



APPLIED RADIO ELECTRONICS

Scientific and Technical Journal 2016 Volume 15 № 3

Special issue devoted to problems of ensuring
information security

CONTENTS

METHODS AND MEANS OF ASYMMETRIC CRYPTOGRAPHIC TRANSFORMATIONS

- Gorbenko I.D., Kachko O.G., Naumenko G.S.* Experimental study of the possibility of using NTRUPrime parameters for asymmetric encryption in accordance with ANSI X9.98 – 2010 standard135
- Bessalov A.V., Oleshko K.A., Porechna D.M., Tsygankova O.V., Chorny O.M.* Secure twisted Edwards curves with minimal complexity of group operations.....141
- Yesina M.V.* Mathematical model of an anonymous electronic signature protocol based on identity151
- Yesina M.V., Kulibaba V.A.* Mathematical and program models of related keys attack implementation on electronic signature IBS-1 mechanism.....157
- Kachko O.G., Televnyi D.K.* Studying the possibility of using functional programming languages in modelling methods of cryptographic transformations162
- Kuznetsov O.O., Lutsenko M.S., Andrushkevych A.V., Melkozerova O.M., Novikova D.V., Loban A.V.* Statistical studies of modern stream ciphers.....167

METHODS AND MEANS OF SYMMETRIC CRYPTOTRANSFORMATIONS

- Rodinko M.Yu., Oliynykov R.V.* A mathematical model of non-injective key schedules properties evaluation of symmetric block ciphers.....179
- Ruzhentsev V.I.* Analysis of the method of proving the resistance of block ciphers to impossible differential attack184
- Torba A.A., Bobuch V.A., Torba M.O., Torba A.O.* Deterministic pseudorandom sequence generators for stream-based encryption D L R R191

POSTQUANTUM AND ELECTRONIC SIGNATURES

- Kovaleva N.V., Gorbenko Yu.I.* Analysis of postquantum digital signature schemes based on hash functions195
- Ponomar V.A., Berezhnyi O.G.* Fast algorithms for calculating isogeny of elliptic curves.....203

CONTENTS

(Continued from front cover)

- Garmash D.V., Baklykov O.O., Filatova N.V., Gorbenko I.D.* Quantum cryptographic algorithms of electronic signature based on multivariate quadratic transformations.....215

METHODS AND MEANS OF INFORMATION SECURITY

- Veklych S.G., Lavrovskaya T.V., Rassomakhin S.G.* Statistical model of functioning an information transmission system using algebraic methods of processing pseudorandom codes.....226
- Stetsenko P.I., Khalimov G.Z.* Method of countering attacks on routing tables based on the botnet architectures for Bitcoin peer-to-peer network232
- Stetsenko P.I., Perekopskiy A.A., Khalimov G.Z.* Infrastructure attack on a Bitcoin peer-to-peer network.....240
- Poluyanenko N.A.* The searching of non-linear feedback shift registers forming a maximal length sequence.....245
- Krasnobayev V.A., Koshman S.A., Yanko A.S.* Methods of data control in a residual class system that are based on the principle of parallel nulevisation253
- In memory of Aleksandr Alekseevich Zelenskiy (24.06 1943 – 15. 05. 2016).....266

Харьковский национальный университет радиоэлектроники

Академия наук прикладной радиоэлектроники

ПРИКЛАДНАЯ РАДИОЭЛЕКТРОНИКА

Научно-технический журнал

И.о. главного редактора

Чурюмов Г.И.

Зам. главного редактора

Дохов А.И.

Редакционный совет

Гузь В.И., Довбня А.Н., Егоров А.М., Калугин В.В., Кравченко В.И.,
Назаренко И.П. (Россия), Неклюдов И.М., Пресняк И.С., Симонов К.Г. (Россия),
Симанков В.С. (Россия), Слипченко Н.И., Чабдаров Ш.М. (Россия),
Яковенко В.М., Ярошенко В.С. (Россия)

Редакционная коллегия

Абрамович Ю.И. (США), Бодянский Е.В., Борисов А.В., Буц В.А., Бых А.И.,
Гомозов В.И., Жуйков В.Я., Зарицкий В.И., Кипенский А.В., Кульпа К. (Польша),
Леховицкий Д.И., Литвинов В.В., Лукин К.А., Мачехин Ю.П.,
Модельский Й. (Польша), Нерух О.Г., Поляков Г.А., Ролинг Г. (Германия),
Седышев Ю.Н., Серков А.А., Сухаревский О.И., Чурюмов Г.И.,
Шифрин Я.С., Шкварко Ю.В. (Мексика)

Адрес редакции:

Редакция журнала «Прикладная радиоэлектроника»
Харьковский национальный университет радиоэлектроники
просп. Науки, 14, 61166, Харьков, Украина
Тел.: + 38 (057) 702 10 57
Факс: + 38 (057) 702 10 13
E-mail: are@nure.ua
<http://www.anpre.org.ua>

СОДЕРЖАНИЕ

МЕТОДЫ И СРЕДСТВА АСИММЕТРИЧНЫХ КРИПТОГРАФИЧЕСКИХ ПРЕОБРАЗОВАНИЙ

<i>Горбенко І.Д., Качко О.Г., Науменко Г.С.</i> Экспериментальне дослідження можливості використання параметрів NTRUPRIME для несимметричного шифруванні згідно з стандартом ansi x9.98 - 2010	135
<i>Бессалов А.В., Олешко К.А., Поречная Д.Н., Цыганкова О.В., Черный О.Н.</i> Криптостойкие скрученные кривые Эдвардса с минимальной сложностью групповых операций	141
<i>Єсіна М.В.</i> Математична модель протоколу анонімного електронного підпису на основі ідентифікаційних даних	151
<i>Єсіна М.В., Кулібаба В.А.</i> Математична та програмна моделі реалізації атаки на зв'язаних ключах відносно механізму електронного підпису IBS-1	157
<i>Качко. Е.Г., Телевний Д.К.</i> Исследование возможности использования языков функционального программирования при моделировании методов криптографических преобразований	162
<i>Кузнецов О.О., Луценко М.С., Андрушкевич А.В., Мелкозерова О.М., Новікова Д.В., Лобан А.В.</i> Статистичні дослідження сучасних потокових шифрів	167

МЕТОДЫ И СРЕДСТВА СИММЕТРИЧНЫХ КРИПТОПРЕОБРАЗОВАНИЙ

<i>Родінко М.Ю., Олійников Р.В.</i> Математична модель оцінки властивостей неін'єктивних схем розгортання ключів симметричних блокових шифрів	179
<i>Руженцев В.И.</i> Проверка метода доказательства стойкости блочных шифров к атаке невыполнимых дифференциалов	184
<i>Торба А.А., Бобух В.А., Торба М.О., Торба А.О.</i> Детерминированные генераторы псевдослучайных последовательностей для потокового шифрования на основе ДЛРР	191

ПОСТКВАНТОВЫЕ И ЭЛЕКТРОННЫЕ ПОДПИСИ

<i>Ковальова Н.В., Горбенко Ю.І.</i> Аналіз постквантових механізмів електронних підписів на основі геш-функцій	195
<i>Пономар В.А., Бережний О.Г.</i> Швидкі алгоритми для обчислення ізогеній на еліптичних кривих	203
<i>Гармаш Д. В., Бакликов О. О., Філатова Н.В., Горбенко І.Д.</i> Квантові криптографічні алгоритми електронного підпису на основі мультіваріативних квадратичних перетворень	215

МЕТОДЫ И СРЕДСТВА ЗАЩИТЫ ИНФОРМАЦИИ

<i>Веклич С.Г., Лавровская Т.В., Рассомахин С.Г.</i> Статистическая модель функционирования системы передачи информации при использовании алгебраических методов обработки псевдослучайных кодов	226
<i>Стеценко П.І., Халімов Г.З.</i> Метод протидії атакам на таблиці маршрутизації на основі архітектур ботнетів для однорангової пірингової мережі Bitcoin	232
<i>Стеценко П.І., Перекопський О.О., Халімов Г.З.</i> Атака інфраструктури на однорангову пірингову мережу Bitcoin	240
<i>Полуянєнко Н.А.</i> Поиск регистров сдвига с нелинейной обратной связью, формирующих последовательность максимального периода	245
<i>Краснобаев В.А., Кошман С.А., Янко А.С.</i> Методы оперативного контроля данных в системе остаточных классов, основанные на принципе параллельной нулевизации	253

Памяти Зеленского Александра Алексеевича (24.06 1943 – 15. 05. 2016).....	266
--	------------

МЕТОДЫ И СРЕДСТВА АСИММЕТРИЧНЫХ КРИПТОГРАФИЧЕСКИХ ПРЕОБРАЗОВАНИЙ

УДК 004.056.55

ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ МОЖЛИВОСТІ ВИКОРИСТАННЯ ПАРАМЕТРІВ NTRUPRIME ДЛЯ НЕСИМЕТРИЧНОГО ШИФРУВАННЯ ЗГІДНО З СТАНДАРТОМ ANSI X9.98 - 2010

І.Д. ГОРБЕНКО, О.Г. КАЧКО, Г.С. НАУМЕНКО

Сучасні атаки використовують спеціальний формат параметрів, які застосовуються в алгоритмі NTRU (ANSI X9.98). Сьогодні знайдені постквантові параметри, для яких ці атаки не можуть бути використані (NTRUPrime). В роботі досліджується можливість та доцільність використання нових параметрів в алгоритмі ANSI X9.98.

Ключові слова: ANSI X9.98-2010, NTRUPrime, NTRU parameters, швидкодія.

ВСТУП

У зв'язку з проблемами, пов'язаними з можливостями використання сучасних несиметричних алгоритмів в умовах наявності квантових комп'ютерів, найбільший інтерес становлять криптографічні алгоритми, криптостійкість яких базується на решітках [1]. Саме до цього класу відносяться NTRU подібні алгоритми.

Для стандарту несиметричного шифрування ANSI X9.98 - 2010[2], який базується на NTRU алгоритмі, визначені різні атаки, пов'язані з параметрами, а саме з модулями, які використовуються. В роботі [3] запропоновано нові параметри, які запобігають цим атакам, а саме, інший поліном й інші модулі для використання (NTRU Prime, подальшому NTRUPrime). Ми дослідили можливість використання цих параметрів для шифрування з боку найбільш важливої характеристики, яка відрізняє методи NTRU від решти несиметричних алгоритмів, а саме швидкісні характеристики. Для цього використовується фактично стандарт ANSI X9.98 – 2010, але з математикою NTRUPrime. Це наступна робота з циклу робіт, присвячених цьому класу алгоритмів [4], [5]. Подальший аналіз з боку криптостійкості, в тому числі, в умовах використання квантових комп'ютерів буде проведено в подальших роботах циклу.

1. ВІДМІННОСТІ КЛАСИЧНОГО NTRU-МЕТОДУ ВІД NTRUPRIME) ТА ЇХ АНАЛІЗ З БОКУ ОБЧИСЛЮВАЛЬНОЇ СКЛАДНОСТІ

1. Замість кільця $(Z/qZ)[X](X^N - 1)$

використовується поле $(Z/qZ)[X](X^N - X - 1)$. Як N запропоновано значення 739.

2. Замість значення параметру $q = 2048$, який використовується для обчислення модуля коефіцієнтів поліному, застосовується значення 9829. Значення параметрів N і q визначили назву методу, яке дали

йому автори [3], а саме Streamlined NTRU Prime 9829739.

3. Кількість 1 та -1 в особистому ключі (значення d_f), та в поліномі для маскування (Blinding Polynomial) (значення d_r) визначається залежно від N та потрібної криптостійкості, але для усіх параметрів $d_f = d_r$. Для NTRUPrime $d_f = d_r = 202$.

4. Повний набір параметрів для класичного NTRU забезпечують криптостійкість від 112 до 256. Обраний набір параметрів ($N=739$, $q = 2048$), як стверджують автори [3], забезпечує криптостійкість не менше, ніж 128.

Використання кільця в класичному методі в ході обчислення добутку (найважчої операції при шифруванні - розшифруванні) просто загортали поліном, тобто індекс коефіцієнта брався за модулем N , використання поля для NTRUPrime потребує обчислення добутку за модулем $X^N - X - 1$.

Значення q визначає модуль, за яким виконується обчислення усіх коефіцієнтів поліному. Обчислення за модулем $q = 2048$ не потребує використовувати операцію ділення, для $q = 9829$ операція ділення потрібна. Крім того, значення q визначає довжину відкритого ключа. Відкритий ключ – це поліном порядку N , граничні значення коефіцієнтів якого визначаються значенням q . Для $q = 2048$ довжина відкритого ключа $11N$ бітів, для $q = 9829$ вона складає $14N$. Для $N = 739$ отримуємо 10346 бітів або 1294 байта. Автори [3] пропонують для завдання відкритого ключа в упакованому форматі використовувати комбіновану систему числення $(2 - 10)$, тоді відкритий ключ потребує 1232 байти. Економія менше, ніж 5% адресного простору для завдання відкритого ключа призведе до витрат часу на розгортання цього ключа, тим більше, що це відкритий ключ стороннього користувача, і таке

перетворення необхідно виконувати для кожного нового користувача.

Попередній аналіз основних параметрів показує, що обчислювальна складність для NTRUPrime очікується більше, ніж для класичного NTRU. В роботі виконується саме цей аналіз.

2. ОБЧИСЛЕННЯ ДОДАТКОВИХ ПАРАМЕТРІВ ДЛЯ NTRUPRIME

У зв'язку з необхідністю реалізації класичного NTRU для нових параметрів необхідно обчислити додаткові параметри, які використовуються для реалізації класичного NTRU.

Ці параметри, та формули для їх обчислення наведені в табл.1

Таблиця 1

Параметри та формули для обчислення

Позначення	Формула для обчислення	NTRU Prime
d_g	$d_g = N/3$	246
bLen	bLen = S	192
maxMsgLen Bytes	$\frac{3(N-1) - bLen}{8} - 1$	113
Llen	$\lceil \log_{256} \max \text{MsgLenByte} \rceil$	1
dm0	$dm0 = d_f$	202
HashLen	Залежить від криптостійкості	SHA 256
c	$\lceil \log_2 N \rceil$	11
minCallsR	$\frac{4d_r c}{\text{HashLen}}$	35
minCallsMask	$\lceil \frac{16N}{5 * \text{HashLen}} \rceil + 1$	11

Автори [3] декларують для обраних параметрів криптостійкість більше, ніж 128, тому значення криптостійкості обране 192, а відповідна геш-функція SHA256.

3 БАЗОВІ ОПЕРАЦІЇ

Як базові операції використовуються операції перетворення між байтовими рядками та поліномами, операції множення поліномів, та обчислення інверсії. Алгоритми для перетворень визначені в [2].

3.1 Операції множення поліномів

$c = a01_1 * b;$ для шифрування;
 $c = (3 * a01_1 + 1) * b$ - для розшифрування та генерації відкритого ключа,

де $a01_1$ – поліном в $\left(\frac{Z}{3Z}\right)[X]$;

b, c – поліноми в $\left(\frac{Z}{qZ}\right)[X]$.

Для забезпечення найбільшої ефективності для кожного з типів операції множення використовується свій алгоритм.

В роботі [3] для підвищення швидкодії множення поліномів пропонується використання методів множення Карацуби, Тоома та їх комбінації. Ми перевірили ефективність цих методів для поліномів з $N = 739$ і не досягли підвищення швидкодії. Це пов'язано з великою кількістю нульових елементів у масивах (не менше, ніж $N/3$), а також фактичною відсутністю операцій множення, замість них використовується операція додавання.

Наш варіант алгоритмів для множення поліномів наведено на рис. 1, 2.

Алгоритм 1. Множення поліномів для шифрування	
Вхід. N, q, a – поліном у полі	$\left(\frac{Z}{3Z}\right)[X]$ $(X^N - X - 1)$
b – поліном у полі	$\left(\frac{Z}{qZ}\right)[X]$ $(X^N - X - 1)$
Вихід. c – поліном у полі	$\left(\frac{Z}{qZ}\right)[X]$ $(X^N - X - 1)$
1. Set $c := 0$	
2. Set $i := 0$	
3. While $i \neq N$	
3.1. if ($a_i \neq 0$)	
3.1.1. Set $pc := \text{Address}(c_i)$	
3.1.2 if ($a_i = 1$)	
3.1.2.1 Set $j := 0;$	
3.1.2.2. While $j \neq N$	
3.1.2.2.1 Set $pc_j := pc_j + b_j$	
3.1.2.2.2 Set $j := j + 1$	
3.1.3 else	
3.1.3.1 Set $j := 0;$	
3.1.3.2. While $j \neq N$	
3.1.3.2.1 Set $pc_j := pc_j + b_j$	
3.1.3.2.2 Set $j := j + 1$	
4. Set $c := c \bmod q$	
5 Set $c := c \bmod (X^N - X - 1)$	
6. Return c	

Рис.1 Множення поліномів для шифрування

Алгоритм 2. Множення поліномів для розшифрування та генерації відкритого ключа
Вхід. N, q, f – поліном, $f = 3a+1, a$ – поліном у полі $\left(\frac{Z}{3Z}\right)[X]$
полі $\left(\frac{Z}{qZ}\right)[X]$, b – поліном у полі $(X^N - X - 1)$
Вихід. c – поліном у полі $\left(\frac{Z}{qZ}\right)[X]$
1. Set p3 := 3b
2. Set p_3 := -3b
3. Set c:=f ₀ b
4. Set i := 1
5. While i ≠ N
5.1 if (a _i ≠ 0)
5.1.1 Set pc := Address(c)
5.1.2 if (a _i = 3) p = p3 else p = p_3
5.1.3. Set j := 0;
5.1.4 While (j ≠ N)
5.1.4.1. Set pc _k :=pc _k +p _j
5.1.4.2. Set j:=j+1
6. Set c:=c mod q
7. Set c:=c mod (X ^N - X - 1)
8. Return c

Рис.2 Множення поліномів для розшифрування та генерації відкритого ключа

Останнє значення використовується як початкове значення c .

3.2 Інверсія поліномів

Операція інверсії використовується для обчислення відкритого ключа, який визначається за формулою: $h = f^{-1}gp$, де F – поліном з коефіцієнтами -1, 0, 1, кількість 1, -1 однакова і дорівнює d_f , $f = 3F + 1$, g – поліном з коефіцієнтами -1, 0, 1, кількість -1 дорівнює $d_g = \frac{N}{3}$ кількість 1 дорівнює $d_g + 1$, $p = 3$, а також для перевірки наявності інверсії відносно $p = 3$.

Як визначено в роботі [6] серед розглянутих алгоритмів обчислення інверсії для класичного NTRU

найбільш ефективним є алгоритм almost inverse, але, на жаль, цей алгоритм не можна використовувати для обчислення інверсії відносно поліному $X^p - X - 1$ для q , яке не є ступенем 2. В цьому разі необхідно використовувати розширений алгоритм Евкліда, який оптимізовано за рахунок заміни копіювань поліномів обміном їх адрес.

Алгоритм обчислення інверсії наведено на рис. 3.

Алгоритм 3. Обчислення f^{-1} такого, що $f^{-1} * f = f * f^{-1} = 1$ в полі $\left(\frac{Z}{qZ}\right)[X]$
Вхід. N, q, f – поліном, $f = 3F+1, F$ – поліном у полі $\left(\frac{Z}{3Z}\right)[X]$
полі $\left(\frac{Z}{qZ}\right)[X]$
Вихід. f^{-1} або Error
1. Set c:=0, y := X ^N
2. Set n3 :=f, n4 := y
3. Set n5 := 1, n6 := 0;
4. Set px := Address (n4), py := Address (n3), pa2 := Address (n6), pa1 := Address (n5)
5. Set RetValue := 1
6. Set n2 := *px / *py; *px := *px - *py * n2;
7.if (px == 0)
7.1 if (c is odd)
7.1.1 RetValue := - RetValue
7.2. Set gcd := *py;
7.3. goto step 16;
8. Set c:=c+1
9.Set RetValue:= *pa1* n2 + *pa2;
10 if (deg (RetValue) >= deg (y))
10.1 Set RetValue:= RetValue - y;
11. swap (pa1, pa2);
12. Set *pa1 = RetValue;
13. swap (px, py)
14. goto Step 8
15. if (deg (gcd) ≠ 0)
Return Error
16. Set gcd_1 = gcd ⁻¹ mod q
17. Set RetValue = gcd_1 * RetValue
18. Return RetValue

Рис. 3. Алгоритм інверсії

Для ділення поліномів використовується алгоритм 11[2]. Для множення поліному на число виконується множення кожного коефіцієнта з урахуванням модуля q . Операція обрання значення за адресою (разова адресація) позначена символом *. Операція обміну адресами позначена swap. Алгоритм повертає помилку Error в разі відсутності зворотного елемента,

тобто відповідне Діафантове рівняння не має цілих розв'язань.

4 АЛГОРИТМИ ШИФРУВАННЯ ТА РОЗШИФРУВАННЯ

Визначені в [2] відповідно до Algorithm 23 та Algorithm 24.

Далі розглянуто основні етапи цих алгоритмів та їхню оптимізацію.

4.1 Алгоритм шифрування

Вхід. Рядок байтів для шифрування m з довжиною l , та відкритий ключ h .

Вихід. Шифротекст – рядок байтів або $Eggr$, якщо довжина l перевищує параметр $maxMsgLenBytes$.

Алгоритм шифрування складається з таких кроків:

Крок 1. Формування рядка M форматом:

$b \parallel octL \parallel m \parallel p0$,

де b – випадковий рядок довжиною $bLen$;

$octL$, m – довжина повідомлення ($octL$) та саме повідомлення (m). Для завдання довжини повідомлення використовується параметр $Llen$;

$p0$ – кількість нульових байтів, які доповнюють повідомлення m до максимальної довжини. Обчислюється за формулою $maxMsgLenBytes + 1 - l$.

Для оптимізації цього кроку заповнення рядка $p0$ нулями не обов'язково, достатньо додати один 0.

Крок 2. Перетворення отриманого рядка M у поліном $MTrin$ з коефіцієнтами $\{-1, 0, 1\}$. Для оптимізації цього кроку замість перетворення трибітних послідовностей перетворюємо шестибітні, в результаті отримуємо 4 байти, які відповідають 4 коефіцієнтам $MTrin$. Для заміни шестибітних даних використовується масив констант розміром 64 елемента, який має вигляд:

0x00000000, // {0, 0, 0, 0},
 0x01000000, // {0, 0, 0, 1},
 0xFF000000, // {0, 0, 0, -1},
 ...

Якщо в результаті обробки рядка M отримуємо не усі коефіцієнти, решта коефіцієнтів заповнюється нулями.

Крок 3. Формування рядка $sData$ з форматом:

$OID \parallel m \parallel b \parallel hTrunc$,

де OID – ідентифікатор, береться з параметрів;

m – повідомлення для шифрування;

b – випадковий рядок довжиною $bLen$;

$hTrunc$ – частина упакованого відкритого ключа довжиною $bLen$.

Для оптимізації цього кроку для рядків $sData$ та M виділяється загальна пам'ять, що дозволяє не копіювати повідомлення для формування рядка $sData$. Відкритий ключ записується в контейнер при його встановленні не тільки у форматі поліному, а і в упакованому форматі, що дозволяє не виконувати його перетворення в бітовий рядок у ході формування рядка $sData$.

Крок 4. Формування полінома r для осліплення (blinding polynomial). Алгоритм формування (Algorithm 18 [2]) використовує IGF алгоритм для формування послідовності, яка далі застосовується для визначення номерів (індексів) коефіцієнтів поліному, які приймають значення 1, а потім -1. В [2] визначено два алгоритми формування IGF: IGF-2 (Алгоритм 20) та IGF-RBG (Алгоритм 21). Згідно з алгоритмом 20 спочатку обчислюється геш для рядка $sData$, а потім $minCallsR$ викликається функція обчислення геша для отриманого геша (постійна частина) та номера виклику функції обчислення геша. В результаті щоразу отримуємо рядок, довжина якого не перевищує розмір блоку, для обчислення геша. Для оптимізації цього алгоритму один раз виконуються постійні операції в ході обчислення геша для блоку і $minCallsR$ разів операції, які залежать від номера.

Крок 5. Найчастіше затратна операція множення $R = r * h$. Тут використовується Алгоритм 1 функції множення.

Крок 6. Обчислення $R \bmod 4$ та їх упакування (4 коефіцієнта на байт), отримуємо $oR4$.

Крок 7. MGF перетворення для рядка $oR4$ з урахуванням параметра $minCallsMask$ і отримання поліному $mask$. Формується бітова послідовність аналогічно кроку 4. Але як вхідна послідовність використовується $oR4$, а як кількість викликів – $minCallsMask$. Отриманий байтів масив використовується для формування поліному. З кожного байта, значення якого не перевищує 3^5 , формується 5 коефіцієнтів поліному. Для оптимізації обчислення коефіцієнтів використовується масив констант довжиною 273 рядка.

Приклади констант:

{ 0, 0, 0, 0, 0 },
 { 1, 0, 0, 0, 0 },
 { -1, 0, 0, 0, 0 },
 ...

Крок 8. Обчислення шифротексту:

$r1 = (r + Mtrin) \% 3$

if ($r1.count(1) < df$ or $r1.count(-1) < df$ or $r1.count(0) < df$) goto 1

$rh := R + r1$

Для оптимізації цього кроку для одного коефіцієнта виконуються усі необхідні операції, і ведеться підрахунок значень коефіцієнтів

Крок 9. Перетворення поліному rh в рядок байтів em .

4.2 Алгоритм розшифрування

Вхід. Шифротекст (рядок байтів, em) та його довжина (len)

Вихід. Відкритий текст (рядок байтів, m) та його довжина (l) або $Eggr$

Складається з наступних кроків.

Крок 1. Перетворення байтового рядка em в поліном e .

Крок 2. Set $cm' := f * e$. Для множення використовується алгоритм 2.

Крок 3. Set $d := cm' \bmod 3$

Крок 4. if $(d.count(-1) < df$ or $d.count(0) < df$

Return Error;

Крок 5. Set $coR4 := ConvertToBytes((e - cm') \bmod 4)$

Крок 6. Генерація поліному для маскування. MGF перетворення для рядка $coR4$ з урахуванням параметра $minCallsMask$ і отримання поліному $mask$ (див. крок 7 алгоритму шифрування) та генерація $cMtrin$.

Крок 7. Перетворення $cMtrin$ у бітовий рядок.

Для оптимізації використовується індексна таблиця: 0000	0001	00FF	0100	0101	01FF	FF00	FF01
0	3	6	1	4	7	2	5

Верхній рядок таблиці задано в шістнадцятковій системі.

Якщо серед вхідних елементів є елемент $0xFFFF$, то return Error.

Крок 8. Визначення відкритого тексту та його довжини.

Під час виконання цього пункту необхідно перевірити довжину відкритого повідомлення (вона не може перевищувати максимальну можливу, та наявність необхідної кількості нульових елементів в кінці повідомлення, якщо його довжина менше, ніж $maxMsgLenBytes$).

5 ЕКСПЕРИМЕНТАЛЬНІ РЕЗУЛЬТАТИ

Усі експерименти виконувались на комп'ютері Intel (R) Core (TM) i5 -4400 CPU @3.10 GHz, Windows 7, 64 bit.

За допомогою профілювання визначені функції, які потребують найбільшого часу в ході шифрування та розшифрування. Це функції множення поліномів.

Визначався час виконання функцій множення для поліномів з коефіцієнтами $(-1, 0, 1)$ для шифрування і коефіцієнтами $(-3, 0, 3)$ за виключенням першого коефіцієнта для дешифрування, а також функцій шифрування та розшифрування.

Для порівняння наведені аналогічні результати для класичного NTRU та для NTRUPrime. З усіх параметрів для класичного NTRU обирались параметри, які задовольняють такі критерії:

— рівень криптостійкості $S = 192$;

— значення N найближче до 739;

— співвідношення кількості ненульових елементів до N найближче до $404/739 \approx 0.55$. Найближчі параметри, які задовольняють усі вимоги, $N = 677$, $d_f = 157$.

В табл. 2 наведені результати обчислювального експерименту.

Для оптимізації кроки 3 – 5 виконуються як один крок для кожного коефіцієнта поліному.

Для операцій множення час задано в мілісекундах. Для операцій шифрування та розшифрування задана швидкість у кілобітах за секунду.

Таблиця 2

Результати обчислювального експерименту

Алгоритм	Множення (ms)	Шифрування		Розшифрування	
		ms	kbit/s	ms	kbit/s
Класичний NTRU	0.04	0.066	12202	0.038	20859
NTRUPrime	0.06	0.098	9224.3	0.078	18200

ВИСНОВКИ

З появою квантових комп'ютерів, одним з найперспективніших асиметричних криптографічних методів, який задовольняє вимоги стійкості і швидкості, стала NTRU криптосистема, яка базується на криптостійкості решіток [2].

Класичний NTRU [2] за час своєї експлуатації з 2010 року має недоліки, які можуть бути усунені завдяки використанню нових параметрів [3].

У роботі показано можливість застосування параметрів з [3] для алгоритму [2] та виконано реалізацію запропонованої комбінованої схеми.

Отримані результати наведені в табл. 2.

Класичний метод має кращі швидкісні характеристики, ніж новий практично вдвічі, що очікувано. Дійсно, використання модулів, «незручних» з боку обчислювальної складності, призводить до такого результату. Але з урахуванням того, що NTRU алгоритм за швидкісними характеристиками обганяє існуючі несиметричні методи шифрування в десятки та сотні разів, таке уповільнення не є суттєвим. Отримані швидкісні характеристики є попередніми, в подальшому буде досліджене більш повне використання SIMD операцій та графічних процесорів для покращення цих характеристик.

У подальших роботах буде також розглянуто детально криптостійкість запропонованого методу та можливість використання інших параметрів з метою збільшення криптостійкості.

Література

- [1] Report on Post-Quantum Cryptography. <http://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8105.pdf>
- [2] American National Standard for Financial Services ANSI X9.98 – 2010. Lattice-Based Polynomial Public Key Establishment Algorithm for the Financial Services Industry
- [3] Daniel J. Bernstein^{1,2}, Chitchanok Chuengsatiansup¹, Tanja Lange¹, and Christine van Vredendaal¹. NTRU Prime, <https://ntruprime.cr.yp.to/ntruprime-20160511.pdf> (visited 22.12.2016)

- [4] <https://github.com/NTRUOpenSourceProject/ntru-crypto> (visited 22.12.2016)
- [5] I. Gorbenko, O. Kachko, K. Pogrebnyak. Features OF parameterS calculation for NTRU algorithm. Прикладная радиоэлектроника, 2015. – Том 14, № 3. – С. 272-277.
- [6] Качко О.Г., Погребняк К.А., Макутонина Л.В. Аналіз, оцінки та пропозиції відносно методу генерації системних параметрів у NTRU-подібних асиметричних системах. Радіотехніка. – 2016. – Т.186. – С. 103 – 110.
- [7] Качко Е.Г., Балагура Д.С., Погребняк К.А., Горбенко Ю.І. Исследование методов вычисления инверсии в алгоритме NTRU. Прикладная радиоэлектроника. – 2013. – Т.12, № 2. – С. 254 – 257.



Горбенко Иван Дмитриевич, доктор технічних наук, професор, Харківський національний університет ім. В.Н.Каразіна, професор кафедри безпеки інформаційних систем і технологій.



Качко Олена Григорівна, кандидат технічних наук, професор кафедри ПІ ХНУРЕ. Наукові інтереси: криптографія, криптоаналіз, паралельні обчислення.



Науменко Гліб Сергійович, студент групи ПІ-13-1 факультету КН ХНУРЕ. Наукові інтереси: розподілені системи, криптографія.

УДК 004.056.55

Экспериментальное исследование возможности использования параметров NTRUPrime для несимметричного шифрования в соответствии со стандартом ANSI X9.98 - 2010 / И. Д. Горбенко, Е. Г. Качко, Г.С.Науменко. // Прикладная радиоэлектроника: науч.-техн. журнал. – 2016. – Том 15, № 3. – С. 135 – 140.

Современные атаки используют специальный формат параметров, которые применяются в алгоритме NTRU (ANSI X9.98). Сегодня найдены постквантовые параметры, для которых эти атаки не могут быть выполнены (NTRUPrime). В работе исследуется возможность и целесообразность использования новых параметров в алгоритме ANSI X9.98-2010.

Ключевые слова: ANSI X9.98, NTRUPrime, NTRU parameters, быстродействие.

Табл.: 02. Ил.: 03. Библиогр.: 07 назв.

UDC 004.056.55

Experimental study of the possibility of using NTRUPrime parameters for asymmetric encryption in accordance with ANSI X9.98 – 2010 standard / I.D. Gorbenko, O.G. Kachko G.S. Naumenko // Applied Radio Electronics: Sci. Journ. – 2016. – Vol. 15, № 3. – P. 135 – 140.

Modern attacks use a special format of the parameters which are used in NTRU algorithm (ANSI X9.98 - 2010). Postquantum parameters have been found for which the attacks can not be performed (NTRUPrime). The paper studies the possibility and advisability of using these parameters for ANSI X9.98-2010.

Keywords: ANSI X9.98-2010, NTRUPrime, NTRU parameters, speed.

Tab.: 02. Fig.: 03. Ref.: 07 items.

