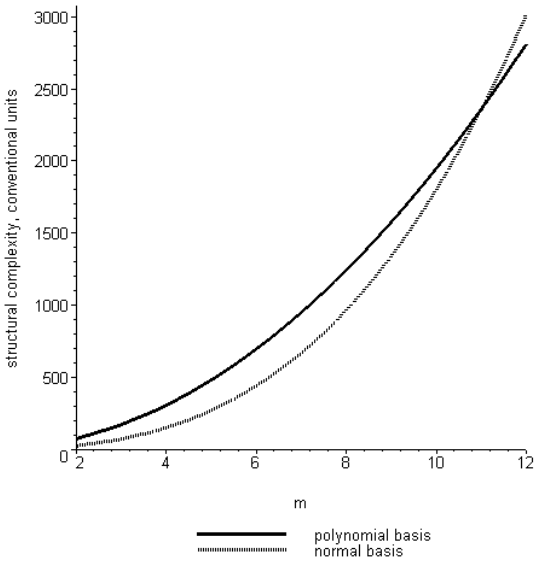

Fig. 8. Structural complexity of multipliers for polynomial and normal bases (m < 12)

For Galois binary fields GF (2^m) with orders m < 12 the multipliers for normal basis have less structural complexity. Multipliers for polynomial basis have a smaller structural complexity with larger orders. For m >> 12 the use of the polynomial basis reduces the structural complexity in comparison with the normal basis in about m times.

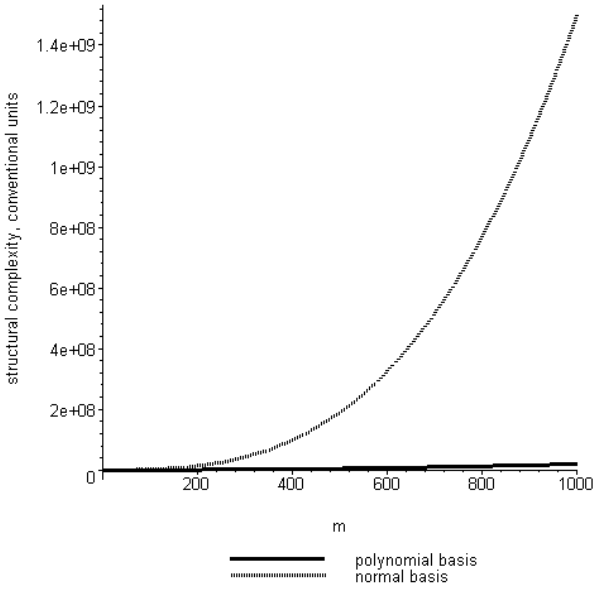

Fig. 9. Structural complexity of multipliers for polynomial and normal bases (m < 1000)

The greater structural complexity of the multipliers for normal basis complicates and makes it impossible to create multi-sectional and parallel multipliers versions [4]. The smaller structural complexity of the multipliers for polynomial basis will allow to create multi-sectional versions with more sections (with a higher level of parallelism and, correspondingly, higher productivity) than those of similar multipliers for a normal basis.

For implementation of hardware multipliers on an FPGA, structural complexity plays a major role, since the hardware and time complexity of multipliers for normal and polynomial bases are approximately the same. As can be seen from the Table 2, hardware costs for the implementation of multi-sectional multipliers for a normal basis may be insignificant.

Table 2

Results of implementation of multisection multipliers in the normal basis

m=998, n=	2	4	8	16
Number of slices (%)		2,396 (11 %)	3,792 (18 %)	6,635 (33%)
Time of implementation, min		3,5	145	102 (**not implemented**)
Structural complexity, conditional units of communication length	559624	1119248	2238496	4476992

The use of the polynomial basis allows us to fully utilize all the resources of the FPGA crystal, since in this case there is no structural complexity limitation.

The following Table 3–6 show the possibility of implementing multipliers for both bases for different types of fields.

