Досліджено теоретичну складність виконання квазіквадратичних алгоритмів обчислення кількості точок еліптичних кривих, визначеної над двійковим полем. Проведено експериментальний аналіз таких методів з використанням програмної моделі. Описано необхідність у збільшенні розмірів загальносистемних параметрів національного стандарту електронного цифрового підпису. Наведено пришвидшений метод за часом свого виконання, що може бути використаний для модернізації стандарту

Ключові слова: метод Сато, метод Харлі, порядок еліптичних кривих двійкове поле, слід Фробеніуса

Исследована теоретическая сложность выполнения квазиквадратичных алгоритмов вычисления количества точек эллиптической кривой, определенной над двоичным полем. Проведено экспериментальный анализ таких методов с использованием программной модели. Описана необходимость в увеличении размеров общесистемных параметров национального стандарта электронной цифровой подписи. Приведен ускоренный метод по времени своего исполнения, который может быть использован для модернизации стандарта

Ключевые слова: метод Сато, метод Харли, порядок эллиптической кривой, двоичное поле, след Фробениуса

UDC 004.056
DOI: 10.15587/1729-4061.2017.95194

EXAMINATION AND IMPLEMENTATION OF THE FAST METHOD FOR COMPUTING THE ORDER OF ELLIPTIC CURVE

I. Gorbenko
Doctor of Technical Sciences, Professor
Department of Security of Information Systems and Technologies
V. N. Karazin Kharkiv National University
Svobody sq., 4, Kharkiv, Ukraine, 61022
E-mail: Gorbenkol@iit.com.ua

R. Hanzia
Postgraduate student
Department of Information Security Technologies
Kharkiv National University of Radio Electronics
Nauky ave., 14, Kharkiv, Ukraine, 61166
E-mail: roman.ganzya@gmail.com

1. Introduction

The quality of providing electronic trusted services, primarily in the functioning of electronic government, is impossible without the use of electronic digital signature (EDS). Its application makes it possible to provide users with such services as integrity, authenticity, irresistibility and authenticity of the source of information, by which it was created. In Ukraine, to solve the indicated tasks, national standards are used – DSTU 4145-2002 and DSTU 7564-2014. In addition to the national standards, such harmonized ones are also used as, for example, DSTU ISO/IEC 14888-3 and DSTU ISO/IEC 9796-3. As shown by the analysis, cryptographic strength of the specified standards largely depends on the properties of general system parameters that are employed, the possibilities of expanding their dimensions and operative generation and replacement. The aforementioned becomes especially relevant in the case of emergence of quantum computers and their application for conducting cryptanalytic attacks, for example, to solve the problems on full disclosure and protection from collisions [1].

Therefore, absolutely relevant is a problem task on the prompt generation and application, with the possibility of replacing the tuples, of strong system-wide parameters for EDS. The EDS used in Ukraine are implemented based on cryptographic transformations in the group of points on elliptic curves. At present, the specified task of rapid replacement of general settings and dimension of these parameters is not solved and there is an essential need to consider and resolve it [2].

2. Literature review and problem statement

The national standard of EDS DSTU 4145-2002 was adopted in 2002 and contains system-wide parameters with size up to 431 bits. At that time, it was considered that such parameters would be able to provide the necessary level of the cryptographic transformations stability on elliptic curves (EC) for decades. However, as shown in studies [1], the current state of technology development in the direction of constructing a quantum computer and existence of the Shor's quantum algorithm [3] open up new threats for the existence of modern cryptography, especially asymmetric, including that on EC.

The National Institute of Standards and Technology in the USA (NIST) plans to adopt in the next three to five years the first standard of asymmetric crypto-transformation. However, it is more likely that before its application there will be a "transition period" when both existing and post-quantum EDS are used [2]. Nevertheless, to ensure cryptographic strength of the existing EDS, for example, of those specified in DSTU 4145-2002, it is necessary to increase the dimensions of the system-wide parameters. Such an increase will not significantly improve stability against quantum cryptanalysis. However, for running such an analysis, a true quantum computer is required with a large number

of qubits, which will not be created for many years to come. Therefore, the use of EDS in line with DSTU 4145-2002 with an increased order of the base point to 1000 and more bits can be a guarantee for the cryptographic stability for a significant number of years [4].

The main problem on forming the general systems parameters for use in the groups of points on EC comes down to a complicated, in terms of calculation, task – to compute the number of points on EC [4]. The problem to calculate the number of points on EC is a non-trivial one and, at present, in Ukraine there are no data in the available sources on the execution order and essence of this stage. However, articles [5–13] present an overview and proof of mathematical methods that can be employed to count the number of points on EC.

The algorithm for computing the trace of Frobenius endomorphism, presented in paper [9], is the first p-adic algorithm that was proposed to resolve this type of problems. Earlier, to calculate the order of EC, they used l-adic algorithms [5]. The article [6] can be considered a basis in the direction of calculation the order of EC for binary fields. Author in paper [6] proposed a modification of the method so that it can be used not only for the characteristic 3 and larger. The computational complexity of the method described there is rather high. In particular, article [8] presents a detailed analysis of existing methods for the calculation of zeta-functions and author of the work proposes certain improvements to the algorithm from paper [9]. All the improvements to the method from article [9] are based on the reduction of its computational complexity. In paper [10], authors propose certain mathematical improvements that will be analyzed in detail later in the present study. Thus, it is proposed to use another basis, analytical method for normalization and the application of inverse Frobenius substitution, in contrast to articles [6, 8, 9], where the ordinary substitution was employed. In [12], author proposes a certain improvement to the algorithm based on arithmetic and geometric progression, which is based on using other modular polynomial. Further, in the present study, there will be a detailed analysis of articles [10, 12] and we shall experimentally test the effectiveness of such improvement. Papers [11, 13] describe a variety of solutions to the Artin-Schreier equation for various bases. In [11], for optimal polynomial basis, in [13], for the Gaussian normal basis. A detailed theoretical and practical analysis of these studies will be presented further in our research.

Paper [14] analyzed the international standard ISO/IEC 15946–5, which defined the methods for generating EC. Research in [14] demonstrated that for the finite field $F(2^m)$, significant advantages by the criterion of time complexity (performance speed) are displayed by the methods that are implemented based on the p-adic numbers. In this case, mathematics in the ring of p-adic numbers needed for constructing a program model was described in articles [7, 8, 15]. As the main criterion in the present study, we shall use the criterion of time complexity (performance speed) in the calculation of the number of points on EC and the guarantee of properties of the appropriate system-wide parameters. We shall also subsequently compare efficiency of the received results with the fundamental results that are presented in papers [8, 14].

3. The aim and tasks of the study

The aim of present research is to determine and implement optimal (by performance speed) algorithm for the computation of EC order, which can be applied to increase the base of system-wide parameters of the Ukrainian EDS standard.

To achieve the set aim, the following tasks were formulated and resolved:

– analysis and substantiation of selecting the most promising methods for computing the EC order out of those existing;

– selection by the criteria of minimization of the computational complexity in the algorithm for generating system-wide parameters of high and super-high levels of stability;

– development of a program model and conducting a program simulation, as well as a comparison of theoretical and obtained experimental results.