КРИПТОСТОЙКИЕ СКРУЧЕННЫЕ КРИВЫЕ ЭДВАРДСА С МИНИМАЛЬНОЙ СЛОЖНОСТЬЮ ГРУППОВЫХ ОПЕРАЦИЙ

А.В. БЕССАЛОВ, К.А. ОЛЕШКО, Д.Н. ПОРЕЧНАЯ, О.В. ЦЫГАНКОВА, О.Н. ЧЕРНЫЙ

Дан анализ оценок сложности групповых операций для скрученных кривых Эдвардса. Предложен метод минимизации вычислений путем выбора минимального значения параметра кривой. Приведены таблицы общесистемных параметров 25 криптостойких рекордно быстрых кривых со значениями модулей поля длиной 192, 224, 256, 384 и 521 бит.

Ключевые слова: скрученные кривые Эдвардса, полные кривые Эдвардса, порядок кривой, порядок точки, квадратичный вычет, квадратичный невычет, сложность операций.

ВВЕДЕНИЕ

Термин «скрученные кривые Эдвардса» был введен авторами работы [2]. В работе [6] мы дали критический анализ противоречий, некорректных определений и статистики распределений числа кривых разных классов в работе [2] и предложили новую классификацию кривых в обобщенной форме Эдвардса, одним из классов которых мы и рассматриваем скрученные кривые Эдвардса. Важным свойством этих кривых является то, что при $p \equiv 1 \pmod{4}$ все они имеют порядок $4n$ (n – нечетное) с минимальным четным кофактором 4. Циклическая подгруппа этих кривых простого порядка n обладает всеми преимуществами полных кривых Эдвардса [1], что открывает пути для их криптографических приложений и стандартизации.

Для полных кривых Эдвардса над простым полем задача поиска криптостойких кривых и их табуляция впервые была решена нами в работе [4]. В данной работе мы решаем ту же задачу для нециклических скрученных кривых Эдвардса. В разделе 1 приведен анализ сложности групповых операций на них и полных кривых Эдвардса в проективных координатах. Далее в разделе 2 мы предлагаем метод минимизации сложности операций путем использования минимального значения параметра a кривой. В разделе 3 описан метод и инструменты поиска быстрых криптостойких скрученных кривых Эдвардса с табуляцией результатов расчетов общесистемных параметров 25 кривых в диапазоне стандартных значений модуля поля.

1. СЛОЖНОСТЬ ГРУППОВЫХ ОПЕРАЦИЙ НА СКРУЧЕННОЙ КРИВОЙ ЭДВАРДСА

В работе [6] мы предложили новую классификацию кривых в обобщенной форме Эдвардса с уравнением

$$E_{a,d}: \begin{cases} x^2 + ay^2 = 1 + dx^2y^2, \\ a, d \in \mathbb{F}_p^*, d \neq 1, a \neq d, p \neq 2. \end{cases} \quad (1)$$

В зависимости от свойств квадратичности параметров a и d в [6] определены 3 непересекающиеся класса кривой (1): полные кривые Эдвардса

$$\left(\frac{ad}{p}\right) = -1, \quad \text{скрученные кривые Эдвардса}$$

$$\left(\frac{a}{p}\right) = -1, \left(\frac{d}{p}\right) = -1) \text{ и квадратичные кривые Эд-$$

$$\text{вардса } \left(\frac{a}{p}\right) = 1 \left(\frac{d}{p}\right) = 1). \text{ В данном разделе мы}$$

приведем оценки сложности групповых операций для первых двух классов, интересных для криптографических задач. Модифицированный универсальный закон сложения точек кривой (1) имеет вид [5]

$$\begin{aligned} (x_1, y_1) + (x_2, y_2) = \\ = \left(\frac{x_1x_2 - ay_1y_2}{1 - dx_1x_2y_1y_2}, \frac{x_1y_2 + x_2y_1}{1 + dx_1x_2y_1y_2} \right). \end{aligned} \quad (2)$$

При совпадении двух точек получим из (2) закон удвоения точек

$$2(x_1, y_1) = \left(\frac{x_1^2 - ay_1^2}{1 - dx_1^2y_1^2}, \frac{2x_1y_1}{1 + dx_1^2y_1^2} \right). \quad (3)$$

Использование модифицированных законов (2), (3) позволяет сохранить общепринятую горизонтальную симметрию (относительно оси x) обратных точек. Нейтральный элемент группы здесь равен $\mathbf{O} = (1, 0)$.

Определяя теперь обратную точку как $-P = -(x_1, y_1) = (x_1, -y_1)$, получим согласно (1) $(x_1, y_1) + (x_1, -y_1) = (1, 0) = \mathbf{O}$. Кроме нейтрально-

го элемента \mathbf{O} на оси x также всегда лежит точка $D_0 = (-1, 0)$ второго порядка, для которой в соответствии с (3) $2D_0 = (1, 0) = \mathbf{O}$. В зависимости от свойств параметров a и d можно получить еще 2 особые точки второго порядка, а также 0, 2, 4, 6, или 8

точек 4-го порядка. Как следует из (1), на оси y могут лежать точки $\pm F_0 = (0, \pm 1/\sqrt{a})$ 4-го порядка, для которых $\pm 2F_0 = D_0 = (-1, 0)$. Эти точки существуют над полем F_p , если параметр a является квадратичным вычетом.

1.1. Сложение точек

Для полных кривых Эдвардса этот анализ приведен в работе [1]. Так как в уравнении кривой (1) появился новый параметр a , требуется оценить, насколько он увеличивает вычислительные затраты. Введем третью координату Z как общий знаменатель

в (2). Пусть $x = \frac{X}{Z}$, $y = \frac{Y}{Z}$, тогда однородное уравнение кривой (1) в проективных координатах имеет вид

$$(X^2 + aY^2)Z^2 = Z^4 + dX^2Y^2,$$

$$X = xZ, \quad Y = yZ.$$

Сумма двух точек теперь записывается как $(X_1 : Y_1 : Z_1) + (X_2 : Y_2 : Z_2) = (X_3 : Y_3 : Z_3)$. С учетом подстановок выразим координаты суммарной точки согласно (2):

$$x_3 = \frac{X_3}{Z_3} = \frac{Z_1 Z_2 (Z_1^2 Z_2^2 + dX_1 X_2 Y_1 Y_2) (X_1 X_2 - aY_1 Y_2)}{(Z_1^2 Z_2^2 + dX_1 X_2 Y_1 Y_2) (Z_1^2 Z_2^2 - dX_1 X_2 Y_1 Y_2)}$$

$$y_3 = \frac{Y_3}{Z_3} = \frac{Z_1 Z_2 (Z_1^2 Z_2^2 - dX_1 X_2 Y_1 Y_2) (X_1 Y_2 + X_2 Y_1)}{(Z_1^2 Z_2^2 + dX_1 X_2 Y_1 Y_2) (Z_1^2 Z_2^2 - dX_1 X_2 Y_1 Y_2)}$$

Обозначим:

$$A = Z_1 Z_2; B = A^2; C = X_1 X_2; D = aY_1 Y_2;$$

$$E = dCD; F = B - E; G = B + E.$$

Тогда:

$$X_3 = A \cdot G \cdot (D - C),$$

$$Y_3 = A \cdot F \cdot ((X_1 + Y_1) \cdot (X_2 + Y_2) - C - D),$$

$$Z_3 = F \cdot G.$$

Подсчет числа элементарных операций здесь дает 10 умножений M , одно возведение в квадрат S и 2 умножения U на параметры a и d кривой. Итак, находим сложность вычисления суммы различных точек, выраженную через число умножений и возведений в квадрат в поле $V_E = 10M + 1S + 2U$ [2].

1.2. Удвоение точек

Используя уравнение кривой (1), закон удвоения (3) запишем в форме, не зависящей от параметра d

$$2(x_1, y_1) = \left(\frac{x_1^2 - y_1^2}{2 - x_1^2 - y_1^2}, \frac{2x_1 y_1}{x_1^2 + y_1^2} \right).$$

Тогда координаты точки удвоения согласно (3):

$$x_3 = \frac{X_3}{Z_3} = \frac{\left(\left(\frac{X_1}{Z_1} \right)^2 - a \left(\frac{Y_1}{Z_1} \right)^2 \right) \left(\left(\frac{X_1}{Z_1} \right)^2 + a \left(\frac{Y_1}{Z_1} \right)^2 \right)}{\left(2 - \left(\frac{X_1}{Z_1} \right)^2 - \left(\frac{Y_1}{Z_1} \right)^2 \right) \left(\left(\frac{X_1}{Z_1} \right)^2 + a \left(\frac{Y_1}{Z_1} \right)^2 \right)} =$$

$$= \frac{(X_1^2 - Y_1^2)(X_1^2 + Y_1^2)}{(2Z_1^2 - X_1^2 - Y_1^2)(X_1^2 + Y_1^2)},$$

$$y_3 = \frac{Y_3}{Z_3} = \frac{2 \frac{X_1}{Z_1} \frac{Y_1}{Z_1} \left(2 - \left(\frac{X_1}{Z_1} \right)^2 - a \left(\frac{Y_1}{Z_1} \right)^2 \right)}{\left(2 - \left(\frac{X_1}{Z_1} \right)^2 - a \left(\frac{Y_1}{Z_1} \right)^2 \right) \left(\left(\frac{X_1}{Z_1} \right)^2 + a \left(\frac{Y_1}{Z_1} \right)^2 \right)} =$$

$$= \frac{2X_1 Y_1 (X_1^2 + Y_1^2)}{(2Z_1^2 - X_1^2 - aY_1^2)(X_1^2 + aY_1^2)}$$

Обозначим

$$A = X_1^2, B = aY_1^2, C = Z_1^2, D = (A + B),$$

$$E = (A - B), F = 2C - A - B,$$

$$G = (X_1 + Y_1)^2, H = G - D.$$

Тогда:

$$X_3 = DE,$$

$$Y_3 = 2XYF,$$

$$Z_3 = DF.$$

Подсчет числа возведений в квадрат и умножений в поле дает суммарную сложность удвоения $T_E = 3M + 4S + 1U$ [2].

Значения сложности групповых операций в проективных координатах для полных кривых Эдвардса [1] и скрученных кривых Эдвардса приведены в таблице 1.

Таблица 1

Класс кривых	Сложность групповой операции	
	Сложение точек	Удвоение точек
Полные кривые Эдвардса	$10M + 1S + 1U$	$3M + 4S$
Скрученные кривые Эдвардса	$10M + 1S + 2U$	$3M + 4S + 1U$

Наименьших вычислительных затрат, как следует из таблицы, требуют операции на полных кривых Эдвардса. Особенно они выигрывают при удвоении, которое обходится без операции умножения $1U$. По сравнению с кривыми в форме Вейерштрасса полные

кривые Эдвардса дают выигрыш в скорости экспоненцирования точки в 1.5 – 1.6 раза [7].

2. МЕТОД ДОСТИЖЕНИЯ МИНИМАЛЬНОЙ СЛОЖНОСТИ ГРУППОВЫХ ОПЕРАЦИЙ НА СКРУЧЕННОЙ КРИВОЙ ЭДВАРДСА

Как следует из таблицы 1, ввод дополнительного параметра a в уравнение скрученной кривой (1) увеличивает вычислительные затраты сложения точек на одну операцию $1U$ и удвоения точек на $1U$ в сравнении с полной кривой Эдвардса. В этом подразделе мы предлагаем простой способ, как можно избавиться от этих дополнительных затрат и достичь максимальной производительности экспоненцирования точки на скрученной кривой Эдвардса.

В работе [6] показано, что квадратичное кручение скрученной кривой Эдвардса дает квадратичную кривую Эдвардса и обратно: $E_{a,d}^t E_{ca,cd}$ (здесь $\left(\frac{c}{p}\right) = -1, \left(\frac{ad}{p}\right) = 1$). Кроме того, внутри классов скрученных и квадратичных кривых Эдвардса имеет место изоморфизм кривых $E_{a,d} E_{d,a}$.

Свойства изоморфизма и квадратичного кручения можно обосновать также, используя j -инвариант кривой в обобщенной форме Эдвардса [2,3]

$$j(a, d) = \frac{16(a^2 + d^2 + 14ad)^3}{ad(a-d)^4}, \quad ad(a-d) \neq 0. \quad (4)$$

Как известно [3,8], изоморфные кривые (с порядком $N_E = p+1-t$) и кривые квадратичного кручения (с порядком $N_E^t = p+1+t$) имеют один и тот же j -инвариант. Из (4) сразу следуют свойства симметрии j -инварианта относительно переменных ca, cd и их инверсий:

$$j(a, d) = j(d, a), \quad (5)$$

$$j(a, d) = j(ca, cd), \quad (6)$$

$$j(a, d) = j(a^{-1}, d^{-1}), \quad (7)$$

$$j(a, d) = j(1, d/a) = j(1, a/d). \quad (8)$$

Внутри класса скрученных кривых Эдвардса нет пар квадратичного кручения, но для каждой кривой имеется изоморфная кривая со свойством (5) или

$$E_{a,d} E_{d,a}, \text{ причем } \left(\frac{a}{p}\right) = -1, \left(\frac{d}{p}\right) = -1.$$

Идея состоит в том, что при поиске подходящей для криптографии скрученной кривой Эдвардса нет смысла в переборе различных значений параметров

a и d . Можно зафиксировать один из этих параметров (например, параметр a) и варьировать другим в области его допустимых значений. Если задать этот фиксированный параметр на минимальном числовом уровне $a = 1$, так, чтобы $\left(\frac{a}{p}\right) = -1$, то можно сэкономить

полевую операцию $1U$ (умножение на параметр кривой) при сложении точек, а также и удвоении точки кривой. Например, если $a = 2$, тогда одно сложение (тождественное умножению на 2) можно считать «бесплатной» операцией. При этом достигается минимальная сложность групповой операции, равная сложности операции для полной кривой Эдвардса. Это же справедливо для всех малых.

Нам требуется доказать, что при фиксации параметра $a = 1$, перебор всех допустимых параметров d дает все возможные значения j -инварианта и, соответственно, порядков скрученной кривой.

Утверждение 1. При фиксированном значении параметра $a = 1$ кривой (1) ее j -инвариант $j(1, d)$ принимает $(p-1)/4$ возможных значений при $p \equiv 1 \pmod{4}$ и $(p-3)/4$ возможных значений при $p \equiv 3 \pmod{4}$ при всех $\left(\frac{d}{p}\right) = -1, d \neq 0$.

Доказательство. Рассмотрим квадратичные кривые Эдвардса с параметрами $\left(\frac{a}{p}\right) = 1, \left(\frac{d}{p}\right) = 1$. Для любой такой кривой существует изоморфизм $E_{a,d} E_{1,d/a} E_{1,a/d}$. Обозначим $a = \frac{d}{a}$. Из всех

$\frac{(p-1)}{2}$ квадратов мультипликативной группы параметр $a \neq 1$ принимает ровно $\frac{(p-3)}{2}$ допустимых значений. При $p \equiv 1 \pmod{4}$ для каждого, кроме квадрата $a = -1$, существует пара изоморфных кривых $E_{1,1}$.

Случай $a = -1 = -1$ вырождает пару изоморфных кривых в одну кривую. Тогда j -инвариант (8) кривой $E_{1,1}$ принимает ровно $\frac{(p-1)}{4}$ значений. При

$p \equiv 3 \pmod{4}$ имеется ровно $\frac{(p-1)}{2}$ квадратов и

$\frac{(p-3)}{2}$ допустимых значений. В этом случае элемент (-1) является квадратичным невычетом и су-

существует ровно $\frac{(p-3)}{4}$ пар изоморфных кривых и такое же число j -инвариантов.

Парой кручения каждой квадратичной кривой Эдвардса является скрученная кривая в форме (1) при

$$\left(\frac{a}{p}\right) = \left(\frac{d}{p}\right) = -1, \text{ т.е. } E_{a,d}^t E_{ca,cd}, \left(\frac{c}{p}\right) = -1.$$

Следовательно, число изоморфных пар скрученных кривых, равное числу j -инвариантов с теми же значениями, что и для квадратичных кривых Эдвардса,

также равно $\frac{(p-1)}{4}$ при $p \equiv 1 \pmod{4}$ и $\frac{(p-3)}{4}$ при $p \equiv 3 \pmod{4}$. Осталось доказать, что

все изоморфные пары скрученных кривых Эдвардса могут быть получены при одном фиксированном значении параметра $a =$.

Воспользуемся свойствами (5), (6), и умножим параметры второго j -инварианта на $c = \frac{a}{d}$,

$$j(a, d) = j(d, a) = j\left(a, \frac{a^2}{d}\right). \quad (9)$$

Отсюда следует, что любая пара изоморфных скрученных кривых Эдвардса определяется единственным параметром $a =$ и множеством всех пар квадратичных невычетов d и $d^{-1}, d \neq$. Объемы множеств таких пар, как и число изоморфизмов скрученных кривых Эдвардса, остается таким же, как и для квадратичных кривых Эдвардса. Утверждение доказано.

Как ранее отмечалось, все скрученные кривые Эдвардса имеют порядок $4n$ при $p \equiv 1 \pmod{4}$, поэтому нам интересен для криптографии лишь этот случай. Минимальное числовое значение квадратичного невычета $= 2$ существует лишь при $p \equiv \pm 3 \pmod{8}$ [8]. Следующее желаемое значение квадратичного невычета $= 3$ требует выполнения $p \equiv \pm 5 \pmod{12}$ [9]. В таблице 2 приведены для примера простые числа $p \equiv 1 \pmod{4}$, для которых $= 2$ и $= 3$ (соответствующие столбцы помечены знаком +).

Таблица 2

p	3	7	29	7	1	3	1	3	9	7
$\alpha=2$	+		+	+		+	+			
$\alpha=3$		+			+				+	

Хотя эта выборка из первой сотни простых чисел с заданными свойствами не репрезентативна, можно сделать предположение, что около 80% простых чисел $p \equiv 1 \pmod{4}$ являются модулями полей, содержащих квадратичные невычеты 2 или 3. В других случаях всегда можно найти минимальное значение параметра $a =$, что позволяет пренебречь сложностью операции $1U$ в оценках сложности групповых операций сложения и удвоения точек.

Пример 1. Пусть $p = 29$ и $a = 2$. Согласно формулы (4) можно сначала найти j -инвариант единственной квадратичной кривой Эдвардса $j(1, -1) = 17$ и соответствующей скрученной кривой Эдвардса $j(2, -2) = 17$. Далее в соответствии с утверждением 1 и формул (9) находим 6 j -инвариантов для изоморфных пар скрученных кривых Эдвардса

$$\begin{aligned} j(2, 3) &= j(2, 11) = 18, & j(2, 8) &= j(2, 15) = 12, \\ j(2, 10) &= j(2, 12) = 16, \\ j(2, 14) &= j(2, 21) = 18, & j(2, 18) &= j(2, 26) = 23, \\ j(2, 19) &= j(2, 17) = 18. \end{aligned}$$

Все скрученные кривые Эдвардса с одинаковым j -инвариантом имеют одинаковый порядок. Но число допустимых порядков в нашем примере почти вдвое меньше числа различных вычисленных j -инвариантов. Действительно, все скрученные кривые при $p \equiv 1 \pmod{4}$ имеют минимальный кофактор 4 порядка кривой. В границах Хассе $[p \pm 2\sqrt{p}]$ имеются лишь 3 значения таких порядков $N_E \in \{20, 28, 36\}$. В данном примере получены такие результаты:

$$\begin{aligned} N_E &= 20 \text{ при } j(2, -2) = 17, \\ N_E &= 28 \text{ при } j(2, d) \in \{16, 18\}, \\ N_E &= 36 \text{ при } j(2, d) \in \{12, 23\}. \end{aligned}$$

Подчеркнем, что кривые с одинаковым j -инвариантом не обязательно изоморфны, но всегда имеют одинаковый порядок. В то же время одинаковый порядок могут иметь кривые с разными значениями j -инвариантов и, разумеется, неизоморфные кривые.

Итак, задавая в скрученной кривой Эдвардса минимальное не квадратичное значение параметра $a =$, можно достичь максимальной производительности вычисления групповых операций и экспоненцирования точек, равной производительности вычислений на полной кривой Эдвардса с параметром $a = 1$.

3. РЕЗУЛЬТАТЫ ВЫЧИСЛЕНИЙ ОБЩЕСИСТЕМНЫХ ПАРАМЕТРОВ КРИПТОСТОЙКИХ СКРУЧЕННЫХ КРИВЫХ ЭДВАРДСА С МИНИМАЛЬНОЙ СЛОЖНОСТЬЮ

В данном разделе мы рассматриваем простые поля с модулями длиной 192, 224, 256, 384 и 521 бит, которые рекомендуются стандартом FIPS–186–4–2013, и приводим перечень параметров скрученных кривых Эдвардса почти простого порядка $N_E = 4n$ (n – простое) над каждым из полей. Результаты расчетов общесистемных параметров кривых в шестнадцатеричной системе чисел сведены в таблицы 3 – 7. Здесь модули длины L обозначены как p_L . Модули полей $p \bmod 4$ выбирались как простые числа с малым двоичным весом Хэмминга 3..5. Для каждой кривой приведены значения p , порядки $n = N_E / 4$ генератора G криптосистемы и его координаты (x_G, y_G) , а также значения параметров a и d .

Надо заметить, что параметр $a = 2$ является квадратичным невычетом лишь при $p = \pm 3 \bmod 8$. Это значит, что двоичное представление числа p заканчивается тремя младшими разрядами 101 = 5_{10} или 011 = 3_{10} , а все более старшие разряды дают 0 $\bmod 8$. В нашем алгоритме случайного поиска простых чисел с малым весом лишь одно значение $p = 2^{255} + 2^{38} + 2^2 + 1$ в таблице 5 отвечает этому условию (здесь $a = 2$), поэтому практически все кривые имеют минимальный параметр $a = 3$.

Проверка чисел p и n на простоту производилась с помощью тестов Миллера-Рабина и Лукаса-Лемера, реализованных в языках программирования C#, Java и в системе Wolfram Mathematica.

Вычисление символов Лежандра для нахождения подходящих параметров a и d производилось с помощью библиотечных функций языка Java и системы Wolfram Mathematica.

Порядки эллиптических кривых рассчитывались по алгоритму SEA (Schoof- Elkies -Atkin), реализованном в библиотеке PARI/GP.

Точки-генераторы были найдены удвоением случайной точки, удовлетворяющей уравнению (1), с использованием системы Wolfram Mathematica и языка Java. Удвоения достаточно, так как на нециклической скрученной кривой порядка $4n$ максимальный порядок точки равен $2n$.

В каждой из приведенных ниже таблиц содержатся параметры 5 скрученных кривых Эдвардса с минимальным значением параметра $a = 2, 3$ или 5. Далее параметр d подбирался как наименьшее из значений, при котором порядок кривой $4n$ становился

почти простым. Порядок кривых по длине сравним с длиной поля.

Таблица 3

Скрученные кривые Эдвардса почти простого порядка над полем с модулем p_{192}

$p = 2^{191} + 2^{16} + 2^{12} + 1$ $p =$ 8001 1001 $n =$ 20000000000000000000000000000000000000004447D62B952200D604 D2BEB9 $a = 3$ $d = DD$ $xG =$ 79FD21DC7BF961D56CDD092798C3016F11C2034F 154A94B5 $yG =$ 17B25C0DADFD379FF3D7CD1EAEFD83328B9454 10BE11D5DB
$p = 2^{191} + 2^{45} + 2^{41} + 1$ $p =$ 800022000000 0001 $n =$ 1FFFFFFFFFFFFFFFFFFFFFFFFFEAA490B04DCC376 0DB69D80B $a = 5$ $d = B7$ $xG =$ 41D4C37EE995324B600442F0DCE31A0EA61C08C9 2D80110B $yG =$ 3EC06FDFB33AFA858E30D898F9C18606CA62E79 F6175FD80
$p = 2^{191} + 2^{46} + 2^{36} + 1$ $p =$ 800040100000 0001 $n =$ 1FFFFFFFFFFFFFFFFFFFFFFFFFC219C1BBD9EF986 92F7D3077 $a = 3$ $d = 4D$ $xG =$ 50707DB9B16BF11C894E2A2F25A2887A6AB3A59 21E9AACDE $yG =$ 1BEF37EF93DBA0D7203F96BD14FCDA0E6B1CF1 18866982B2

<p>$p = 2^{191} + 2^{49} + 2^{27} + 1$ $p =$ 80000000000000000000000000000000200000800 0001 $n =$ 1FFFFFFFFFFFFFFFFFFFFFFFFFD4622687DA3DDCB F47806983 $a = D$ $d = AC$ $xG =$ 3BFD9D07301A8A3A9BEC18540B0EEDC0F7C3C D21F652EFE $yG =$ 3C48E70B7F04C446A8CC208696DA2592A56FB29C 79888D52</p>
<p>$p = 2^{191} + 2^{158} + 1$ $p =$ 80000000400 0001 $n =$ 2000000010000000000000004DE2B37DC449E40E33 C414B9 $a = 5$ $d =$ 800000003FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFC0F $xG =$ 209295DA12CEDAE57617AF4911C57DDCE7043EB 18687E13C $yG =$ 3847775699DF8A7F431D8DA8FE993A28A6D4F108 B6502917</p>

<p>2D4F15BA1A686CAEDB9D43F9525BF78683DAA3 9B82301FCA8A7874BE</p>
<p>$p = 2^{223} + 2^{38} + 2^{36} + 1$ $p =$ 800 005000000001 $n =$ 1FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFBCE784913191 A6362578E1CD28C9 $a = 3$ $d = 215$ $xG =$ 60536D73F2A4EF1F54C1048734301E01306FF7F331 719201335D5A55 $yG =$ DDBFD9D1AFD09CD322F639F524CC60F9A7A727 139F56BBF8D127940</p>
<p>$p = 2^{223} + 2^{61} + 2^{41} + 1$ $p =$ 8002000 020000000001 $n =$ 200 000 BF010CB396EA9 $a = 5$ $d = 3D$ $xG =$ 59507C9FEF622459507C9FEF622459507C9FEF623 AAD7109DDBA6AF7 $yG =$ 1366A5B9781878270245AAA111E53CDC079B0322 CB4A8C805309452A</p>
<p>$p = 2^{223} + 2^{72} + 2^{20} + 1$ $p =$ 800 0000001000001 $n =$ 1FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE0DAC7710567 C631D4CA120783E9 $a = 3$ $d = 15C$ $xG =$ 5C611714E8FD05D966F07BD978DF524642C21CF3 BDBB6BA1FE037DD8 $yG =$ 3499A41BF2C767BD41A045CDB7285F9E49075984 21C0B78D29D5A7E0</p>
<p>$p = 2^{223} + 2^{66} + 2^{14} + 1$</p>

Таблица 4
Скрученные кривые Эдвардса почти простого
порядка над полем с модулем p_{224}

<p>$p = 2^{223} + 2^{24} + 2^{20} + 1$ $p =$ 800 000001100001 $n =$ 200 9E831F641A2B9 $a = 3$ $d = 93$ $xG =$ 6C7CC9C10F4259CBEB0D1973AF0E4FC64AE442A 301A90DFEEB5BC081 $yG =$</p>

p =
800000000000000000000000000000004000
000000004001
n =
200000000000000000000000000014E4DE701954CB4
3404E060EAC01
a = 3
d = 26
xG =
2A795ABA13A5D9DC2BEB32468049F7E8E287393
711EA0A66DCBFF040
yG =
532C172BF052220CDE0B63A2F421DB65E8A676B9
6D7FA0307DDC0E53

BD3CF60CD27D370FF3265
a = 3
d = 6C
xG =
7E5D3187BA7FF18EF3066E57C722DCE95279A019
45D6B4C2E918B56C32FF35D1
yG =
3AC8B7A5A64DA05FF1F28870506E451F103DA6E
E32FAB89D3D903E073660572E

$p = 2^{255} + 2^{38} + 2^2 + 1$
p =
8000
0000000004000000005
n =
20031F23720CCC
C83EF9F44858B6E952E4F
a = 2
d = 1CC
xG =
25091FBF427205B62204FD7FE48236752C1FE497E
F2DB3197938BA36D9A27554
yG =
52A7749533218F16BB1F32137CE731DC0F26149CE
2E1CF1378B74E066B95253

$p = 2^{255} + 2^{70} + 2^{66} + 1$
p =
8000
0044000000000000001
n =
2002B81184994F3
F0880BB2B78314AFD2E9
a = 3
d = 13A
xG =
1C32148FB9F80F78D2E84879553296DA0858C29EF
EAD3EDCCBFC55E8FEE15C85
yG =
2D4ED6C59052B9113E76AEDACBC668F3BB0EE5
819C5B5221A311118DECAAE65E

Таблица 5

Скрученные кривые Эдвардса почти простого
порядка над полем с модулем p_{256}

$p = 2^{255} + 2^{46} + 2^{42} + 1$
p =
8000
00000000440000000001
n =
200330A60EA43D
F5957AC18C44AB8122EB7
a = 3
d = 1AF
xG =
25091FBF427205B62204FD7FE48236752C1FE497E
F2DB3197938BA36D9A27554
yG =
52A7749533218F16BB1F32137CE731DC0F26149CE
2E1CF1378B74E066B952532

$p = 2^{255} + 2^{41} + 1$
p =
8000
0000000020000000001
n =
2002498CDC14E
B2676199F9EFB8C86EA9D1
a = 3
d = BC
xG =
1306A0056F9B6F44758D8146286E140B8D2A4C717
9CCB2B515E9EAE4A679F81
yG =
33CD1A3853C6059B39DE2485F320CC00D97E1BB
77D5C79025BF01D1F3C0D8868

$p = 2^{255} + 2^{66} + 2^{60} + 1$
p =
8000
0004100000000000001
n =
2002BC056BB954

Таблица 6
Скрученные кривые Эдвардса почти простого
порядка над полем с модулем p_{384}

$p = 2^{383} + 2^{155} + 1$
p =
8000
00000000000080000000000000000000000000000000
00000001
n =
1FFF
FFFFFFFFE935B9B743CFBDC890CDB6FC507DB7
744384DF7B5FAF81D1
a = 3

Эдвардса, здесь найдены кривые для модулей поля p_{521} с наивысшим стандартным уровнем стойкости.

Литература

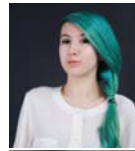
- [1] *Bernstein D.J., Lange T.* Faster Addition and Doubling on Elliptic Curves // *Advances in Cryptology – ASIACRYPT 2007* (Proc. 13th Int. Conf. On the Theory and Application of Cryptology and Information Security. Kuching, Malaysia. December 2–6, 2007). *Lect. Notes Comp. Sci.* V. 4833. Berlin: Springer, 2007. P. 29 – 50.
- [2] *Bernstein Daniel J., Birkner Peter, Joye Marc, Lange Tanja, Peters Christiane.* Twisted Edwards Curves. // *IST Programme under Contract IST–2002–507932 ECRYPT*, and in part by the National Science Foundation under grant ITR–0716498, 2008. P. 1 – 17.
- [3] *Morain F.* Edwards curves and CM curves. *ArXiv* 0904/2243v1 [Math.NT] Apr.15, 2009.
- [4] *Бессалов А.В., Дихтенко А.А.* Криптостойкие кривые Эдвардса над простыми полями. *Прикладная радиоэлектроника*, 2013, Том 12, №2. – С. 285-291.
- [5] *Бессалов А.В., Цыганкова О.В.* Взаимосвязь семейств точек больших порядков кривой Эдвардса над простым полем. *Проблемы передачи информации*. – Том 51, вып 4, 2015. – С.92 – 98.
- [6] *Бессалов А.В., Цыганкова О.В.* Классификация кривых в форме Эдвардса над простым полем. *Прикладная радиоэлектроника: научно-техн. журнал*. – 2015. – Том 14. – №4. – С.197 – 203.
- [7] *Бессалов А.В., Цыганкова О.В.* Производительность групповых операций на скрученной кривой Эдвардса над простым полем. *Радиотехника*, №181, 2015. – С.58 – 63.
- [8] *Бессалов А.В., Телиженко А.Б.* Криптосистемы на эллиптических кривых: учеб. пособие. – К.: ИВЦ «Політехніка», 2004. – 224с.
- [9] *Дэвенпорт Г.* Высшая арифметика: введение в теорию чисел // Пер. с англ. под редакцией Ю.В.Линника. – М: «Наука», 1965. – 176с.



Бессалов Анатолий Владимирович, д-р. техн. наук, профессор, профессор физико-технического института НТУУ «КПИ им. Игоря Сикорского».



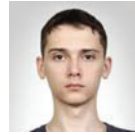
Олешко Константин Андреевич, магистрант физико-технического института НТУУ «КПИ им. Игоря Сикорского».



Поречная Дарья Никитична, магистрант физико-технического института НТУУ «КПИ им. Игоря Сикорского».



Цыганкова Оксана Валентиновна, аспирант физико-технического института НТУУ «КПИ им. Игоря Сикорского».



Черный Олег Николаевич, магистрант физико-технического института НТУУ «КПИ им. Игоря Сикорского».

УДК 681.3.06

Криптостійкі скручені криві Едвардса з мінімальною складністю групових операцій / А.В. Бессалов, К.А. Олешко, Д.Н. Поречна, О.В. Цыганкова, О.М. Чорний // *Прикладна радіоелектроніка: наук.-техн. журнал*. – 2016. – Том 15, № 3. – С. 141 – 150.

Дано аналіз оцінок складності групових операцій для скручених кривих Едвардса. Запропоновано метод мінімізації обчислень за допомогою вибору мінімального значення параметра a кривої. Наведено таблиці загальносистемних параметрів 25 криптостійких рекордно швидких кривих зі значеннями модулів поля довжиною 192, 224, 256, 384 і 521 біт.

Ключові слова: скручені криві Едвардса, повні криві Едвардса, порядок кривої, порядок точки, квадратичний лишок, квадратичний нелишок, складність операцій.

