АНАЛИЗ И СРАВНЕНИЕ АЛГОРИТМОВ ЭЛЕКТРОННОЙ ЦИФРОВОЙ ПОДПИСИ ГОСТ Р 34.10-1994, ГОСТ Р 34.10-2001 И ГОСТ Р 34.10-2012

Комарова А.В.[1], Менщиков А.А.[2], Коробейников А.Г.[3]

Современные требования к электронной цифровой подписи регламентируются в нашей стране национальными стандартами линейки ГОСТ Р 34.10. В работе был выполнен сравнительный анализ национального стандарта ГОСТ Р 34.10–2012 с его предшественниками ГОСТ Р 34.10–2001 и ГОСТ Р 34.10–1994 на предмет актуальности, криптостойкости и трудоемкости. Для этого были произведены оценки скорости и производительности процедур генерации и проверки подписи, выполнены оценки сложности взлома российских государственных стандартов, рассчитаны предположительные оценки продолжительности взлома, обосновано преимущество нового ГОСТ Р 34.10–2012 перед его предшественниками. Отмечено, что именно использование аппарата эллиптических кривых позволяет в значительной степени повысить стойкость нового алгоритма электронной цифровой подписи. При этом доказано, что новый стандарт требует в несколько раз больше времени и для формирования хэш–функции и для процесса формирования самой подписи. В работе отмечена актуальность оптимизации реализаций библиотек данных криптографических алгоритмов под существующие языки программирования.

Ключевые слова: *хэш–функция, дискретное логарифмирование, асимметричная криптография, эллиптическая кривая, криптостойкость, криптографическая защита информации, национальный стандарт, средства криптографической защиты, контроль целостности*

DOI: 10.21681/2311-3456-2017-1-51-56

Введение

Информационные технологии на сегодняшний день имеют огромное значение для функционирования различных предприятий [1]. Безопасные информационные технологии применяются в таких областях, как производственная, экономическая, финансовая, военная [2]. Вся информация, обрабатывающаяся в компьютерных сетях и в интернете, должна быть хорошо защищена. Этот факт требует непрерывного совершенствования механизмов защиты информации.

Наиболее востребованы на сегодняшний день криптосистемы с открытым ключом. Наиболее стойкими и трудоемкими из них, по мнению ученых, являются криптосистемы на эллиптических кривых [3]. Данный факт послужил одной из причин смены ГОСТ Р 34.10–1994 «Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи» его аналогом в 2001 году и далее последующим усовершенствованным аналогом в 2012 году. Алгоритмы ЭЦП ГОСТ Р 34.10-2001 и ГОСТ Р 34.10-2012 являются усовершенствованием алгоритма ЭЦП ГОСТ Р 34.10-1994. В отличие от старого стандарта, основанного на сложности дискретного логарифмирования в конечном простом поле, новые алгоритмы построены на базе математического аппарата эллиптических кривых.

Исследование хэш-функций по ГОСТ Р 34.11-1994 и ГОСТ Р 34.11-2012

Стойкость ЭЦП складывается из надежностей используемой хэш-функции и непосредственно алгоритма ЭЦП [4]. Исследуем хэш-функции, используемые в старом и новых алгоритмах ЭЦП.

Алгоритмы ЭЦП ГОСТ Р 34.10-1994 и ГОСТ Р 34.10-2001 использует хэш-функцию по ГОСТ Р 34.11-1994. Для нового стандарта ЭЦП ГОСТ Р 34.10-2012 был разработан и новый стандарт выработки хэш-функции ГОСТ Р 34.11-2012.