Table 3

Implementation of multipliers on the FPGA Virtex 6vlx130t (m = 515)

	Normal basis	The polynomial basis, the number of slices is limited by the number of slices for a normal basis	Polynomial basis with as many slices as possible
m	515	515	515
n	16		
The implemented part of the multiplier, k=n/m	16/515	16/515	0,28
Increase in the number of slices in the implementation of a multiplier for a polynomial basis, times			9
Number of slices	2307	2307	20763
Part of the slices (%)	11	11	99
Estimated structural complexity, imaginary units of length of connections	4242224	8237	74136

Table 4

Implementation of multipliers on the FPGA Virtex 6vlx130t (m=519)

	Normal basis	The polynomial basis, the number of slices is limited by the number of slices for a normal basis	Polynomial basis with as many slices as possible
m	519	519	519
n	16		
The implemented part of the multiplier, k=n/m	0,03	0,03	0,18
Increase in the number of slices in the implementation of a multiplier for a polynomial basis, times			6
Number of slices	3240	3240	19440
Part of the slices (%)	16	16	96
Estimated structural complexity, imaginary units of length of connections	4295296	8276	49657

Table 5

Implementation of multipliers on the FPGA Virtex 6vlx130t (m=998)

	Normal basis	The polynomial basis, the number of slices is limited by the number of slices for a normal basis	Polynomial basis with as many slices as possible
m	998	998	998
n	8		
The implemented part of the multiplier, k=n/m	0,01	0,01	0,04
Increase in the number of slices in the implementation of a multiplier for a polynomial basis, times			5
Number of slices	3792	3792	18960
Part of the slices (%)	18	18	90
Estimated structural complexity, imaginary units of length of connections	2238496	2243	11215

As can be seen, the use of a polynomial basis allows, in contrast to the normal basis, to completely realize on the FPGA a parallel multiplier for the fields GF(2515) and GF(2519) and increase the realized part of the multiplier for the field GF (2998) by 5–12 times.

Table 6

Implementation of multipliers on the FPGA Spartan xc6slx150t (m=998)

	Normal basis	The polynomial basis, the number of slices is limited by the number of slices for a normal basis	Polynomial basis with as many slices as possible
m	998	998	998
n	4		
The implemented part of the multiplier, k=n/m	0,004	0,004	0,05
Increase in the number of slices in the implementation of a multiplier for a polynomial basis, times			12
Number of slices	1896	1896	22752
Part of the slices (%)	8	8	96
Estimated structural complexity, imaginary units of length of connections	1119248	1121	13458

V. TIME COMPLEXITY COMPARISON OF MULTIPLIERS FOR POLYNOMIAL BASIS PROGRAM REALIZATION

One of the methods of hacking the cryptographic information security system is the brute-force method [14], in which the general-purpose computer selects all sorts of keys or passwords until one of them fits. The same operations on the Galois fields elements are performed both during the execution of the hack program and in the hardware crypto processors. For general-purpose computers, one can estimate the time of execution of the main operation, multiplication of the elements of the Galois fields, for extended fields with different bases, but with approximately the same number of elements of the field. The basis for such a check was the field $GF(2^{999})$. This field is recommended by standard [15]. The calculations were made using the Maple 2017 package [16]. During the time complexity check, the execution time of 10.000 multiplication operations over the elements of each $GF(d^m)$ field selected for testing was recorded 5 times. The fields were chosen so that a condition $2^{999} \approx d^m$ (d is a simple integer, m is an integer) was fulfilled for them, that is, fields had approximately the same number of elements. The average time value was calculated after 5 experiments. The relative time complexity was also determined by the ratio of the multiplication time in the field $GF(d^m)$ to the multiplication time in the field $GF(2^{999})$. The times of execution of such number of multiplications with respect to the time of execution of the same number of operations in the binary field $GF(2^{999})$ are shown in the Table 7 and Table 8 and in the Fig. 10. The Table 7 shows the multiplication time in the field GF (2^{999}) with the field polynomial recommended by IEEE 1363 [15] and with the field polynomial that was found using the Maple package.