4. Materials and methods for examining the algorithms to compute the order of elliptic curves

4. 1. Overview of canonical lift of EC

Article [14] presents a general approach to count the number of EC points (EC order) E, assigned by equation:

$$y^2 + xy = x^3 + a_6,$$

with element a_6 over F_q with j-invariant

$$j(E) = a_6^{-1}$$

and $q=p^n$. Such a general approach includes phases of canonical lift of EC, conducting the normalization of coefficients of lifted curve to the base field and the computation of trace of Frobenius endomorphism. Further attention will be paid to the optimal methods of canonical lift of EC that possess quasi-quadratic complexity of their execution or close to such. The stage of normalization is described in more detail in papers [7, 8] and will be described in the subsequent studies on the given subject.

The first algorithm for canonical lift of EC using the p-adic methods was proposed in article [9] (described in detail with certain modifications in paper [8]) and is based on the fulfillment of the Lublin-Serre-Tate theorem.

Theorem. Assume E is not a supersingular elliptic curve over the field F_q. Then, up to isomorphism, there is one curve ε, defined above Z_q, such that:

1. EC equation ε is equal to EC equation E by module 2.
2. $\mathrm{End}(\varepsilon) \cong \mathrm{End}(E)$.

A corollary to this theorem is the following commutative diagram:

$$\begin{array}{ccccccc} \varepsilon_0 & \xrightarrow{\phi_{p,0}} & \varepsilon_1 & \cdots\cdots & \xrightarrow{\phi_{p,d-1}} & \varepsilon_0 \\ \downarrow\psi & & \downarrow\psi & & \downarrow\psi & \downarrow\psi \\ E_0 & \xrightarrow{\varphi_{p,0}} & E & \cdots & \xrightarrow{\varphi_{p,d-1}} & E_0, \end{array} \quad (2)$$

where ε_i is the canonical lift E and $\varphi_{p,I}$ is the corresponding lift $\phi_{p,i}$.

Such isogeny of degree 2 allows us to associate modular equations also of the 2 degree j-invariants of canonically lifted curves:

$$\Phi_2(j(\varepsilon), \Sigma(j(\varepsilon))) = 0, \quad (3)$$

where $\Phi_2(X, Y)$ is the symmetric bivariate polynomial:

$$\Phi_2(X,Y)=X^3+Y^3-X^2Y^2+2^4\cdot 3\cdot 31\left(X^2Y+XY^2\right)-$$
$$-2^4 3^4 5^3\left(X^2+Y^2\right)+3^4 5^3\cdot 4027XY+$$
$$+2^8 3^7 5^6\cdot(X+Y)-2^{12}3^9 5^9. \quad (4)$$

A more detailed description of the canonical lift is given in articles [8, 9]. In the following chapters, we shall demonstrate the application of basic properties of canonical lift, as well as their evolution to rapid counting of the number of points on EC.

4. 2. The Satoh-Skjernaa-Taguchi method (SST)

In the beginning of counting the number of EC points, j(E) in F_q/F_4 is known and it is required to compute j(ε) according to (3). The value of j(ε) can be calculated by receiving in turns one by one bit of the entire magnitude: assume that we know J as:

$$J=j(\varepsilon)\bmod 2^k,$$

then we can record computation:

$$j(\varepsilon)=J+2^k e$$

and insert it formally into equation:

$$\Phi_2(j(\varepsilon),\Sigma(j(\varepsilon)))=0,$$

hence, we obtain equation e by module 2. Thus, receiving one bit when approximating to j(ε) [8].

In a general case, this method can be used for different characteristics of finite fields, but in the present study we shall carry out the adaptation towards extending the finite fields of characteristic 2. This is due to the existence of acting standard on the electronic digital signature based on elliptic curves DSTU 4145-2002. In a general case, expressions of type:

$$j(\varepsilon)=J+2^k e,$$

take the form:

$$j(\varepsilon)=J+p^k e$$

and others on the analogy.

For a more accurate tuning, let us take its decomposition:

$$\Phi_2(j(\varepsilon),\Sigma(j(\varepsilon)))=0,$$

in a Taylor series:

$$0=\Phi_2(j(\varepsilon),\Sigma(j(\varepsilon)))=\Phi_2(J+2^k e,\Sigma(J+2^k e))=$$
$$=\Phi_2(J,\Sigma(J))+2^k e*\frac{\partial\Phi_2}{\partial X}(J,\Sigma(J))+$$
$$+2^k e*\frac{\partial\Phi_2}{\partial Y}(J,\Sigma(J))+2^{2k}r(e,\Sigma(e)), \quad (5)$$

with such element r that $r\in Z_q[X,Y]$.

In this equation:

$$\Phi_2(j(\varepsilon),\Sigma(j(\varepsilon)))=0,$$

by module 2 in degree k, and hence we can divide expression by 2^k and obtain ratio e by module 2. In addition, from the Kronecker relationship:

$$\Phi_p(X,Y)=(X^p-Y)(X-Y^p), \quad (6)$$

it follows that:

$$\frac{\partial\Phi_2}{\partial X}(J,\Sigma(J))+2^k=0,$$

$$e*\frac{\partial\Phi_2}{\partial Y}(J,\Sigma(J))\neq 0,$$

module 2. Finally, since $\Sigma(e)=e^2 \bmod 2$, we obtain:

$$e^2\equiv-\frac{\Phi_2(J,\Sigma(J))}{2^k\frac{\partial\Phi_2}{\partial Y}(J,\Sigma(J))}\bmod 2. \quad (7)$$

Taking the unique root of power 2 ($e\in F_q$) leads to a better approximation of j(ε) that is assigned by:

$$J+2^k e\equiv j(\varepsilon)\bmod 2^{k+1}.$$

In paper [10], authors proposed a different approach to solve such problem (difficult in terms of computational complexity) as finding the root for the situation of degree 2.

To avoid having to find the root, it was proposed to replace the system of equations in (2) with such $J=j(\varepsilon)$ by module 2 with:

$$\Phi_2(\Sigma^{-1}(J),J)=0 \text{ and } J\equiv j(\bar{E})\bmod 2. \quad (8)$$

In this case, expression in (7) can be converted into another form and thus avoid having to calculate the root:

$$e\equiv-\frac{\Phi_2(\Sigma^{-1}(J),J)}{2^k\frac{\partial\Phi_2}{\partial Y}(\Sigma^{-1}(J),J)}\bmod 2. \quad (9)$$

It is necessary to note that:

$$\Phi_2(\Sigma^{-1}(J),J)\equiv 0\bmod 2^k,$$

and we only need to compute the inversion:

$$\frac{\partial\Phi_2}{\partial Y}(\Sigma^{-1}(J),J),$$

by module 2. The only problem that remains is the calculation of inverse Frobenius substitution.

Authors of the national standard of electronic digital signature DSTU 4145-2002 propose to use the representation when $f(x)\in\mathbf{Z_q}[x]$ is used as a module. It means that such a polynomial is the discharged one, irreducible to $F_p[x]$ of degree n. Using such a representation makes it possible to effectively perform algebraic operations in a ring, especially the operation of taking by the module. However, for the operation, which is necessary to compute (7) or (9), in particular calculation of Frobenius substitution value, it is very slow.

To reduce computational complexity when calculating the Frobenius substitution, authors the SST method in article [10] proposed to use different representation of the ring of p-adic numbers. Such representation makes it possible to significantly accelerate computing the Frobenius substitution, but requires certain precomputations.