Табл.: 07. Бібліогр.: 09 найм.

UDC 681.3.06

Secure twisted Edwards curves with minimal complexity of group operations / A.V Bessalov, K.A. Oleshko, D.M. Porechna, O.V. Tsygankova, O.M. Chorny // *Applied Radio Electronics: Sci. Journ.* – 2016. – Vol. 15, № 3. – P. 141 – 150.

An analysis of evaluations of group operations complexity for twisted Edwards curves is given. A method of minimizing calculations by selecting the minimum value of the curve parameter (a) is suggested. Tables of system-wide settings of 25 record fast cryptographically secure curves in finite fields with modules of lengths 192, 224, 256, 384, and 521 bits are provided.

Keywords: twisted Edwards curves, complete Edwards curves, order of a curve, order of a point, quadratic residue, non-quadratic residue, complexity of operations.

Tab.: 07. Ref.: 09 items.

МАТЕМАТИЧНА МОДЕЛЬ ПРОТОКОЛУ АНОНІМНОГО ЕЛЕКТРОННОГО ПІДПISУ НА ОСНОВІ ІДЕНТИФІКАЦІЙНИХ ДАНИХ

М.В. ЄСІНА

У роботі розглядається математична модель протоколу анонімного електронного підпису на основі алгоритмів ДСТУ ISO/IEC 14888-3:2014 IBS-1 та IBS-2. Розглядається можливість застосування механізмів електронного підпису на основі ідентифікаційних даних у протоколі анонімного підпису.

Ключові слова: анонімний підпис, електронний підпис, ідентифікаційні дані.

ВСТУП

З метою надання у різноманітних інформаційних технологіях електронних довірчих послуг на міжнародному, регіональних та національних рівнях застосовуються значне число стандартизованих механізмів електронних підписів (ЕП). При цьому у розробників та користувачів додатків електронних довірчих послуг є можливість вибору ЕП із значного числа існуючих міжнародних та національних стандартів, наприклад, ДСТУ ISO/IEC 14888-3:2014 [1,9]. У ряді додатків електронних довірчих послуг обов'язковою є вимога надання електронної послуги анонімності (невідстежуваності), наприклад, у системах таємного електронного голосування, електронних грошей тощо. Визнаним механізмом надання послуги анонімності є застосування механізму анонімного підпису. Анонімним (сліпим) називається підпис, який накладається третьою стороною на попередньо замасковане повідомлення [3 – 4,8,11 – 12].

Зважаючи на актуальність, на даний момент комітетом ISO/IEC JTC 1/SC 27 (одним з учасників якого є Україна) розробляється пакет стандартів стосовно електронних довірчих послуг. Анонімний підпис є однією з таких послуг і стосовно нього розробляється міжнародний стандарт ISO/IEC DIS 18370-2 [2], що регламентуватиме види анонімного підпису, їх використання та стандартизуватиме конкретні механізми і протоколи анонімного підпису.

Сьогодні широке розповсюдження отримують ЕП, стійкість яких ґрунтується на складності дискретного логарифмування в скінченних полях та групах точок еліптичних кривих (ЕК). Також пройшли дослідження та рекомендуються до застосування ЕП з додатком, що ґрунтуються на ідентичності – спарюванні точок ЕК [1,5 – 7,9 – 10].

На сьогоднішній день вже існують деякі механізми анонімних підписів. Всі вони ґрунтуються на еліптичних кривих. Але сьогодні також існують механізми ЕП, що базуються на ідентифікаційних даних, і вони рекомендуються до застосування. Тому важливою є задача розробки та детального дослідження даного виду механізмів ЕП з точки зору можливості застосування у механізмі анонімного підпису [3 – 12].

Метою цієї статті є визначення можливостей та умов реалізації, а також обґрунтування використання у протоколі анонімного підпису алгоритмів ЕП згідно з ДСТУ ISO/IEC 14888-3:2014, що базуються на ідентифікаційних даних.

1. СУТНІСТЬ ЕЛЕКТРОННИХ ПІДПISІВ IBS-1 ТА IBS-2, ЩО ВИЗНАЧЕНІ ТА РЕАЛІЗОВАНІ В ДСТУ ISO/IEC 14888-3:2014

Розглянемо спочатку сутність механізмів ЕП IBS-1 та IBS-2 та етапи налаштування. Для застосування ЕП IBS-1 та IBS-2 спочатку мають бути введені та налаштовані загальні параметри та згенеровані асиметричні пари ключів.

Загальними параметрами ЕП IBS-1 та IBS-2 є [1,7,10]:

- U – секретний майстер-ключ – ціле число, $U \in [1, q-1]$;
- V – відкритий майстер-ключ – точка ЕК, $V = [U]P \bmod q$, $V \in G_1$;
- X – особистий (секретний) ключ підписувача – точка ЕК, $X = [U]Y \bmod q$, $X \in G_1$;
- Y – відкритий ключ (перевіряння) підписувача – точка ЕК, $Y = H_1(ID) \bmod q$, $Y \in G_1$;
- P – базова точка центру сертифікації ключів порядку q .

Генерація чи обчислення загальних параметрів мають здійснюватися з дотриманням таких умов:

- особистий ключ користувача X обчислюється за його запитом у центрі генерації ключів (ЦГК) та надається користувачеві по захищеному каналу;
- відкритий ключ користувача Y може обчислити кожен користувач домену;
- ID – є рядок даних, що містить ідентифікатор підписувача;
- H_1 – функція гешування, яка перетворює рядок даних у елемент групи G_1 ;
- H_2 – функція гешування, що визначена у ДСТУ ISO/IEC 10118-3:2005;
- G_1 – циклічна група простого порядку q , елементами якої є точки на ЕК над $GF(p)$;

- G_2 – циклічна група простого порядку q ,
 - елементами якої є елементи скінченного поля $GF(p^m)$.
- В таблицях 1 та 2 наведені механізми IBS-1 та IBS-2 підписування та перевірки [1,7,10].

Таблиця 1

Механізм ЕП IBS-1

Підпис повідомлення	Перевірка підпису
1. Генерування випадкового чи псевдовипадкового одноразового таємного ключа – цілого числа K , $1 < K < (q-1)$.	1. Перевірник отримує цілісні загальні параметри та відкритий ключ підписувача.
2. Здійснення спарювання: $\Pi = \langle X, P \rangle^K$, $\Pi \in G_2$ над полем $GF(p^m)$, Π – передпідпис	2. Відновлення одноразового відкритого ключа: – R та S відновлюються з доповнення; – бітова довжина R має дорівнювати довжині виходу функції H_2 ; – $S \in G_1$. Якщо хоча б одна з цих умов не виконується, підпис відхиляється.
3. Повідомлення у вигляді цілого M розбивається на його частини: M_2 – порожня частина, $M_1 = M$ – повідомлення, що треба підписати.	3. Підготування повідомлення до перевірки: – відновлення M з підписаного повідомлення; – розбиття повідомлення на M_1 та M_2 : M_2 – порожнє, $M_1 = M$.
4. Обчислення одноразового відкритого ключа: $R = H_2(M_1 \parallel FE2BS(\Pi))$, $R \in G_2$.	4. Відновлення призначення: $T = (T_1, T_2)$, $T_1 = -Y$, $T_2 = [R]Y$.
5. Обчислення призначення: $T = (T_1, T_2) = (-Y, [R]Y)$.	5. Здійснення спарювання: $\bar{\Pi} = \langle S, P \rangle \times \langle Y, V \rangle^R$.
6. Обчислення компоненти підпису: $S = [K - R]X \text{ mod } q$, $S \in G_1$. Підписом є $\Sigma = (R, S)$.	6. Обчислення одноразового відкритого ключа перевірки: $\bar{R} = H_2(M_1 \parallel FE2BS(\bar{\Pi}))$.
7. Побудова доповнення з конкатенуванням тексту у вигляді $(R, S) \parallel \text{text}$.	7. Порівняння $\bar{R} = R$: якщо не співпадають, то підпис хибний, інакше – істинний.
8. Побудова підписаного повідомлення у вигляді $M((R, S) \parallel \text{text})$.	

Таблиця 2

Механізм ЕП IBS-2

Підпис повідомлення	Перевірка підпису
1. Генерування випадкового чи псевдовипадкового одноразового таємного ключа – цілого числа K , $1 < K < (q-1)$.	1. Перевірник отримує цілісні чинні загальні параметри та чинний відкритий ключ підписувача.
2. Здійснення скалярного множення: $\Pi = [K]Y \text{ mod } q$, $\Pi \in G_1$, Π – передпідпис, точка ЕК.	2. Відновлення одноразового відкритого ключа: – R та S відновлюються з доповнення; – $R \in G_1$, $S \in G_1$. Якщо хоча б одна з цих умов не виконується, підпис відхиляється.

3. Повідомлення у вигляді цілого M розбивається на його частини: M_1 – порожня частина, $M_2 = M$ – повідомлення, що треба підписати.	3. Підготування повідомлення до перевірки: – відновлення M з підписаного повідомлення; – розбиття повідомлення на M_1 та M_2 : M_1 – порожнє, $M_2 = M$.
4. Обчислення одноразового відкритого ключа: $R = \Pi$, $R \in G_1$.	4. Відновлення призначення: $T = (T_1, T_2)$, $T_1 = -Y$, $T_2 = [-H]Y$, $H = H_2(M_2 \parallel FE2BS(R_x))$.
5. Обчислення призначення: $T = (T_1, T_2) = (-Y, [-H]Y)$, $H \in G_2$, $H = H_2(M_2 \parallel FE2BS(\Pi_x))$.	5. Обчислення передпідпису: $\bar{\Pi} = R$, $\bar{\Pi} \in G_1$.
6. Обчислення компоненти підпису: $S = [K + H]X \bmod q$, $S \in G_1$. Підписом є $\Sigma = (R, S)$.	6. Обчислення: $\bar{R}_1 = \langle P, S \rangle$ та $\bar{R}_2 = \langle V, \bar{\Pi} + [H]Y \rangle$.
7. Побудова доповнення: $(R, S) \parallel text$.	7. Порівняння $\bar{R}_1 = \bar{R}_2$: якщо не співпадають, то підпис хибний, інакше – вірний.
8. Побудова підписаного повідомлення: $M((R, S) \parallel text)$.	

2. ЗАГАЛЬНИЙ ОПИС МЕХАНІЗМУ АНОНІМНОГО ЕЛЕКТРОННОГО ПІДПISУ НА ЕЛІПТИЧНИХ КРИВИХ

Нехай у механізмі (схемі) анонімного (сліпого) ЕП на еліптичних кривих (ЕК) взаємодіють три сторони [3 – 4,8,10 – 12]: А – підписувач, В – абонент (емітент документу/повідомлення m), С – валідатор. При цьому валідатором може виступати будь-хто з них, або довірена третя особа. Емітент створює документ m , який підписувач має підписати анонімно, тобто не мати доступу до його семантичного змісту – на практиці – до реального геш-значення. Для цього емітент, отримавши згоду підписувача, маскує документ, а реально – геш-значення, за допомогою певного криптографічного перетворення та пересилає його підписувачу.

Після підпису замаскованого документу, підписувач надсилає його емітенту. Емітент здійснює зворотне, відносно маскування, перетворення та знімає його, залишивши ЕП неушкодженим. Перевірник, після отримання підписаного документу, перевіряє його цілісність, справжність та встановлює авторство за допомогою відкритого ключа підписувача.

Для забезпечення безпечності механізму, попередньо мають бути згенеровані та захищеним шляхом розповсюджені певні загальні параметри аналізу криптографічних перетворень на еліптичних кривих. Перелік перетворень та вимоги до них визначені у відповідних стандартах [1]. Також мають бути згенеровані асиметричні пари ключів для підписувачів А, а перевірник С повинен мати доступ до відкритих ключів (сертифікатів) підписувачів. Емітент повинен мати загальні параметри та ключі замаскування і розмаскування.

3. ПЕРЕВІРКА ЗАХИЩЕНОСТІ МЕХАНІЗМУ ЗА КРИТЕРІЄМ АНОНІМНОСТІ

Для схем анонімного підпису, на відміну від інших різновидів ЕП, актуальною є атака порушення анонімності. Якщо вважати, що ЕП, який застосовується, є стійким проти усіх відомих та потенційних атак, то для доведення безпечності механізму анонімного підпису необхідно довести ще його стійкість до атаки порушення анонімності.

Сутність атаки на анонімність полягає в тому, що вона може бути здійснена підписувачем за умови, що він матиме для кожної сесії постановки підпису всі відомі йому параметри схеми анонімного підпису разом із ідентифікатором емітента. Накопичена таким чином база даних (БД) може бути використана в атаці, яка полягає у спробі визначення автора певного документу m із підписом $\langle r, s \rangle$, який пройде перевірку за допомогою відкритого ключа підписувача Q .

Більш детально реалізація атаки порушення анонімності описана у [3].

4. ПРОТОКОЛ АНОНІМНОГО ЕЛЕКТРОННОГО ПІДПISУ НА ОСНОВІ ІДЕНТИФІКАЦІЙНИХ ДАНИХ

Спочатку визначимо основні сторони протоколу, що взаємодіють [3 – 4,7 – 8,10 – 12]:

А – підписувач (у ролі підписувача виступає ЦГК);

В – емітент документу m ;

С – перевірник (валідатор).

Ключову пару (U, V) та ключову пару (X, Y) обчислює ЦГК;

4.1 Протокол анонімного підпису на основі IBS-1

Генерація ключів: ЦГК у ролі підписувача А створює пару ключів (X, Y) :

$$X = [U]Y \bmod q, Y = H_1(ID) \bmod q.$$

Постановка засліпленого підпису [3]:

Абонент А:

– вибирає одноразовий таємний ключ K : $1 < K < (q-1)$;

– обчислює передпідпис Π : $\Pi = \langle X, P \rangle^K$,

$\Pi \in G_2$ над полем $GF(p^m)$;

– відправляє точку P емітенту В;

– передає обчислений передпідпис Π абоненту В.

Абонент В:

– розбиває повідомлення M на дві частини: M_2 – порожня частина, $M_1 = M$ – повідомлення, що треба підписати;

– обирає параметр маскування α :

$1 < \alpha < (q-1)$;

– обчислення одноразового відкритого ключа R (також можна вважати і одночасним його засліпленням): $R = H_2(M_1 \parallel FE2BS(\Pi))$, $R \in G_2$ і передача його абоненту А.

Абонент А:

– формування засліпленого підпису S' : $S' = [K - R]X \bmod q$, $S' \in G_1$. Підписом є $\Sigma = (R, S')$;

– передає засліплений підпис S' на перевірку абоненту В.

Перевірка засліпленого підпису [3]:

Емітент В перевіряє справжність засліпленого підпису S' за допомогою звичайної перевірки електронного підпису IBS-1:

$$\Pi = \langle X, P \rangle^K; \bar{\Pi}' = \langle S', P \rangle \times \langle Y, V \rangle^R;$$

$\bar{R} = R$, якщо $\bar{\Pi}' = \Pi$, тоді:

$$\begin{aligned} \bar{\Pi}' &= e(S', P) \times e(Y, V)^R = e([K - R]X, P) \times \\ &\times e(Y, [U]P)^R = e([K - R][U]Y, P) \times \\ &\times e([U]Y, RP) = e(X, [K - R]P) \times \\ &\times e(X, RP) = e(X, KP - RP + RP) = \\ &= e(X, KP) = e(X, P)^K = \Pi \end{aligned}$$

Таким чином, засліплений підпис проходить стандартну перевірку.

Постановка фінального підпису [3]:

Якщо S' проходить перевірку, то абонент В формує з нього фінальний анонімний підпис повідомлення M у вигляді (R, S) , попередньо перетворивши S' у S :

$$S = \alpha S' \bmod q,$$

$\Sigma = (R, S)$ – фінальний анонімний підпис.

Перевірка фінального підпису [3]:

$$\Pi = \langle X, P \rangle^{\alpha K}; \bar{\Pi} = \langle S, P \rangle \times \langle Y, V \rangle^{\alpha R};$$

$\bar{R} = R$, якщо $\bar{\Pi} = \Pi$, тоді:

$$\begin{aligned} \bar{\Pi} &= e(S, P) \times e(Y, V)^{\alpha R} = e(\alpha S', P) \times e(\alpha Y, [U]P)^R = \\ &= e(\alpha[K - R]X, P) \times e(\alpha[U]Y, RP) = \\ &= e(\alpha[K - R]X, P) \times e(\alpha X, RP) = \\ &= e(\alpha X, KP - RP + RP) = e(\alpha X, KP) = \\ &= e(X, P)^{\alpha K} = \Pi \end{aligned}$$

Фінальний анонімний підпис проходить стандартну перевірку.

Перевірка за критерієм анонімності [3]:

$$\alpha' = \frac{S}{S'} \bmod q; \Pi = \langle X, P \rangle^{\alpha K}; \tilde{\Pi} = \langle X, P \rangle^{\alpha' K}$$

якщо $\tilde{\Pi} = \Pi$, тоді підпис є стійким за критерієм анонімності:

$$\begin{aligned} \tilde{\Pi} &= e(\alpha' X, KP) = e\left(\frac{S}{S'} X, KP\right) = \\ &= e\left(\frac{\alpha S'}{S'} X, KP\right) = e(\alpha X, P)^K = \Pi \end{aligned}$$

Отже, анонімний електронний підпис на основі IBS-1 є стійким за критерієм анонімності.

4.2 Протокол анонімного підпису на основі IBS-2

Генерація ключів: ЦГК у ролі підписувача А створює пару ключів (X, Y) :

$$X = [U]Y \bmod q, Y = H_1(ID) \bmod q.$$

Постановка засліпленого підпису [3]:

Абонент А:

– вибирає одноразовий таємний ключ K : $1 < K < (q-1)$;

– обчислює передпідпис Π : $\Pi = [K]Y \bmod q$, $\Pi \in G_1$;

– відправляє точку P емітенту В;

– передає обчислений передпідпис Π абоненту В.

Абонент В:

– розбиває повідомлення M на дві частини: M_1 – порожня частина, $M_2 = M$ – повідомлення, що треба підписати;

– обирає параметр маскування α : $1 < \alpha < (q-1)$;

– обчислення одноразового відкритого ключа R : $R = \Pi$, $R \in G_2$;

– обчислює геш-значення H повідомлення: $H = H_2(M_2 \parallel FE2BS(\Pi_x))$, $H \in G_2$ (і таким чином засліплює його);

– передає геш-значення H підписувачу А.

Абонент А:

– формування зашліпленого підпису S' :
 $S' = [K + H]X \bmod q$, $S' \in G_1$. Підписом є $\Sigma = (R, S')$;
 – передає зашліплений підпис S' на перевірку абоненту В.

Перевірка зашліпленого підпису [3]:

Емітент В перевіряє справжність зашліпленого підпису S' за допомогою звичайної перевірки електронного підпису IBS-2:

$$\bar{\Pi} = R; \bar{R}_1 = \langle P, S' \rangle, \bar{R}_2 = \langle V, \bar{\Pi} + [H]Y \rangle,$$

тоді:

$$\begin{aligned} \bar{R}_2 &= e([U]P, [K]Y + [H]Y) = e(P, [K + H][U]Y) = \\ &= e(P, [K + H]X) = e(P, S') = \bar{R}_1 \end{aligned}$$

Таким чином, зашліплений підпис проходить стандартну перевірку.

Постановка фінального підпису [3]:

Якщо S' проходить перевірку, то абонент В формує з нього фінальний анонімний підпис повідомлення M у вигляді (R, S) , попередньо перетворивши S' у S :

$$S = \alpha S' \bmod q,$$

$\Sigma = (R, S)$ – фінальний анонімний підпис.

Перевірка фінального підпису [3]:

$$\bar{\Pi} = R; \bar{R}_1 = \langle P, S \rangle, \bar{R}_2 = \langle V, \bar{\Pi} + [H]Y \rangle^{\alpha},$$

тоді:

$$\begin{aligned} \bar{R}_2 &= e(V, \bar{\Pi} + [H]Y)^{\alpha} = e([U]P, \alpha([K]Y + [H]Y)) = \\ &= e(P, \alpha[K + H][U]Y) = e(P, \alpha[K + H]X) = \\ &= e(P, \alpha S') = e(P, S) = \bar{R}_1 \end{aligned}$$

Фінальний анонімний підпис проходить стандартну перевірку.

Перевірка за критерієм анонімності [3]:

$$\alpha' = \frac{S}{S'} \bmod q; \bar{R}_1 = \langle P, S \rangle; \tilde{R} = \langle V, \bar{\Pi} + [H]Y \rangle^{\alpha'},$$

якщо $\tilde{R} = \bar{R}_1$, тоді підпис є стійким за критерієм анонімності:

$$\begin{aligned} \tilde{R} &= e(V, \alpha'(\bar{\Pi} + [H]Y)) = e([U]P, \frac{S}{S'}([K]Y + [H]Y)) = \\ &= e(P, \frac{S}{S'}[K + H][U]Y) = e(P, \frac{S}{S'}[K + H]X) = \\ &= e(P, \frac{S}{S'}S') = e(P, S) = \bar{R}_1 \end{aligned}$$

Отже, анонімний електронний підпис на основі IBS-2 є стійким за критерієм анонімності.

ВИСНОВКИ

1. Механізм анонімного підпису забезпечує підтвердження справжності документів без розкриття їхнього авторства і може бути реалізований з використанням стандартних ЕП, що ґрунтуються на спарюванні точок еліптичної кривої.

2. До критеріїв перевірки захищеності механізму анонімного (сліпого) підпису додається критерій анонімності. При його застосуванні доводиться неможливість визначити підписувачу автора документа, як-

що він використовуватиме всі відомі йому параметри, які використовувались при постановці підпису.

3. Визначено та обґрунтовано можливості використання механізмів ЕП, що базуються на ідентифікаційних даних, у протоколі анонімного підпису. Доведено, що в ході використання даних механізмів ЕП, анонімний (сліпий) підпис є стійким за критерієм анонімності, тобто неможливо визначити автора документу, що підписується.

Література

- [1] Information technology – Security techniques – Digital signatures with appendix – Part 3: Discrete logarithm based mechanisms : ISO/IEC 14888-3 (Edition 2) : 2014. – 130 p.
- [2] Information technology – Security techniques – Blind digital signatures – Part 2: Discrete logarithm based mechanisms : ISO/IEC DIS 18370-2:2014(E):2015. – 70 p.
- [3] Gorbenko I. Blind electronic signature mechanisms on elliptic curves improvement / I. Gorbenko, M. Yesina, V. Ponomar // COMPUTER SCIENCE AND CYBERSECURITY. – Харківський національний університет імені В.Н. Каразіна, Випуск 1(1), 2016. Електронний ресурс. Режим доступу: <http://periodicals.karazin.ua/cscs/article/view/6205/5744>
- [4] Gorbenko I. Anonymous electronic signature method / I. Gorbenko, M. Yesina, V. Ponomar // Problems of Information Science and Technology (PIC S&T 2016), October 4–6, 2016, Kharkov National University of Radio Electronics, Kharkiv, Ukraine.
- [5] Акользіна О. С. Сутність та порівняльний аналіз криптографічної стійкості електронного підпису на ідентифікаційних даних / О. С. Акользіна, І. Д. Горбенко // Комп'ютерне моделювання в наукоємких технологіях (КМНТ-2016) : Труды научно-технической конференции с международным участием, 26-31 мая 2016 г. – Х. : Харьковский национальный университет им. В. Н. Каразина, 2016. – С. 89–92.
- [6] Горбенко Ю. І. Електронні підписи на основі ідентифікаторів та бінарного відображення / Ю. І. Горбенко, Р. С. Ганзя, О. С. Акользіна // Прикладная радиоэлектроника. – Х. : Харьковский национальный университет радиоэлектроники, 2015. – Т. 14, № 4 – С. 284–290.
- [7] Горбенко Ю.І. Сутність та умови здійснення атаки на зв'язаних ключах відносно електронних підписів IBS-1 та IBS-2 ДСТУ ISO/IEC 14888-3 / Ю.І. Горбенко, М.В. Єсіна, В.А. Кулібаба // Системи обробки інформації. – Х. : Харківський університет Повітряних Сил, 2016. – № 7(144) – С.113–118.
- [8] Єсіна М. В. Математична модель протоколу сліпого електронного підпису на еліптичних кривих / М. В. Єсіна // Прикладная радиоэлектроника. – Х. : Харьковский национальный университет радиоэлектроники, 2015. – Т. 14, № 4 – С.300–305.
- [9] Єсіна М. В. Порівняльний аналіз та умови застосування електронних підписів з додатком ДСТУ ISO/IEC 14888 / М. В. Єсіна // Комп'ютерне моделювання в наукоємких технологіях (КМНТ-2016) : Труды научно-технической конференции с международным участием, 26-31 мая 2016 г. – Х. : Харьковский национальный университет им. В.Н. Каразина, 2016. – С. 141–144.
- [10] Єсіна М. В. Реалізація атаки на зв'язаних ключах відносно електронних підписів IBS-1 та IBS-2 ДСТУ

ISO/IEC 14888-3/ М. В. Єсіна // V Міжнародна науково-технічна конференція “Захист інформації і безпека інформаційних систем” : Праці Науково-технічної конференції, 02–03 червня 2016 р. – Л. : Національний університет “Львівська політехніка”, 2016. – С. 100 – 101.

- [11] Пономар В. А. Удосконалення механізмів сліпого електронного підпису на еліптичних кривих / В.А. Пономар, М. В. Єсіна // Науково-практична конференція “Проблеми кібербезпеки інформаційно-телекомунікаційних систем”: Праці Науково-практичної конференції, 10-11 березня 2016 р. – К. : Київський національний університет імені Тараса Шевченка, 2016. – С. 67 – 68.
- [12] Пономар В. А. Механізми та умови реалізації анонімних електронних підписів на основі стандартних та перспективних алгоритмів / В. А. Пономар, М. В. Єсіна // Безпека інформації в інформаційно-телекомунікаційних системах : Матеріали міжнародної науково-практичної конференції, Випуск 18, 25-26 травня 2016 р. – К. : Державна служба спеціального зв’язку та захисту інформації України, 2016 – С. 24.



Єсіна Марина Віталіївна, аспірантка факультету комп’ютерних наук, кафедри безпеки інформаційних систем і технологій Харківського національного університету імені В.Н. Каразіна. Наукові інтереси: електронний підпис, анонімний підпис, протоколи анонімного електронного підпису, криптографічний захист інформації.

УДК 004.056.55

Математическая модель протокола анонимной электронной подписи на основе идентификационных данных / М.В. Есина // Прикладная радиоэлектроника: науч.-техн. журнал. – 2016. – Том 15, № 3. – С. – 151 – 156.

В работе рассматривается математическая модель протокола анонимной электронной подписи на основе алгоритмов DSTU ISO/IEC 14888-3:2014 IBS-1 и IBS-2. Рассматривается возможность применения механизмов электронной подписи на основе идентификационных данных в протоколе анонимной подписи.

Ключевые слова: анонимная подпись, электронная подпись, идентификационные данные.

Табл.: 02. Библиогр.: 12 назв.

UDC 004.056.55

Mathematical model of an anonymous electronic signature protocol based on identity / M.V. Yesina // Applied Radio Electronics: Sci. Journ. – 2016. – Vol. 15, № 3. – P. 151 – 156.

The paper deals with a mathematical model of an anonymous electronic signature protocol based on the algorithms DSTU ISO/IEC 14888-3:2014 IBS-1 and IBS-2. A possibility of using electronic signature mechanisms based on the identity in the anonymous signature protocol is considered.

Keywords: anonymous signature, electronic signature, identity.

Tab.: 02. Ref.: 12 items.

МАТЕМАТИЧНА ТА ПРОГРАМНА МОДЕЛІ РЕАЛІЗАЦІЇ АТАКИ НА ЗВ'ЯЗАНИХ КЛЮЧАХ ВІДНОСНО МЕХАНІЗМУ ЕЛЕКТРОННОГО ПІДПISУ IBS-1

М.В. ЄСІНА, В.А. КУЛІБАБА

У роботі розглядається стан захищеності електронних підписів на основі спарювання точок еліптичної кривої від атаки на зв'язаних ключах. Визначаються умови та можливості організації та реалізації цього виду атак. Будуються математична та програмна моделі реалізації атаки на зв'язаних ключах щодо механізму електронного підпису IBS-1. Надаються рекомендації відносно захисту від вказаних уразливостей, у тому числі у постквантовий період.

Ключові слова: атака, електронний підпис, зв'язані ключі, спарювання точок еліптичної кривої.

ВСТУП

На сьогоднішній день ретельно досліджуються та рекомендуються до застосування електронні підписи (ЕП) з додатком, що ґрунтуються на ідентичності – спарюванні точок еліптичної кривої (ЕК) [4, 5]. Відомі умови здійснення атак на зв'язаних ключах відносно ЕП, що ґрунтуються на стандартизованих криптографічних перетвореннях в скінченних полях та циклічних групах супернесингулярних кривих. Проведений аналіз джерел дозволив зробити висновок, що відносно захищеності та умов здійснення атак на зв'язаних ключах відносно ЕП IBS-1 даних практично немає [4 – 6]. Водночас попередні дослідження стійкості алгоритму ЕП IBS-1 показали, що атака на зв'язаних ключах може бути реалізована [5]. Тому важливими є дослідження стійкості вказаного ЕП від атаки на зв'язаних ключах, а також побудова математичних та програмних моделей реалізації визначеної атаки.

Проблеми стійкості стали особливо актуальними після заяв та виступів провідних спеціалістів про потенціальні уразливості на ЕП у постквантовий період. Технічний звіт АНБ США [1] стверджує, що ЕП, алгоритми яких ґрунтуються на перетворенні в кільці та в скінченному полі [3, 6] будуть нестійкими за появи

квантових комп'ютерів. Аналогічні припущення висловлені також відносно криптографічних перетворень в групі точок еліптичної кривої [3,6]. Таким чином важливими є задачі та їх вирішення, відносно стійкості ЕП, що базуються на ідентичності [4 – 6].

Отже, метою цієї статті є аналіз стану захищеності механізму ЕП IBS-1 від атаки на зв'язаних ключах та демонстрація можливості практичного здійснення такої атаки на прикладі бібліотеки PBC [2], що свідчить про недопустимість застосування цього алгоритму як перспективного стандарту ЕП в Україні в тому вигляді, в якому він був наведений, або про необхідність використання сертифікованих апаратних або апаратно-програмних засобів генерації ключів..

1. СУТНІСТЬ ЕЛЕКТРОННОГО ПІДПISУ IBS-1

Зважаючи на новизну та необхідність постановки задачі дослідження ЕП IBS-1, спочатку розглянемо сутність цього механізму ЕП та етапи налаштування.

Для застосування ЕП IBS-1 спочатку мають введеними та налаштовані загальні параметри та сгенеровані асиметричні пари ключів [4 – 6].

В таблиці 1 наведено процес підписування та перевіряння за механізмом ЕП IBS-1 [5, 6].

Таблиця 1

Механізм ЕП IBS-1

Підпис повідомлення	Перевірка підпису
1. Генерування випадкового чи псевдовипадкового одноразового таємного ключа – цілого числа K , $1 < K < (q-1)$.	1. Перевірник отримує цілісні загальні параметри та відкритий ключ підписувача.
2. Здійснення спарювання: $\Pi = \langle X, P \rangle^K$, $\Pi \in G_2$ над полем $GF(p^m)$, Π – передпідпис	2. Відновлення одноразового відкритого ключа: – R та S відновлюються з доповнення; – бітова довжина R має дорівнювати довжині виходу функції H_2 ; – $S \in G_1$. Якщо хоча б одна з цих умов не виконується, підпис відхиляється.

3. Повідомлення у вигляді цілого M розбивається на його частини: M_2 – порожня частина, $M_1 = M$ – повідомлення, що треба підписати.	3. Підготування повідомлення до перевірки: – відновлення M з підписаного повідомлення; – розбиття повідомлення на M_1 та M_2 : M_2 – порожнє, $M_1 = M$.
4. Обчислення одноразового відкритого ключа: $R = H_2(M_1 \parallel FE2BS(\Pi))$, $R \in G_2$.	4. Відновлення призначення: $T = (T_1, T_2)$, $T_1 = -Y$, $T_2 = [R]Y$.
Підпис повідомлення	Перевірка підпису
5. Обчислення призначення: $T = (T_1, T_2) = (-Y, [R]Y)$.	5. Здійснення спарювання: $\bar{\Pi} = \langle S, P \rangle \times \langle Y, V \rangle^R$.
6. Обчислення компоненти підпису: $S = [K - R]X \bmod q$, $S \in G_1$. Підписом є $\Sigma = (R, S)$.	6. Обчислення одноразового відкритого ключа перевірки: $\bar{R} = H_2(M_1 \parallel FE2BS(\bar{\Pi}))$.
7. Побудова доповнення з конкатенуванням тексту у вигляді $(R, S) \parallel text$.	7. Порівняння $\bar{R} = R$: якщо не співпадають, то підпис хибний, інакше – істинний.
8. Побудова підписаного повідомлення у вигляді $M((R, S) \parallel text)$.	

2. МАТЕМАТИЧНА МОДЕЛЬ РЕАЛІЗАЦІЇ АТАКИ НА ЗВ'ЯЗАНИХ КЛЮЧАХ НА МЕХАНІЗМ ЕП IBS-1

Нехай криптоаналітик перехопив та має повний доступ до i підписаних повідомлень [3, 5]:

$$\begin{cases} S_1 = [K_1 - R_1]X \bmod q \\ \dots \\ S_i = [K_i - R_i]X \bmod q \end{cases} \quad (1)$$

У систему (1) входить i рівнянь та $i+1$ невідомих.