As can be seen from the Table 7, the timing of operations for these two cases varies insignificantly. Therefore, all the studies were continued for field polynomials that were found using the Maple package. A similar study was also conducted for a simple field $GF(P^1)$, where P is the nearest prime number that is more than 2^{999}. As indicated in the Table 8, software multiplication of triple extended field elements has the longest execution time. It provides hardware cryptoprocessors based on such fields of additional protection against hacking. Software-implemented operations on simple field elements are executed the fastest, that indicates the inappropriateness of cryptographic processors based on such fields.

Table 7

The time complexity of multiplying in a binary field, s

Field Base	1	2	3	4	5	Polynomial
2	5,30	5,42	5,25	5,38	5,27	IEEE
2	5,67	5,77	5,61	5,83	5,55	Maple

Table 8

The time complexity of multiplying in a binary field, s

	Time, conventional units						
Field Degree	1	2	3	4	5	average	relative
1	0,16	0,16	0,14	0,30	0,16	0,18	0,03
2	5,67	5,77	5,61	5,83	5,55	5,68	1,00
3	8,42	8,17	8,34	8,27	8,22	8,28	1,46
5	6,70	6,67	6,75	6,64	6,72	6,70	1,18
7	4,94	4,77	4,75	4,66	4,75	4,77	0,84
11	3,44	3,36	3,34	3,27	3,23	3,33	0,59
13	3,03	2,97	2,97	3,06	2,97	3,00	0,53
17	2,66	2,42	2,63	2,42	2,63	2,55	0,45
19	2,47	2,25	2,44	2,28	2,50	2,39	0,42
23	2,06	2,14	2,27	2,22	2,05	2,15	0,38
29	2,08	1,78	1,92	1,75	1,97	1,90	0,33

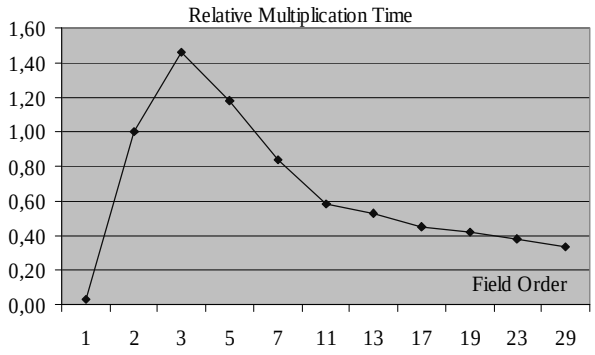

Fig. 10. Relative time complexity of multiplying in a binary field

VI. CONCLUSIONS

The time complexity of software implementations of multipliers for the Galois fields GF (2m) and GF(dn) with approximately the same number of elements in the field and with representation of these elements in the polynomial basis is investigated.

Software multiplication of triple extended field elements has the longest execution time. It provides hardware cryptoprocessors based on such fields of additional protection against hacking. Software-implemented operations on simple field elements are executed the fastest, that indicates the inappropriateness of cryptographic processors based on such fields.

The use of a polynomial basis allows, in contrast to the normal basis, to completely realize on the FPGA a parallel multiplier for the fields GF(2^{515}) and GF(2^{519}) and increase the realized part of the multiplier for the field GF (2^{998}) by 5–12 times.

REFERENCES

[1] DSTU 4145-2002. Informatsiyni tekhnolohiyi. Kryptohrafichnyy zakhyst informatsiyi. Tsyfrovyy pidpys, shcho gruntuyet'sya na eliptychnykh kryvykh. Formuvannya ta pereviryannya [Information Technology. Cryptographic Techniques. Digital Signatures Based on Elliptic Curves. Generation and Verification]. *Derzhavnyy komitet Ukrayiny z pytan' tekhnichnoho rehulyuvannya ta spozhyvchoyi polityky*, Kiyv, Ukraine, 2003 (In Ukrainian).

[2] Hlukhov V. S. Porivnyannya polinomial"noho ta normal"noho bazysiv predstavlennya elementiv poliv Halua. [Comparison of polynomial and normal bases of Galois fields elements presentation.]. *Visnyk Nacional"noho universytetu "L"vivs"ka politexnika" "Komp'yuterni systemy proektuvannya. Teoriya i praktyka"*. Vol. 591, Lviv, Ukraine, 2007, pp. 22–27.