The essence of using alternative representation is in the following. Assume $\bar{\theta}\in F_q$ is zero in f(x)mod p, in other

words, $F_q = F_p[\bar{\theta}]$. It is obvious that $\bar{\theta}$ is the (q−1)th roots of unity because $\bar{\theta} \in F_q$. Assume θ is the Teichmuller lift from $\bar{\theta}$, in other words, the (q−1)th root of unity in Z_q and $\bar{\theta} \equiv \theta \bmod p$, then we define $m(x) \in Z_p[x]$ as a formative polynomial. In this case, m(x)=f(x) by modulo 2, and hence it follows that it is possible to assign a ring of p-adic numbers (Q_q as $Q_q[x]/m(x)$) [8]. More details on the procedure for calculating the Teichmuller module can be found in [7].

Such a representation of the ring allows us to reduce the computation of Frobenius substitution. Since $\Sigma(\theta)=\theta^p$ by module 2 and $\Sigma(\theta)$ is the (q−1)th root of unity in Z_q, the authors conclude that:

$$\Sigma(\theta) \equiv \theta^p \bmod p.$$

Therefore, efficient computation of the Frobenius substitution is reduced to:

$$\sum\left(\sum_{i=0}^{n-1} a_i\theta^i\right) = \sum_{i=0}^{n-1} a_i\theta^{ip}, \quad (10)$$

the result of the calculation in this case should always be reduced by module m(x). For a binary field, characteristic p takes value 2.

Returning to (9), in particular to the problem on computing the inverse Frobenius substitution and employing new presentation of the ring, such computation takes the following form:

$$\sum{}^{-1}\left(\sum_{i=0}^{n-1} a_i\theta^i\right) = \sum_{j=0}^{p-1}\left(\sum_{0\le pk+j<n} a_{pk+j}\theta^k\right)C_j(\theta), \quad (11)$$

where

$$C_j(\theta) = \Sigma^{-1}(\theta^j) = \theta^{jp^{n-1}}.$$

This value can be precomputed (as well as the value of the Teichmuller module). Thus, if we perform all the necessary precomputations, then finding the value of inverse Frobenius substitution, in terms of computational complexity, will be simple. Interpretation of formula (11) to the binary field will take the following form:

$$\sum{}^{-1}\left(\sum_{i=0}^{n-1} a_i\theta^i\right) = \sum_{0\le 2k<n} a_{2k}\theta^k * \sum_{0\le 2k+1<n} a_{2k+1}\theta^k * C_1(\theta). \quad (12)$$

For a binary field, actually, calculating the inverse Frobenius substitution does not require a large amount of computations, and precomputations come down to finding only one element:

$$C_1(\theta) = \theta^{2^{n-1}},$$

that is also a trivial task.

If we take an idea to use in particular the inverse Frobenius substitution and, accordingly, other space, and combine with the Satoh algorithm, then it is possible to represent the idea of the SST method ("grey" method) as follows:

Algorithm. Lift_J_Naive
Input: j–invariant $j \in F2^n \backslash F2^2$ and accuracy N
Output: $J \in Z_q, J \equiv j \bmod 2$ and $\Phi_2(\Sigma^{-1}(J), J) \equiv \bmod 2^N$.

1. $d \equiv \left(\frac{\partial \Phi_2}{\partial Y}(\Sigma^{-1}(j), j)\right)^{-1} \bmod 2;$
2. $y \equiv j \bmod 2;$
3. for $i = 2$ to N do
3. 1. $x \equiv \Sigma^{-1}(y) \bmod 2^i;$
3. 2. $y \equiv y - d * \Phi_2(x,y) \bmod 2^i;$
4. return y. (13)

Substitution of lift step of the j-invariant with algorithm (13) in the Satoh algorithm will yield the acceleration of computing the EC order by about 5 times. Authors of the SST method pointed that at every step of such algorithm (13), it is necessary to recalculate $\Phi_2(x,y)$, though values of x and y at step i+1 are very close to the values that are used at step 1. Therefore, we can compute:

$$y \equiv j(E) \bmod 2^W$$

and

$$x \equiv \Sigma^{-1}(y) \bmod 2^W$$

for certain W. Assume $m \ge 1$ and $i \ge 0$:

$$\Phi_2(x + 2^{mW+i}\Delta x, y + 2^{mW+i}\Delta y,) \equiv \Phi_2(x,y) +$$
$$+2^{mW+i}\left(\frac{\partial \Phi_2}{\partial X}(x,y)\Delta x + \frac{\partial \Phi_2}{\partial Y}(x,y)\Delta y\right) \bmod 2^{(m+1)W}, \quad (14)$$

so it remains to find out the values:

$$\frac{\partial \Phi_2}{\partial X}(x,y) \text{ and } \frac{\partial \Phi_2}{\partial Y}(x,y).$$

Thus, it is possible to use the indicated equation to compute:

$$\Phi_2(x + 2^{mW+i}\Delta x, y + 2^{mW+i}\Delta y),$$

from value $\Phi_2(x,y)$ while i<W. This idea is represented in the following algorithm, which conditionally consists of two parts: in the first, we compute $y \equiv j(E) \bmod 2^W$ using algorithm (13), and in the next, we use expression from (14). Such expression is used to incrementally update the value of $\Phi_2(x,y)$ without its recalculation at each step [10].

Algorithm. Lift_J
Input: j–invariant $j \in F2^n \backslash F2^2$ and accuracy N
Output: $J \in Z_q, J \equiv j \bmod 2$ and $\Phi_2(\Sigma^{-1}(J), J) \equiv \bmod 2^N$.

1. $d \equiv \left(\frac{\partial \Phi_2}{\partial Y}(\Sigma^{-1}(j), j)\right)^{-1} \bmod 2;$
2. $y \equiv j \bmod 2;$
3. for $i = 2$ to W do
3. 1. $x \equiv \Sigma^{-1}(y) \bmod 2^i;$
3. 2. $y \equiv y - d * \Phi_2(x,y) \bmod 2^i;$
4. $x \equiv \Sigma^{-1}(y) \bmod 2^W;$
5. $D_x \equiv \frac{\partial \Phi_2}{\partial X}(x,y) \bmod 2^W;$
6. $D_y \equiv \frac{\partial \Phi_2}{\partial Y}(x,y) \bmod 2^W;$
7. for $m = 1$ to $\lfloor (N-1)/W \rfloor$ do (15)
7. 1. $x \equiv \Sigma^{-1}(y) \bmod 2^{(m+1)W};$
7. 2. $V \equiv \Phi_2(x,y) \bmod 2^{(m+1)W};$
7. 3. for $i = 0$ to W - 1 do
7. 3. 1. $\Delta y \equiv -d * 2^{-(mW+i)} V \bmod 2;$

7. 3. 2. $\Delta x \equiv \Sigma^{-1}(\Delta y) \bmod 2^{W-i}$;
7. 3. 3. $y \equiv y + 2^{mW+i}\Delta y \bmod 2^{(m+1)W}$;
7. 3. 4. $V \equiv V + 2^{mW+i}(D_x\Delta x + D_y\Delta y) \bmod 2^{(m+1)W}$;
8. return y.