Непосредственно на скорость выработки ЭЦП и процесса подписания документа влияет и время, затраченное на выработку хэша сообщения. Для оценки времени генерации хэш-функции был написан программный код на языке Python с использованием библиотеки Pygost, функциональные выдержки из которого приведены ниже:

1 Комарова Антонина Владиславовна, аспирант, Университет ИТМО, Санкт-Петербург, piter-ton@mail.ru

2 Менщиков Александр Алексеевич, аспирант, Университет ИТМО, Санкт-Петербург, menshikov@corp.ifmo.ru

3 Коробейников Анатолий Григорьевич, доктор технических наук, профессор, Университет ИТМО, Санкт-Петербург, korobeynikov_a_g@mail.ru

```
from os import urandom
import time
from pygost.gost34112012 import GOST34112012
from pygost.gost341194 import GOST341194

def test_hash_94(message):
 GOST341194(message).digest()

def test_hash_2012(message):
 GOST34112012256(message).digest()

def test(callback):
 for ll in [32, 64, 128, 256, 512]:
  start = time.clock()
  callback(urandom(ll))
  finish = time.clock()
```

Для каждой длины шифротекста было выполнено по 100 повторов генерации хэш-функции для каждого ГОСТ. Вычисления проводились на персональном компьютере с процессором Intel Core i5-3317U. На (Рис.1) можно видеть экспоненциальную зависимость времени работы от длины шифруемых данных. Так же, не трудно заметить, что время, требуемое на выработку хэша при одинаковой длине шифротекста, в ГОСТ Р 34.11-2012 возросло в два раза по сравнению с предыдущим ГОСТ. Так, при длине сообщения в 512 байт (4096 бит) по старому ГОСТ потребуется столько же временных ресурсов, что и при длине 80 байт (640бит) по новому стандарту. Пояснить такое положение вещей можно особенностями реализации Python модуля. Необходимо упомянуть, что в зависимости от требований к среде исполнения результаты могут быть разными, поэтому актуальным является также и дальнейшие исследования в области оптимизации реализаций библиотек данных алгоритмов под существующие языки программирования.

Далее проверим вычислительную стойкость хэш-функций.

Как известно, стойкая хэш-функция должна обладать следующими тремя свойствами:

1. Стойкость к вычислению прообраза – невозможность нахождения неизвестного прообраза для любых предварительно заданных хэш-значений, т.е. для заданной хэш-функции h вычислительно невозможно найти неизвестный прообраз x при предварительно заданном хэш-значении y =h(x) для любого значения y.

2. Стойкость к вычислению второго прообраза – невозможность нахождения любого другого прообраза, который давал бы такое же хэш-значение, как и заданный, т.е. для заданной хэш-функции h и прообраза x вычислительно невозможно найти другой прообраз x′≠ x, для которого выполнялось бы условие h(x) =h(x′).

3. Стойкость к коллизиям – невозможность нахождения двух прообразов, для которых вырабатывалось бы одинаковое значение, т.е. для заданной хэш-функции h вычислительно невозможно найти два прообраза x и x′, x ≠ x′, для которых выполнялось бы условие h(x) =h(x′) [5].

Самая распространенная атака на хэш-функции – это атака методом перебора [6]. Реализовать данную атаку можно двумя способами. Первый способ предполагает взлом второго свойства хэш-функции: то есть, создание другого документа с таким же хэш-значением. Второй способ предполагает взлом третьего свойства хэш-функции: поиск двух документов с одинаковыми хэш-значениями.

Вероятность взлома хеш-функции можно оценить на основании статьи [7], в которой были выведены формулы для оценки вероятности взлома хэш-функции методом перебора. Первый способ,