Знайдемо особистий довгостроковий ключ X – невідому точку ЕК, який для усіх підписів є постійним. У результаті отримаємо систему вигляду:

$$\begin{cases} X = [K_1 - R_1]^{-1} S_1 \bmod q \\ \dots \\ X = [K_i - R_i]^{-1} S_i \bmod q \end{cases} \quad (2)$$

У системі (2) невідомими є особистий довгостроковий ключ X та i невідомих K_1, K_2, \dots, K_i . Для повного розкриття, тобто визначення секретного ключа X за i ЕП, необхідно розв'язати систему i -го порядку з $i+1$ невідомими. Дану систему (2) можна розв'язати тільки за допомогою силового методу пониження порядку системи, але проведений аналіз показав, що таким чином понизити систему рівнянь практично неможливо. Тому можна вважати, що атака на основі підписаних даних має експоненційну складність [3].

Як показав проведений аналіз, одним із можливих варіантів пониження порядку системи рівнянь може бути зв'язування ключів, наприклад, у вигляді [3]:

$$K_1 + K_2 = q \quad (3)$$

чи іншим способом. Розглянемо атаку на зв'язаних ключах.

Запишемо систему (1) для випадку двох рівнянь та розглянемо алгоритми підписування для двох повідомлень M_1 та M_2 , та ключів, що задовольняють умови (3).

Для повідомлення M_1	Для повідомлення M_2
$K_1 \in [1, q-1]$	$K_2 = (q - K_1) \in [1, q-1]$
$\Pi_1 = \langle X, P \rangle^{K_1}$	$\Pi_2 = \langle X, P \rangle^{K_2}$
$R_1 = H_2(M_1 \parallel FE2BS(\Pi_1))$	$R_2 = H_2(M_2 \parallel FE2BS(\Pi_2))$
$S_1 = [K_1 - R_1]X \bmod q$	$S_2 = [(q - K_1) - R_2]X \bmod q$

Після цього знайдемо умову, за якої $S_1 = S_2$, тобто знайдемо особистий ключ X , при якому ЕП повідомлень M_1 та M_2 будуть однаковими. У результаті маємо:

$$[K_1 - R_1]X \bmod q = [(q - K_1) - R_2]X \bmod q. \quad (4)$$

Скоротимо в (4) на X , у результаті отримаємо:

$$[K_1 - R_1] \bmod q = [(q - K_1) - R_2] \bmod q; \quad (5)$$

$$[K_1 - R_1] \bmod q = [-K_1 - R_2] \bmod q; \quad (6)$$

$$2K_1 \bmod q = [R_1 - R_2] \bmod q. \quad (7)$$

Із (7) знайдемо одноразовий ключ K_1 , оскільки R_1 та R_2 відомі і містяться у підписі:

$$K_1 = \frac{R_1 - R_2}{2} \bmod q. \quad (8)$$

Таким чином, порядок системи рівнянь понижено на невідомий одноразовий таємний ключ, у нашому випадку K_1 :

$$\begin{cases} X = [K_1 - R_1]^{-1} S_1 \bmod q \\ \dots \\ X = [K_i - R_i]^{-1} S_i \bmod q \end{cases} \quad (9)$$

Підставивши K_1 , а взагалі K_j , у систему (9), маємо систему з i рівнянь з i невідомими, яка має розв'язок.

Далі розглянемо інший підхід до реалізації атаки на зв'язаних ключах на алгоритм ЕП IBS-1. Нехай криптоаналітик перехопив та має повний доступ до i підписаних повідомлень аналогічно до (1) [3, 5].

Знайдемо невідому точку ЕК – особистий довгостроковий ключ X , який для усіх підписів є постійним. Вихідні дані аналогічні даним, що наведені для першого випадку реалізації атаки на зв'язаних ключах.

У результаті отримаємо для IBS-1 систему вигляду:

$$\begin{cases} S_1 = [K_1 - R_1]X \bmod q \\ S_2 = [-K_1 - R_2]X \bmod q \\ S_1 + S_2 = [(K_1 - R_1) + (-K_1 - R_2)]X \bmod q \\ S_1 + S_2 = [-R_1 - R_2]X \bmod q \end{cases}, \quad (10)$$

$$X = (S_1 + S_2)[-R_1 - R_2]^{-1} \bmod q$$

$$X = -[R_1 + R_2]^{-1}(S_1 + S_2) \bmod q$$

де $[R_1 + R_2]^{-1}$ – обернений елемент у полі до $R_1 + R_2$.

3. ПРОГРАМНА МОДЕЛЬ РЕАЛІЗАЦІЇ АТАКИ НА ЗВ'ЯЗАНИХ КЛЮЧАХ НА МЕХАНІЗМ ЕП IBS-1

Розглянемо програмну модель реалізації атаки на зв'язаних ключах. Програмне моделювання виконувалось мовою програмування C із використанням бібліотеки зі спарюванням точок ЕК PBC [2]. Нижче наведемо фрагмент лістингу програми та результати виконання програми (рис. 1 – 4).

```
printf("Протокол электронной подписи IBS-1
(Hess) \n");
printf("ГЕНЕРАЦИЯ КЛЮЧЕЙ\n");
element_random(P); //P базова точка
element_random(U); //U особистий майстер
ключ
element_from_hash(Y, "ID", 2); //Y від-
критий ключ користувача Y=H1(ID)
element_mul_zn(V, P, U); //V вироб-
лення відкритого майстер ключа
element_mul_zn(X, Y, U); //X вироб-
лення особистого ключа користувача
element_printf("Y = %B\n", Y);
element_printf("P = %B\n", P);
element_printf("V = %B\n", V);
element_printf("X = %B\n", X);
printf("ПОДПИСЬ\n");
//element_random(k); //K особис-
тий сеансовий ключ, set()
element_set_str(k,
"83877189269548132578866982247987537472729789753
", 10);
element_printf("K1 = %B\n", k);
element_pairing(t1, X, P); //спарювання
element_pairing(t1, X, P)
element_pow_zn(pi, t1, k); //П передпід-
пис
element_to_mpz(t2, pi); //FE2BS()
element_from_hash(t3, "Message", 7); //M
- елемент - ціле mod r
element_mul_mpz(R, t3, t2); //R відкритий
сеансовий ключ
element_mul_zn(t4, X, R);
element_mul_zn(t5, X, k);
element_neg(t4, t4);
```

```
element_add(S, t4, t5); //S друга час-
тина підпису
//(M || (R,S))
printf("Подпись сообщения \"Message\"
:\n");
element_printf("t4 = %B\n", t4);
element_printf("t5 = %B\n", t5);
element_printf("S = %B\n", S);
element_printf("R = %B\n", R);
mpz_init(t22);
printf("ПОДПИСЬ 2 ----- \n");
// k2 = r - k
element_set_str(k2,
"64687362939590348878225226332351736393324676986
4", 10);
element_printf("K2 = %B\n", k2);
element_pairing(t12, X, P); //спарювання
element_pairing(t1, X, P)
element_pow_zn(pi2, t1, k2); //П передпі-
дпис
element_to_mpz(t22, pi2); //FE2BS()
element_from_hash(t32, "Message2", 8);
//M - елемент - ціле mod r
element_mul_mpz(R2, t32, t22); //R від-
критий сеансовий ключ
element_mul_zn(t42, X, R2);
element_mul_zn(t52, X, k2);
element_neg(t42, t42);
element_add(S2, t42, t52); //S друга час-
тина підпису
//(M || (R,S))
printf("Подпись сообщения \"Message2\"
:\n");
element_printf("t42 = %B\n", t42);
element_printf("t52 = %B\n", t52);
element_printf("S2 = %B\n", S2);
element_printf("R2 = %B\n", R2);
element_t R1R2, PROB_X;
element_init_Zr(R1R2, pairing);
element_add(R1R2, R, R2);
element_invert(R1R2, R1R2);
element_t S12;
element_init_G1(S12, pairing);
element_init_G1(PROB_X, pairing);
element_add(S12, S, S2);
element_mul_zn(PROB_X, S12, R1R2);
element_neg(PROB_X, PROB_X);
element_printf("PROB_X = %B\n", PROB_X);
if (!element_cmp(X, PROB_X)) { //R' !=
R
printf("Атака удалась!\n");
} else {
printf("Атака НЕ удалась!!\n");
}
```

Як видно із рис. 1 та 4, отримуємо однакові значення особистого секретного ключа підписувача X . Отже, атака на зв'язаних ключах на механізм ЕП IBS-1 є реалізованою.

Для захисту від атаки такого типу можна використовувати, наприклад, такі механізми захисту ЕП IBS-1 [3, 5]:

1. На основі шифрування підписаних повідомлень з використанням симетричних чи асиметричних шифрів. З точки зору складності (швидкодії) шифрування та стійкості, краще застосовувати симетричні шифри – блокові чи потокові. Тоді криптоаналітику потрібно буде розв'язувати систему із $2i+1$ невідомими, але для системи з i рівняннями. Така задача

при реальних значення параметрів є експоненційно складною.

2. Іншим механізмом захисту від атак на зв'язаних ключах ЕП IBS-1 є виключення можливостей зв'язування одноразових ключів K у процесі

здійснення підписування потоку повідомлень. Вказане може бути здійснене на основі застосування апаратних чи апаратно-програмних засобів ЕП, які виключали б можливість втручання в процес підписування повідомлень. Можливі і інші механізми ЕП.

```

ГЕНЕРАЦИЯ КЛЮЧЕЙ
Y = [593230625408230976423088865774888542966342760890442488500364728071955217145
5968630184954193772346790969340560575632423336371811405065993426223701420164427,
4262049904246927426128441331377983508893642561995061504433261674220480551590029
12162089661697059026635829505986683515385077853979316744506279398301219993 ]
P = [2701975567765339491488496188447629317655507077705374885908045583262796995876
0818260934457746667124645145442917011030032304690396710084139129332492321384852,
8036212800796297638135061300878953762244258801764873766627391069449902392124124
827677452893129075187573347974634231250930069902301959150246651301329392124 ]
U = [334100411354909943715821277186015471679068985537637432523981817175740324358
9917100836271445511960524742713576835425723662282401080145092731530098142727404,
2540139100324430192798965226764816271267295020974582596605402795233577244598215
697132142232180614415261091892650257641785623116236840709446738740161446625 ]
X = [419071389125103182337569222029489718612847118512671340260097228739273491777
1313619812225656535043189306749274658236941331989035146601906646406092896901253,
8438819796088812178168911716336244359364391724668701944168765977096160411910890
60023068649150814988266464599089663087694367231347196662518713780770022382 ]
    
```

Рис. 1. Генерація ключів підпису

```

ПОДПИСЬ
K1 = 83877189269548132578866982247987537472729789753
Подпись сообщения "Message" :
t4 = [67347877286011863388207668409711979007138030523294987792862173651973228873
55379918932248992347728993781717261641966975860127312340012985990456531112494312
349595228196708259844157526108969597365885085001941779941040519518670468561415
6387094967033539953265802651113131268450882581817017007880658595139768400149 ]
t5 = [64449811951918345212021211946098316205527379590633843557430788576771095756
92014758805375358475424131313343924428048179719551060913105168648727654774974884
843699517765632470414053650876960880553588029219036373489117908385307596952208
6948196033698409118828774698789062062384848587345210660172031578083013836672 ]
S = [427062646567887909956623039611779902681213584947567876323045428744330082084
5673319362159055044324936578530303225829549265053878580620897336837283077507944,
4996985722636512529485146925358583302373753475472300417820804314508216485924225
837213246608986771664206682022465894629985355230385824635485673205495818777 ]
R = 300653797268392864499462591559411719048525745039
    
```

Рис. 2. Підпис для повідомлення "Message"

```

ПОДПИСЬ 2 -----
K2 = 646873629395903488782252263323517363933246769864
Подпись сообщения "Message2" :
t42 = [2548204586908278482664579611626562107836296954696684853044919159778340128
32432680925968407571017729060511502710953026679821579735627878324181878924456223
7, 35967025223452103198406163061521922456824847444185637832971292651460071925028
3886994656999072352428170549737318646989293590786112496644058890935254967810 ]
t52 = [6444981195191834521202121194609831620552737959063384355743078857677109575
692014758805375358475424131313343924428048179719551060913105168648727654774974884
4, 34371562200698781829724547598444101027100290722384447613747431541339966135813
6008882591481013543392648457069707519932610690368156657309293347046984388119 ]
S2 = [74061812041416760208377069138075687317902398556899055310386085908472843324
72836524689384395000015167204783546198055839029741288259386614317991416470293884
788363026006614953814318957678503014377939545974110748516792516384007921903411
522080876373423626480752774466555536460304574212792435280360284361366980758 ]
R2 = 599166079184009640143826120256023298147531419708
    
```

Рис. 3. Підпис для повідомлення "Message2"

```

PROB_X = [4190713891251031823375692220294897186128471185126713402600972287392734
91777131361981222565653504318930674927465823694133198903514660190664640609289690
1253, 84388197960888121781689117163362443593643917246687019441687659770961604119
1089060023068649150814988266464599089663087694367231347196662518713780770022382 ]
    
```

Рис. 4. Результат атаки на зв'язаних ключах (знаходження особистого ключа X)

ВИСНОВКИ

1. В процесі удосконалення ЕП, запропоновано алгоритм ЕП IBS-1 на ідентифікаторах зі спарюванням точок ЕК, де як особистий ключ запропоновано використовувати точку еліптичної кривої X . В результаті при перехопленні i підписаних повідомлень для визначення довгострокового ключа X необхідно розв'язувати систему рівнянь з $i+1$ невідомим, i з яких є великими випадковими числами, одне – X є

точкою ЕК. У процесі аналізу не було виявлено ефективних методів розв'язку такої системи.

2. У процесі досліджень виявлено, що криптоперетворення ЕП IBS-1 не забезпечує криптографічної стійкості проти атак на зв'язаних ключах. При чому було отримано дві різні математичні моделі здійснення атаки на зв'язаних ключах – системи (8–9) та (10). Вказані атаки мають поліноміальну складність.

3. Реалізовано програмну модель здійснення атаки на зв'язаних ключах на механізм ЕП IBS-1. Дана

програмна модель базується на математичній моделі атаки, що описується системою (10).

4. Таким чином, як математично, так і програмно показано, що алгоритм ЕП IBS-1 є нестійким проти атаки на зв'язаних ключах, тому в ході його застосування потрібно використовувати механізми захисту від таких атак. Механізми захисту також запропоновані – основними з них є шифрування підписаних повідомлень та застосування кваліфікованих апаратно-програмних засобів ЕП.

Література

- [1] Neal Koblitz A riddle wrapped in an enigma / Neal Koblitz, Alfred J. Menezes // Режим електронного доступу: <https://eprint.iacr.org/2015/1018.pdf>.
- [2] PBC Library: The Pairing-Based Cryptography Library [E-resource]. – Access mode: <https://crypto.stanford.edu/pbc/manual/>.
- [3] Горбенко І.Д., Горбенко Ю.І. Прикладна криптологія. Теорія. Практика. Застосування: монографія. Харків: «Форт», 2012. – 870 с.
- [4] Горбенко Ю.І. Електронні підписи на основі ідентифікаторів та бінарного відображення / Ю.І. Горбенко, Р.С. Ганзя, О.С. Акользіна // Прикладна радіоелектроніка. – Х. : Харківський національний університет радіоелектроніки, 2015. – Т. 14, № 4 – С.284–290.
- [5] Горбенко Ю.І. Сутність та умови здійснення атаки на зв'язаних ключах відносно електронних підписів IBS-1 та IBS-2 ДСТУ ISO/IEC 14888-3 / Ю.І. Горбенко, М.В. Єсіна, В.А. Кулібаба // Системи обробки інформації. – Х. : Харківський університет Повітряних Сил, 2016. – № 7(144) – С.113–118.
- [6] Інформаційні технології – Методи захисту – Цифрові підписи з доповненням – Частина 3. Механізми, що ґрунтуються на дискретному логарифмі : (ISO/IEC 14888-3:2008, IDT) ДСТУ ISO/IEC 14888-3:2014 : 2014. – 113 с.



Єсіна Марина Віталіївна, аспірантка факультету комп'ютерних наук, кафедри безпеки інформаційних систем і технологій Харківського національного університету імені В.Н. Каразіна. Наукові інтереси: електронний підпис, атаки на електронний підпис, криптографічний захист інформації.



Кулібаба Владислав Андрійович, магістр факультету комп'ютерних наук, кафедри безпеки інформаційних систем і технологій Харківського національного університету імені В.Н. Каразіна. Наукові інтереси: криптографічний захист інформації, математичні та програмні моделі реалізації атак на електронні підписи.

УДК 004.056.55

Математическая и программная модели реализации атаки на связанных ключах относительно механизма электронной подписи IBS-1 / М.В. Єсіна, В.А. Кулібаба // Прикладна радіоелектроніка: науч.-техн. журнал. – 2016. – Том 15, № 3. – С. 157 – 161.

В работе рассматривается состояние защищенности электронных подписей на основе спаривания точек эллиптической кривой от атаки на связанных ключах. Определяются условия и возможности организации и реализации этого вида атак. Строятся математическая и программная модели реализации атаки на связанных ключах относительно механизма электронной подписи IBS-1. Предоставляются рекомендации относительно защиты от указанных уязвимостей, в том числе в пост квантовый период.

Ключевые слова: атака, электронная подпись, связанные ключи, спаривание точек эллиптической кривой.

Ил.: 04. Табл.: 01. Библиогр.: 06 назв.

UDC 004.056.55

Mathematical and program models of related keys attack implementation on electronic signature IBS-1 mechanism / M.V. Yesina, V.A. Kulibaba // Applied Radio Electronics: Sci. Journ. – 2016. – Vol. 15, № 3. – P. 157 – 161.

The paper deals with the security of electronic signatures based on an elliptic curve points pairing from a related keys attack. The conditions and possibilities of this attack type organization and implementation are defined. Mathematical and program models of related key attack implementation on the electronic signature IBS-1 mechanism are constructed. Recommendations on protection from these vulnerabilities, including those in the post quantum period are provided.

Keywords: attack, electronic signature, related keys, elliptic curve points pairing.

Fig.: 04. Tab.: 01. Ref.: 06 items.

ИССЛЕДОВАНИЕ ВОЗМОЖНОСТИ ИСПОЛЬЗОВАНИЯ ЯЗЫКОВ ФУНКЦИОНАЛЬНОГО ПРОГРАММИРОВАНИЯ ПРИ МОДЕЛИРОВАНИИ МЕТОДОВ КРИПТОГРАФИЧЕСКИХ ПРЕОБРАЗОВАНИЙ

Е.Г. КАЧКО, Д.К. ТЕЛЕВНЫЙ

Статья посвящена проблеме возможности моделирования методов криптографических преобразований в контексте функционального программирования. Проанализированы основные особенности чисто функциональных языков, а также математический базис симметричных шифров. Выявлены и обоснованы преимущества и недостатки функционального программирования при реализации существующих алгоритмов криптографических преобразований, моделировании и верификации создаваемых. На основе предлагаемого исследования авторами приведены доказательства о целесообразности использования такого подхода как инструмента при моделировании и верификации преобразований.

Ключевые слова: функциональный язык, Haskell, C++, симметричные шифры, AES.

ВВЕДЕНИЕ

Для современных криптографических алгоритмов к условиям хорошего алгоритма относят не только открытость, криптостойкость, но и возможность легкой реализации на различных программных и аппаратных архитектурах. Так одним из способов повышения быстродействия является распараллеливание вычислений на мульти- и многопроцессорной конфигурации. [1]

Отличным выбором при моделировании распараллеливаний новых методов является использование функциональных языков для более точного описания возможности мультипоточного выполнения при имплементации алгоритма на разных языках. В работе исследуется возможность использования функциональных языков на примере криптографического алгоритма AES.

1. ОСНОВНАЯ ИДЕЯ ИССЛЕДОВАНИЯ

Парадигма функционального программирования трактует процесс вычисления функций в математическом понимании последних, в отличие от понимания функции как подпрограммы в процедурном программировании.

Некоторые подходы программирования специфичны только функциональной парадигме и чужды другим подходам (процедурному и особенно ООП). Но учитывая то, что большинство современных языков реализуют гибридные парадигмы, возможно использование особенностей ФП.

К особенностям ФП можно отнести:

– Представление функции в ЯП – в виде математических функций высших порядков, т.е. основная идея состоит в том, что функция не отличается от других объектов данных. Это значит что ФВП могут принимать функции в качестве аргументов и возвращать функцию в качестве результата. Подходу возможен благодаря использованию карринга (каррирования) предложенного Хаскелом Керри.

– Использование чистых функций как таких, что не имеют побочных эффектов на состояние процесса выполнения. Такие функции легко могут быть мемоизированы (результаты вычислений заносятся в таблицу и при повторном вызове подставляются в виде константы). Именно эти функции позволяют ценой небольшого увеличения расхода памяти оптимизировать процесс выполнения.

– Рекурсия является одним из возможных средств организации цикла в ФЯП (в идее ФП отсутствует понятие цикла), может потребоваться увеличение стека вызовом, что можно обойти используя хвостовую рекурсию.

Как говорилось ранее pure функции, являясь независимыми от состояния программы, позволяют оптимизировать код в различных потоках. Любая такая функция, не имея побочных эффектов, может быть представлена в виде абстрактного листа дерева выполнения, где количество листьев в узле – потоки, которые могут выполняться независимо. Кроме этого аргументом функции может быть ссылка на другую функцию. [2]

Далее приведен пример того, как правильное применение идеи функционального программирования может помочь при моделировании новых криптографических преобразований.

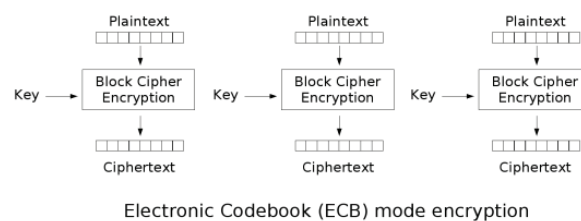


Рис. 1. Схема режима электронной книги

Начать следует с того, что любая криптографическая функция (допустим симметричный шифр) является математической функцией.

$$C = E(P, K); \quad (1.1)$$

$$P = D(C, K). \quad (1.2)$$

Формулы (1.1) и (1.2) представляют собой функции из двух параметров с возвращаемым значением. С точки зрения ФП, E и D – функции высшего порядка, где P – блок открытого текста (длина зависит от шифра), C – блок шифротекста, K – криптографический ключ. Функции (1.1) и (1.2) являются также чистыми (англ. Pure), что значит они не обладают состоянием и контекстно-безопасны.

Рассмотрим это на примере режимов шифрования. Каждый режим представляет собой функцию над массивом блоков (блочных шифров) данных (допустим выравнивание и дописывание до кратности уже сделано) и ключа. Также встречается применение вектора инициализации в сцепленных режимах.

Режим ECB (режим электронной книги) не предполагает сцепления блоков. На рис.1.1 изображена схема режима электронной книги. [3]

Так, преобразование в этом режиме (используя симметричный шифр в качестве параметра) можно изобразить в виде функции высшего порядка.

$$C = ECB(E, P, K); \quad (1.3)$$

$$P = ECB(D, P, K). \quad (1.4)$$

где E, D – функции (1.1) и (1.2), P – открытый текст, C – шифротекст, K – симметричный ключ. Стоит учитывать, что расписание ключей для этого режима одинаково для всех данных и в императивных языках его стоит вычислять перед началом шифрования для более быстрого доступа. С точки зрения ФП (1.3) – pure, поэтому K может быть не только массивом из слов или байт, но и функцией.

Давайте рассмотрим реализацию этого режима на языке F#. Далее приведен код объявления этой функции и ее тела. Пусть уже задан симметричный шифр в виде функции заглушки, принимающий два аргумента и возвращающий значение зашифрованного блока.

```
let aesEncipher (cipherKey: int array)
(plaintext: int array) =
    //cryptotransformation over one
    block
```

Аргумент cipherKey может быть смело заменен функцией, возвращающей ключ для каждого раунда. Не стоит забывать, что функции ФЯ могут быть каррированы и декарированы.

Так общая сигнатура функции генерации расписания ключа выглядит как:

```
let rec scheduleKey (key: int array)
(numberOfRound: int) =
```

```
// return new key for specific
round
```

Рассмотрим реализацию режима ECB. В приведенном коде в качестве первого параметра представлено объявление отображения функции. Таким образом в качестве этого параметра может быть передана как заранее определенная функция, так и лямбда-функция.

```
let ecbMode (f: int array -> int array ->
int array) (plaintext: int array array)
(key: int array) =
    let curriedKeyFunc = scheduleKey
key
    let curriedCipher = f
curriedKeyFunc
    Parallel.map curriedCipher
plaintext;
```

Функция curriedCipher будет применена ко всем элементам plaintext. Чтобы распараллелить обработку массива, нужно место apply.map вызвать Parallel.map.

2. МОДЕЛИРОВАНИЕ ШИФРА RIJNDAEL В ФУНКЦИОНАЛЬНОМ КОНТЕКСТЕ

Представим реализацию основного раунда AES-128 в Haskell и реализация на C.

В Haskell обозначим представление, объявленное в следующем виде и представленное, как список списков 8-битных слов (байт). Благодаря декларативным возможностям функциональных языков, функция aesMainRound, как показано ниже, реализована композицией составных функций преобразований.

```
type State = [[Word8]]
aesMainRound :: State -> State -> State
aesMainRound rk = addRoundKey rk .
    MixColumns. shiftRows
.
    subBytes
```

В языке C данный полный алгоритм шифрования блока выглядит следующим образом

```
AddRoundKey(0, state);
for (unsigned round = 1; round < Nr;
round++){
    SubBytes(state);
    ShiftRows(state);
    MixColumns(state);
    AddRoundKey(round, state);
}
SubBytes(state);
ShiftRows(state);
AddRoundKey(Nr, state);
```

Рассмотрим функцию реализацию subBytes в этих языках. Так, в Haskell данная функция, определена таким образом.

```
subBytes :: State -> State
subBytes = map row
           where row = map subByte

subByte :: Word8 -> Word8
subByte x = sbox ! x

sbox :: Array Word8 Word8
sbox = array (0, 255) $ zip [0..] vals
      where vals = [ 0x63, 0x7c, ...,
0x16 ]
```

Функция `subBytes` использует один из подходов в функциональных языках, известный как маппинг. Так, `map` для всех элементов списка `State` вызывает функцию `row`, которая объявлена внутри `subBytes`, `where` – ключевое слово для определения локальных функций. Таким образом из кода видно, что функции находятся в суперпозиции, объявленной декларативно. Функция `zip` – объявлена в библиотеке. Результатом этой функции является список пар элементов двух входящих списков. `$` – правоассоциативный оператор приложения, в данном контексте подставляет значение возвращаемое, `zip` в `array`. `!` – оператор индексирования.

В языке C данная реализация выглядит следующим образом

```
// case for matrix of bytes
unsigned char* t = (unsigned
char*)state;
for (int i = 0; i < 16; i++){
    t[i] = sbox[t[i]];
}

// case for 32word array
unsigned value, result, temp;
for (auto i = 0; i < Nb; i++){
    temp = state[i];
    value = 0xff & temp;
    result = sbox[value];
    value = 0xff & (temp >> 8);
    result |= (unsigned)sbox[value] << 8;
    value = 0xff & (temp >> 16);
    result |= (unsigned)sbox[value] << 16;
    value = 0xff & (temp >> 24);
    result |= (unsigned)sbox[value] << 24;
    state[i] = result;
}
```

В обоих вариантах массив подстановок уже определен заранее для оптимизации. В случае с AES, где эти значения предопределены, это оправдано. Однако для других алгоритмов, где используется другая схема генерации блока постановок, необходимо использовать функцию генерации. В Haskell это делается заменой параметра в функции `zip`. [4]

Другим важным этапом в SP – модели является линейное преобразование перестановкой. Можно привести две реализации заданного этапа.

```
shiftRows :: State -> State
```

```
shiftRows xss = [ shift n xs | (n,
xs)<-
zip [0..] xss ]
where shift n xs =
drop n xs ++ take n xs

shiftRows :: State -> State
shiftRows [[a1, a2, a3, a4],
[b1, b2, b3, b4],
[c1, c2, c3, c4],
[d1, d2, d3, d4]] =
[[a1, a2, a3, a4],
[b2, b3, b4, b1],
[c3, c4, c1, c2],
[d4, d1, d2, d3]]
```

Первая функция достает каждую строку из `xss`, выполняем `zip` с функцией `shift` для промежутка значений от 0 до 3. Shift проводит циклический сдвиг.

Второй вариант выполнен в декларативном стиле и использует подход, известный как `pattern matching`. Такой подход потребляет больше памяти, однако его можно использовать, как наглядный пример в модульном тестировании при верификации частей алгоритма.

В языке C данная функция выглядит следующим образом.

```
unsigned temp;
//shifting 2st row
temp = state[1];
state[1] =(temp >> 8) | (state[1] <<
24);
//shifting 3nd row This can be modified
by
//adding XOR oper to 2 opernads;
temp = state[2];
state[2] = (temp << 16)|(state[2] >>
16);
//shifting 4th row;
temp = state[3];
state[3] = (temp >> 24)|(state[3] << 8);
```

В приведенном выше коде показано, что для циклического сдвига используются побитовые операции.

Функция `mixColumns` является камнем преткновения в данном шифре. Ее трудно реализовывать в обоих языках, однако Haskell позволяет упростить процесс за счет декларативных возможностей. Пример функции приведен ниже.

```
mixColumns :: State -> State
mixColumns = transpose . map mixColumn
. Transpose

mixColumn :: [Word8] -> [Word8]
mixColumn [a0, a1, a2, a3] =
[b0, b1, b2, b3]
where b0 = mult2 ! a0 `xor` mult3
```

A1

```

        `xor` a2 `xor` a3
    b1 = a0 `xor` mult2 ! a1
`xor`
        mult3 ! a2 `xor` a3
    b2 = a0 `xor` a1 `xor` mult2
!
        a2 `xor` mult3 ! a3
    b3 = mult3 ! a0 `xor` a1
`xor`
        a2 `xor` mult2 ! a3

mult2, mult3 :: Array Word8 Word8
mult2 = array (0, 255) $ zip [0..] vals
    where vals = [0x00, 0x02 ..., 0xe5
]

mult3 = array (0, 255) $ zip [0..] vals
    where vals = [0x00, 0x03 ..
0x1a]

```

Определение `mixColumns` демонстрирует преимущества функционального подхода.

В языке C данное преобразование имеет следующую реализацию:

```

#define xtime(x) ((x<1) ^ ((x>7) & 1) *
0x1b))
#define Multiply(x,y) (((y & 1) * x) ^
((y > 1 & 1) * xtime(x)) ^ ((y > 2 & 1) *
xtime(xtime(x))) ^ ((y > 3 & 1) *
xtime(xtime(xtime(x)))) ^ ((y > 4 & 1) *
xtime(xtime(xtime(xtime(x))))))

void MixColumns(unsigned* state)
{
    int i;
    unsigned char* st;
    unsigned char Tmp, Tm, t;
    for (i = 0; i < 4; i++)
    {
        st = (unsigned char*)state;
        t = st[i];
        Tmp = st[i] ^ st[4 + i] ^
            st[8 + i] ^ st[12 + i];
        Tm = st[i] ^ st[4 + i];
        Tm = xtime(Tm);
        st[i] ^= Tm ^ Tmp;
        Tm = st[4 + i] ^ st[8 + i];
        Tm = xtime(Tm);
        st[4 + i] ^= Tm ^ Tmp;
        Tm = st[8 + i] ^ st[12 + i];
        Tm = xtime(Tm);
        st[8 + i] ^= Tm ^ Tmp;
        Tm = st[12 + i] ^ t;
        Tm = xtime(Tm);
        st[12 + i] ^= Tm ^ Tmp;
    }
}

```

3. ПРЕИМУЩЕСТВА ФУНКЦИОНАЛЬНОГО ПОДХОДА С ТОЧКИ ЗРЕНИЯ БЕЗОПАСНОСТИ

Доказано, что, и функциональные языки, и императивные являются Тьюринг-полными, т.е. множество вычислений включает множество, которое использует Тьюринг-машина. Тем не менее это свойство не значит, что программа безопасна. Пять принципов функциональных языков предоставляют функции, которые заставляют или улучшают безопасность ПО, построенного на этих языках.

3.1 Функции с параметрами и результатами

Все процедуры в функциональных языках представляют из себя функции и четко выделяют входящие значения от возвращаемых. Функции следуют моделям функций в математической теории: параметры принадлежат определенному домену значений, а результаты находятся в определенном множестве. Следовательно – функция это математическое отображение параметров на значение. Это заставляет функциональное программирование реализовывать вычисления таким образом, что вне зависимости от порядка выполнения выражений программа возвращает один и тот же результат для разных параметров. Это обеспечивает безопасное выполнение с детерминированным поведением, обеспечиваемым строго алгоритмом, описанным программно.

3.2 Привязка параметров

Функциональная парадигма не имеет понятия присваивания значений переменной. Эта операция имеет соответствующую команду загрузки в ассемблерных языках. Такие действия тесно связывают программу и аппаратное обеспечение. Хотя некоторые языки 3-го поколения имеют ограничения на присваивание, но оно имеет значение лишь для компилятора, но не для ОС. Следовательно языки содержащие такие конструкции, не могут полностью контролировать значение, т.е. содержимое памяти или отслеживать изменения значения другим источником.