Authors in paper [10] proved that the complexity of calculation in this method, algorithm in (15), for $W \approx n^{\mu/(1+\mu)}$ and $N \approx n/2$ is $O(n^{2\mu+1/(1+\mu)})$. In practice, authors also recommend using such a value for W that is multiple to the word size of central processor. Of course, the above-described complexity does not account for the time to perform precomputations for the polynomial, which assigns field m(x), also the value $C_j(\theta)$ from expression (11). One has to point that the complexity of computing $C_1(\theta) = \theta^{2^{n-1}}$ is $O(n^{2\mu+1})$, demonstrated in [8].

In general, the full version of this algorithm can be shown in algorithm (16). This version is the full version of the SST algorithm and includes the following steps:

1. Computing a polynomial that assigns the field (can be recalculated).

2. Computing element $C_j(\theta)$ for finding the inverse Frobenius substitution (can be precomputated).

3. Lifting the j-invariant of curve to the required accuracy.

4. Finding value c_0 from expression:

$$V_p^*(\tau) = c_0\tau_0 + O(\tau_0^2).$$

This step for the characteristic of field 2 is described in more detail in articles [8, 10].

5. Normalization of coefficients of value from step 4 and finding a trace of Frobenius endomorphism.

6. Obtaining value of the EC order as $\#E(F_q)=1+q\pm t$, where t is the value obtained in the previous step.

Algorithm. SST
Input: Elliptic curve E: $y^2 + xy = x^3 + \overline{c}$ *over* $F2^d$
Output: Number of points on curve $E(F2^d)$

1. $N = \left\lceil \frac{d}{2} \right\rceil + 13;$
2. $M = N - 10;$
3. $j = j(E);$
4. $m(x) = \text{GenTeichmullerModule}(f(x));$
5. $C = \text{GenC}_1(x) \bmod 2^N;$
6. $J = \text{Lift\_J}(j) \bmod 2^N;$
7. $J' = \Sigma^{-1}(J) \bmod 2^N;$
8.

$$Z \equiv -\frac{(J'^2 + 195120J' + 4095J + 660960000)}{8(J'^2 + J'(563760 - 512J) + 372735J + 8981280000)} \bmod 2^N; \quad (16)$$

9. $T' \equiv ((12Z^2 + Z)(J - 1728) - 36) \bmod 2^M;$
10. $CN \equiv (J - (504 + 12096Z)T) \bmod 2^M;$
11. $CD \equiv (T(240T + J)) \bmod 2^M;$
12. $c = \text{Sqrt}(CN / CD) \bmod 2^{M-1};$
13. $t = \text{Norm}(c) \bmod 2^{M-1};$
14. if $t^2 > 2^{d+2}$ then $t \leftarrow t - 2^{N-1};$
15. return $2^d + 1 - t.$

The algorithm presented requires certain explanations and clarifications, in particular $j(E) = 1/\overline{c}$ for the curve equation:

$$E: y^2 + xy = x^3 + \overline{c} \text{ over } F_{2^d}.$$

Function of the Teichmuller module generation:

$$m(x) = \text{GenTeichmullerModule}(f(x)),$$

described in detail in [7, 10], and by function:

$$C = \text{GenC}_1(x),$$

we mean the computation:

$$C_1(\theta) = \theta^{2^{n-1}},$$

to find the value for the inverse Frobenius substitution (12). Function Lift_J(j) in the 6th step of algorithm (16) is described in more detail in (15). The functions of finding a square root in the ring of p-adic numbers used in step 12, as well as other mathematical functions that are defined in the ring of p-adic numbers, are described in detail in [7]. The operation of normalization in the 13th step of the algorithm implies the recovery of trace of Frobenius endomorphism [15]:

$$t \equiv \prod_{i=0}^{n-1} \Sigma^i(c_0) \equiv N_{Q_q/Q_p}(c_0) \bmod q. \quad (17)$$

A fast algorithm for computing the norm and proof of its work was proposed by authors of the SST method in the same paper [10]:

$$N_{Q_q/Q_p}(a) = \exp(\text{Tr}_{Q_q/Q_p}(\log(a))). \quad (18)$$

This algorithm works fast for finite fields of characteristic 2, but for finite fields of other characteristics its computational complexity increases significantly. This is primarily caused by the complexity of calculating the logarithm in the ring of p-adic numbers. It should also be noted that this algorithm in the wording proposed in [10] for the binary fields will work only for the sparse generating polynomials. For the polynomials that form the Teichmuller basis, expression from (18) with the possibility of rapid computation of the logarithm will not work out. This causes the need to transform element *c* after step 12 of algorithm (16) back to the optimal polynomial basis.

4. 3. The modified Satoh-Skjernaa-Taguchi method (MSST)

Paper [12] proposed to combine the idea of arithmetic-geometric method and the Satoh-Skjernaa-Taguchi method. The AGM method was analyzed in detail in article [14]. It is worth noting that for the MSST work, one should use a one shift variation of the AGM method, though in terms of computational complexity, one shift and two shift AGM are almost indistinguishable. However, using the one shift variation implies applying the AGM-sequence $(a_k, b_k)_{k=0}^{\infty}$, in which:

$$a_k \equiv b_k \equiv 1 \bmod 4, \quad a_k \equiv b_k \bmod 8,$$

in the next variant, $\lambda_k \equiv a_k/b_k$, which matches EC:

$$E_{\lambda_k}: y^2 = x(x-1)(x-\lambda_k^2). \quad (19)$$

As each preceding AGM-sequence makes it possible to compute the next one, in other words, such sequence is iterative:

$$(a_{k+1}, b_{k+1}) = ((a_k + b_k)/2, \sqrt{a_k b_k}),$$

then we may represent the iterative function of the one shift AGM-sequence in the form

$$\lambda_{k+1} = \frac{2\sqrt{\lambda_k}}{1+\lambda_k} \text{ [12].}$$

Initialization of the one shift AGM-sequence is performed as follows:

$$\lambda_1 \equiv (1+8c) \bmod 16, \quad (20)$$

where $c \equiv \bar{c} \bmod 2$ is the free EC coefficient. Gaudrì also proves that sequence λ_{k+1} converges similar to a_k/b_k, it is implied that:

$$\lambda_{k+1} \equiv \Sigma(\lambda_k) \bmod 2^{k+3},$$

and if we substitute this value into the value of iterative function of the AGM-sequence, we shall obtain:

$$\Sigma(\lambda_k)^2(1+\lambda_k)^2 - 4\lambda_k \equiv 0 \bmod 2^{k+3}. \quad (21)$$

The above-indicated expression is solved with the use of the main idea of the SST algorithm. Paper [12] presents the solution to this problem through the change of expression for modular polynomial that is used for EC lift (described in (3)).

Assume E(X,Y) is the module expression of AGM, then:

$$E(X,Y) = Y^2(1+X)^2 - 4X = 0. \quad (22)$$

The above representation allows us to use the condition of modular polynomials (Lublin-Serre-Tate theorem) in the form:

$$E_2(X, \Sigma(X)) \equiv 0 \bmod 2^{k+3}.$$

It should be noted that both partial derivatives are equal to zero by module 2 in this expression, which is why it is not possible to directly use the SST algorithm. To eliminate such a problem, [12] proposes the following substitutions:

$$X \leftarrow 1+8X,$$

$$Y \leftarrow 1+8Y$$

and, as a result, we shall obtain a modified modular polynomial for finite fields of characteristic 2 in the form of:

$$\tilde{E} = (X + 2Y + 8XY)^2 + Y + 4XY = 0, \quad (23)$$

that can be solved when X is known and it is not equal to zero by module two, so that $Y = \Sigma(X)$. In this case, the partial derivatives are equal to:

$$\frac{\partial \tilde{E}}{\partial X}(X,Y) = 2(X + 2Y + 8XY)(1+8Y) + 4Y, \quad (24)$$

$$\frac{\partial \tilde{E}}{\partial Y}(X,Y) = (4(X + 2Y + 8XY) + 1)(1+4X),$$

it is not difficult to confirm that each partial derivative by X is indeed equal to zero by module two, and by Y is equal to unity that satisfies the requirements of the SST algorithm [10].