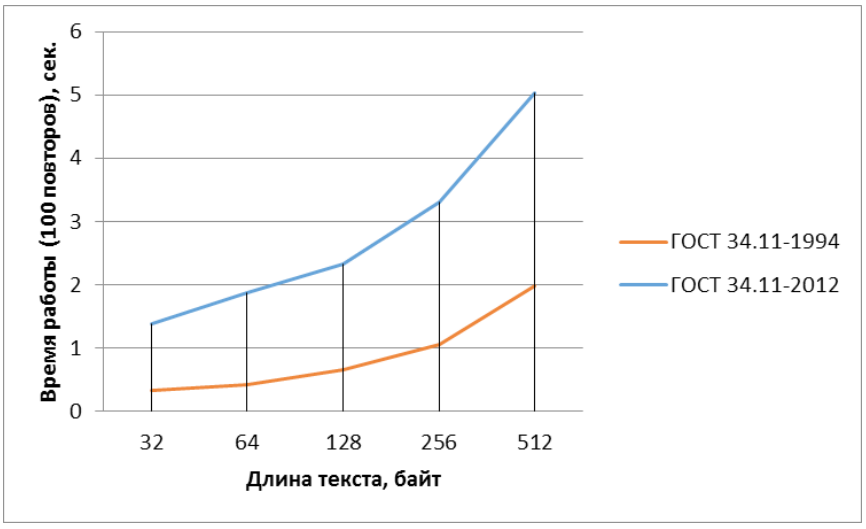

***Рис.1**. Сравнительное время выработки хэш-функций по ГОСТ*

осуществляющий взлом второго свойства хэш-функций дал вероятность $P = 2^{1-k}$, где k - число разрядов хэш-кода. Второй метод, направленный на взлом третьего свойства хэш-функций дал вероятность $P = 2^{-k/2}$, где k-число разрядов хэш-кода.

Если одна MIPS (Million Instruction Per Second) машина хэширует миллион сообщений в секунду. При таких условиях число хэш-кодов, вычисленных одной MIPS машиной за один год составит D=60×60×24×365×10⁶=3,15×10¹³[6].

В (табл. 1) приведена оценка вероятности взлома хэш-функции для двух рассмотренных способов атаки при различных значениях длины выходного хэш-кода.

Российский стандарт хэш-функций ГОСТ Р 34.11-1994 использовал 256-битное хэш-значение сообщения. Его преемник стандарт хэш-функции ГОСТ Р 34.11-2012 позволяет создавать как 256-битное, так и 512-битное хэш-значение сообщения, что позволяет утверждать, что при современных вычислительных мощностях его вскрытие вычислительно невозможно. При длине хэш-значения 512 бит, продолжительность взлома вторым, более быстрым методом, составит 3,68×10⁶³ MIPS-лет. А для взлома ГОСТ Р 34.11-1994 при длине хэш-значения 256 бит соответственно потребуется **1,08×10²⁵** MIPS-лет, что в значительной степени уступает предыдущему значению.

Оценка и сравнение алгоритмов формирования ЭЦП

Основной отличительной особенностью нового ГОСТ Р 34.10-2012 от ГОСТ Р 34.10-2001 является наличие дополнительных вариантов параметров схем (соответствующего длине секретного ключа порядка 512 бит).

Применительно для практического использования того или иного алгоритма ЭЦП, конечно, одним из первых вопросов поднимается проблема быстродействия, а именно времени, требующегося на процессы генерации ключа, подписывания документа, и, в последующем, проверки подписи. Эти три аспекта и были нами рассмотрены. Для этого был создан программный код так же на языке Python с использованием библиотеки Pygost, функциональные выдержки из которого приведены ниже:

```
from os import urandom
import time

from pygost.gost3410 import CURVE_PARAMS,
\          prv_unmarshal, pub_marshal, verify
from pygost.gost3410 import GOST3410Curve
from pygost.gost3410 import public_key
from pygost.gost3410 import sign

def test_generate_key(n):
    prv_raw = urandom(n)
    prv = prv_unmarshal(prv_raw)
    c = GOST3410Curve
          (*CURVE_PARAMS[«GostR3410_2001_
ParamSet_cc»])
    pub = public_key(c, prv)
    return c, pub, prv

def test_sign_key(c, prv):
    sign(c, prv, b»some data»)

def test_verify_key(c, pub, s):
    verify(c, pub, b»some data», s)
```

Для каждого из рассматриваемых стандартов было выполнено по 100 повторов генерации

Таблица 1.
Оценка вероятности взлома хэш-функции двумя разными способами

k - длина хэш-кода, бит	Первый способ (создание другого документа с таким же хэш-значением)		Второй способ (поиск двух документов с одинаковыми хэш-значениями)	
	Вероятность взлома	Продолжительность взлома, MIPS-лет	Вероятность взлома	Продолжительность взлома, MIPS-лет
	$P = 2^{1-k}$	$T = 2^{k-1} \div D$	$P = 2^{-k/2}$	$T = 2^{k/2} \div D$
64	$1,08 \times 10^{-19}$	$2,93 \times 10^{5}$	$2,33 \times 10^{-10}$	$1,36 \times 10^{-4}$
128	$5,88 \times 10^{-39}$	$5,40 \times 10^{24}$	$5,42 \times 10^{-20}$	$5,86 \times 10^{5}$
256	$1,73 \times 10^{-77}$	$1,84 \times 10^{63}$	$2,94 \times 10^{-39}$	$1,08 \times 10^{25}$
512	$1,49 \times 10^{-154}$	$2,13 \times 10^{140}$	$8,64 \times 10^{-78}$	$3,68 \times 10^{63}$
1024	$1,11 \times 10^{-308}$	$2,85 \times 10^{294}$	$7,45 \times 10^{-155}$	$4,26 \times 10^{140}$