Чисто функциональные языки напротив не имеют синтаксических и семантических средств для присваивания. Параметры, видимые в теле функции – всего лишь аргументы и любые другие локальные параметры, созданные через «where» or «let» clause. В сущности параметры содержат константы, которые только применяются к выражению и не изменяются, пока функция не завершается. Как только параметр в теле функции привязан к значению — оно сохраняется до конца жизни вызова функции. Таким образом ведут себя и параметры функции. При следующем вызове функции параметры не содержат информации о ранних вызовах и не могут меняться пока не закончится выполнение. Это обеспечивает безопасное программирование с гарантией, что ни какие изменения в переменных окружения, не

связанных с параметрами, не могут повлиять на результат.

3.3 Рекурсия

Хвостовая рекурсия позволяет алгоритму быть оптимизированным компилятором. При применении рекурсивных вызовов безопасность обеспечивается на основе доказательной техники, основанной на математической индукции.

3.4 Ссылочная прозрачность

Все функции являются ссылочно-прозрачными. Это значит, что любая функция возвращает один и тот же результат для одинакового набора параметров независимо от контекста вызовов. Это также значит что порядок вычисления аргументов не влияет на результат. Это позволяет инкапсулировать целый алгоритм в функцию, зная что никакое внешнее влияние прямо или косвенно затронет результат.

3.5 Первичные функции

При таком подходе функции могут быть не только переданы в качестве параметра, но и возвращаемы как результат. Это позволяет безопасному программированию создавать программы, которые моделируют математическую композицию. Программа в этом случае становится большим математическим вычислением.

Таким образом при правильном применении функциональной парадигмы основные криптографические атаки могут быть выполнены только используя уязвимость ресурсов, на которых выполняется программа (уязвимости ОС, уязвимости в аппаратном обеспечении). [5]

ЗАКЛЮЧЕНИЕ

В работе была изучена возможность использования функционального подхода для разработки и моделирования блочных шифров. Имплементация алгоритма на Haskell позволяет упростить процесс, сделав разработку более наглядной и соответствующей математической базе данного алгоритма. Разработка приложения в функциональной парадигме позволяет упростить процесс нахождения возможностей оптимизации программного кода. Кроме этого такой подход делает разработку более гибкой – конструирование новых функций на базе уже существующих.

В дальнейшем планируется проводить разработку на более низком уровне математической модели. Такой подход позволит произвести оптимизацию выполнения путем объединения составных частей преобразования.

Литература

- [1] Рябко Б. Я., Фионов А. Н. Криптографические методы защиты информации [Текст].— М.;— Изд-во Горяч.Линия-Телеком, 2005.
- [2] А. Филд, П. Харрисон Функциональное программирование: Пер. с англ. — М.: Мир, 1993. — 637 с, ил. ISBN 5-03-001870-0. Стр. 120 [Глава 6: Математические основы: λ -исчисление].

- [3] NIST Recommendation for Block Cipher Modes [Электронный ресурс]. – Режим доступа: <http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf> – 16.11.2001 г. – Загл. с экрана.
- [4] Cryptographic Block Ciphers in Functional Programming: A Case Study on Feldspar and AES [Электронный ресурс].– Режим доступа: https://gitlab.com/gregor_ulm/dat085_feldspar/blob/master/GregorUlm.FinalReport.May31.pdf – 26.06.2010 г. – Загл. с экрана.
- [5] Арто Мустайоки. Теория функционального синтаксиса. От семантических структур к языковым средствам [Текст]. – Litres, 2014.



Качко Елена Григорьевна, кандидат технических наук, профессор кафедры ПИ ХНУРЭ. Научные интересы: криптография, криптоанализ, параллельные вычисления.



Телевный Дмитрий Константинович, студент группы ИПЗм-15-1 факультета КН ХНУРЭ. Научные интересы: применение функционального подхода в разработке систем, методы симметричного криптопреобразования.

УДК 004.056.55

Дослідження можливості використання мов функціонального програмування для моделювання методів криптографічних перетворень / О.Г. Качко, Д.К. Телевний. // Прикладна радіоелектроніка: наук.-техн. журнал. – 2016. – Том 15, № 3. – С 162 – 166.

Стаття присвячена дослідженню можливості моделювання методів криптографічних перетворень в контексті функціонального програмування. На основі проведеного дослідження автори показали доцільність використання такого підходу, як інструменту під час моделювання та верифікації перетворень.

Ключові слова: функціональна мова, Haskell, C++, симетричні шифри, AES

Лл.: 01. Бібліогр.: 05 найм.

UDC 004.056.55

Studying the possibility of using functional programming languages in modelling methods of cryptographic transformations / O.G. Kachko, D.K. Televnyi // Applied Radio Electronics: Sci. Journ. – 2016. – Vol. 15, № 3. – P. 162 – 166.

The paper is devoted to the possibility of modelling methods of cryptographic transformations in functional programming context. The main peculiarities of purely functional languages as well as the mathematical basis of symmetric ciphers are analysed. Advantages and disadvantages of functional programming at realizing the existing algorithms of cryptographic transformations, modeling and verifying the ones being created have been revealed and substantiated. On the basis of the proposed study the author have presented evidence of the feasibility of such an approach as a tool for modelling and verifying transformations.

Keywords: functional language, Haskell, C++, symmetric ciphers, AES.

Fig.: 01. Ref.: 05 items.

СТАТИСТИЧНІ ДОСЛІДЖЕННЯ СУЧАСНИХ ПОТОКОВИХ ШИФРІВ

О.О. КУЗНЕЦОВ, М.С. ЛУЦЕНКО, А.В. АНДРУШКЕВИЧ, О.М. МЕЛКОЗЕРОВА, Д.В. НОВИКОВА,
А.В. ЛОБАН

Розглядається математична структура нового потокового симетричного шифру «Струмок». Досліджуються його криптографічні властивості шляхом статистичного тестування вихідних послідовностей (гами шифрувальної). Проводиться порівняльний аналіз показників статистичної безпеки з відомими світовими потоковими шифрами.

Ключові слова: потоковий симетричний шифр, криптографічні властивості, статистичне тестування.

ВСТУП

Сучасні симетричні криптоперетворення знайшли найбільше застосування для захисту інформаційно-телекомунікаційних систем і технологій, зокрема, важливої інформації, що є власністю держави, персональних даних, таємної та комерційної інформації та інших даних, які мають підлягати захисту відповідно до законів, наказів і постанов та інших нормативно-правових актів [1 – 25]. Симетричні криптоперетворення застосовуються для захисту інформації практично в усіх криптографічних додатках, зокрема: забезпечення конфіденційності та цілісності інформації та повідомлень на усіх етапах їх життєвого циклу; шифрування в інформаційно-телекомунікаційних системах в різних режимах роботи залежно від вимог, що висуваються; генерація псевдовипадкових послідовностей; криптографічні протоколи автентифікації, встановлення таємниці та ключів, узгодження таємниці та ключів, розподілу таємниці тощо, коли висуваються складні вимоги до складності (швидкодії); криптографічні протоколи електронного цифрового підпису тощо.

Серед симетричних криптоперетворень особливе місце займають потокові алгоритми [1, 2], в яких інформація подається та обробляється у вигляді нескінченного потоку, тобто послідовності, що гіпотетично може бути нескінченної довжини. Головною перевагою такого перетворення є встановлення певної залежності між окремими символами потоку даних, що дозволяє забезпечити додатковий захист від нав'язування хибної інформації, або хибних режимів роботи апаратури захисту чи кінцевого обладнання телекомунікаційних систем і мереж. Відповідно до цього криптографічне потокове перетворення зазвичай користується більшою довірою у користувачів, оскільки потоки даних, що захищаються поточковим алгоритмом, не можуть бути спотворені будь-яким чином, за результатами навмисної або ненавмисної дії користувачів та зловмисників, або якихось випадкових природних чинників чи факторів [1, 2].

Запропонований у [21 – 24] потоковий симетричний шифр «Струмок» застосовує базову структуру

алгоритму шифрування «SNOW2.0» та дозволяє збільшити швидкість формування ключового потоку в ході забезпечення високих та надвисоких показників криптографічної безпеки. Збільшення швидкості досягається за рахунок застосування таблиць передобчислень та перетворень над 64-бітними словами, які розглядаються як елементи скінченного поля $GF(2^{64})$. Використання РЗЛЗЗ з відводами зворотного зв'язку за примітивним поліномом над полем $GF(2^{64})$ дозволяє формувати послідовності максимального періоду, що у сукупності із високонелінійними перетвореннями над послідовністю станів генератору забезпечує властивості випадковості та непередбачуваності формованих послідовностей. Зокрема нелінійний шар перетворень алгоритму «Струмок» засновано на компонентах із національного блокового шифру «Калина» [18, 19], який був стандартизований як ДСТУ 7624:2014 наприкінці 2014 року після тривалих та ретельних досліджень.

Метою цієї роботи є дослідження статистичних властивостей нового потокового симетричного шифру «Струмок» та інших сучасних потокових алгоритмів. Під час дослідження застосовуються відомі методики статистичного тестування [1, 2, 26, 27], які дозволяють шляхом виконання певних розрахунків оцінити показники статистичної безпеки алгоритму, визначити непередбачуваність та випадковість формованих послідовностей.

1. ПОТОКОВИЙ СИМЕТРИЧНИЙ ШИФР «СТРУМОК»

В роботах [20, 21] проведено аналіз та порівняльні дослідження сучасних алгоритмів симетричного криптоперетворення, на основі узагальнення певних математичних моделей і методів потокового шифрування запропоновано новий алгоритм «Струмок». Цей шифр за своєю структурою подібний до стандартизованого у ISO/IEC 18033-4 [7] алгоритму потокового шифрування «SNOW 2.0».

В основі потокового шифру «Струмок» [22 – 24] лежить класична схема підсумовуючого генератора. Криптоалгоритм орієнтований на 64-розрядні

обчислювальні системи, і, відповідно, розмір слова в шифрі визначено рівним 64 бітам. Як вхідні дані використовується 512 (або 1024)-бітний секретний ключ K та 512-бітний вектор ініціалізації IV .

Основними структурними компонентами шифру є регістр зсуву з лінійним зворотним зв'язком (РЗЛЗЗ) та кінцевий автомат (finite-state machine – FSM), в якому виконується нелінійне перетворення. Вхідні дані використовуються для ініціалізації змінної стану $S_i (i \geq 0)$, яка складається з вісімнадцяти 64-бітових блоків, до складу яких входить дві компоненти: 16 змінних $s^{(i)}$ – комірок регістра зсуву з лінійним зворотним зв'язком: $s^{(i)} = (s_{15}^{(i)}, s_{14}^{(i)}, \dots, s_0^{(i)})$ і двох регістрів кінцевого автомату $r^{(i)} : r^{(i)} = (r_2^{(i)}, r_1^{(i)})$. На виході отримуємо ключовий потік (гаму шифрувальну), який формується з 64-бітових слів Z_i .

Схематичне зображення потокового шифру «Струмок» у режимі генерації гами шифрувальної наведено на рис. 1. На рисунку зображено функціонування генератора в довільний момент часу i . Змінну часової залежності i не наведено.

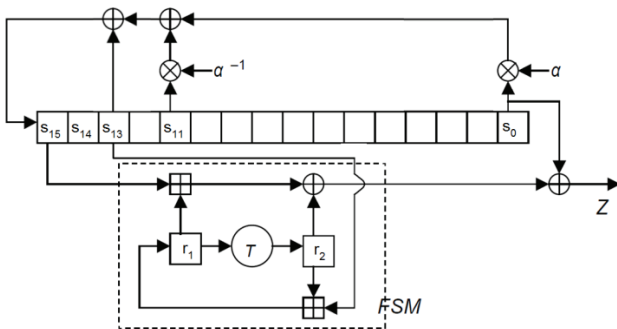


Рис. 1. – Схематичне зображення потокового шифру «Струмок» у режимі генерації ключового потоку

Відводи зворотного зв'язку у РЗЛЗЗ будуються за примітивним над полем $GF(2^{64})$ поліномом $f(x) = x^{16} + x^{13} + \alpha^{-1}x^{11} + \alpha$, де α є коренем примітивного над полем $GF(2^8)$ поліному

$$g(z) = z^8 + g_7z^7 + \dots + g_1z + g_0.$$

В свою чергу поле $GF(2^8)$ будується за примітивним над полем $GF(2)$ поліномом

$$p(y) = y^8 + y^4 + y^3 + y^2 + 1,$$

а коефіцієнти g_0, g_1, \dots, g_7 подаються через ступінь примітивного елементу β поля $GF(2^8)$, тобто β – корінь поліному $p(y)$.

Таким чином, маємо вежу полів:

$$GF(2) \subset GF(2^8) \subset GF(2^{64}) \subset GF(2^{1024}),$$

де

– поле $GF(2^{1024})$ задається відводами зворотного зв'язку РЗЛЗЗ як факторкільце $GF(2^{64})[x]/(f(x))$,

– поле $GF(2^{64})$ задається як факторкільце $GF(2^8)[z]/(g(z))$,

– поле $GF(2^8)$ задається як факторкільце $GF(2)[y]/(p(y))$.

Отже період вихідної послідовності РЗЛЗЗ є максимальним і дорівнює $2^{1024} - 1$. Нижче розглянуто різні варіанти побудови примітивного многочлена $g(z)$ із дослідженням властивостей відповідних вихідних послідовностей.

В ході дослідження були сформовані чотири варіанти поліному $g(z)$:

$$1) \quad g(z) = z^8 + \beta^{170}z^7 + \beta^{166}z^6 + \beta^2z^5 + \beta^{224}z^4 + \beta^{70}z^3 + \beta^2,$$

або при шістнадцятковому поданні коефіцієнтів:

$$g(z) = z^8 + D7z^7 + 3Fz^6 + 04z^5 + 12z^4 + 5Ez^3 + 04;$$

$$2) \quad g(z) = x^8 + \beta^{153}z^7 + \beta^{63}z^6 + \beta^{172}z^5 + \beta^{186}z^4 + \beta^{123}z^3 + \beta^{184}z^2 + \beta^{242},$$

або при шістнадцятковому поданні коефіцієнтів:

$$g(z) = z^8 + 92z^7 + A1z^6 + 7Bz^5 + 6Ez^4 + C5z^3 + 95z^2 + B0;$$

$$3) \quad g(z) = z^8 + \beta^{228}z^7 + \beta^{237}z^6 + \beta^{200}z^5 + \beta^{37}z^4 + \beta^{64}z^3 + \beta^{64}z^2 + \beta^{149},$$

або при шістнадцятковому поданні коефіцієнтів:

$$g(z) = z^8 + 3Dz^7 + 8Bz^6 + 1Cz^5 + 4Az^4 + 5Fz^3 + 5Fz^2 + A4;$$

$$4) \quad g(z) = z^8 + \beta^{14}z^7 + \beta^{151}z^6 + \beta^{158}z^5 + \beta^{117}z^4 + \beta^{95}z^3 + \beta^8z^2 + \beta^{112},$$

або при шістнадцятковому поданні коефіцієнтів:

$$g(z) = z^8 + 13z^7 + AAz^6 + B7z^5 + EDz^4 + E2z^3 + 1Dz^2 + 70,$$

де $\beta = y$ – примітивний елемент поля $GF(2^8)$, корінь двійкового поліному $p(y) = y^8 + y^4 + y^3 + y^2 + 1$ (у шістнадцятковому поданні $\beta = 02$).

Структурно в алгоритмі потокового шифрування «Струмок» можна виділити три основні функції:

– функція ініціалізації *Init*, яка приймає як вхідні дані ключ K (512 біт або 1024 біта) і вектор ініціалізації IV (256 біт або 512 біт), і виробляє початкове значення змінної стану $S_0 = (s^{(0)}, r^{(0)})$;

– функція наступного стану *Next*, яка приймає на вхід змінну стану $S_i = (s^{(i)}, r^{(i)})$ і виробляє наступне значення змінної стану $S_{i+1} = (s^{(i+1)}, r^{(i+1)})$. Функція *Next* може виконуватися в двох режимах, залежно від способу виконання ітерації – як частини реалізації або як частини нормального режиму генерації вихідних даних;

– функція ключового потоку $Strm$, що приймає на вході змінну стану $S_i = (s^{(i)}, r^{(i)})$ і виробляє на виході 64-бітний ключовий потік Z_i .

Схематичне зображення потокового шифру «Струмок» у ході виконання функції $Next$ у режимі ініціалізації $INIT$ наведено на рис. 2.

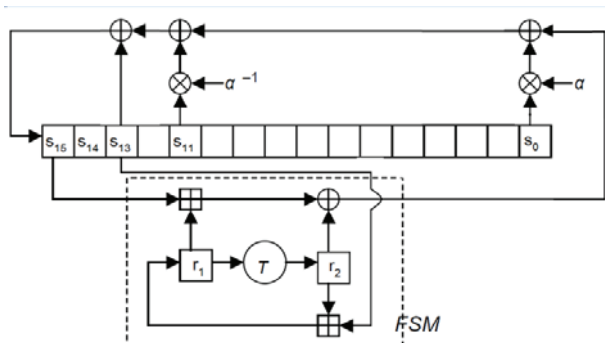


Рис. 2. Схематичне зображення потокового шифру «Струмок» у режимі ініціалізації функції $Next$

Одним з важливих показників криптографічної стійкості генератора ключових потоків є період формованих псевдовипадкових послідовностей. Розглянутий потоковий шифр «Струмок» виконано за схемою 64-розрядного слово-орієнтованого синхронного поточного криптоалгоритму. За рахунок застосування перетворень над 64-бітними словами, які розглядаються як елементи скінченного поля $GF(2^{64})$, та використання РЗЛЗЗ із 16 64-бітними словами забезпечується формування псевдовипадкових послідовностей максимального періоду, що дорівнює $(2^{64})^{16} - 1 = 2^{1024} - 1$ бітів. У сукупності із нелінійними перетвореннями над послідовністю станів генератора забезпечуються властивості випадковості та непередбачуваності формованих послідовностей.

2. МЕТОДИКА СТАТИСТИЧНОГО ТЕСТУВАННЯ

Для проведення експериментальних досліджень криптографічних властивостей потокового шифру «Струмок» було використано статистичне тестування вихідних послідовностей (ключового потоку або гами шифрувальної). До найбільш відомих наборів статистичних тестів належать [1, 2, 26]: DIEHARD, NIST Statistical Test Suite (NIST STS), DieHarder. Їх сутність полягає в перевірці гіпотези про випадковий характер вихідної послідовності досліджуваного криптоалгоритму, тобто в перевірці припущення про те, що сформовані дані не відрізняються в статистичному сенсі від деякої гіпотетичної «випадкової» послідовності.

Під час проведенні експериментальних досліджень було застосовано пакет статистичного тестування NIST STS, який був розроблений в ході проведення конкурсу AES для дослідження генераторів випадкових або псевдовипадкових чисел.

За методикою NIST STS гіпотеза перевіряється за 15 незалежними статистичними тестами (з урахуванням різних вхідних параметрів виконуються 188 тестів), по кожному з яких обчислюється відповідна ймовірність $P_j, j=1, \dots, 188$ проходження тесту. Ця ймовірність використовується правилом прийняття суджень (критерієм згоди) про істинність чи хибність гіпотези, тобто якщо значення ймовірності P_j проходження j -го тесту не нижче деякого порогового значення $\alpha \in [0.96, 0.99]$, наприклад, якщо $P_j \geq 0.99$, тоді гіпотеза H_0 приймається (на j -му тесті). В іншому випадку приймається альтернативна гіпотеза. Аналіз елементів P_j вектора P дозволяє вказати на конкретні дефекти «випадковості» протестованої послідовності, тобто низькі значення P_j вказують на явну відмінність досліджуваної послідовності від реалізації випадкового процесу, виявлене j -м статистичним тестом.

Отже, пакет статистичного тестування NIST STS містить 15 статистичних тестів, але, фактично, залежно від вхідних параметрів обчислюються 188 значень ймовірності P , які можна розглядати як результат роботи окремих тестів. До 15 тестів належать наступні.

1. *Частотний побітовий тест.* Спрямований на визначення співвідношення між нулями та одиницями у двійковій послідовності певної довжини. Для дійсно випадкової бінарної послідовності кількість нулів та одиниць має бути майже однакова. Отже, тест оцінює, на скільки близькою є доля одиниць до 0,5.

2. *Частотний блоковий тест.* Суть тесту полягає у визначенні долі одиниць всередині блоку довжиною m бітів, тобто необхідно з'ясувати, чи дійсно частота повторення одиниць в блоці довжиною m бітів приблизно є рівною $m/2$, як можна було б припустити у випадку випадкової послідовності.

3. *Тест на послідовність однакових бітів.* У цьому тесті відбувається пошук рядків, тобто неперервних послідовностей однакових бітів. Ряд (серія) довжиною k бітів складається з k абсолютно ідентичних бітів, починається та закінчується з біту, який містить протилежне значення. В даному тесті необхідно з'ясувати швидко чи повільно чергуються одиниці та нулі у початковій послідовності.

4. *Тест на найдовшу послідовність одиниць в блоці.* В даному тесті визначається найдовший рядок одиниць всередині блоку довжиною m бітів. Необхідно з'ясувати відхилення від теоретичного закону розподілу максимальної довжини серії одиниць.

5. *Тест рангів бінарних матриць.* Тут здійснюється розрахунок рангів неперетинних підматриць, побудованих з початкової двійкової

послідовності. Метою цього тесту є перевірка на лінійну залежність підрядків фіксованої довжини, що складають початкову послідовність..

6. *Спектральний тест.* Суть тесту полягає в оцінці висоти піків дискретного перетворення Фур'є початкової послідовності. Метою є виявлення періодичних властивостей вхідної послідовності, наприклад, близько розташованих один до одного повторюваних ділянок.

7. *Тест на співпадіння шаблонів, що не перекриваються.* У даному тесті підраховується кількість заздалегідь визначених шаблонів, які знайдені в початковій послідовності. Необхідно виявити генератори псевдовипадкових чисел, що формують занадто часто задані неперіодичні шаблони. Як і в тесті №8 на співпадіння шаблонів, що перекриваються, для пошуку конкретних шаблонів довжиною m бітів використовується вікно також довжиною m бітів. Якщо шаблон не знайдено, вікно зсувається на один біт. Якщо ж шаблон знайдено, тоді вікно пересувається на біт, який є наступним за знайденим шаблоном, та пошук продовжується далі.

8. *Тест на співпадіння шаблонів, що перекриваються.* Суть даного тесту полягає в підрахунку кількості заздалегідь визначених шаблонів, які знайдені в початковій послідовності. Пошук проводиться майже аналогічним способом як у тесті №7.

9. *Універсальний статистичний тест Маурера.* Тут визначається число бітів між однаковими шаблонами в початковій послідовності (міра, що має безпосереднє відношення до довжини стиснутої послідовності). Необхідно з'ясувати, чи може дана послідовність бути значно стиснута без втрат інформації. У разі, якщо це можливо зробити, то вона не є дійсно випадковою.

10. *Тест на лінійну складність.* В основі тесту лежить принцип роботи лінійного регістра зсуву зі зворотним зв'язком. Необхідно з'ясувати, чи є вхідна послідовність досить складною для того, щоб вражатися абсолютно випадковою. Абсолютно випадкові послідовності характеризуються довгими лінійними регістрами зсуву зі зворотним зв'язком. Якщо ж такий регістр занадто короткий, то передбачається, що послідовність не є повною мірою випадковою.

11. *Тест на періодичність.* Даний тест полягає в підрахунку частоти всіх можливих перекривань шаблонів довжини m бітів протягом початкової послідовності бітів. Метою є визначення, чи дійсно кількість появ $2m$ шаблонів, що перекриваються, довжиною m бітів, є приблизно такою як і у випадку абсолютно випадковою вхідної послідовності бітів. Остання, як відомо, володіє одноманітністю, тобто кожен шаблон довжиною m біт з'являється в послідовності з однаковою ймовірністю.

12. *Тест приблизної ентропії.* В даному тесті акцент робиться на підрахунку частоти всіх можливих

перекривань шаблонів довжини m бітів у початкової послідовності бітів. Необхідно порівняти частоти перекривання двох послідовних блоків початкової послідовності з довжинами m та $m+1$ з частотами перекривання аналогічних блоків в абсолютно випадковій послідовності. Цей тест виявляє регулярність властивостей генератора.

13. *Тест кумулятивних сум.* Тест полягає в максимальному відхиленні (від нуля) при довільному обході, визначеному кумулятивною сумою заданих $(-1,+1)$ цифр у послідовності. Необхідно визначити, чи є кумулятивна сума часткових послідовностей, що виникають у вхідній послідовності, занадто великою або занадто маленькою порівняно з очікуваною поведінкою такої суми для абсолютно випадкової вхідної послідовності.

14. *Тест на довільні відхилення.* Суть даного тесту полягає в підрахунку числа циклів, що мають суворо k відвідувань при довільному обході кумулятивної суми. Довільний обхід кумулятивної суми починається з часткових сум після послідовності $(0,1)$ перекладеної у відповідну послідовність $(-1,+1)$. Цикл довільного обходу складається з серії кроків одиначної довжини, виконаних у випадковому порядку. Мета даного тесту полягає у визначенні того, чи відрізняється число відвідувань певного стану всередині циклу від аналогічного числа в разі абсолютно випадкової вхідної послідовності. Фактично даний тест є набором, що складається з восьми тестів, які проводяться для кожного з восьми станів циклу: $-4, -3, -2, -1$ та $+1, +2, +3, +4$.

15. *Інший тест на довільні відхилення.* У цьому тесті підраховується загальна кількість відвідувань певного стану при довільному обході кумулятивної суми. Метою є визначення відхилень від очікуваного числа відвідувань різних станів при довільному обході. Цей тест складається з 18 тестів, що проводяться для кожного стану: $-9, -8, \dots, -1$ та $+1, +2, \dots, +9$.

Проходження кожного з тестів є важливим критерієм оцінки псевдовипадкового генератора [20]. Тому не відповідність за одним чи більше критеріями означає, що ключовий потік не може на високому рівні протистояти криптоаналізу. Якщо, з іншого боку, генератор проходить всі тести, це зовсім не означає захищеність генератора, оскільки такі тести не враховують особливостей реальної конструкції генератора.

Більшість сучасних криптоалгоритмів мають значення P_j перевищують порогове значення і як результат тестування використовують лише число пройдених тестів, тобто число ймовірностей $P_j \geq 0.99$ з множини $P = \{P_1, P_2, \dots, P_n\}$. Позначимо число пройдених тестів для конкретної i -ї вибірки символом X_i , $0 \leq X_i \leq n$, $i=1, \dots, N$, де N – кількість протестованих вихідних послідовностей

криптоалгоритму. Слід зазначити, що значення X_i , так само як і значення з множини $P = \{P_1, P_2, \dots, P_n\}$, що характеризують статистичну безпеку досліджуваного криптоалгоритму, мають стохастичну природу. Ці значення безпосередньо залежать як від властивостей досліджуваного генератора, так і від початкових даних під час проведення експериментальних досліджень. Іншими словами, значення елементів P_j визначаються для конкретної i -ї вибірки, $i=1, \dots, N$, тобто для конкретної вихідної послідовності криптоалгоритму заданої довжини. Різні початкові дані (різні вихідні послідовності заданої довжини в i -му експерименті) можуть давати і різні значення елементів P_j , при цьому відмінності у власних значеннях можуть бути істотними.

Таким чином, кількість пройдених тестів досліджуваного генератором, безпосередньо залежить від обраної вихідної послідовності криптоалгоритму. Для забезпечення заданої достовірності результатів статистичного тестування в роботах [20, 21, 27] запропоновано оцінити математичне сподівання числа пройдених тестів X_i досліджуваного генератором (криптоалгоритмом), розглядаючи при цьому кожне i -те тестування як одне спостереження, тобто як конкретну реалізацію деякої випадкової величини X . Саме цю методику було застосовано під час проведення експериментальних досліджень, отримані результати мають високу точність та достовірність статистичного тестування.

3. РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ

Відповідно до методики статистичного тестування були проведені експериментальні дослідження криптографічних властивостей потокового шифру «Струмок» (були протестовані усі чотири версії алгоритму).

Для порівняння показників статистичної безпеки обрано всесвітньовідомі криптоалгоритми, які стандартизовані на міжнародному або національному рівні та, які на сьогоднішній день мають найбільшу довіру та розповсюдження. Зокрема, були протестовані ключові потоки сучасних потокових шифрів [6 – 16, 25]: «Encoro», «HC-128», «HC-256», «Grain», «MICKEY 2», «MUGI», «Rabbit», «Salsa20», «SNOW 2.0», «Sosemanuk», «Trivium», та вихідні послідовності блокового симетричного шифру «AES» із довжиною ключа 128 та 256 бітів (у режимі зворотного зв'язку за виходом цей шифр можна використовувати як потоковий). Наведемо стислі відомості щодо досліджених криптографічних алгоритмів.

Потоковий симетричний шифр «SNOW 2.0» є генератором ключових потоків [7], який використовує як вхідні дані 128 або 256-бітовий секретний ключ K і 128-бітовий вектор ініціалізації IV . Шифр є слово-

орієнтованим. Автори алгоритму – Томас Йохансон та Патрік Екдаль. Алгоритм було стандартизовано у ISO/IEC 18033-4. Для «SNOW 2.0» максимально рекомендовану кількість біт ключового потоку, виробленого на одній парі (K, IV) , дорівнює $23 \cdot 2^{50}$ біт. Це обмеження виправдане з точки зору забезпечення стійкості алгоритму проти криптоаналітичних атак.

Потоковий симетричний шифр «Sosemanuk» – це синхронний програмно-орієнтований потоковий шифр, який відповідає першому профілю конкурсу eCRYPT [14]. Його довжина ключа може бути обрана між 128 і 256 бітами. Шифр працює з 128 бітовим початковим значенням, при цьому, як стверджується розробниками алгоритму, будь-яка довжина ключа досягає 128-бітного захисту. Алгоритм Sosemanuk використовує деякі основні принципи потокового шифру «SNOW 2.0» і деякі перетворення, отримані з блокового шифру SERPENT.

Потоковий симетричний шифр «Trivium» – це симетричний апаратно-орієнтований паралельний потоковий шифр. Авторами шифру є Крістоф Де Канн'єр і Барт Пренел [15]. Trivium найбільш простий шифр проекту eSTREAM (другий профіль), який демонструє відмінні результати криптостійкості. За специфікацією алгоритм Trivium – це паралельний потоковий шифр, призначений для генерації 2^{64} біт ключового потоку з 80 біт секретного ключа і 80 біт вектора ініціалізації. Шифр є біт-орієнтованим.

Потоковий симетричний шифр «Encoro» – апаратно-орієнтований криптоалгоритм, який описано у [25]. Це байт-орієнтований шифр із довжиною ключа 128 біти та вектору ініціалізації 64 біти. Незважаючи на те, що «Encoro» є апаратно-орієнтованим шифром, він також має і ефективну програмну реалізацію. Для досягнення різних вимог, використовуються байтові операції.

Потоковий симетричний шифр «HC-256», який було розроблено у 2004 році [9]. HC-256 простий, безпечний, програмно-орієнтований шифр з ефективною реалізацією і може вільно використовуватися. Спрощену версію HC-128 було представлено на eSTREAM у першому профілі. Для ініціалізації використовується 256-бітний ключ та вектор ініціалізації довжиною 256 біт. Рекомендована максимальна довжина ключової послідовності – 2^{128} .

Потоковий симетричний шифр «Grain», який було представлено Мартіном Хеллом, Томасом Юханссоном та Віллі Мейєром у 2004 на міжнародному конкурсі eSTREAM за другим профілем (апаратно орієнтовані шифри) [10]. Симетричний алгоритм синхронного поточного шифрування, який орієнтований на використання на обчислювальних машинах з обмеженою кількістю вентилів (gate), невеликими потужністю та обсягом пам'яті. Залежно від апаратної реалізації шифр Grain може бути біт-орієнтованим або слово-орієнтованим.

В Grain v1 на вхід подається ключ довжиною 80 біт та вектор ініціалізації довжиною 64 біти. В основі конструкції алгоритму лежать 2 регістри зсуву – з лінійним та нелінійним зворотним зв'язком та вихідна функція. Рекомендована довжина ключового потоку, який може бути вироблений на одній парі ключ/вектор – 2^{44} біт.

Потоковий симетричний шифр «Mickey», вдосконалену версію 2.0 якого було представлено у 2005 році Стивом Беббіджем та Метью Доддом [11] (розшифровується як Mutual Irregular Clocking KEYstream generator – генератор ключового потоку із взаємно нерівномірним рухом). Його призначено для апаратних платформ з обмеженими ресурсами, тобто потоковий шифр MICKEY був розроблений за другим профілем, як апаратно-орієнтований шифр. Для ініціалізації початкового стану використовуються ключ довжиною 80 біт та вектор ініціалізації довжиною до 80 біт. Максимально можлива довжина ключового потоку дорівнює 2^{40} біт на одному ключі, але з використанням різних векторів ініціалізації однієї довжини. Алгоритм шифрування MICKEY має просту апаратну реалізацію, але при цьому забезпечує високий рівень безпеки. Завдяки використанню нерегулярного руху регістрів зсуву, а також нових методів, забезпечується висока стійкість до певних криптоаналітичних атак.