The last thing that is necessary to do is to align the expression from which a Frobenius trace is computed, for the two-shift AGM sequence it takes the following form:

$$Tr\bar{F} = t_k + q/t_k \bmod^{k+3} \quad (25)$$

from

$$t_k = N_{Q_q/Q_p}(a_k / a_{k+1})$$

substituting in this expression the values that were accepted for the one-shift AGM, in other words:

$$\lambda_k \equiv a_k / b_k, \quad a_{k+1} = (a_k + b_k)/2$$

and

$$\lambda_1 \equiv 1+8c,$$

we shall obtain:

$$t_k = N_{Q_q/Q_p}\left(\frac{1}{1+4\lambda_k}\right). \quad (26)$$

Upon presenting the SST method, the algorithm described above does not need special explanations. If one compares two similar methods of SST and MSST in terms of computational complexity, the MSST method is usually works faster. For $W \approx n^{\mu/(1+\mu)}$ and $N \approx n/2$, its complexity is $O(n^{2\mu+0,5})$. This is caused by the fact that the modular polynomial used in the MSST method requires 1 multiplication and 1 squared raising. The SST method needs 3 multiplications and 2 squared raisings (meaning in the ring of p-adic numbers). From the point of view of the spatial complexity, the two methods require similar resources. Spatial complexity for them is $O(n^2)$. Although the constants that are used in the MSST modular polynomial are significantly lower than those in the SST method. The phase of precomputations for two algorithms is the same and requires $O(n^{2\mu+1})$ computing resources for its implementation [8].

4. 4. The Harley method

Article [13] proposed a p-adic algorithm for finding the EC order without precomputations and with computing complexity $O(n^{2\mu}\log n)$. The main idea of this method was applying the method for solving equations by Artin-Schreier, that is, equation of the form:

$$x^p - x + \alpha = 0,$$

with element $\alpha \in F_q$ (by the field characteristic – p). Positive version of Hilbert space claims that such equation has the solution in F_q only in the case when:

$$Tr_{F_q/F_p}(\alpha) = 0.$$

Since:

$$\Sigma(x) = x^p,$$

then the given equation for Z_q takes the form:

$$\Sigma(x) - ax - b = 0, \quad (27)$$

with a, b∈Z_q. Let us define:

$$\Sigma^k(x) = a_k x + b_k,$$

for all k=2,...,n. As

$$\Sigma^n(x) = x,$$

the authors conclude that the above-given equation is assigned as:

$$b_n / (1 - a_n),$$

and for computing a_k, b_k, the authors propose the following formula

$$\Sigma^{k+l}(x) = \Sigma^l(a_k x + b_k) = \Sigma^l(a_k)(a_l x + b_l) + \Sigma^l(b_k). \quad (28)$$

It is the above-given expression that is used to solve the Artin-Schreier equation. Next, to lift EC, a generalized Newton's algorithm is used [13], the very basic idea of lifting remains the same as in the SST algorithm (2). However, [13] proposed these parameters (solving the Artin-Schreier equation and the generalized Newton's algorithm) for the Gaussian normal basis only.

[11] presents a similar variant for solving the Artin-Schreier equation and a different generalized Newton's algorithm that can be used for the optimal polynomial basis and the Teichmuller basis. Assume we have the following problem: given

$$\Phi(X, Y) \in Z_q[X, Y],$$

and it is required to find the root x∈Z_q such as:

$$\Phi(X, \Sigma(X)) = 0. \quad (29)$$

Assume we know that:

$$x^m \equiv x \bmod p^m$$

and let it be

$$\delta_m = (x - x_m) / p^m,$$

then the decomposition in a Taylor series around x_m for p=2 yields:

$$0 = \Phi_2(x, \Sigma(x)) = \Phi_2(x_m + 2^m \delta_m, \Sigma(x_m + 2^m \delta_m)) \equiv$$
$$\equiv \Phi_2(x_m, \Sigma(x_m)) + 2^m \left(\delta_m \Delta x + \Sigma(\delta_m)\Delta y\right) \bmod 2^{2m}, \quad (30)$$

with

$$\Delta x = \frac{\partial \Phi_2}{\partial X}(x_m, \Sigma(x_m)) \bmod 2^m$$

and

$$\Delta y = \frac{\partial \Phi_2}{\partial Y}(x_m, \Sigma(x_m)) \bmod 2^m.$$

Hence, it follows that δ_m has a solution of the following type:

$$-\frac{\Phi_2(x, \Sigma(x))}{2^m} \equiv \delta \Delta x + \Sigma(\delta)\Delta y \bmod 2^m. \quad (31)$$

Assume:

$$k = ord_2(\Delta y),$$

and then if $ord_2(\Delta x) \geq k$ and

$$ord_2(x_m, \Sigma(x_m)) \geq k + m$$

and m>k, we obtain the following equation:

$$\alpha \Sigma(\delta) + \beta\delta + \gamma \equiv 0 \bmod 2^{m-k}, \quad (32)$$

coefficients α, β, γ∈Z_q and α are the unity in Z_q. As the Frobenius substitution maintains the values, then the above-given equation computes $\delta_m \bmod 2^{m-k}$ unequivocally and, in addition:

$$x \equiv x_m + 2^m \delta_m \bmod 2^{2m-k}.$$

We shall assume that there is an algorithm that returns the null value $\delta_{t'}$ from equation (32) with accuracy $t' = \lceil t/2 \rceil$ (half precision), then there is a possibility to employ the same algorithm for computing δ_t with accuracy t (full precision). Let us substitute:

$$\delta_t = \delta_{t'} + 2^{t'} \Delta t$$

we shall receive in (31):

$$\alpha \Sigma(\Delta t) + \beta \Delta t + \frac{\alpha \Sigma(\delta_{t'}) + \beta \delta_{t'} + \gamma}{2^{t'}} \equiv 0 \bmod 2^{t-t'}, \quad (33)$$

that is why, since $t - t' \leq t'$, then it is possible to apply the same algorithm for computing Δt by module $2^{t-t'}$ and, therefore, we may receive the value δ_t. Thus, it gives us a recursive algorithm that allows us to find value (32). If we assume that:

$$ord_2(\Delta x) > ord_2(\Delta y),$$

then it follows that $ord_2(\beta) > 0$ and everything is reduced to solving:

$$\alpha \Sigma(\delta) + \gamma \equiv 0 \bmod 2.$$

And since α is the unity, it allows unambiguously computing δ mod 2. To calculate the Artin-Schreier equation, it is proposed to employ the following algorithm [11].

Algorithm. Artin-Schreier-root
Input: α, β, γ∈Z_q, α∈Z^*_q, ord2(β)>0, accuracy N.
Output: x∈Z_q so as αΣ(x)+βx+γ≡0mod 2^N.