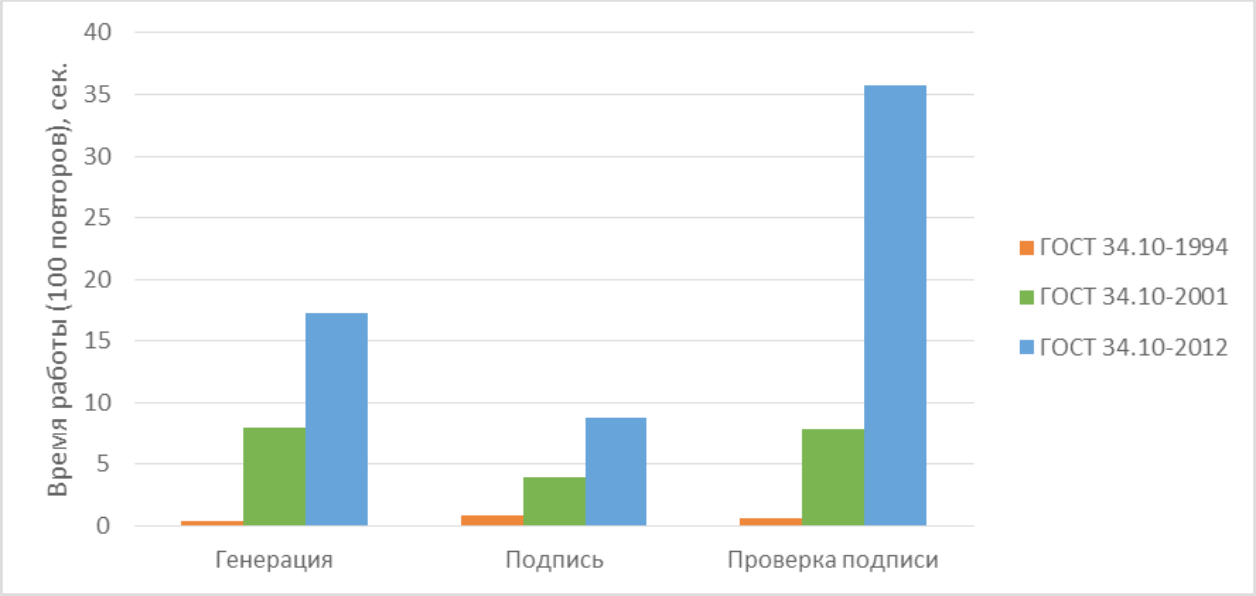

***Рис.2**. Сравнительное время работы процессов ГОСТ*

подписи, непосредственно подписания документа и проверки подписи. Вычисления проводились на персональном компьютере с процессором Intel Core i5-3317U. Подпись вырабатывалась длиной 256 бит, так как лишь этот параметр остался прежним (ГОСТ Р 34.10-2012 позволяет вырабатывать подпись длиной и 256 бит и 512 бит). На основании проведенных экспериментов выяснилось, что ГОСТ Р 34.10.2012 требует самых больших временных ресурсов для всех трех процессов (Рис.2), в несколько раз больше, чем его предшественники.

Как уже отмечалось выше, надежность цифровой подписи определяется стойкостью к криптоаналитическим атакам алгоритма ЭЦП [2].

Далее определим криптостойкость российских стандартов ЭЦП. Стойкость стандарта ГОСТ Р 34.10-1994 основана на сложности решения частной задачи дискретного логарифмирования в простом поле GF(p). Стойкость же стандартов ГОСТ Р 34.10-2001 и ГОСТ Р 34.10-2012 основана на сложности решения задачи дискретного логарифмирования в группе точек эллиптической кривой.