Потоковий симетричний шифр «MUGI» є генератором ключових потоків, який було рекомендовано проектом CRYPTREC для використання у 2003 році урядом Японії [7]. Алгоритм було стандартизовано у ISO/IEC 18033-4. Як початкові дані MUGI використовує 128-бітовий секретний ключ, 128-бітовий вектор ініціалізації. MUGI використовує нелінійні блоки підстановки та лінійні трансформації з використанням MDS матриці алгоритму AES. Основні конструкції шифру подібні до конструкцій шифру Rapamata. Шифр MUGI є слово-орієнтованим.

Потоковий симетричний шифр «Rabbit», розробниками алгоритму є Мартін Боегсаард, Метте Вестерагер, Томас Педерсен, Йеспер Крістіансен та Ове Скавіньюс [12]. У травні 2005р., цей шифр був представлений на конкурсі eStream у першому профілі – програмно-орієнтовані алгоритми. Алгоритм використовує 128-бітний ключ і 64-бітний вектор ініціалізації. На одній парі ключ/вектор може бути вироблено до 2^{67} бітів ключового потоку.

Потоковий симетричний шифр «Salsa 20», який було розроблено Даніелем Бернштейном [13]. Алгоритм став переможцем конкурсу eSTREAM в першому профілі (програмно-орієнтовані алгоритми). Для ініціалізації внутрішнього стану використовується ключ довжиною 256 біт, 64-бітний nonce та 64-бітна позиція блоку ключового потоку. Максимальна довжина псевдовипадкової ключової послідовності дорівнює 2^{70} біт.

Блоковий симетричний шифр «AES», який стандартизовано в США як FIPS-197 [5]. На міжнародному рівні стандартизовано у ISO/IEC 18033-3 [6]. Використовує ключ довжиною 128, 192 або 256 біт. Залежно від довжини ключа відбувається 10, 12 або 14 раундів шифрування. AES базується на принципі, відомому як мережа заміни-перестановок та, завдяки цьому, має швидку апаратну та програмну реалізацію. У режимі зворотного зв'язку за виходом цей шифр можна використовувати як потоковий.

Для усіх протестованих шифрів було складено статистичні портрети, які наведено на рис. 3 – 18. Під статистичним портретом мають на увазі гістограму, на якій на осі ординат знаходяться вірогідності проходження j-го тесту, а на осі абсцис – номер j-го тесту.

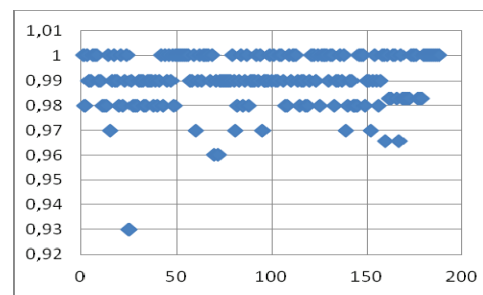


Рис. 3. Статистичний портрет шифру AES-128

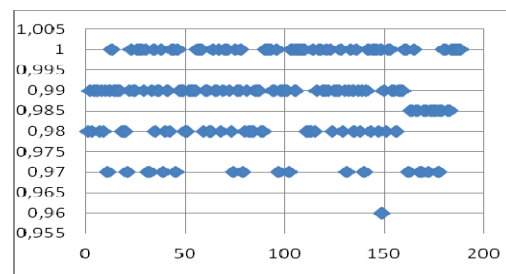


Рис. 4. Статистичний портрет шифру AES-256

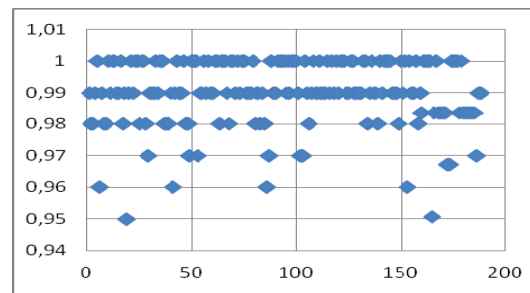


Рис. 5. Статистичний портрет шифру Eponogo

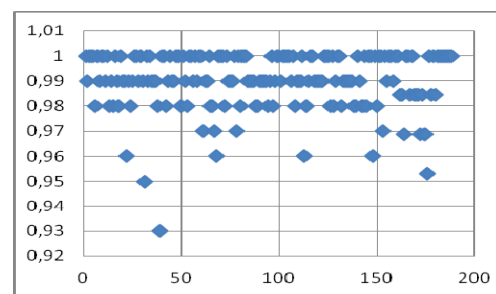


Рис. 6. Статистичний портрет шифру Grain

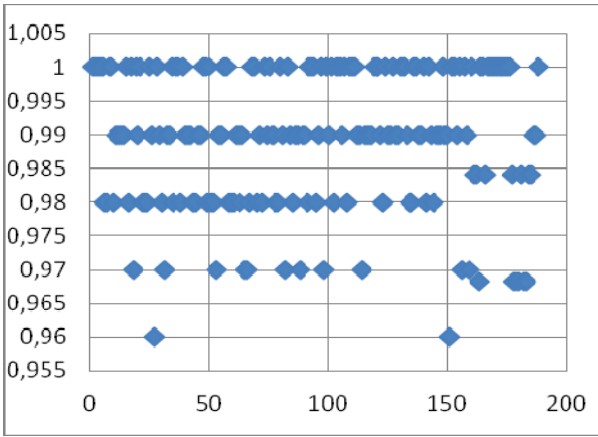


Рис. 7. Статистичний портрет шифру HC-128

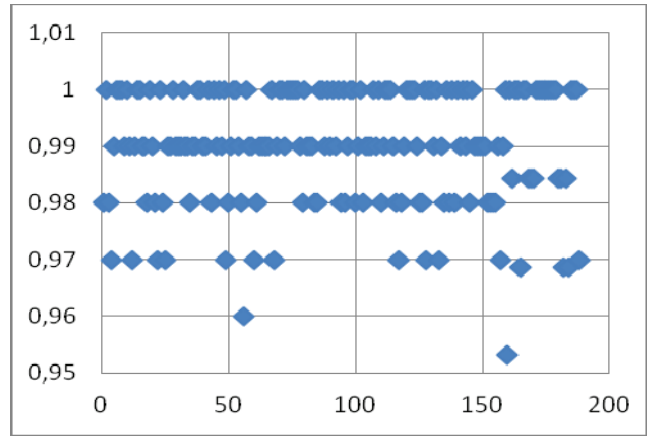


Рис. 11. Статистичний портрет шифру Rabbit

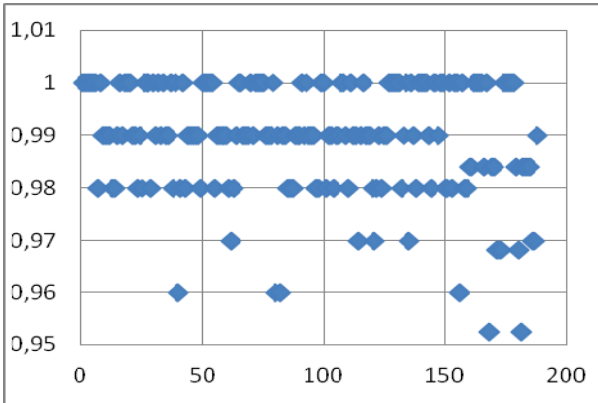


Рис. 8. Статистичний портрет шифру HC-256

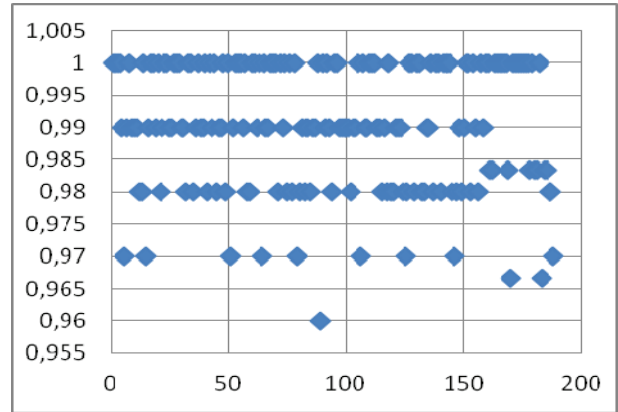


Рис. 12. Статистичний портрет шифру Salsa20

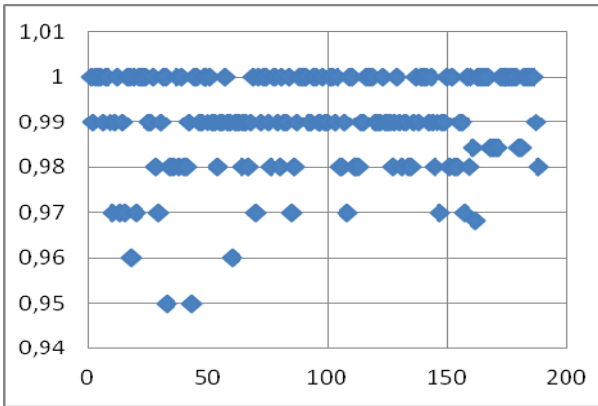


Рис. 9. Статистичний портрет шифру Mickey 2

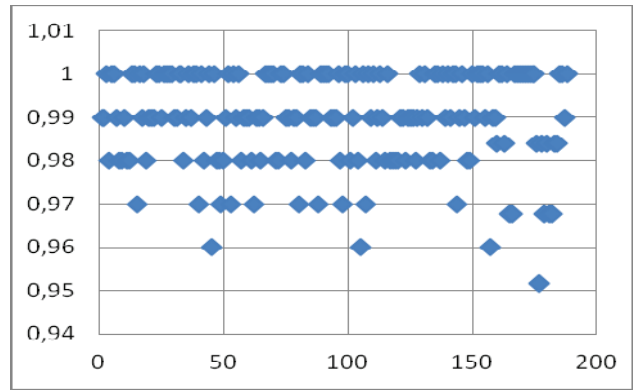


Рис. 13. Статистичний портрет шифру Sosemanuk

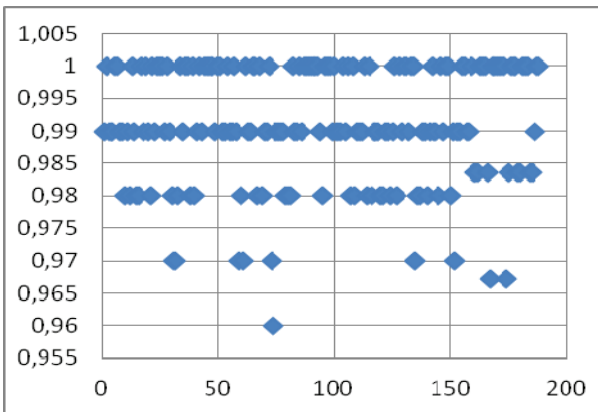


Рис. 10. Статистичний портрет шифру MUGI

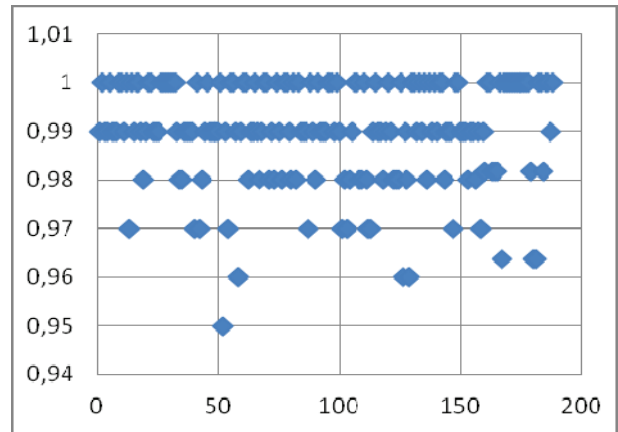


Рис. 14. Статистичний портрет шифру Trivium

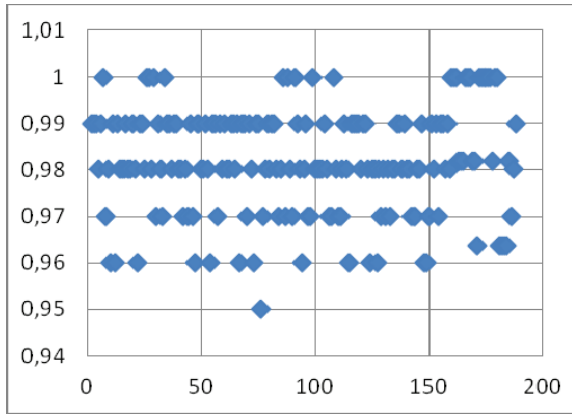


Рис. 15. Статистичний портрет шифру SNOW2.0

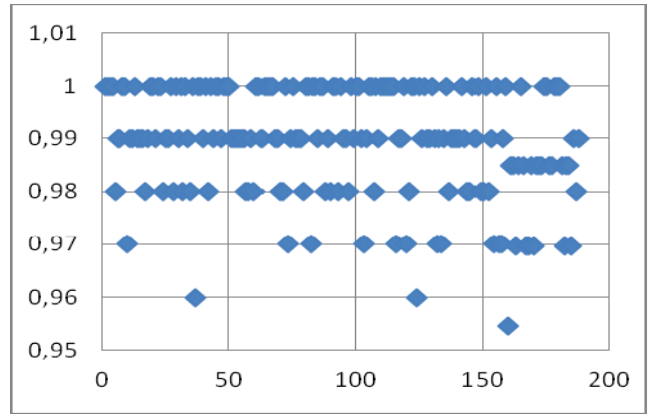


Рис. 19. Статистичний портрет шифру «Струмук»

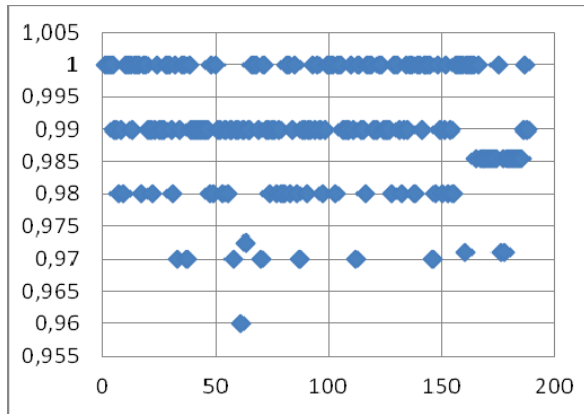


Рис. 16. Статистичний портрет шифру «Струмук»

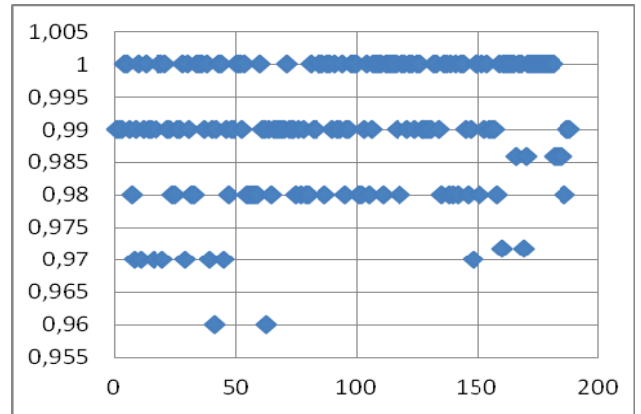


Рис. 20. Статистичний портрет шифру «Струмук»

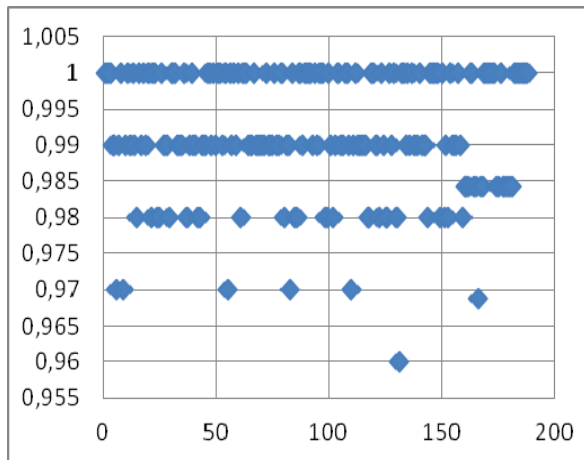


Рис. 17. Статистичний портрет шифру «Струмук»

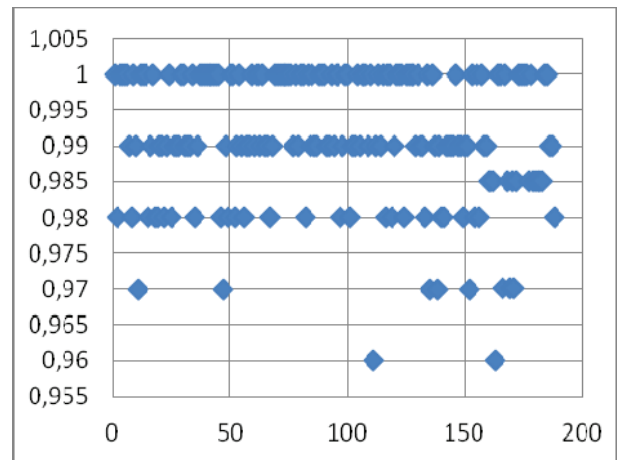


Рис. 21. Статистичний портрет шифру «Струмук»

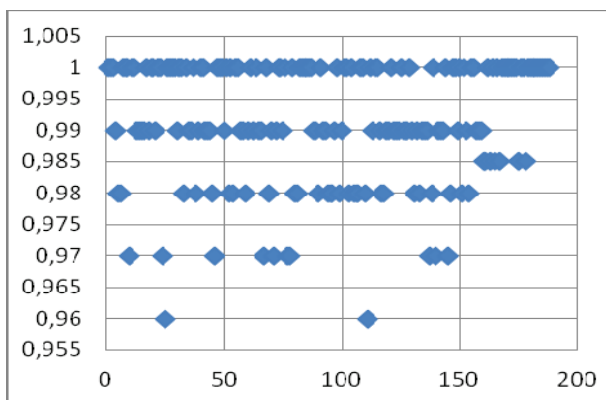


Рис. 18. Статистичний портрет шифру «Струмук»

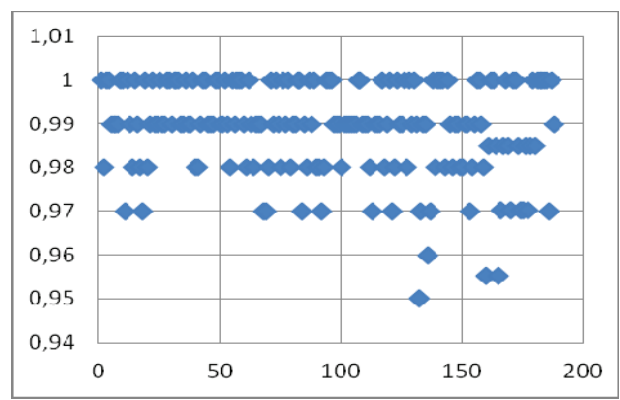


Рис. 22. Статистичний портрет шифру «Струмук»

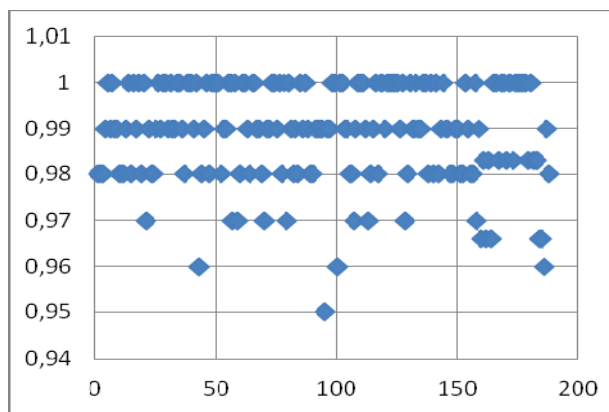


Рис. 23. Статистичний портрет шифру «Струмок»

На рис. 16 – 23 наведено статистичний портрет шифру «Струмок» із параметрами:

- рис. 16, 17 – ключ 512 та 1024 бітів, поліном $g(z)$ за першим варіантом;
- рис. 18, 19 – ключ 512 та 1024 бітів, поліном $g(z)$ за другим варіантом;
- рис. 20, 21 – ключ 512 та 1024 бітів, поліном $g(z)$ за третім варіантом;
- рис. 22, 23 – ключ 512 та 1024 бітів, поліном $g(z)$ за четвертим варіантом.

Нижче у таблиці 1 наведено результати статистичного тестування послідовності, яку було згенеровано парою випадковий ключ K / випадковий вектор ініціалізації IV , для всіх чотирьох варіантів обрання поліному $g(z)$ з довжиною ключа 512 і 1024 біта. Кожна послідовність завдовжки 10^6 біт.

Аналіз даних таблиці 1 показує, що наведені результати тестування приблизно однакові. Але для першого варіанта обрання поліному $g(z)$ показники виявилися дещо вищими, саме цю версію реалізації алгоритму і обрано для подальшого порівняльного аналізу з іншими криптоалгоритмами.

Результати порівняльного аналізу наведено у таблицях 2, 3 з такими позначеннями:

- «IV_const» – послідовності для тестування сформовані парою випадковий ключ K / вектор ініціалізації IV ;
- «K_const» – послідовності для тестування сформовані парою ключ K / випадковий вектор ініціалізації IV ;
- «K_IV» – послідовності для тестування сформовані парою випадковий ключ K / випадковий вектор ініціалізації IV .

В таблицях 1 – 4 наведено такі дані:

- «M096» та «M099» – оцінки математичного сподівання (вибіркові середні) числа пройдених статистичних тестів за критерієм $P_j \geq 0.96$ та за критерієм $P_j \geq 0.99$, відповідно;
- «D096» та «D099» («S096» та «S099») – оцінки дисперсій (середньоквадратичних відхилень) результатів тестування числа пройдених статистичних тестів за критеріями $P_j \geq 0.96$ та $P_j \geq 0.99$, відповідно;
- «P099» – значення довірчої ймовірності для числа пройдених статистичних тестів за критерієм $P_j \geq 0.99$ та при точності $\varepsilon = 2$;
- «P096» – значення довірчої ймовірності для числа пройдених статистичних тестів за критерієм $P_j \geq 0.96$ та при точності $\varepsilon = 1$;
- «Min096» – мінімальні значення числа пройдених статистичних тестів за критерієм $P_j \geq 0.96$.

Наведені результати тестування різних потокових криптоперетворень підтверджують їхні високі криптографічні показники. Досліджувані шифри показали високе число успішно пройдених тестів: 130 – 133 за критерієм $P_j \geq 0.99$ та 186-187 за критерієм $P_j \geq 0.96$. Ці оцінки отримані з високою достовірністю ($P_0 = 0,99$ для $P_j \geq 0.99$ та $P_0 \approx 1$ для $P_j \geq 0.96$).

Таблиця 1

Результати статистичного тестування різних версій алгоритму «Струмок»

Назва алгоритму	M099	D099	S099	P099	M096	D096	S096	P096	MIN
Strumok_1_512	130,01	23,6	4,86	1.00	186,45	1,4555	1,206	1.00	184
Strumok_2_512	132,58	50,20	7,086	1.00	186,63	2,054	1,433	1.00	184
Strumok_3_512	133,75	36,44	6,037	1.00	186,66	1,907	1,381	1.00	182
Strumok_4_512	131,38	55,702	7,463	1.00	186,71	1,452	1,205	1.00	184
Strumok_1_1024	132,83	56,516	7,518	1.00	186,90	0,802	0,896	1.00	185
Strumok_2_1024	132,125	66,22	8,1376	1.00	186,792	1,0162	1,0081	1.00	185
Strumok_3_1024	131,204	29,443	5,4261	1.00	186,122	0,5156	0,7181	1.00	184
Strumok_4_1024	134,14	34,932	5,9102	1.00	186,4	2,516	1,586	1.00	183

Таблица 2

Результаты статистического тестирования алгоритму «Струмок»

	M099	D099	S099	P099	M096	D096	S096	P096	MIN
Strumok-512_K	133,475	77,62	8,81	1,00	186,96	1,6222	1,274	1,00	185
Strumok-512_IV	133,34	35,71	5,9757	1,00	186,921	1,4195	1,191	1,00	184
Strumok-512_K_IV	130,01	23,614	4,8594	1,00	186,455	1,416	1,1901	1,00	184
Strumok-1024_K	130,158	32,866	5,733	1,00	187,099	1,099108	1,0484	1,00	184
Strumok-1024_IV	132,1	45,967	6,7799	1,00	186,911	1,289089	1,1354	1,00	184
Strumok-1024_K_IV	132,56	56,06	7,4873	1,00	186,891	0,791	0,889	1,00	185

Таблица 3

Результаты статистического тестирования современных потоковых шифров

	M099	D099	S099	P099	M096	D096	S096	P096	MIN
AES-128_K_IV	127,07	20,456	4,438	1,00	186,63	0,3191	0,554	1,00	185
Enocoro_K_IV	132,92	51,22	7,157	1,00	187,17	0,79	0,89	1,00	185
HC_256_K_IV	133,75	36,44	6,04	1,00	186,66	1,93	1,381	1,00	182
Snow2.0_K_IV	132,78	23,93	4,89	1,00	186,79	0,43	0,656	1,00	183
Strumok-512_K_IV	130,01	23,6	4,86	1,00	186,45	1,4555	1,206	1,00	184
Strumok-1024_K_IV	132,83	56,516	7,518	1,00	186,90	0,802	0,896	1,00	185
Grain_K_IV	132,36	57,32	7,571	1,00	186,921	1,414	1,185	1,00	182
Mickey_2_K_IV	133,53	61,65	7,85	1,00	186,6	2,302	1,51	1,00	179
MUGI_K_IV	132,227	56,279	7,3295	1,00	186,5	1,0238	0,9886	1,00	185
Rabbit_K_IV	132,65	16,87	4,017	1,00	187,22	0,451	0,657	1,00	185
Salsa20_K_IV	134,16	28,055	5,27	1,00	187,001	1,01	0,99	1,00	183
Sosemanuk_K_IV	131,73	49,36	6,991	1,00	186,8	2,240	1,49	1,00	184
Trivium_K_IV	130,24	99,683	9,935	1,00	187,15	1,49	1,214	1,00	182

Таблица 4

Результаты статистического тестирования алгоритму «Струмок» с разной количеством начальных тактов

Кількість тактувань	M099	D099	S099	P099	M096	D096	S096	P096	MIN
0	130,381	20,141	4,488	1,00	186,905	1,5147	1,2307	1,00	183
16	133,095	87,5147	9,355	1,00	186,714	0,966	0,9828	1,00	185
32	132,238	24,753	4,975	1,00	187,143	0,6939	0,833	1,00	185
64	135	35,333	5,9442	1,00	187,095	1,0386	1,019	1,00	184
128	133,905	73,4195	8,5685	1,00	186,762	0,84807	0,921	1,00	185

Слід відмітити високі показники статистичної безпеки алгоритму шифрування «Струмок», який виявив певні властивості генератора випадкових бітів. Зокрема за результатами даних таблиці видно, що формовані послідовності за своїми властивостями не поступаються всесвітньо відомим потоковим криптографічним алгоритмам, зокрема шифрами HC-256, Salsa20, Mickey та SNOW 2.0. Крім того, для шифру «Струмок» мінімальні значення числа пройдених статистичних тестів за критерієм $P_j \geq 0.96$ є вищі, ніж у цих алгоритмах, що свідчить про незначну перевагу показників статистичної безпеки дослідженого алгоритму.

Процес ініціалізації алгоритму «Струмок» включає в себе окремі етапи: встановлення ключа і уста-

новку вектора ініціалізації, початкове тактування. Для того, щоб генератор вийшов в робочий стан необхідно зробити 64 ініціюючих такти без генерації ключового потоку, тобто 4 повних циклів. Тому перед початком генерації ключового потоку необхідно зробити 64 зсуви вмісту комірок без генерації потоку, вихід з кінцевого автомата братиме участь у формуванні вмісту комірок регістра, а не ключового потоку. Вибір саме такої кількості початкових тактувань виходить в тому числі із міркувань поліпшення статистичних властивостей послідовності, яку буде згодом згенеровано. Для перевірки доцільності такої кількості тактувань було проведено тестування послідовності довжиною 10^6 бітів після різної кількості початкових тактувань. Отримані результати наведені в таблиці 4, які свідчать

про те, що після 64 тактувань забезпечуються високі статистичні властивості формованої послідовності.

ВИСНОВКИ

Алгоритм поточкового шифрування «Струмок» виконано за схемою 64-розрядного слово-орієнтованого синхронного поточного криптоалгоритму, що заснований на ідеї класичного сумуючого генератора. Він застосовує базову структуру алгоритму «SNOW2.0» та призначений для збільшення швидкості формування ключового потоку при збереженні високих криптографічних властивостей. Це досягається за рахунок застосування перетворень над 64-бітними словами, які розглядаються як елементи скінченного поля $GF(2^{64})$, із використанням РЗЛЗЗ над цим полем для формування послідовності максимального періоду. У сукупності із нелінійними перетвореннями над послідовністю станів генератора це забезпечує властивості випадковості та непередбачуваності формованих послідовностей.

Методика, яка розглянута в даній роботі, та отримані з її використання результати можуть розглядатися як первинний аналіз криптографічних властивостей генератора, оскільки такі статистичні тести не враховують власну структуру генератора.

За результатами експериментальних досліджень слід відмітити високі показники статистичної безпеки алгоритму шифрування «Струмок». Формовані послідовності за своїми властивостями не поступаються всесвітньо відомим поточковим криптографічним алгоритмам. Крім того, для шифру «Струмок» мінімальні значення числа пройдених статистичних тестів є вищі ніж у цих алгоритмах, що свідчить про незначну перевагу показників статистичної безпеки дослідженого алгоритму.

Статистичні дослідження вихідних послідовностей з різною кількістю початкових тактувань (на етапі ініціалізації шифру) показали, що після 64 ітерацій забезпечуються високі показники статистичної безпеки. Це опосередковано підтверджує правильність обрання кількості початкових тактувань в специфікації алгоритму поточкового шифрування.

Перспективним напрямком подальших досліджень є аналіз криптографічних властивостей алгоритму поточкового шифрування «Струмок», обґрунтування практичних рекомендацій з його застосування, в тому числі і на постквантовий період.

Література

- [1] Горбенко І.Д., Горбенко Ю.І. Прикладна криптологія. Теорія. Практика. Застосування: Підручник для вищих навч. закладів. – Харків: Вид-во «Форт», 2013. – 880 с.
- [2] Горбенко Ю.І. Побудовання та аналіз систем, протоколів і засобів криптографічного захисту інформації: монографія. – Частина 1: Методи побудовання та аналізу, стандартизація та застосування криптографічних систем / За заг. ред. д.т.н., професора І.Д. Горбенка / Ю.І. Горбенко // Харків, Видавництво «Форт», 2016. – 960 с.
- [3] Кузнецов О.О., Сватовський І.І. та ін., всього 13 осіб. Аналіз, розробка та дослідження постквантових криптографічних примітивів та обґрунтування умов їхнього застосування в Україні: звіт про НДР (проміжний). Том 1. – Аналіз та порівняльні дослідження симетричних криптографічних перетворень на постквантовий період / ХНУ ім. В.Н. Каразіна; кер. Кузнецов О.О.; вик.: Сватовський І.І. [та інш., всього 13 осіб]. Х.: ХНУ ім. В.Н. Каразіна. – 2016. – 119 с.
- [4] Кузнецов О.О., Сватовський І.І. та ін., всього 6 осіб. Аналіз, розробка та дослідження постквантових криптографічних примітивів та обґрунтування умов їхнього застосування в Україні: звіт про НДР (проміжний). Том 2. – Аналіз та порівняльні дослідження постквантових алгоритмів електронного цифрового підпису та направленої шифрування / ХНУ ім. В.Н. Каразіна; кер. Кузнецов О.О.; вик.: Сватовський І.І. [та інш., всього 6 осіб]. Х.: ХНУ ім. В.Н. Каразіна. – 2016. – 66 с.
- [5] FIPS-197: Advanced Encryption Standard (AES). National Institute of Standards and Technology. - 2001. [Електронний ресурс]. – Режим доступу: <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>.
- [6] Information technology – Security techniques – Encryption algorithms, Part 3: Block ciphers (ISO/IEC 18033-3) - 80 p.
- [7] ISO/IEC 18033-4:2011. Information technology – Security techniques – Encryption algorithms – Part 4: Stream ciphers. [Електронний ресурс]. – Режим доступу: http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=54532
- [8] ISO/IEC 29192-3:2012. Information technology – Security techniques – Lightweight cryptography – Part 3: Stream ciphers. [Електронний ресурс]. – Режим доступу: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=56426
- [9] The eSTREAM Project - eSTREAM Phase 3. HC (Portfolio Profile 1). [Електронний ресурс]. – Режим доступу: <http://www.ecrypt.eu.org/stream/hcpf.html>
- [10] The eSTREAM Project - eSTREAM Phase 3. Grain (Portfolio Profile 2). [Електронний ресурс]. – Режим доступу: <http://www.ecrypt.eu.org/stream/grainpf.html>
- [11] The eSTREAM Project - eSTREAM Phase 3. MICKEY (Portfolio Profile 2). [Електронний ресурс]. – Режим доступу: <http://www.ecrypt.eu.org/stream/mickeypf.html>
- [12] The eSTREAM Project - eSTREAM Phase 3. Rabbit (Portfolio Profile 1). [Електронний ресурс]. – Режим доступу: <http://www.ecrypt.eu.org/stream/rabbitpf.html>
- [13] The eSTREAM Project - eSTREAM Phase 3. Salsa20 (Portfolio Profile 1). [Електронний ресурс]. – Режим доступу: <http://www.ecrypt.eu.org/stream/salsa20pf.html>
- [14] The eSTREAM Project - eSTREAM Phase 3. SOSEMANUK (Portfolio Profile 1). [Електронний ресурс]. – Режим доступу: <http://www.ecrypt.eu.org/stream/sosemanukpf.html>
- [15] The eSTREAM Project - eSTREAM Phase 3. Trivium (Portfolio Profile 2). [Електронний ресурс]. – Режим доступу: <http://www.ecrypt.eu.org/stream/triviumpf.html>
- [16] Горбенко Ю.І., Потий А.В., Избенко Ю.А., Орлова С.Ю. Анализ схем поточкового шифрования, представленных на европейский конкурс NESSIE // Правове, нормативне та метрологічне забезпечення

системи захисту інформації в Україні: науково-технічний збірник. – 2002. – Вип. 5. – С. 92-110.