[3] H. H. Guild. Fully iterative fast array for binary multiplication and addition. Electronics Letters, Vol. 5, Issue 12, 12 June 1969, pp. 263 (In English).

[4] V. S. Hlukhov, R. M. Elias, A. O. Mel'nyk. Osoblyvosti realizatsiyi na PLIS sektsiynykh pomnozhuvachiv elementiv poliv Halua GF(2m) z nadvelykym stepenem [Features of the FPGA-based Galois Field GF(2^m) Elements Sectional Multipliers with Extra Large Exponent]. *Komp'yuterno-intehrovani tekhnolohiyi: osvita, nauka, vyrobnytstvo – naukovyy zhurnal, Luts'kyy natsional'nyy tekhnichnyy universytet*. Luts'k, Ukraine, 2013, Vol. 12, pp. 103–106 (In Ukrainian).

[5] Hlukhov V. S., Hlukhova O. V. Rezul'taty otsinky strukturnoyi skladnosti pomnozhuvachiv elementiv poliv Halua [Structural Complexity of Galois Field Elements Multipliers Evaluation Results]. *Visnyk Natsional'noho universytetu "L'vivs'ka politekhnika" "Komp'yuterni systemy ta merezhi"*. Lviv, Ukraine, 2013, Vol. 773, pp. 27–32 (In Ukrainian).

[6] Hlukhov V. S., Trishch H. M. Otsinka strukturnoyi skladnosti bahatosektsiynykh pomnozhuvachiv elementiv poliv Halua [Evaluation of structural complexity of multisection multiplier for Galois field elements]. *Visnyk Natsional'noho universytetu "L'vivs'ka politekhnika" "Komp"yuterni systemy ta merezhi"*. Lviv, Ukraine, 2014, Vol. 806, pp. 27–33 (In Ukrainian).

[7] Sholohon O. Z. Obchyslennya strukturnoyi skladnosti pomnozhuvachiv u polinomial'nomu bazysi elementiv poliv Halua GF(2^m) [Structural Complexity of Galois Field GF(2^m) Elements Multipliers in Polynomial Basis Calculation]. *Visnyk Natsional'noho universytetu "L'vivs'ka politekhnika" "Komp'yuterni systemy ta merezhi"*. Lviv, Ukraine, 2014, Vol. 806, pp. 284–289 (In Ukrainian).

[8] Sholohon Yu. Z. Otsinyuvannya strukturnoyi skladnosti pomnozhuvachiv poliv Halua na osnovi elementarnykh peretvoryuvachiv [Based on Elementary Transducers Structural Complexity of Galois Field Multipliers Evaluation]. *Visnyk Natsional'noho universytetu "L'vivs'ka politekhnika" "Komp'yuterni systemy ta merezhi"*. Lviv, Ukraine, 2014, Vol. 806, pp. 290–295 (In Ukrainian).

[9] Hlukhova O. V., Lozynskyi A. Ya., Yaremkevych R. I., Ihnatovych A. O. Analitychna otsinka strukturnoi skladnosti pomnozhuvachiv elementiv poliv Halua. [Analytical evaluation of Galois field elements multipliers structural complexity]. *Materialy V Vseukrainskoi shkoly-seminaru molodykh vchenykh i studentiv. Suchasni kompiuterni informatsiini tekhnolohii. ACIT'2015*. 22-23 may 2015 year. Ternopil. Ukraine. TNEU, 2015. – pp. 166–167 (In Ukrainian).

[10] Hlukhov V. S., Elias R. Umenshenie strukturnoy slozhnosti mnogosektsionnyih umnozhiteley elementov poley Galua [Galois Fields Elements Multisection Multipliers Structural Complexity Reduction]. *Elektrotehnicheskie i kompyuternyie sistemyi*. – 2015. – No. 19 (95). – pp. 222–226 (In Russian).