Рассчитаем вычислительную сложность алгоритмов ЭЦП нового и старых стандартов. Одним из наиболее эффективных алгоритмов решения задачи дискретного логарифмирования в простом поле GF(p) является обобщенный метод решета числового поля [8]. Сложность данного метода оценивается как

$$O\left(\exp\left(\left(\sqrt[3]{\frac{64}{9}}+O(1)\right)(\ln p^q)^{\frac{1}{3}}(\ln\ln p^q)^{\frac{2}{3}}\right)\right),$$

где $O(1)\to 1$, а $p\to\infty$ [9].

Для вычисления задачи дискретного логарифмирования в группе точек эллиптической кривой воспользуемся методом p-Полларда, сложность которого оценивается как $O\left(\sqrt{p}\right)$ [10].

Таблица 2.
Сложность взлома российских стандартов ЭЦП

Порядок числа p и порядок q группы точек эллиптической кривой	ГОСТ Р 34.10-1994	ГОСТ Р 34.10-2001, ГОСТ Р 34.10-2012
128	$1,35\times 10^{10}$	$1,84\times 10^{19}$
256	$1,12\times 10^{14}$	$3,40\times 10^{38}$
512	$1,76\times 10^{19}$	$1,16\times 10^{77}$
1024	$1,32\times 10^{26}$	$1,34\times 10^{154}$
2048	$1,53\times 10^{35}$	$1,80\times 10^{308}$

В (табл. 2) приведена оценка вычислительной сложности решения задач дискретного логарифмирования в простом поле и в группе точек эллиптической кривой.

Так, при 512-разрядных p и q, сложность взлома нового стандарта составляет $\mathbf{1{,}16\times10^{77}}$, а старого – $\mathbf{1{,}76\times10^{19}}$.

Заключение

Согласно оценкам, полученным при анализе, можно сделать вывод, что схемы ЭЦП ГОСТ Р 34.10-2001 и ГОСТ Р 34.10-2012 являются более стойкими по сравнению со схемой ЭЦП ГОСТ Р 34.10-1994. Использование аппарата эллиптических кривых позволяет в значительной степени повысить стойкость алгоритма ЭЦП при одинаковом порядке p и q. Это означает, что в новом российском стандарте ЭЦП можно использовать меньшую длину ключа, чем в старом, без понижения безопасности всей системы. В свою очередь, новый ГОСТ Р 34.10-2012 является более функциональным по сравнению с ГОСТ Р 34.10-2001 благодаря возможности применения разных порядков q циклической подгруппы группы точек эллиптической кривой. Но, не смотря на все преимущества в стойкости, новый стандарт требует в несколько раз больше времени и для формирования хэш-функции и для процесса формирования самой подписи.

Эту проблему можно решить попыткой подбора стойкой и в то же время доступной для использования в реальных криптографических приложениях эллиптической кривой. Данный вопрос показывает актуальность дальнейших исследований и разработок в области ЭЦП и асимметричной криптографии в целом.

Во многом время и производительность зависят от устройств, на которых они выполняются. В связи с этим так же актуальной задачей для будущих исследований является оптимизация реализаций библиотек данных алгоритмов под существующие языки программирования.