- [17] Дослідження режимів застосування блокових симетричних шифрів: звіт про НДР (заключний). / ХНУ ім. В.Н. Каразіна; кер. Кузнецов О.О.; вик.: Шлокін В.М. [та ін., всього 4 особи]. Х.: ХНУ ім. В.Н. Каразіна. – 2014. – 89 с.
- [18] Розробка нового блокового симетричного шифру: звіт за перший етап НДР «Алгоритм» (проміжний) / АТ «ІТ»; кер. І.Д. Горбенко – Харків, 2014, Том 4. – 304 с.
- [19] Інформаційні технології. Криптографічний захист інформації. Алгоритм симетричного блокового перетворення: ДСТУ 7624:2014. – К.: Мінекономрозвитку України, 2015. – 238 с.
- [20] Аналіз та порівняльні дослідження сучасних алгоритмів потокового криптоперетворення: звіт про НДР (проміжний). / ХНУ ім. В.Н. Каразіна; кер. Кузнецов О.О.; вик.: Малахов С.В. [та ін., всього 11 осіб]. Х.: ХНУ ім. В.Н. Каразіна. – 2015. – 254 с.
- [21] Розробка пропозицій до проекту алгоритму потокового симетричного шифрування та обґрунтування його властивостей: звіт про НДР (заключний). / ХНУ ім. В.Н. Каразіна; кер. Кузнецов О.О.; вик.: Малахов С.В. Х.: ХНУ ім. В.Н. Каразіна. – 2015. – 73с.
- [22] Кузнецов О.О., Іваненко Д.В., Белозерцев І.М., Андрушкевич А.В. Алгоритм потокового криптоперетворення «Струмок» // Труды научно-технической конференции с международным участием «Компьютерное моделирование в наукоемких технологиях», 26-31 мая 2016 г. – Х.: ХНУ имени В.Н. Каразина – 2016. – С. 187 – 190.
- [23] Kuznetsov O. O., Ivanenko D.V., Lutsenko M.S. Strumok stream cipher: specification and basic properties // Third International Scientific-Practical Conference «Problems of Infocommunications. Science and Technology» (PICS&T-2016). October 4 - 6, 2016 Ukraine, Kharkiv. – Kharkiv: Ministry of Education and Science of Ukraine, Kharkov National University of Radioelectronics. – 2016. – С. 15-28
- [24] Андрушкевич А.В., Іваненко Д.В., Луценко М.С., Кухар Ю.В. Аналіз властивостей перспективного потокового шифру «Струмок» // 71-ша науково-технічна конференція професорсько-викладацького складу, науковців, аспірантів та студентів. м. Одеса 6-8 грудня 2016 р. – м. Одеса: ОНАЗ.
- [25] Pseudorandom number generator Enoogo. [Электронный ресурс]. – Режим доступу: <http://www.hitachi.com/rd/yrl/crupto/enogo/>
- [26] Special Publication 800-22. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. [Электронный ресурс]. Режим доступа: <http://csrc.nist.gov/publications/nistpubs/800-22-rev1a/SP800-22rev1a.pdf>
- [27] Кузнецов А.А., Мордвинов Р.И., Колованова Е.П., Самойлова А.В. Методика статистического тестирования криптографических алгоритмов // Спеціальні телекомунікаційні системи та захист інформації. – Київ – 2014. – №1(25). – С.54-61



Кузнецов Олександр Олександрович, доктор технічних наук, професор, професор кафедри БІСТ ХНУ імені В.Н. Каразіна. Область наукових інтересів: Криптографія та автентифікація, теорія передачі даних, стеганографічні методи захисту інформації.



Луценко Марія Сергіївна, студентка факультету комп'ютерних наук ХНУ імені В.Н. Каразіна. Область наукових інтересів: Криптографічні методи захисту інформації, потокові симетричні шифри.



Андрушкевич Аліна Вадимівна, молодший науковий співробітник кафедри БІСТ ХНУ імені В.Н. Каразіна. Область наукових інтересів: аналіз стійкості симетричних шифрів, криптографія і автентифікація.



Мелкозерова Ольга Михайлівна, кандидат технічних наук, магістрант факультету комп'ютерних наук ХНУ імені В.Н. Каразіна. Область наукових інтересів: Криптографічні методи захисту інформації, потокові симетричні шифри.



Новікова Дарина Вікторівна, студентка радіотехнічного факультету ХНУРЕ. Область наукових інтересів: Криптографічні методи захисту інформації, потокові симетричні шифри.



Лобан Анна Володимирівна, студентка радіотехнічного факультету ХНУРЕ. Область наукових інтересів: Криптографічні методи захисту інформації, потокові симетричні шифри.

УДК 004.056.55

Статистические исследования современных потоковых шифров / А.А. Кузнецов, М.С. Луценко, А.В. Андрушкевич, О.М. Мелкозерова, Д.В. Новикова, А.В. Лобан // Прикладная радиоэлектроника: науч.-техн. журнал. – 2016. – Том 15, № 3. – С. 167 – 178.

Рассматривается математическая структура нового потокового симметричного шифру «Струмок». Исследуется його криптографические свойства путем статистического тестирования выходных последовательностей (гаммы шифровальной). Проводится сравнительный анализ показателей статистической безопасности с известными мировыми потоковыми шифрами.

Ключевые слова: потоковый симметричный шифр, криптографические свойства, статистическое тестирование. Табл.: 04. Ил.: 23. Библиогр.: 27 назв.

UDC 004.056.55

Statistical studies of modern stream ciphers / O.O. Kuznetsov, M.S. Lutsenko, A.V. Andrushkevych, O.M. Melkozherova, D.V. Novikova, A.V. Loban // Applied Radio Electronics: Sci. Journ. – 2016. – Vol. 15, № 3. – P. 167 – 178.

The paper considers the mathematical structure of the new stream symmetric cipher "Strumok". Its cryptographic properties are studied by statistical tests of initial sequences (gamma encryption). A comparative analysis of statistical indicators of security of the new cipher and those of known world stream ciphers has been performed.

Keywords: stream symmetric cipher, cryptographic properties, statistical testing.

Tab.: 04. Fig.: 23. Ref.: 27 items.

МЕТОДЫ И СРЕДСТВА СИММЕТРИЧНЫХ КРИПТОПРЕОБРАЗОВАНИЙ

УДК 621.3.06

МАТЕМАТИЧНА МОДЕЛЬ ОЦІНКИ ВЛАСТИВОСТЕЙ НЕІН'ЕКТИВНИХ СХЕМ РОЗГОРТАННЯ КЛЮЧІВ СИМЕТРИЧНИХ БЛОКОВИХ ШИФРІВ

М.Ю. РОДІНКО, Р.В. ОЛІЙНИКОВ

В роботі подано математичну модель оцінки ймовірності співпадіння потужностей множини послідовностей циклових ключів і множини ключів шифрування для неін'єктивних (схем розгортання ключів) СРК. Зокрема, сформульована та доведена теорема, що визначає ймовірність співпадіння потужностей множини послідовностей циклових ключів, які формуються неін'єктивною СРК, і множини ключів шифрування. Показано, що для повномасштабного шифру ця ймовірність практично дорівнює 1. Доведено, що складність атак переборного типу на неін'єктивні СРК практично дорівнює складності атак на ін'єктивні схеми (складність перебірних атак не знижується).

Ключові слова: симетричний блоковий шифр, схема розгортання ключів, ДСТУ 7624:2014.

ВСТУП

Одним із основних компонентів симетричного блокового шифру є схема розгортання циклових ключів. Схема розгортання ключів – це алгоритм, що розширює відносно короткий майстер-ключ (як правило, довжиною від 128 до 512 бітів) до відносно великого розширеного ключа (як, правило декілька сотень чи тисяч бітів) для подальшого застосування в алгоритмах зашифрування та розшифрування [1].

Класичним підходом до проектування схем розгортання ключів (СРК) вважається застосування бієктивного перетворення для відображення ключа шифрування у послідовність циклових ключів. Перші СРК були дуже простими і включали, наприклад, просту перестановку бітів ключа шифрування (DES, IDEA) або пряме чи рекурсивне лінійне перетворення з майстер-ключа [2]. Із розвитком технологій криптоаналізу розробники почали додавати до СРК нелінійні операції (такі, як підстановки) з метою уникнення атак на зв'язаних ключах.

Застосування простої бієктивної функції дозволяє забезпечити компактну реалізацію, достатньо високу швидкодію та відсутність еквівалентних ключів шифрування. Суттєвим недоліком подібних СРК є відсутність властивості односпрямованості, тобто складність відновлення ключа шифрування при знанні одного або декількох підключів є не вищою за поліноміальну. Ця властивість робить шифр більш уразливим до атак на реалізацію.

СРК шифру «Калина» (ДСТУ 7624:2014) розроблялася з урахуванням необхідності захисту від атак на реалізацію та атаки на зв'язаних ключах [3]. З цією метою була розроблена односпрямована СРК, що забезпечує неможливість відновлення циклового

ключа при знанні ключа шифрування або інших підключів. Особливістю односпрямованих СРК є те, що вони є неін'єктивними, тобто теоретично припускається існування еквівалентних ключів (таких, що формують однакову послідовність циклових ключів). Односпрямовані СРК застосовуються і в інших відомих блокових шифрах таких, як FOX [4], Twofish [5] та ін. Оцінка ймовірності співпадіння потужностей множини послідовностей циклових ключів, які формуються неін'єктивною СРК, і множини ключів шифрування дозволила б додатково обґрунтувати стійкість односпрямованих СРК та доцільність їх використання у блокових шифрах.

1. ВИМОГИ ДО СХЕМ РОЗГОРТАННЯ КЛЮЧІВ

На сьогоднішній день не досягнуто консенсусу щодо необхідних та достатніх умов, які мають задовольняти СРК. Розробники приділяють більше уваги основному шифруючому перетворенню, ніж СРК [2]. Багато з існуючих правил проектування СРК є далекими від практичного застосування. Деякі з них є занадто слабкими з точки зору безпеки, деякі фокусуються лише на певних типах атак і є однобічними, інші є емпіричними і не мають достатнього обґрунтування [2].

Загальними принципами побудування СРК є відсутність слабких, напівслабких та еквівалентних ключів. Застосування циклових констант необхідно для попередження симетрії шифру, яка призводить до можливості реалізації слайд-атак [3].

Метою побудування сильної СРК є усунення будь-якої слабкості, яка гіпотетично або практично може бути використана для атаки на блоковий шифр [6]. Як і при проектуванні блокових шифрів, при роз-

робці СРК часто застосовуються методи досягнення перемішування та розсіювання.

У [7] показано, що шифри зі складними СРК є більш стійкими до атак диференціального та лінійного криптоаналізу, ніж шифри з більш простими СРК. Показано, що деякі ітеративні шифри з дуже простими СРК навіть при повному наборі циклів не досягають рівномірного розподілу ймовірностей диференціалів та лінійних корпусів. Водночас показано, що добре спроектовані шифри зі складними СРК досягають рівномірного розподілу швидше, ніж шифри з поганими СРК.

Л. Кнудсен [8] вважає, що сильна СРК повинна мати такі загальні властивості, які можуть бути досягнуті водночас:

а) односпрямована функція, стійка до колізій (функція, яку неможливо інвертувати);

б) мінімальна взаємна інформація (між всіма бітами підключа та бітами майстер-ключа);

в) ефективна реалізація.

Під час розробки шифру «Калина» до СРК перспективного шифру були висунуті такі вимоги [3]:

а) нелінійна залежність кожного біта кожного циклового ключа від кожного біта ключа шифрування;

б) циклові ключі суттєво відрізняються і мають складну нелінійну залежність;

в) захист від відомих криптоаналітичних атак, що орієнтовані на схему розгортання ключів;

г) відсутність слабких ключів, за яких погіршуються криптографічні властивості або знижується стійкість перетворення;

д) обчислювальна складність формування всіх циклових ключів не перевищує складності зашифрування трьох блоків;

е) простота програмної, програмно-апаратної і апаратної реалізації.

Як додаткові вимоги, розглядалися такі [3]:

а) неможливість отримання ключа шифрування за один або декілька цикловими ключами, що є доступними для криптоаналітика;

б) можливість формування циклових ключів у довільному порядку (однакова обчислювальна і просторова складність для зашифрування і розшифрування).

2. АТАКИ НА СХЕМИ РОЗГОРТАННЯ КЛЮЧІВ

Слайд-атака [9]. Слайд-атака була вперше описана А. Бірюковим та Д. Вагнером у 1999 р. та є криптографічною атакою на основі підбраного відкритого тексту. У більшості випадків атака дозволяє проводити криптоаналіз багатоциклових шифрів незалежно від числа циклів. Слайд-атака експлуатує степінь самоподоби блокового шифру та в основному застосовується до ітеративних блокових шифрів з періодичною СРК.

Шифр розглядається як результат застосування ідентичних перетворень $F(x,k)$, де k є секретним ключем (при цьому F може складатися більше, ніж з одного циклу шифру) [9].

Ідея атаки полягає у зсуві однієї копії процесу зашифрування відносно іншої копії процесу зашифрування так, що два процеси є зсунутими на один цикл. Це дає можливість легко отримати ключ шифрування після однієї ітерації F . Згідно з парадоксом про день народження для здійснення атаки необхідно набрати $2^{n/2}$ пар (M_i, C_i) [9].

Атака на зв'язаних ключах. Атака припускає [10], що криптоаналітику відоме деяке математичне співвідношення, що зв'язує між собою ключі. Наприклад, співвідношення може бути простим значенням XOR з відомою константою $K_1=K_2 \oplus C$ або більш складним зв'язком. Атака вперше була запропонована Е. Біхамом та нагадує слайд-атаку.

Атаки типу «зустріч посередині» [1]. Атаки цього типу виникають, коли перша половина циклів шифру та друга половина циклів шифру залежать від різних наборів ключових бітів. Це дозволяє зловмиснику атакувати дві частини незалежно одна від одної і протидіє подвійному шифруванню з блоковим шифром та двома різними ключами.

Слабкі ключі [1]. Слабким вважається ключ K , для якого зашифрування є ідентичною функцією до розшифрування. Напівслабкими вважається пара ключів K та K' , для яких зашифрування за допомогою K ідентичне розшифруванню за допомогою K' і навпаки. Якщо число слабких ключів відносно мале, вони можуть не представляти загрози для шифру, якщо той використовується для забезпечення конфіденційності. Однак у деяких режимах гешування, що використовують блокові шифри, зловмисник може обрати вхідне значення ключа при спробі пошуку колізії. В таких режимах блоковий шифр не повинен мати слабких та напівслабких ключів.

Класи ключів, що виявляються [1]. Одним зі способів зменшення ефективного ключового простору є його поділ на класи і подальший пошук атаки, які показують, до якого класу належить ключ. У деяких випадках об'єм роботи із визначення приналежності ключа до певного класу дуже малий. Подібні ключі іноді також називають слабкими [1]. Наприклад, певні ключі в алгоритмі Blowfish призводять до однакових входів у S-блок і можуть бути виявлені у зменшених за кількістю циклів варіантах шифру. Шифр IDEA має декілька класів ключів, що виявляються, лише за допомогою двох зашифрувань обраних відкритих текстів [1].

Прості зв'язки та еквівалентні ключі [1]. Простий зв'язок виникає між двома різними ключами і проявляється як співвідношення між відкритими текстами та шифртекстами. Блокові шифри DES та LOKI мають простий зв'язок, який виражається у тому, що: якщо K зашифровує P у C , тоді побітове доповнення

K зашифрує побітове доповнення P у побітове доповнення C . Це зменшує ефективний ключовий простір на один біт. Алгоритми DES та LOKI мають пари ключів, для яких простий зв'язок існує, щонайменше, для частини всіх відкритих текстів [1]. Два ключа є еквівалентними, якщо вони зашифровують усі відкриті тексти ідентично. Це може розглядатися як спеціальний вид простого зв'язку.

Атаки на СРК, що не є односпрямованими [1]. СРК не є односпрямованою, якщо маючи декілька циклових підключів, зловмисник може отримати інформацію про ключ шифрування або інші невідомі підключі. Наприклад, відновлення декількох циклових підключів дозволяє відновити більшу частину майстер-ключа в СРК DES. Е. Біхам та А. Шамір використали це для оптимізації їхньої диференціальної атаки на DES. Крім того, це може зробити простішим пошук слабких та зв'язаних ключів для СРК, що не є односпрямованими.

3. МАТЕМАТИЧНА МОДЕЛЬ ОЦІНКИ ВЛАСТИВОСТЕЙ НЕІН'ЄКТИВНИХ СХЕМ РОЗГОРТАННЯ КЛЮЧІВ БЛОКОВИХ ШИФРІВ

Розв'язання задачі оцінки потужності множини послідовностей циклових ключів було започатковано у [11]. Запропонуємо новий підхід до її вирішення, який дозволяє отримати більш точні оцінки та довести, що для неін'єктивних схем розгортання циклових ключів потужність множини циклових ключів не зменшується порівняно з ін'єктивними схемами.

Введемо математичну модель оцінки ймовірності співпадіння потужностей множини послідовностей циклових ключів, які формуються неін'єктивною СРК, і множини ключів шифрування. Математична модель використовує припущення, що схема розгортання ключів є випадковим відображенням, що впливає із конструкції та підтверджується результатами статистичного тестування. Тоді справедливою є наступна теорема.

Теорема 1 (про ймовірність співпадіння потужностей множини послідовностей циклових ключів, які формуються неін'єктивною СРК, і множини ключів шифрування).

Нехай τ – неін'єктивна схема розгортання ключів, що реалізує випадкове відображення та має такі параметри:

k – довжина ключа шифрування в бітах;

l – довжина циклового ключа в бітах;

t – кількість циклових ключів;

$K = 2^k$ – потужність множини ключів шифрування;

нея;

$L = 2^l$ – потужність множини послідовностей циклових ключів.

Тоді ймовірність співпадіння потужностей множини послідовностей циклових ключів, які формуються неін'єктивною СРК, і множини ключів шифрування для τ обчислюється через таке співвідношення:

$$P_{\theta} = \left(\frac{L}{L-K}\right)^{L-K+\frac{1}{2}} \cdot e^{(-K)}. \quad (1)$$

Доведення.

Нехай задано множину ключів шифрування $\Psi = \left\{K^{(i)} \mid i = 0, 1, \dots, 2^k - 1\right\}$ з потужністю $K = |\Psi|$ та множину послідовностей циклових ключів $\Lambda = \left\{L^{(i)} \mid i = 0, 1, \dots, 2^l - 1\right\}$ з потужністю $L = |\Lambda|$.

Послідовність циклових ключів, що формується СРК, має вигляд $L^{(i)} = (L_0^{(i)}, L_1^{(i)}, \dots, L_{t-1}^{(i)})$.

Генерація кожної послідовності ключів $L^{(i)}$ виконується за допомогою випадкової функції $f: \Psi \rightarrow \Lambda$.

Для кожного ключа шифрування $K^{(i)}$ формується послідовність циклових ключів, тобто всього з множини Λ обирається K послідовностей. Першу послідовність $L^{(0)} = (L_0^{(0)}, L_1^{(0)}, \dots, L_{t-1}^{(0)})$ можна обрати L способами, другу $L^{(1)} = (L_0^{(1)}, L_1^{(1)}, \dots, L_{t-1}^{(1)})$ (так, щоб вона не співпадала з першою) – $(L-1)$ способами, останню $L^{(K)} = (L_0^{(K)}, L_1^{(K)}, \dots, L_{t-1}^{(K)})$ – $(L-(K-1))$ способами.

Таким чином, загальна кількість варіантів K послідовностей, що не повторюються, дорівнює

$$L \cdot (L-1) \cdot \dots \cdot (L-(K-1)) = \frac{L!}{(L-K)!}$$

Число всіх можливих варіантів послідовностей циклових ключів дорівнює L^K . Тоді ймовірність співпадіння потужностей множини послідовностей циклових ключів, які формуються неін'єктивною СРК, і множини ключів шифрування визначається як

$$P_{\theta} = \frac{L!}{(L-K)!L^K}$$

Згідно з формулою Стирлінга, наближене значення факторіала обчислюється як

$$n! \approx \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$$

Таким чином, отримуємо:

$$\begin{aligned} P_{\theta} &= \frac{L!}{(L-K)!L^K} = \frac{\sqrt{2\pi L} \cdot \left(\frac{L}{e}\right)^L}{\sqrt{2\pi(L-K)} \cdot \left(\frac{L-K}{e}\right)^{L-K} \cdot L^K} = \\ &= \sqrt{\frac{L}{L-K}} \cdot \frac{e^{-L} \cdot L^L}{e^{-L+K} \cdot (L-K)^{L-K} \cdot L^K} = \\ &= \sqrt{\frac{L}{L-K}} \cdot e^{-K} \cdot (L-K)^{K-L} \cdot L^{L-K} = \\ &= \sqrt{\frac{L}{L-K}} \cdot e^{-K} \cdot \left(\frac{L}{L-K}\right)^{L-K} = \left(\frac{L}{L-K}\right)^{L-K+\frac{1}{2}} \cdot e^{-K}. \end{aligned}$$

Доведення закінчено.

В таблиці 1 наведено результати розрахунків за формулою (1).

Таблиця 1

Результати розрахунків за формулою (1)

k , біт	l , біт	t	P_θ
4	4	2	0,6197211
		3	0,9710922
		4	0,9981705
		5	0,9998856
		6	0,9999928
		7	0,9999996
		8	0,9999997
		9	0,99999998
		10	0,9999999989
		8	8
3	0,9980564		
4	0,9999924		
5	0,9999997		
6	0,999999988		
16	16	2	0,6065337
		3	0,9999924

Як видно з таблиці вже при $K = 2^4$ та $L = 2^{40}$ ймовірність співпадіння потужностей множини послідовностей циклових ключів, які формуються неін’єктивною СРК, і множини ключів шифрування дорівнює 0,9999999989. Зі зростанням довжин ключа шифрування та циклового ключа значення P_θ зростає. Для повномасштабного шифру практично $P_\theta \approx 1$. Це означає, що складність атак переборного типу на неін’єктивні СРК практично дорівнює складності атак на ін’єктивні СРК.

ВИСНОВКИ

Застосування сильних схем розгортання ключів дозволяє усунути вразливості, які теоретично або практично можуть бути використані для атаки на шифр. Крім того, використання сильних СРК покращує характеристики шифру, зокрема диференціальні та лінійні. До основних вимог, яким повинна задовольняти сильна СРК відносяться односпрямованість, нелінійна залежність між всіма бітами циклових ключів та ключа шифрування і ефективна реалізація.

Більшість атак на схеми розгортання ключів не становлять практичної небезпеки, проте вони показують теоретичну слабкість, яка може бути використана за певних обставин. Найбільш небезпечними можна вважати атаку на зв’язаних ключах та атаки на СРК, що не є односпрямованими.

В роботі подано математичну модель оцінки ймовірності співпадіння потужностей множини послідовностей циклових ключів, які формуються неін’єктивною СРК, і множини ключів шифрування. Зокрема, отримано співвідношення, що дозволяє обчислити цю ймовірність.

За допомогою запропонованої математичної моделі доведено, що складність атак переборного типу на неін’єктивні схеми розгортання ключів практично

дорівнює складності атак на ін’єктивні схеми (складність перебірних атак не знижується). При цьому неін’єктивні СРК забезпечують додаткову стійкість до атак на реалізацію та деяких інших криптоаналітичних атак. Таким чином, при побудованні нового перспективного симетричного блокового шифру доцільно використовувати саме неін’єктивну схему розгортання циклових ключів.

Література

[1] *Kelsey J.* Key-Schedule Cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES [Text] / J. Kelsey, B. Schneier, D. Wagner // *Advances in Cryptology – CRYPTO’96.*: Springer Berlin Heidelberg, 1996. – P. 237 – 251.

[2] *Huang J.* Revisiting Key Schedule’s Diffusion In Relation With Round Function’s Diffusion [Text] / J. Huang, X. Lai // *Designs, codes and cryptography.* – 2014. – V.73. – No.1. – P. 85 – 103.

[3] *Олійников Р.* Принципи побудови і основні властивості нового національного стандарту блокового шифрування України [Текст] / Р. Олійников, І. Горбенко, О. Казимиров, В. Руженцев, Ю. Горбенко // *Захист інформації.* – 2015. – Т. 17. – №2. – С. 142 – 157.

[4] *Junod P.* FOX: a new family of block ciphers [Text] / P. Junod, S. Vaudenay // *Selected Areas in Cryptography.* – Springer Berlin Heidelberg, 2005. – pp. 114– 129.

[5] *Schneier B.* Twofish: A 128-Bit Block Cipher [Text] / B. Schneier, et al. // *AES algorithm submission.* – June 15, 1998. – 68 с.

[6] *May L.* Strengthening the Key Schedule of the AES [Text] / L. May, M. Henricksen // *Information Security and Privacy.*: Springer Berlin Heidelberg, 2002. –P. 226 – 240.

[7] *Knudsen R.* Lars. On the Role of Key Schedules in Attacks on Iterated Ciphers [Text] / Lars R. Knudsen, John E. Mathiassen // *Computer Security–ESORICS 2004.*: Springer Berlin Heidelberg, 2004. – P. 322 – 334.

[8] *Knudsen L. R.* Practically secure Feistel ciphers / L.R. Knudsen // *Fast Software Encryption.*: Springer Berlin Heidelberg, 1993. – pp. 211-221.

[9] *Biryukov A.* Slide attacks [Text] / A. Biryukov, D. Wagner // *In Fast Software Encryption.*: Springer Berlin Heidelberg, 1999. – P 245 – 259.

[10] *Biham Eli.* New types of cryptanalytic attacks using related keys [Text] / Eli Biham // *Journal of Cryptology.*: Springer-Verlag, 1994. – V. 7. – No. 4 – P. 229 – 246.

[11] *Олійников Р.В.* Методи аналізу і синтезу перспективних симетричних криптографічних перетворень / Р.В. Олійников // *Дисертація на здобуття наукового ступеня доктора технічних наук по спеціальності 05.13.05 – комп’ютерні системи та компоненти.* ХНУРЕ. – Харків. – 2014. – 423 с.



Родінко Марія Юрївна, аспірантка кафедри безпеки інформаційних систем і технологій ХНУ ім. В.Н. Каразіна. Наукові інтереси: симетрична криптографія та криптоаналіз.



Олійников Роман Васильович, доктор технічних наук, професор кафедри безпеки інформаційних систем і технологій ХНУ ім. В.Н. Каразіна. Наукові інтереси: симетрична криптографія та криптоаналіз, мережна безпека.

шифрования. Показано, что для полномасштабного шифра эта вероятность практически равна 1. Доказано, что сложность атак переборного типа на неинъективные СРК практически равна сложности атак на инъективные схемы (сложность переборных атак не снижается).

Ключевые слова: симметричный блочный шифр, схема разворачивания ключей, ДСТУ 7524:2014.

Табл.: 01. Библиогр: 11 назв.

УДК 621.3.06

Математическая модель оценки свойств неинъективных схем разворачивания ключей симметричных блочных шифров / М.Ю. Родинко, Р.В. Олейников // Прикладная радиоэлектроника: научн.-техн. журнал. – 2016. – Том 15, № 3. – С. 179 – 183.

В работе представлена математическая модель оценки вероятности совпадения мощностей множества последовательностей цикловых ключей и множества ключей шифрования для неинъективных схем разворачивания ключей (СРК). В частности, сформулирована и доказана теорема, которая определяет вероятность совпадения мощностей множества последовательностей цикловых ключей, которые формируются неинъективной СРК, и множества ключей

UDC 621.3.06

A mathematical model of non-injective key schedules properties evaluation of symmetric block ciphers / M.Yu. Rodinko, R.V. Oliynykov // Applied Radio Electronics: Sci. Journ. – 2016. – Vol. 15, № 3. – P. 179 – 183.

This paper presents a mathematical model of probability evaluation of matching of round keys (formed by non-injective key schedule) set and encryption keys set cardinalities. A theorem that determines such a probability is formulated and proved. It is shown that for non-injective key schedules the probability evaluation of matching of round keys set and encryption keys set cardinalities is almost equal to 1. Using the presented model it is proved that exhaustive search attacks complexity on non-injective key schedules is almost equal to injective ones.

Keywords: symmetric block cipher, key schedule, DSTU 7624:2014.

Tab.: 01. Ref.: 11 items.

ПРОВЕРКА МЕТОДА ДОКАЗАТЕЛЬСТВА СТОЙКОСТИ БЛОЧНЫХ ШИФРОВ К АТАКЕ НЕВЫПОЛНИМЫХ ДИФФЕРЕНЦИАЛОВ

В.И. РУЖЕНЦЕВ

Обсуждаются результаты вычислительных экспериментов по поиску невыполнимых дифференциалов для уменьшенных моделей блочных симметричных шифров. Подтверждается справедливость выводов, полученных с помощью предложенного в работе [1] метода обоснования отсутствия невыполнимых дифференциалов. Демонстрируются новые найденные для отдельных видов шифров невыполнимые дифференциалы, которые покрывают большее число циклов, чем известные.

Ключевые слова: блочный шифр, атака невыполнимых дифференциалов, невыполнимый дифференциал, Rijndael-подобные преобразования.

ВВЕДЕНИЕ

В работе [1] был предложен метод, который позволяет обосновать отсутствие невыполнимых дифференциалов (НД) для блочных симметричных шифров (БСШ). Сложность метода, в отличие от известных, в меньшей степени зависит от размера блока. Метод был применен к Rijndael-подобным SPN шифрам и фейстель-подобным шифрам. В этой работе представлены результаты вычислительных экспериментов по поиску НД для уменьшенных моделей БСШ. Полученные результаты, с одной стороны, подтверждают справедливость предложенного в [1] метода, с другой стороны, на наш взгляд, демонстрируют новые потенциальные возможности в организации атак невыполнимых дифференциалов.

1. АТАКА НЕВЫПОЛНИМЫХ ДИФФЕРЕНЦИАЛОВ. ВИДЫ НЕВЫПОЛНИМЫХ ДИФФЕРЕНЦИАЛОВ

Атака невыполнимых дифференциалов (НД) является одним из наиболее эффективных нападений на современные блочные симметричные шифры (БСШ). Этот криптоаналитический метод успешно позволяет атаковать как SPN-шифры [2 – 4], так и шифры, построенные с использованием цепи фейстеля и других структур [5 – 9]. Подтверждением сказанного является большое количество работ, появившихся за последнее десятилетие и направленных, главным образом, на поиск невыполнимых дифференциалов [2 – 12]. Для шифра Rijndael с уменьшенным количеством циклов данную атаку можно считать одной из самых успешных.

Атака НД на блочные симметричные шифры, как большинство криптоаналитических нападений, относится к классу атак на цикловую функцию, и для ее реализации необходимо иметь некоторое количество пар открытый текст-криптограмма, полученных на одном и том же секретном ключе.

Данная криптоаналитическая методика называется атакой невыполнимых дифференциалов, поскольку

в атаке используются дифференциалы специального вида – те, которые не могут выполняться, т. е. имеющие нулевую вероятность. Атака невыполнимых дифференциалов на r -цикловый шифр обычно становится возможной, когда имеется $(r-1)$ -цикловый невыполнимый дифференциал.

На рис. 1 представлена схема выполнения атаки НД.

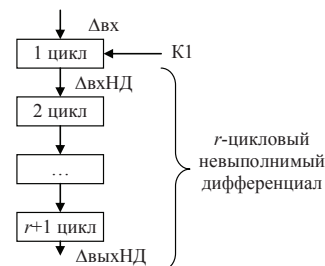


Рис. 1. Схема атаки НД

При наличии r -циклового НД с входной разностью $\Delta_{вхНД}$ и выходной разностью $\Delta_{выхНД}$ атака на $(r+1)$ -цикловый шифр состоит из следующих шагов. Выполняется поиск пары с некоторой входной разностью $\Delta_{вх}$ и выходной разностью $\Delta_{выхНД}$. При этом переход $\Delta_{вх}$ в $\Delta_{вхНД}$ на первом цикле должен быть возможен лишь для некоторых значений ключа первого цикла $K1$. Если такая пара найдена, то, в соответствии с НД, после первого цикла не могла быть разность $\Delta_{вхНД}$. И все ключи первого цикла, которые будут приводить к этой разности после одноциклового шифрования, являются неверными. Путем отсева всех неверных ключей определяется правильный подключ первого цикла $K1$.

Один из вариантов атаки – атака байтовых или усеченных невыполнимых дифференциалов (БНД) – была предложена в работах [2 – 4]. В ходе атаки через преобразования шифра пытаются провести вектора активизации. Каждый бит вектора активизации отражает активность одного байта в обычной разности. Таким образом, вектор активизации содержит столько

битов, сколько байтов в блоке, а значение бита определяется активностью байта: «1» – байт активный, «0» – байт пассивный.