1. if(N == 1) then
1.1. $x \equiv (-\gamma/\alpha)^{1/2} \bmod 2$;
2. else
2. 1. $N' = \lceil N/2 \rceil$;
2. 2. $M = N - N'$;
2. 3. $x' = Artin - Shreier - root(\alpha, \beta, \gamma, N')$;
2. 4. $\gamma' = (\alpha \Sigma(x') + \beta x' + \gamma) / 2^{N'} \bmod 2^M$;
2. 5. $\Delta' = Artin - Shreier - root(\alpha, \beta, \gamma', M)$; (34)

2. 6. $x \equiv x' + 2^{N'}\Delta' \bmod 2^N$;
3. return x.

Paper [8] demonstrates certain optimization regarding computing in step 1. 1. of algorithm 4 (34), in other words, root calculation takes the following form:

$$\left(\sum_{i=0}^{n-1} a_i\theta^i\right)^{1/p} = \sum_{j=0}^{p-1}\left(\sum_{0\le pk+j<n} a_{pk+j}\theta^k\right)C_j(\theta), \qquad (35)$$

where

$$C_j(\theta) = (\theta^j)^{1/p} = \theta^{jp^{n-1}}$$

and version for the binary field takes the form:

$$\left(\sum_{i=0}^{n-1} a_i\theta^i\right)^{1/2} = \sum_{0\le 2k<n} a_{2k}\theta^k * \sum_{0\le 2k+1<n} a_{2k+1}\theta^k * C_1(\theta), \qquad (36)$$

where

$$C_1(\theta) = \theta^{2^{n-1}}.$$

Assessment of the complexity of the above-given algorithm is also provided in article [11] and amounts to:

$$O((nN)^M \log N).$$

The complexity of the algorithm as a whole is based on recursive challenges in steps 2. 3. and 2. 5. of algorithm 5 (34), multiplication and computation of the Frobenius substitution in step 2.4. The complexity of computing the Frobenius substitution is $O((nN)^\mu)$ bit operations [8].

The above-given algorithm allows us to effectively solve equation (33), and to solve the main problem outlined in (30) and reduced to (32) in paper [11], proposes to use a generalized Newton's lift somewhat different from the one described in article [13].

Algorithm. Generalized-Newton-Lift

Input: Modular polynomial $x_0 \in Z_q$, *which satisfies* $\Phi_2(x_0, \Sigma(x_0)) \equiv 0 \bmod 2^{k+1}$ *and* $\left(\frac{\partial\Phi_2}{\partial X}(x_0, \Sigma(x_0))\right) > k$, $k = \mathrm{ord}_2\left(\frac{\partial\Phi_2}{\partial Y}(x_0, \Sigma(x_0))\right)$, *accuracy N.*

Output: $x_N \in Z_q$, $\Phi_2(x_N, \Sigma(x_N)) \equiv 0 \bmod 2^{N+k}$ *and* $x_N \equiv x_0 \bmod 2^{k+1}$.

1. if $(N \le 2k+1)$ then
1. 1. $x \equiv x_0$;
2. else
2. 1. $N' = \lceil N/2 \rceil + k$;
2. 2. $M = N - k$; (37)
2. 3.

$$\Delta x \equiv \frac{\partial\Phi_2}{\partial X}(x', y') \bmod 2^{N'}) / 2^k \bmod 2^M;$$

$$x' = \mathrm{Generalized} - \mathrm{Newton} - \mathrm{Lift}(x_0, N');$$

2. 4. $\gamma' = \Sigma(x') \bmod 2^N$;
2. 5. $V \equiv (\Phi_2(x', \gamma') \bmod 2^N) / 2^{N'} \bmod 2^M$;
2. 6. $V \equiv (\Phi_2(x', \gamma') \bmod 2^N) / 2^{N'} \bmod 2^M$;
2. 7. $\Delta y \equiv \frac{\partial\Phi_2}{\partial Y}(x', y') \bmod 2^{N'}) / 2^k \bmod 2^M$;
2. 8. $\Delta' = \mathrm{Artin} - \mathrm{Schreier} - \mathrm{Root}(\Delta y, \Delta x, V, M)$;
2. 9. $x = x' + 2^M\Delta' \bmod 2^N$;
3. return x.

The complexity of algorithm (37) is similar to algorithm (35). The main complexity is based on the recursive challenges of the algorithm from (35).

Multiplication of two integers that consist of n bits is performed in $O(n^\mu)$ operations, where μ is the constant that defines the period of multiplying two m bit integers with time complexity $O(m^\mu)$. Thus, for classical algorithms of multiplication, values $\mu=2$, and for the fast Karatsuba algorithm, $\mu=\log_2 3$.

It should be noted that each of the presented methods finds only the order of the curve and does not tackle the issue on the possibility of its use in cryptographic transformations. Based on this, upon computing the order of the curve, it is necessary to verify the feasibility of its application in the cryptographic systems. And to choose the criteria that make it possible to select elliptic curves for constructing system-wide parameters at the required level of stability.

5. Results of exploring time complexity of the methods for computing the order of elliptic curves

When exploring the mechanisms for constructing strong cryptographic system-wide parameters of EC, an important criterion is the time required for the construction of such parameters and the dimensionality of field, over which the curve is defined. Parameters obtained in this way will be suitable for their further use in the EDS mechanisms according to DSTU 4145-2002 (or similar, in other words those that employ a binary field). The Ukrainian standard sets the parameters of size up to 431 bits while the standard FIPS 186-3 contain system-wide parameters up to 521 bits. Important is the estimation of time needed to build strong cryptographic parameters at the super high level of stability ($509 \le \#E \le 1031$, bits) [4].

The largest complexity and resource consumption when generating the parameters is displayed exactly by the step in counting the points on EC, the theoretical information about this stage is presented in the previous chapters of present article. Those chapters also describe theoretical evaluation of complexity in the phase of EC lift for all presented algorithms. The complexity of computing the norm is given for only one of the existing algorithms that was used to calculate the order of EC.

For the canonical lift, we applied the SST method [10], the MSST method [12] and the Harley method [11] (described in chapters 4.2–4.4 of the present article). For the normalization, an analytical method (proposed in paper [10]) was employed.

To count the number of points on EC, we developed a software tool in the C++ language using the library NTL and gmp. A research into algorithm execution time was carried out in the program that was compiled using gcc 4.84 in the operating system Ubuntu 14.04 (USA) and the processor CPU Intel Core i5-2300 (USA). As all the operations for this class of algorithms are conducted sequentially, the parallelization of implementation of the algorithm is impossible, and the number of cores in the processor will not change execution time of the algorithm.

Table 1 gives the period of computing the lift phase for the SST, MSST, Harley methods and the norm. An analysis of Table 1 allows us to argue that there is an advantage of the MSST method over its standard modification in the SST method; the practical results of present study confirm theoretical research into the given methods. The Harley method is the most optimal among the three presented methods from the point of view of computational complexity.