[11] M. Zholubak, A. T. Kostyk, V. S. Hlukhov. Osoblyvosti opratsyuvannya elementiv triykovykh poliv Halua na suchasniy elementniy bazieskye y komp'yuternыe systemы [Features of processing Binary Galois fields elements on modern hardware base]. *Visnyk Natsional'noho universytetu "L'vivs'ka politekhnika" "Komp'yuterni systemy ta merezhi"*. Lviv, Ukraine, 2015, Vol. 830, pp. 27–33 (In Ukrainian).

[12] Elias R., Rahma M., Hlukhov V. Multipliers for Galois fields time complexity. *Elektrotehnicheskie i kompyuternyie sistemyi*. – 2016. – No. 22 (98) – pp. 323–327 (In Ukrainian).

[13] Zholubak I. M., Hlukhov V. S. Vyznachennia rozshyrenoho polia Halua GF(dm) z naimenshoiu aparatnoiu skladnistiu pomnozhuvacha [Definition of the extended Galois field GF(dm) with multiplier minimal hardware complexity]. *Visnyk Natsionalnoho universytetu «Lvivska politekhnika» "Informatsiini systemy ta merezhi"/* – Lviv, Ukraine, 2016. Vol. 854. – pp. 63–69 (In Ukrainian).

[14] Password cracking. https://en.wikipedia.org/wiki/Password_ cracking

[15] IEEE 1363-2000. Standard Specifications for Public-Key Cryptography. Copyright © 2000 IEEE. All rights reserved.

[16] Maple User Manual. Copyright © Maplesoft, a division of Waterloo Maple Inc. 2017

[17] V. S. Hlukhov, R. Elias, M. Rahma. Structural Complexity of Multipliers for Galouan Fields Elements in Normal and Polynomial Bases. Electrotechnic and Computer Systems. – Odessa, 2017. Astroprint. – No. 25(101). – pp. 324–331.

Valerii S. Hlukhov is a professor of the Department of Computer Engineering in Lviv Polytechnic National University, Ukraine. He graduated from Lviv Polytechnic Institute with the engineer degree in computer engineering in 1977. In 1991 he obtained his Ph.D. from the Institute of Modeling Problems in Power Engineering of the National Academy of Science of Ukraine. He was recognized for his outstanding contributions into special-purpose computer systems design as a Senior Scientific Researcher in 1995.

He was awarded the academic degrees of doctor of technical sciences in 2013 in Lviv Polytechnic National University. He became a Professor of Computer Engineering in 2014. He has scientific, academic and hands-on experience in the field of computer systems research and design, proven contribution into IP Cores design methodology and high-performance reconfigurable computer systems design methodology. He is experienced in computer data protection, including cryptographic algorithms, cryptographic processors design and implementation. Mr. Hlukhov is an author of more than 100 scientific papers, patents and monographs.

Andrii Kostyk was born in 1989 in Lviv, Ukraine. He received the B.S. degree in computer engineering from Lviv Polytechnic National University in 2010 and M.S degree in 2011. Since 2011 he works at the Computer Engineering Department at Lviv Polytechnic National University. In 2016 he graduated from postgraduate studies. His research interests include

Ivan Zholubak was born in 1991 in Lviv, Ukraine. He received the B.S. degree in computer engineering from Lviv Polytechnic National University in 2013 and M.S degree in system programming from the Lviv Polytechnic National University in 2014 . He has been doing scientific and research work since 2014. Currently, he is a graduate student of the Computer Engineering Department at Lviv Polytechnic National University. His research interests include algorithms of hardware data protection in cryptography.

Mohammed Kadhim Rahma was born in 1979 in Al-Qadisiyyah, Iraq. He received the B.S. degree in "computer engineering" from College of Engineering, Al-Mustansiriya University in (2004-Baghdad, Iraq) and M.S degree in "computer systems and networks" from Lviv Polytechnic National University in 2015-Lviv. He has been doing scientific and research work since 2015. Currently, he is a PhD student of the Computer Engineering Department at Lviv Polytechnic National University. His research interests include algorithms of hardware data protection in cryptography.