***Рецензент**: Бондаренко Игорь Борисович, кандидат технических наук, доцент Университета ИТМО, igorlitmo@rambler.ru*

Литература

1. Сабанов А.Г. Аутентификация при электронном обмене документами // Доклады Том. гос. ун-та систем управления и радиоэлектроники. 2011. № 2(24). С. 263-266.
2. Елисеев Н.И. Анализ и синтез перспективной системы электронного документооборота Министерства обороны Российской Федерации // Наукоемкие технологии в космических исследованиях Земли. 2016. Т. 8. № 2. С. 76-84.
3. Пискова А.В. Разработка алгоритма электронной цифровой подписи, основанного на задачах факторизации и дискретного логарифмирования на эллиптических кривых // Сборник трудов IV Всероссийского конгресса молодых ученых – СПб: Университет ИТМО, 2015. С. 322-326.
4. Функции хэширования: классификация, характеристика и сравнительный анализ [Электронный ресурс] / В.Н. Вервейко, А.И. Пушкарев, Т.В. Цепурит. Режим доступа: http://bezopasnik.org/article/book/94.pdf (дата обращения 06.02.2015).
5. Вихман В.В., Панков М.А. Исследование криптостойкости алгоритмов многоитерационного хэширования в подсистемах аутентификации МИС // Актуальные проблемы электронного приборостроения (АПЭП–2014): труды 12 международной конференции. Новосибирск: Изд-во НГТУ. 2014. Т. 2. С. 193–199.
6. Ferguson N., Lucks S., Schneier B., Whiting D., Bellare M., Kohno T., Callas J., Walker J. The Skein Hash Function Family. Submisson to NIST. 2008. Available at: http://www.skem-hash.info/sites/default/files/skein1.1.pdf, accessed 27.11.2016.
7. Ключарев П.Г. Криптографические хэш-функции, основанные на обобщённых клеточных автоматах // Наука и образование. Электронное научно-техническое издание. 2013. №1. DOI: 10.7463/0113.0534640
8. Вихман В.В., Панков М.А. Повышение стойкости хеш-функций в информационных системах на основе алгоритма многоитерационного хеширования с несколькими модификаторами // Труды СПИИРАН. 2014. Вып. 5(36).
9. Горбенко, И.Д. Методы распараллеливания алгоритма Полларда решения задачи дискретного логарифмирования для систем с общей памятью / И.Д. Горбенко, Е.Г. Качко, К.А. Погребняк // Высокопродуктивные вычисления (HPC-UA'2012): труды международной научной конференции (Киев, 8-10 октября, 2012 г.). - Киев: НАНУ, 2012. -С. 152-157.
10. Качко Е.Г., Погребняк К.А. Параллельный метод Полларда решения задачи дискретного логарифмирования в группе точек эллиптической кривой // Параллельные вычислительные технологии (ПАВТ–2012): труды международной научной конференции (Новосибирск, 26–30 марта, 2012 г.). Челябинск: Издательский центр ЮУрГУ, 2012. – С. 723.

ANALYSIS AND COMPARISON OF ELECTRONIC DIGITAL SIGNATURE ALGORITHMS GOST R 34.10-1994, GOST R 34.10-2001 AND GOST R 34.10-2012

Komarova A.V.[4], Menshchikov A.A.[5], Korobeynikov A.G.[6]

Abstract. *Nowadays information systems that use asymmetric cryptography or public key cryptography are widely spread all over the world. The reason of this fact is a rather high reliability of existing algorithms. In the direction of information security we can distinguish the following important tasks: ensuring the availability of information, ensuring the authenticity of information,ensuring the confidentiality of information. All these problems can be solved with the help of electronic digital signature (hereinafter–EDS). It ensures the integrity of information, its confidentiality, authenticity and the authenticity of authorship. EDS is widely used in various commercial and governmental organizations which in their activities are obliged to follow the state standard GOST R 34.10–2012 «Information technology. Cryptographic protection of information. The processes of forming and checking electronic digital signature». In view of the above a comparative analysis of the GOST R 34.10–2012 with its predecessors the GOST R 34.10–2001 and the GOST R 34.10–1994 was performed for finding its relevance, reliability and complexity. According to these the estimation of speed and performance procedures generation and signature verification were made. Also the assessments of the hacking complexity of the Russian state standards were realized. The approximate durations of forced entry were calculated and the advantage of the new GOST R 34.10–2012 before its predecessors was justified .*

Keywords: *digital signature, hash function, discrete logarithm, elliptic curve, resistance, asymmetric cryptography, GOST R 34.10–2012.*