Table 1

Computational complexity of p-adic methods

The degree of extension of the field d, bits	Lift phase			Normalization by SST, s
	SST, s	MSST, s	The Harley method, s	
7	0,006265	0,003216	0,000925	0,000248
23	0,023238	0,010162	0,004397	0,00088
79	0,198349	0,077477	0,054023	0,007455
107	0,251051	0,123306	0,07934	0,010746
173	0,680349	0,400715	0,219232	0,028311
199	0,822255	0,501402	0,278724	0,035182
257	0,950504	0,698594	0,290267	0,054191
307	1,55134	1,17776	0,431548	0,088748
383	2,39215	1,91847	0,614475	0,133243
433	2,78954	2,245	0,766383	0,159998
503	3,97648	3,2511	1,00376	0,198481
601	7,15564	6,00326	1,60194	0,424093
709	10,44	9,02212	2,16152	0,532264
787	13,2826	11,6431	2,61117	0,649224
827	15,0722	13,1966	2,86254	0,715296
929	19,5872	17,5986	3,54738	0,920241
1021	24,9435	22,4928	4,35925	1,06235
1049	34,2923	30,0397	5,27848	1,7509

It can be argued that the canonical lift of EC using the Harley method is about six times more efficient than the SST method (the idea of which was employed in the Harley method). If we refer to the results of studying the AGM method and Satoh method in the previous article [14], then it can be argued that the Harley method is about 25 times more effective than the AGM method and is 75 times more efficient than the best modification of the Satoh method.

Fig. 1 shows a chart of dependence of the size of the field and the execution time for different methods of the canonical lift. A difference in the execution time between the algorithms of lift is observed for all the examined sizes of field extension.

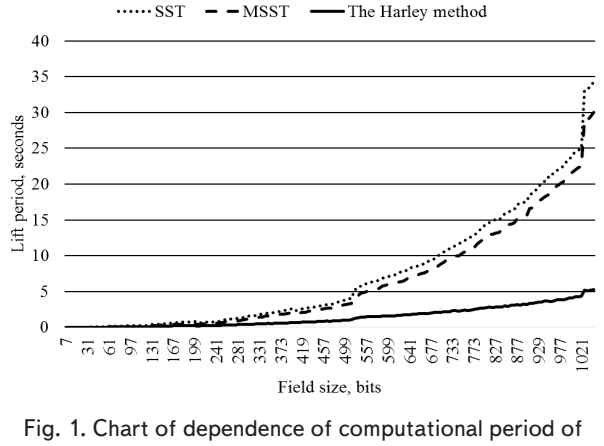

Fig. 1. Chart of dependence of computational period of canonical lift on the size of field extension for different algorithms

Fig. 2 shows a chart of dependence of the field size and the norm computation period for different methods of canonical lift of elliptic curves, although the same method of normalization was actually used.

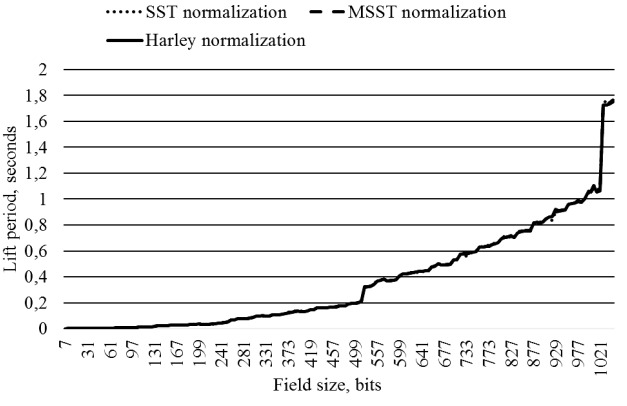

Fig. 2. Chart of dependence of the norm computation period for different size of the field extension

The chart in Fig. 2 demonstrates that such a method for computing the norm is efficient and results for all methods were about the same each time with a difference of 0.01 s.

6. Discussion of results of exploring complexity in the methods for computing the order of elliptic curves

The results obtained can be compared to the results received in article [8]. To count the number of points on EC, the author used the processor AMD XP 1700+ (USA) and the operating system Linux Redhat 7.1 (USA). The algorithms were written in the programming language C, and basic mathematical operations in the ring of p-adic numbers were performed on Assembler. Results of comparing the time indicators from [8] to the ones received in the present study are given in Table 2 (columns with data from article [8] contain the designation "B").

Table 2

Comparison of the time complexity of computing the order of curve

Extension of the field d, bits	Total time for computing the order of curve					
	SST, s	SST "B", s	MSST, s	MSST "B", s	The Harley method, s	The Harley method "B", s
144	0,44	0,13	0,24	0,06	0,17	0,06
168	0,63	0,26	0,37	0,08	0,2	0,08
192	0,74	0,29	0,45	0,13	0,26	0,12
240	0,77	0,65	0,55	0,28	0,23	0,25
288	1,3	0,72	0,97	0,39	0,37	0,38
336	1,83	1,17	1,43	0,64	0,49	0,6
384	2,4	1,76	1,91	0,97	0,6	0,92
480	3,44	3,56	2,87	2,03	0,91	1,87

Results in Table 2 demonstrate that the basic mathematical operations, written in the low-level programming language (implemented in paper [8]), yield large effectiveness when computing the order of EC. The most effective method for computing the order of EC, as shown in Table 2,

is the Harley method, in other words, research results of the present work and of the studies in article [8] regarding the method that is optimal in terms of computational complexity, coincide.

One should consider the size of 1031 bits, because this is exactly the size of elliptic curves required for the super high level of stability (512 bits for symmetric cipher). The total time for counting the number of points on EC of size 1031 bits for the Harley method at computing the norm by SST is approximately 10 s. This period slightly exceeds that of Table 1 because there are certain operations performed, which were not dealt with in the present work in detail. For example, present article does not address the process of polynomial generation for the field, converting elements before the normalization, etc.

It can be argued based on the presented findings that there is a convergence between the theoretical computational complexity of the examined algorithms and the compared experimental results. Similar to article [8], the present study observes reduction in the computational complexity from the SST, MSST methods to the Harley method. The obtained experimental assessments confirm the analytical complexity that was described by authors of the estimated methods in articles [10–13]. We can state that a combination of the Harley algorithm for EC lift and the normalization method from the SST algorithm are the best candidates to modify the Ukrainian EDS standard. Here by modification we mean an expansion in the base of general parameters. And, of course, the given combination is the best one only by the criterion of computational complexity.

The given article presents an optimal algorithm to count the number of points on EC that applies the Harley method [11] and the method for normalization form paper [10], as well as certain adaptation for a binary field used in the Ukrainian and world standards. Results of research into these algorithms demonstrated that they might be employed to modify the Ukrainian standard DSTU 4145-2002 in the direction of extending the number of general parameters and their size. The software model makes it possible to generate system-wide parameters at the super high level of stability in seconds.

In addition, present study shows the time required to generate parameters at the good level of stability (for example, 257 bits) – it is less than 0.5 s. At such characteristics, users of information systems can generate common parameters all by themselves prior to the phase of assigning keys between the parties. Previously, such a situation was impossible due to the high complexity of computations, but modern computing power and efficient mathematics provide the users with such possibility.

7. Conclusions

1. The conducted analysis of promising methods for computing the EC order revealed that to solve this problem, at present, the most efficient (in terms of computational complexity) is the Harley method. This conclusion was made based on the performed theoretical and experimental studies and comparison between the Satoh, AGM, SST, MSST and Harley methods.

2. Analytical complexity in the execution of the SST, MSST and Harley methods was demonstrated and, according to it, we determined the fastest (by execution time) algorithm for computing the order of the curve. It is demonstrated that the Harley method is the fastest, due to applying the Artin-Schreier equation for canonical lift of EC. By reducing the number of computations for the large size accuracy, the Harley method is more efficient than the SST and MSST methods. Present study shows that using the Harley method in practice makes it possible to accelerate the computation of EC order by approximately 7 times compared with the SST method.

3. Based on the explored data, we constructed a program model for the methods of canonical lift of EC and normalization. Development of a software model made it possible to perform experimental analysis of the examined algorithms. By the data obtained, the present work experimentally confirmed the quasi quadratic dependence of the field size, over which EC is defined, and the time required for the EC canonical lift.

References

1. Horbenko, Yu. I. Analysis of the possibility of quantum computers and quantum computings for cryptanalysis of modern cryptosystems [Text] / Yu. I. Horbenko, R. S. Hanzia // Eastern-European Journal of Enterprise Technologies. – 2014. – Vol. 1, Issue 9 (67). – P. 8–16. – Available at: http://journals.uran.ua/eejet/article/view/19897/18759
2. Hanzia, R. S. Analiz shlyakhiv rozvytku kryptohrafiyi pislya poyavy kvantovykh kompyuteriv [Text] / R. S. Hanzia, Yu. I. Horbenko // Visnyk Natsional'noho universytetu "L'vivs'ka politekhnika": Kompyuterni systemy ta merezhi. – 2014. – Issue 806. – P. 40–48.
3. Shor, P. W. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer [Text] / P. W. Shor // SIAM Journal on Computing. – 1997. – Vol. 26, Issue 5. – P. 1484–1509. doi: 10.1137/s0097539795293172
4. Horbenko, I. D. Prykladna kryptolohiya [Text]: monohrafiya / I. D. Horbenko, Yu. I. Horbenko; KhNURE. – Kharkiv: Fort, 2012. – 868 p.
5. Schoof, R. Counting points on an elliptic curve over finite fields [Text] / R. Schoof // Proc. Journees Arithmetiques. – 1995. – Issue 93. – P. 219–252.
6. Skjernaa, B. Satoh's algorithm in characteristic 2 [Text] / B. Skjernaa // Mathematics of Computation. – 2003. – Vol. 72, Issue 241. – P. 477–488. doi: 10.1090/s0025-5718-02-01434-5
7. Handbook of Elliptic and Hyperelliptic Curve Cryptography [Text] / H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen, F. Vercauteren (Eds.). – NW.: Chapman & Hall/CRC, 2005. – 807 p. doi: 10.1201/9781420034981
8. Vercauteren, F. Computing zeta functions of curves over finite fields [Text]: diss. for the degree of PhD / F. Vercauteren. – Katholieke Universiteit Leuven, 2003. – 195 p.
9. Satoh, T. The Canonical Lift of an Ordinary Elliptic Curve over a Finite Field and its Point Counting [Text] / T. Satoh // J. Ramanujan Math. Soc. – 2000. – Vol. 15, Issue 4. – P. 247–270.

10. Satoh, T. Fast computation of canonical lifts of elliptic curves and its application to point counting [Text] / T. Satoh, B. Skjernaa, Y. Taguchi // Finite Fields and Their Applications. – 2003. – Vol. 9, Issue 1. – P. 89–101. doi: 10.1016/s1071-5797(02)00013-8
11. Harley, R. Asymptotically optimal p-adic point-counting [Electronic resource] / R. Harley // E-mail to NMBRTHRY list. – 2002. – Available at: https://listserv.nodak.edu/cgi-bin/wa.exe?A2=ind0212&L=NMBRTHRY&F=&S=&P=7824
12. Gaudry, P. A Comparison and a Combination of SST and AGM Algorithms for Counting Points of Elliptic Curves in Characteristic 2 [Text] / P. Gaudry // Lecture Notes in Computer Science. – 2002. – P. 311–327. doi: 10.1007/3-540-36178-2_20
13. Lercier, R. Counting Points on Elliptic Curves over Finite Fields of Small Characteristic in Quasi Quadratic Time [Text] / R. Lercier, D. Lubicz // Lecture Notes In Computer Science. – 2003. – P. 360–373. doi: 10.1007/3-540-39200-9_22
14. Hanzia, R. S. Otsinka obchyslyuval'noyi skladnosti metodiv pidrakhunku kil'kosti tochok na eliptychniy kryviy [Text] / R. S. Hanzia // Systemy obrobky informatsiyi. – 2016. – Issue 8. – P. 92–99.
15. Satoh, T. Asymptotically fast algorithm for computing the Frobenius substitution and norms over unramied extension of p-adic number fields [Text] / T. Satoh. – Department of Mathematics, Faculty of Science, Saitame University, 2001. – P. 1–21.

В зв'язку з незадовільною стійкістю стандартних криптографічних алгоритмів з відкритим ключем до методів квантового криптоаналізу, проведено дослідження можливості використання постквантових криптографічних алгоритмів. Проведено порівняльну оцінку таких алгоритмів в залежності від умов використання та проаналізовано переваги різних механізмів криптографічних перетворень, що є стійкими до методів квантового криптоаналізу

Ключові слова: постквантові криптографічні алгоритми, порівняльна оцінка криптоалгоритмів, критерії порівняння криптоалгоритмів

В связи с неудовлетворительной стойкостью стандартных криптографических алгоритмов с открытым ключом к методам квантового криптоанализа, проведено исследование возможности использования постквантовых криптографических алгоритмов. Проведена сравнительная оценка таких алгоритмов в зависимости от условий применения и выполнен анализ преимуществ разных механизмов криптографических преобразований, стойких к методам квантового криптоанализа

Ключевые слова: постквантовые криптографические алгоритмы, сравнительная оценка криптоалгоритмов, критерии сравнения криптоалгоритмов

UDC 004.056.55
DOI: 10.15587/1729-4061.2017.96321

EXAMINING A POSSIBILITY TO USE AND THE BENEFITS OF POST-QUANTUM ALGORITHMS DEPENDENT ON THE CONDITIONS OF THEIR APPLICATION

I. Gorbenko
Doctor of Technical Sciences, Professor*
E-mail: Gorbenkol@iit.com.ua
V. Ponomar
Postgraduate student*
E-mail: Laedaa@gmail.com
*Department of Security of Information Systems and Technologies
V. N. Karazin Kharkiv National University
Svobody sq., 4, Kharkiv, Ukraine, 61022

1. Introduction

Due to the development of technologies for quantum computing and the introduction of quantum computer, there is a threat to the current state of protection of cryptographic systems with a public key [1]. With an advent of quantum computer that would have the volume of register required for the methods of quantum cryptanalysis, the stability of existing crypto algorithms will significantly degrade [2, 3]. This necessitates the creation of algorithms resistant to the methods of quantum cryptanalysis. The European project "New European Schemes for Signatures, Integrity, and Encryptions " (NESSIE) and the National Institute of Standards and Technologies (NIST) of the USA announced a start of recruiting the applicants for the contest of post-quantum algorithms whose standards are planned to be adopted over 2020–2022 [4, 5].

A peculiarity of this task is that the contest will accept the algorithms whose cryptographic transformations are based on the latest information or insufficiently tested mathematical methods that will require considerable time to prove their stability in terms of quantum cryptanalysis. That is why the choice of the new standard will affect not only the algorithm that will be employed but also further development of the post-quantum cryptography.

Another feature is that the universal algorithms are lacking that can be used both for electronic signature (ES) and the encryption. Therefore, it is necessary for each of the security