References

1. Sabanov A.G. Autentifikacija pri jelektronnom obmene dokumentami // Doklady Tom. gos. un-ta sistem upravlenija i radiojelektroniki [The Reports Tom. state University of control systems and Radioelectronics]. - 2011. - № 2(24). - S. 263-266.
2. Eliseev N.I. Analiz i sintez perspektivnoj sistemy jelektronnogo dokumentooborota Ministerstva oborony Rossijskoj Federacii // Naukoemkie tehnologii v kosmicheskih issledovanijah Zemli [Science intensive technologies in space research Earth]. 2016. T. 8. № 2. C. 76-84.
3. Piskova A.V. Razrabotka algoritma ehlektronnoj cifrovoj podpisi, osnovannogo na zadachah faktorizacii i diskretnogo logarifmirovaniya na ehllipticheskih krivyh // Sbornik trudov IV Vserossijskogo kongressa molodyh uchenyh [Proceedings of the IV all-Russian Congress of young scientists] – SPb: ITMO university, 2015. – S. 322-326.
4. Funkcii hehshirovaniya: klassifikaciya, harakteristika i sravnitel'nyj analiz [Elektronnyj resurs]/ V.N. Vervejko, A.I. Pushkarev, T.V. Cepurit. Rezhim dostupa: http://bezopasnik.org/article/book/94.pdf (data obrashcheniya 06.02.2015).
5. Vihman V.V., Pankov M.A. Issledovanie kriptostojkosti algoritmov mnogoiteracionnogo hjeshirovanija v podsistemah autentifikacii MIS // Aktual'nye problemy jelektronnogo priborostroenija (APJeP–2014) [Actual problems of electronic instrument making (of APEP–2014)]: tr. 12 mezhdunar. konf.Novosibirsk : Izd-vo NGTU. 2014. T. 2. S. 193–199.
6. Ferguson N., Lucks S., Schneier B., Whiting D., Bellare M., Kohno T., Callas J., Walker J. The Skein Hash Function Family. Submisson to NIST. 2008. Available at: http://www.skem-hash.info/sites/default/files/skein1.1.pdf, accessed 27.11.2016.
7. Kljucharjov P.G. Kriptograficheskie hjesh-funkcii, osnovannye na obobshhennyh kletochnyh avtomatah // Nauka i obrazovanie. MGTU im. N.Je. Baumana [Science and education. MGTU im. N. Uh. Bauman]. Elektron. zhurn. 2013. №1. DOI: 10.7463/0113.0534640
8. Vihman V.V., Pankov M.A. Povyshenie stojkosti hesh-funkcij v informacionnyh sistemah na osnove algoritma mnogoiteracionnogo heshirovaniya s neskol'kimi modifikatorami // Trudy SPIIRAN. [SPIIRAS Proceedings]. - 2014. Issue 5(36). ISSN 2078-9181
9. Gorbenko, I.D. Metody rasparallelivanija algoritma Pollarda reshenija zadachi diskretnogo logarifmirovanija dlja sistem s obshhej pamjat'ju / I.D. Gorbenko, E.G. Kachko, K.A. Pogrebnjak // Vysokoproduktivnye vychislenija (HPC-UA'2012): trudy mezhdunarodnoj nauchnoj konferencii (Kiev, 8-10 oktjabrja, 2012 g.) [Highly productive computing (HPC-UA'2012): proceedings of international scientific conference (Kyiv, 8-10 October, 2012).]- Kiev: NANU, 2012. -S. 152-157.
10. Kachko E.G., Pogrebnjak K.A. Parallel'nyj metod Pollarda reshenija zadachi diskretnogo logarifmirovanija v gruppe tochek jellipticheskoj krivoj // Parallel'nye vychislitel'nye tehnologii (PAVT–2012): trudy mezhdunarodnoj nauchnoj konferencii (Novosibirsk, 26–30 marta, 2012 g.) [Parallel computational technologies (PCT ' –2012): proceedings of international scientific conference (Novosibirsk, March 26-30, 2012)]. Cheljabinsk: Izdatel'skij centr JuUrGU, 2012. – S. 723.

4 Antonina Komarova, postgraduate student, St. Petersburg National Research University of Information Technologies, Mechanics and Optics, St. Petersburg, piter-ton@mail.ru

5 Aleksandr Menshchikov, postgraduate student, St. Petersburg National Research University of Information Technologies, Mechanics and Optics, St. Petersburg, menshikov@corp.ifmo.ru

6 Anatoly Korobeynikov, Dr.Sc., Professor, St. Petersburg National Research University of Information Technologies, Mechanics and Optics, St. Petersburg, korobeynikov_a_g@mail.ru