Преимуществом БНД перед просто НД является то, что каждая найденная правильная пара позволяет отсеять не один или несколько неправильных ключей первого цикла, а сразу несколько сотен или тысяч неправильных ключей первого цикла.

2. ПРЕДЛАГАЕМЫЙ МЕТОД

В отличие от большинства известных подходов [2 – 12], которые были рассмотрены в [1] и которые направлены на поиск НД, наш подход направлен на обоснование отсутствия НД.

Основная идея предлагаемого подхода заключается в том, чтобы обосновать существование некоторой разности на промежуточном этапе шифрования, которая может быть получена для любой входной или выходной разности. Рис. 2 поясняет эту идею.

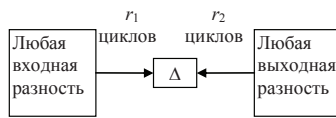


Рис. 2. Предлагаемый подход к обоснованию отсутствия НД

В [1] сформулирована и доказана следующая теорема, которую можно считать критерием отсутствия НД.

Теорема 1 ([1]). Если для БСШ существует некоторая разность Δ , которая может быть получена из любой ненулевой входной разности за r_1 циклов преобразований и которая может быть получена из любой ненулевой выходной разности за r_2 циклов, выполняемых в направлении дешифрования, то для такого БСШ не существует НД с r_1+r_2 и более циклами.

Таким образом, для доказательства отсутствия НД необходимо определить количество циклов r_1 и r_2 , за которые любая входная разность и любая выходная могут прийти к некоторому значению разности Δ .

Когда речь идет о векторах активизации или о байтовой разности на входе, выходе и на промежуточных этапах, то Δ обычно содержит сразу все активные байты (вектор активизации состоит из всех «1»). Такие НД будем называть байтовыми НД (БНД). Теорему 1 можно переформулировать следующим образом для БНД.

Теорема 2. Если для БСШ существует некоторая байтовая разность Δ , которая может быть получена для любого входного вектора активизации за r_1 циклов преобразований и которая может быть получена для любого выходного вектора активизации за r_2 циклов, выполняемых в направлении дешифрования, то для такого БСШ не существует БНД с r_1+r_2 и более циклами.

С помощью теоремы 1 в [1] объясняется отсутствие БНД для многих Rijndael-подобных шифров, в том числе для шифра Rijndael со 128-битным блоком.

При этом, область использования теоремы 2 не ограничивается только этим видом шифров. Далее будут рассмотрены также фейстель-подобные шифры и шифры, построенные с использованием схемы Лея-Мэсси (Lai-Massey).

Для каждой из рассматриваемых разновидностей шифров была построена уменьшенная модель, на которой с помощью вычислительных экспериментов выполнялась проверка справедливости полученных теоретических результатов.

В таблицах 1 и 2 приведены алгоритмы, которые были использованы для вычислительных экспериментов по поиску НД и БНД для уменьшенных 16 битовых моделей шифров.

Таблица 1

Алгоритм поиска НД

Входные данные: Шифрующее преобразование E. Пустая строка таблицы разности соответствующего размера.	
1	Перебор всех вариантов входной разности d
2	Обнуление строки таблицы разности
3	Перебор вариантов ключа k
4	Перебор всех вариантов входного значения x
5	Инкрементируем ячейку с индексом $E_k(x)+E_k(x+d)$
6	Проверяем строку таблицы разности на наличие «0». Каждый такой «0» соответствует НД
Выходные данные: Найденные НД.	

Таблица 2

Алгоритм поиска байтовых НД (БНД)

Входные данные: Шифрующее преобразование E. Пустая строка таблицы разности соответствующего размера.	
1	Перебор всех вариантов входного вектора активизации
2	Обнуление строки таблицы разности активизации
3	Перебор всех вариантов входной разности d, отвечающих выбранному входному вектору активизации
4	Перебор вариантов ключа k
5	Перебор всех вариантов входного значения x
6	Инкрементируем ячейку с индексом $E_k(x)+E_k(x+d)$
7	Перебор элементов полученной таблицы разности
8	Наличие элемента с ненулевым значением свидетельствует об отсутствии БНД с данным выходным вектором активизации
9	Если отсеяны все возможные выходные вектора активизации, то БНД не найдены и переходим к следующему значению входного вектора активизации
10	Если остались неотсеянные выходные вектора активизации, то каждый из них соответствует найденному БНД
Выходные данные: Найденные БНД.	

3. ИСПОЛЬЗУЕМЫЕ УМЕНЬШЕННЫЕ МОДЕЛИ

В рамках проводимых исследований будут рассматриваться криптографические свойства фейстель-подобных, SPN блочных шифров и шифров, построенных с использованием схемы Lai-Massey, с уменьшенным размером блока и ключа (8 или 16 битов). Целесообразность рассмотрения именно уменьшенных моделей шифров объясняется тем, что полноценный поиск НД можно провести только для шифров с небольшим размером блока. В качестве операций перемешивания и рассеивания были взяты преобразования, предложенные в [13] для уменьшенной версии шифра Rijndael. На рис. 3 и 4 схематически представлены преобразования, которые выполняются в рассматриваемых моделях SPN и фейстель-подобных шифров.

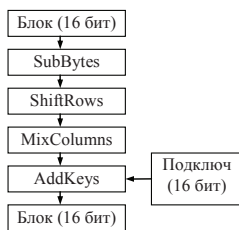


Рис. 3. Схема одного цикла SPN-шифра

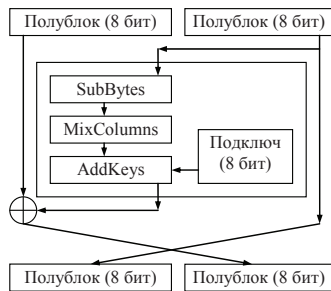


Рис. 4. Схема одного цикла фейстель-подобного шифра

К основным особенностям предложенных уменьшенных моделей шифров следует отнести:

- размер блока 16 бит, размер ключа 8 или 16 бит;
- структура блока для SPN: 2 колонки по 24-битовых элемента;

структура полублока для фейстель-подобного: 24-битовых элемента;

- умножение элементов каждой колонки на фиксированную МДР-матрицу размером 2 на 2 над $GF(2^4)$ (MixColumns);
- подстановка 4 в 4 бита (SubBytes);
- число ветвей активизации линейного преобразования MixColumns $B = 3$.

4. АНАЛИЗ RIJNDAEL-ПОДОБНЫХ ШИФРОВ

В работе [1] выполнен анализ условий, при которых теоремы 1 и 2 могут быть применены к Rijndael-подобным SPN шифрам. Доказано следующее утверждение.

Утверждение 1 ([1]). Для Rijndael-подобных шифров с блоком, в котором количество строк m не меньше, чем количество колонок n ($m > n$), не существует байтовых НД для 4 и более циклов с полным набором преобразований.

Полученный результат полностью согласуется с известными результатами для шифра Rijndael со 128-битным блоком, т. к. наилучшие НД, которые были найдены или использованы в известных работах, покрывают 3 полных и один (последний) неполный циклы [2 – 4, 8 – 10].

Для Rijndael-подобных шифров с блоком, в котором строк меньше, чем колонок ($m < n$), для того, чтобы гарантировать отсутствие НД потребуется, по крайней мере, два дополнительных цикла преобразований (по одному с каждой стороны). То есть, для таких шифров можно говорить о доказательстве отсутствия НД не менее, чем для 6 полных циклов.

С помощью представленных в таблицах 1 и 2 алгоритмов был проведен поиск НД и БНД для уменьшенной 16-битной версии алгоритма AES. Результаты представлены в таблицах 3 и 4.

Таблица 3

Результаты поиска НД для уменьшенной версии AES

Количество циклов	Количество найденных НД	Комментарии
4	510	Для каждой вх. разности с 1 активным S-блоком
5	0	

Таблица 4

Результаты поиска БНД для уменьшенной версии AES

Количество циклов	Количество найденных БНД	Комментарии
4 неполных (без MC в последнем цикле)	24	По 6 для каждого вх. вектора активизации с 1 активным S-блоком
4	0	

Результаты вычислительных экспериментов из табл. 4 подтверждают справедливость доказанного утверждения 1.

Результаты, представленные в табл. 3, показывают, что при отсутствии БНД могут присутствовать обычные НД. Для 4 полных циклов уменьшенного AES не найдено БНД, но найдены НД. Найденные НД можно назвать полубайтовыми, т.к., входную разность можно описать вектором активизации с одним активным битом, а для выходной разности важны сами значения в каждом из активных байтов. В выходной разности должно быть два активных полубайта, которые до последнего ShiftRow находятся в одной колонке. Значение разности в этих полубайтах должно быть таким, чтобы при выполнении последней

операции МС в обратном направлении была получена ненулевая разность только в одном полубайте. Подобные полубайтовые НД для четырех полных циклов существуют и для полноразмерных Rijndael-подобных шифров, но сведений о них в доступной литературе найдено не было. Возможно, использование этих полубайтовых НД может сделать известные атаки более эффективными. Однако данный вопрос требует более тщательного исследования.

Под условия утверждения 1 попадают все Rijndael-подобные SPN шифры, что позволяет впервые доказать отсутствие байтовых НД для 4 и более циклов шифра «Калина» (ДСТУ 7624:2014) со всеми размерами блоков, для 512 битных блочных шифров, которые используются в хеш-функциях Whirlpool, Groestl и «Купина» (ДСТУ 7564:2014).

5. АНАЛИЗ ШИФРОВ, ПОСТРОЕННЫХ С ИСПОЛЬЗОВАНИЕМ ЦЕПИ ФЕЙСТЕЛЯ

Схема фейстеля – одна из наиболее распространенных схем современных БСШ. В качестве шифра, для исследований взят алгоритм, который по структуре близок к шифрам Торнадо [14] и Лабиринт [15]. В каждом цикле выполняется SL-преобразование, схема которого представлена на рис. 5.

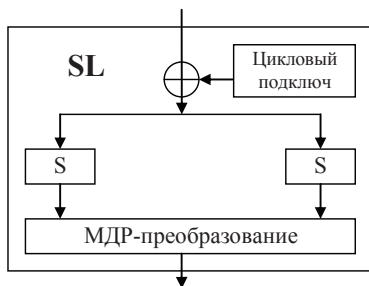


Рис. 5. SL-преобразование

Важным моментом является то, что МДР-преобразование (аналог MixColumn в Rijndael-подобных шифрах) охватывает весь обрабатываемый полублок. Поэтому за один цикл такое SL-преобразование может любую ненулевую разность на входе трансформировать в разность со всеми активными байтами в полублоке на выходе. Общая схема трех циклов преобразований представлена на рис. 6.

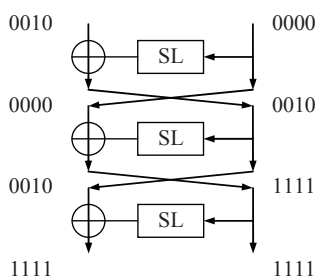


Рис. 6. Схема трансформации байтовой разности для трех циклов

В [1] показана справедливость следующего утверждения.

Утверждение 2. Для рассматриваемого шифра (схема фейстеля и в цикловом преобразовании МДР-преобразование покрывает весь полублок) не существует БНД, покрывающих 6 и более циклов.

Как и в предыдущей части для проверки полученных теоретических выводов с помощью представленных в таблицах 1 и 2 алгоритмов был проведен поиск НД и БНД для уменьшенной 16-битной версии алгоритма. Результаты представлены в таблицах 5 и 6.

Таблица 5
Результаты поиска НД для уменьшенной версии фейстель-подобного шифра

Количество циклов	Количество найденных НД	Комментарии
7 (S-блок max_dif = 10)	12	Например, 0x0100-0x0001
7 (S-блок max_dif = 4)	8	Например, 0x0100-0x0001
8	0	

Таблица 6
Результаты поиска БНД для уменьшенной версии фейстель-подобного шифра

Количество циклов	Количество найденных БНД	Комментарии
5	4	По два для входных векторов активизации 1000 и 0100
6	0	

Эксперименты по поиску обычных НД были проведены для подстановок с различным максимальным значением в таблице разности. Это значение указано в первой колонке табл. 5 для случаев, когда были найдены НД. В первом случае (первая строка табл. 5) максимальная вероятность прохождения ненулевой разности для подстановки 4 в 4 бита составляет 10/16 (S-блок max_dif = 10), а во втором (вторая строка табл. 5) - 4/16 (S-блок max_dif = 4).

Представленные результаты показывают, что дифференциальные свойства нелинейных подстановок не оказывают решающего влияния на стойкость БСШ к атаке НД. При этом, вполне ожидаемо, что при большем максимальном значении в таблице разности найдено больше НД, т. к., большее максимальное значение свидетельствует о большем количестве нулей (запрещенных переходов) в таблице разности.

Как и в предыдущем подразделе, отсутствие БНД не означает отсутствие НД для рассматриваемого шифра. В отличие от SPN шифров, где разница в числе циклов, необходимых для отсутствия БНД, от чис-

ла циклов, необходимых для отсутствия НД, составляет 1 цикл (см. табл. 3 и 4), в данном случае эта разница составляет 2 цикла. При этом результаты из табл. 6 подтверждают справедливость утверждения 2.

Под условия утверждения 2 попадают фейстель-подобные шифры с цикловой функцией, в которой используется МДР-преобразование, покрывающее весь полублок, что позволяет доказать отсутствие байтовых НД для 6 и более циклов шифров Торнадо и Лабиринт с размером блока 128 битов.

6. АНАЛИЗ ШИФРОВ, ПОСТРОЕННЫХ С ИСПОЛЬЗОВАНИЕМ СХЕМЫ ЛЕЯ-МЭССИ (LAI-MASSEY)

К наиболее известным шифрам, которые используют схему Lai-Massey, относятся шифры семейства Fox [16], шифр Мухомор [17]. Схема не является очень распространенной, поэтому и недостаточно изучена. В частности, один из пробелов – стойкость к атаке невыполнимых дифференциалов. В доступной литературе мы не встретили обоснование стойкости шифров с использованием схемы Lai-Massey к атаке невыполнимых дифференциалов. В этом подразделе продемонстрировано, как с использованием теоремы 2 может быть обоснована стойкость шифра, который использует схему Lai-Massey.

В качестве шифра, стойкость которого будем исследовать, взят алгоритм, в каждом цикле которого выполняется такое же SL-преобразование как и в предыдущем подразделе (см. рис. 5).

Схема двух циклов схемы Lai-Massey представлена на рис. 7.

Используя теорему 2 и лемму 3.4 из [16], покажем справедливость следующего утверждения.

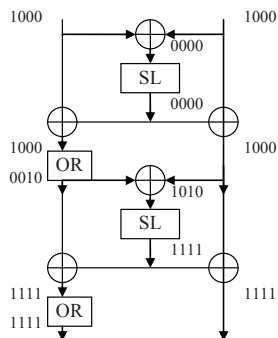


Рис. 7.Схема трансформации байтовой разности для двух циклов

Утверждение 3. Для рассматриваемого шифра (схема Lai-Massey и в цикловом преобразовании МДР-преобразование покрывает весь полублок) не существует БНД, покрывающих 4 и более цикла.

Доказательство. В соответствии с леммой 3.4 [16], в одном из двух подряд идущих циклов всегда будет ненулевая разность на входе SL-преобразования. Тогда МДР-преобразование может

распространить эту разность на все байты полублока. После такого цикла всегда может быть получена разность со всеми активными байтами.

Дешифрование выполняется по такой же схеме, как и шифрование (отличие только в OR преобразовании), поэтому два цикла дешифрования также гарантируют возможность получения из любой ненулевой разности разность со всеми активными байтами. Тогда в соответствии с теоремой 2, для такого шифра не существует БНД, покрывающих 4 и более цикла. Утверждение доказано.

Для проверки полученных теоретических выводов с помощью представленных в таблицах 1 и 2 алгоритмов был проведен поиск НД и БНД для уменьшенной 16-битной версии алгоритма. Результаты представлены в таблицах 7 и 8.

Для уменьшенной модели шифра, который использует схему Lai-Massey и Rijndael-подобное цикловое преобразование, вычислительные эксперименты подтвердили отсутствие 4-цикловых БНД (см. табл. 8). В то же время, обычные НД не были найдены лишь для 7 циклов.

Как и при анализе фейстель-подобных шифров, в табл. 7 для максимального количества циклов, когда еще существуют НД, представлены результаты проверки шифров с подстановками, обладающими различной максимальной вероятностью прохождения ненулевой разности. В первом случае максимальная вероятность прохождения ненулевой разности для подстановки 4 в 4 бита составляет 10/16 (S-блок max_dif = 10), а во втором – 4/16 (S-блок max_dif = 4). Представленные результаты демонстрируют отсутствие решающего влияния дифференциальных свойств нелинейных подстановок на стойкость БСШ данного вида к атаке НД.

Таблица 7

Результаты поиска НД для уменьшенной версии шифра, который использует схему Lai-Massey

Количество циклов	Количество найденных НД	Комментарии
6(S-блок max_dif = 10)	Около 50	Для каждой входной разности типа $0y0y_{16}$, где y – произвольное 16-ричное значение
6 (S-блок max_dif = 4)	Около 20	Для каждой входной разности типа $0y0y_{16}$, где y – произвольное 16-ричное значение
7	0	

Таблица 8

Результаты поиска БНД для уменьшенной версии шифра, который использует схему Lai-Massey

Количество циклов	Количество найденных НД	Комментарии
3	1	Вх. вект. активиз. 1000, вых. вект. активиз. 0010
4	0	

ЗАКЛЮЧЕНИЕ

Продемонстрировано применение предложенного в [1] метода для шифров, построенных по схеме SPN, по схеме фейстеля и по схеме Lai-Massey. Во всех случаях вычислительные эксперименты по поиску байтовых невыполнимых дифференциалов для уменьшенных моделей шифров подтвердили справедливость полученных теоретических выводов.

Показано, что при отсутствии БНД могут присутствовать обычные НД. Для уменьшенных шифров, построенных по схеме SPN, по схеме фейстеля и по схеме Lai-Massey обычные НД покрывают на 1, 2 и 3 цикла больше, чем БНД.

Представленные результаты показали, что дифференциальные свойства нелинейных подстановок не оказывают решающего влияния на стойкость БСШ к атаке НД и максимальное количество циклов, покрываемых НД, не меняется при различных параметрах подстановок. При этом, вполне ожидаемо, что при большем максимальном значении в таблице разности найдено больше НД, так как большее максимальное значение свидетельствует о большем количестве нулей (запрещенных переходов) в таблице разности.

Одним из перспективных направлений будущих исследований представляется изучение возможностей использования найденных обычных НД, которые покрывают большее количество циклов, чем БНД, в атаках на БСШ.

Литература.

[1] Руженцев В.И. О методе доказательства стойкости блочных шифров к атаке невыполнимых дифференциалов [Текст] / Руженцев В.И. // Прикладная радиоэлектроника. Тематический выпуск, посвященный проблемам обеспечения безопасности информации. Харьков. Том 12, №2, 2013. – С. 215 - 219.

[2] Biham E. Cryptanalysis of Reduced Variant of Rijndael [Electronic resource] / E. Biham, N. Keller // The Third Advanced Encryption Standard Candidate Conference, New York, USA, April 13–14, 2000. – Mode of access : www. URL: <http://csrc.nist.gov/archive/aes/index.html>.

[3] Improved Impossible Differential Cryptanalysis of Rijndael and Crypton [Text]/ Cheon, J.H., Kim, M., Kim, K., Lee, J.-Y., Kang, S. // In: Kim, K.-c. (ed.) ICISC 2001. LNCS, vol. 2288, pp. 39–49. Springer, Heidelberg (2002).

[4] New Impossible Differential Attacks on AES [Electronic resource] / J. Lu, O. Dunkelman, N. Keller, J. Kim // IACR Cryptology ePrint Archive 2008: 540 (2008).

[5] Biham E. Cryptanalysis of Skipjack Reduced to 31 Rounds

using Impossible Differentials [Text] / Biham E., Biryukov A., Shamir A. // Technion, CS Dept, Tech Report CS0947 (1998).

[6] Improving the efficiency of impossible differential cryptanalysis of reduced Camellia and MISTY1 [Electronic resource] / Lu, J., Kim, J., Keller, N., Dunkelman, O. // Mode of access : <http://jiqiang.googlepages.com>.

[7] Wu W. Impossible differential cryptanalysis of reduced-round ARIA and Camellia [Text] / W. Wu, W. Zhang, D. Feng. // Journal of Computer Science and Technology, 22(3):449-456, 2007. Springer.

[8] Impossible differential cryptanalysis for block cipher structures [Text] / J. Kim, S. Hong, J. Sung, S. Lee, J. Lim // INDOCRYPT 2003, LNCS 2904, pp. 82-96, 2003.

[9] A Unified Method for Finding Impossible Differentials of Block Cipher Structures [Electronic resource] / Y. Luo, Z. Wu, X. Lai, G. Gong // IACR Cryptology ePrint Archive 2009: 627 (2009).

[10] Li R. Impossible Differential Cryptanalysis of SPN Ciphers [Electronic resource] / Ruilin Li, Bing Sun, Chao Li // IACR Cryptology ePrint Archive 2010: 307 (2010).

[11] Yap H. Impossible Differential Characteristics of Extended Feistel Networks with Provable Security against Differential Cryptanalysis [Text]/ H. Yap // SecTech 2008, CCIS 29, pp. 103-121, 2009.

[12] Biham E. Miss in the Middle Attacks on Idea and Khufu [Text] / E. Biham, A. Biryukov, A. Shamir // Fast Software Encryption : proceedings of the 6th International Workshop, FSE'99, Rome, Italy, March 24–26, 1999. – Berlin ; Heidelberg : Springer, 1999. – P. 124–138. – (Lecture Notes in Computer Science ; vol. 1636).

[13] Kleiman E. The XL and XSL attacks on Baby Rijndael [Electronic resource] / E. Kleiman // Thesis, 2005. Mode of access : <http://orion.math.iastate.edu/dept/thesisarchive/MS/EKleimanMSSS05.pdf>.

[14] Горбенко И.Д. Алгоритм блочного симметричного шифрования «Горнадо». Спецификация преобразования [Текст] / Горбенко И.Д., Головашич С.А. // Радиотехника. 2003, № 134. – С. 60 – 80.

[15] Головашич, С. А. Спецификация алгоритма блочного симметричного шифрования «Лабиринт» [Текст] / С. А. Головашич // Прикладная радиоэлектроника. Тематический выпуск, посвященный проблемам обеспечения безопасности информации. Харьков. Том 6, №2, 2007. – С. 230 – 240.

[16] Junod P. FOX: a new family of block ciphers [Text] / P. Junod, S. Vaudenay // Selected Areas in Cryptography. – Berlin ; Heidelberg : Springer, 2005. – P. 114–129.

[17] Перспективний блоковий симетричний шифр «Мухомор»: основні положення та специфікація [Текст] / Р. В. Олійников, І. Д. Горбенко, М. Ф. Бондаренко, В. І. Руженцев // Прикладная радиоэлектроника. – 2007. – Т. 6, № 2. – С. 147–157.



Руженцев Виктор Игоревич, доктор технических наук, доцент, профессор кафедры БИТ ХНУРЭ. Научные интересы: симметричная криптография, криптоанализ.

УДК 004.056.55

Перевірка методу доведення стійкості блокових шифрів до атаки нездійснених диференціалів / В.І. Руженцев // Прикладна радіоелектроніка: наук.-техн. журнал. – 2016. Том 15, № 3. – С. 184 – 190.

Обговорюються результати обчислювальних експериментів з пошуку нездійснених диференціалів для зменшених моделей блокових шифрів. Підтверджується справедливість висновків, отриманих за допомогою запропонованого в роботі [1] методу. Демонструються нові знайдені нездійснені диференціали, які покривають більшу кількість циклів, ніж відомі.

Ключові слова: блоковий шифр, атака нездійснених диференціалів, нездійснений диференціал, Rijndael-подібні перетворення.

Табл.: 08. Іл.: 07. Бібліогр.: 17 назв.

UDC 004.056.55

Analysis of the method of proving the resistance of block ciphers to impossible differential attack / V.I. Ruzhentsev // Applied Radio Electronics: Sci. Journ. 2016. – Vol. 15, № 3. – P. 184 – 190.

The results of computing experiments on impossible differentials searching for reduced models of block symmetrical ciphers are discussed. The validity of the conclusions obtained with the help of the method of substantiating the lack of impossible differentials, that is suggested in the work [1], is confirmed. New impossible differentials which cover more rounds than the known ones are presented for some types of ciphers.

Keywords: block cipher, impossible differential attack, impossible differential, Rijndael-like transformations.

Tab.: 08. Fig.: 07. Ref.: 17 items.

ДЕТЕРМИНИРОВАННЫЕ ГЕНЕРАТОРЫ ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ДЛЯ ПОТОКОВОГО ШИФРОВАНИЯ НА ОСНОВЕ ДЛРР

А.А. ТОРБА, В.А. БОБУХ, М.О. ТОРБА, А.О. ТОРБА

В работе проанализированы алгоритмы потокового шифрования на основе динамических линейных рекуррентных регистров (ДЛРР) и методы повышения их криптостойкости. Эти алгоритмы используют рандомизированный подход, основанный на создании объёмной задачи; криптограф тем самым пытается сделать решение задачи дешифрования физически невозможным.

Ключевые слова: потоковый шифр, динамический линейный рекуррентный регистр, гаммирующая последовательность, рандомизированный подход.

ВВЕДЕНИЕ

Максимальная скорость передачи информации в каналах связи с ограниченным доступом определяется быстродействием аппаратных (или программно-аппаратных) алгоритмов шифрования и расшифрования сообщений.

Наибольшим быстродействием среди известных симметричных алгоритмов криптографических преобразований обладают потоковые алгоритмы, которые позволяют формировать каждый очередной бит псевдослучайной гаммы за один такт синхронизации.

Потоковые шифры, которые шифруют и дешифруют данные по одному биту, не очень подходят для программных реализаций. А блочные шифры легче реализовывать программно, т. к. они позволяют избежать трудоемких манипуляций с битами и оперируют удобными для компьютера блоками данных, соизмеримыми с разрядностью регистров общего назначения (РОН). С другой стороны, потоковые шифры на регистрах сдвига больше подходят для аппаратной реализации.

Согласно Райнеру Рюппелю можно выделить четыре основных подхода к проектированию потоковых шифров (ПШ):

- Системно-теоретический подход основан на создании для криптоаналитика сложной, ранее неисследованной проблемы.

- Сложностно-теоретический подход основан на сложной, но известной проблеме (например, факторизация чисел или дискретное логарифмирование).

- Информационно-технический подход основан на попытке утаить открытый текст от криптоаналитика – вне зависимости от того сколько времени потрачено на дешифрование, криптоаналитик не найдёт однозначного решения.

- Рандомизированный подход основан на создании объёмной задачи; криптограф тем самым пытается сделать решение задачи дешифрования физически невозможным.

Большое количество реальных потоковых шифров основано на регистрах сдвига с линейной обратной связью – линейных рекуррентных регистрах (ЛРР). Основные преимущества ЛРР:

- Высокое быстродействие криптографических алгоритмов;

- Применение только простейших операций сложения и умножения, аппаратно реализованных практически во всех вычислительных устройствах;

- Хорошие криптографические свойства (генерируемые последовательности имеют большой период и хорошие статистические свойства);

- Легкость анализа с использованием алгебраических методов за счет линейной структуры.

Сами по себе ЛРР являются хорошими генераторами псевдослучайных последовательностей, но они обладают некоторыми нежелательными неслучайными свойствами. Для ЛРР с количеством разрядов « n » внутреннее состояние представляет собой предыдущие « n » выходных битов генератора. Даже если параметры рекуррентны (номера отводов m_k обратной связи) и хранятся в секрете, то они могут быть определены по $2n$ выходным битам генератора с помощью алгоритма Берлекэмп-Мэсси.

Существует несколько методов проектирования генераторов псевдослучайного ключевого потока, которые разрушают линейные свойства ЛРР и тем самым делают такие системы криптографически более стойкими:

- использование нелинейной функции, объединяющей выходы нескольких ЛРР (генератор Геффа и др.);

- использование нелинейной фильтрующей функции для содержимого каждой ячейки единственного ЛРР;

- использование выхода одного ЛРР для управления синхросигналом одного (или нескольких) ЛРР (алгоритм А5 и др.);

- динамическое изменение параметров рекурренты (длины регистра « n » и номеров отводов m_k) в

процессе формирования псевдослучайной гаммирующей последовательности, – так называемые динамические линейные рекуррентные регистры (ДЛРР).

ОСНОВНАЯ ЧАСТЬ

Простейший детерминированный генератор псевдослучайных последовательностей для потокового шифрования на основе ДЛРР «AUGUST-1», описанный в патенте Украины [1,2], позволяет динамически изменять параметры рекурренты в процессе формирования псевдослучайной гаммы.

Скорость формирования псевдослучайной последовательности определяется быстродействием программируемых логических интегральных схем (ПЛИС) и может составлять от 10 МГц до 1 ГГц.

Длина секретного ключа K_s в битах определяет криптостойкость алгоритма потокового шифрования и равняется разрядности « n » сдвигающего регистра RG1. При использовании современных ПЛИС разрядность регистра RG1 (и секретного ключа K_s) может составлять от 100 до нескольких тысяч бит.

Выходная псевдослучайная последовательность гаммы является детерминированной (т.е. может быть полностью восстановлена на приемной стороне канала связи) и зависит от секретного значения кратковременного сеансового ключа K_s , от случайного значения инициализации IV и долговременных секретных параметров (ключей):

- длины секретного ключа « n »,
 - таблицы коммутации мультиплексора MS и
 - коэффициента деления первого счетчика СТ1.
- Секретное значение длины « n » кратковременного

сеансового ключа K_s делает бесполезной лобовую атаку по перебору всех значений ключа.

В алгоритме потокового шифрования «AUGUST-2» [3,4] для увеличения криптостойкости генератора гаммирующей последовательности на основе ДЛРР предложено изменять величины интервалов времени между сменами параметров рекурренты в псевдослучайном порядке.

Эти временные интервалы задаются счетчиком с программируемым коэффициентом деления, информационные входы которого подключены в произвольном порядке к выходам сдвигающего регистра. Поэтому величины временных интервалов будут зависеть от начального значения сеансового ключа K_s , значения инициализации IV и текущего состояния сдвигающего регистра.

Это позволяет реализовать один из критериев Райнера Рюппеля: «Каждый бит гаммирующей последовательности должен быть сложным преобразованием большинства битов ключа».

Также преимуществом алгоритма «AUGUST-2» является введение второго выходного элемента «ИСКЛЮЧАЮЩЕЕ ИЛИ» (элемента «XOR»), входы которого подключены в произвольном порядке к выходам ДЛРР. С выхода этого элемента «ИСКЛЮЧАЮЩЕЕ ИЛИ» снимается псевдослучайная гаммирующая последовательность. Это улучшает статистические свойства формируемой гаммы, а именно: уменьшает разность вероятностей «нулей» и «единиц» выходной последовательности, а также уменьшает нормированные коэффициенты автокорреляционной функции [5].

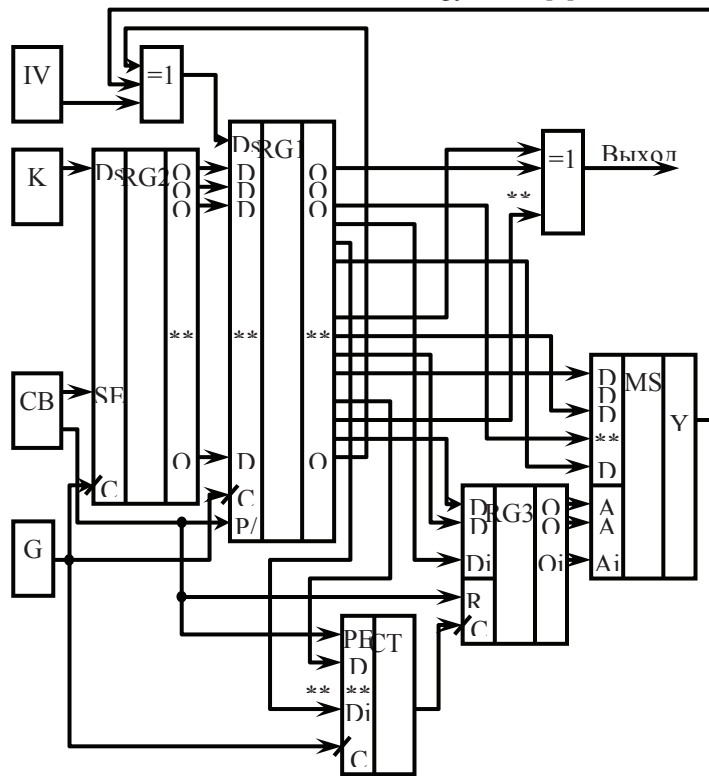


Рис. 1. Алгоритм потокового шифрования «AUGUST-4»

В этом алгоритме добавлены новые секретные долговременные параметры:

- диапазон изменения коэффициента деления счетчика СТ1;
- номера выходов регистра RG1, которые подключены к информационным входам счетчика СТ1.

В алгоритме потокового шифрования «AUGUST-3» [4,6] для увеличения криптостойкости генератора гаммирующей последовательности на основе ДЛРР предложено изменять параметры рекурренты в псевдослучайном порядке.

Для этого на адресные входы мультиплексора подаются двоичные коды с выходов дополнительного параллельного регистра RG3, в котором через фиксированные интервалы времени сохраняются коды с произвольных выходов RG1.

В алгоритме потокового шифрования «AUGUST-4» [7] (рис.1) для увеличения криптостойкости генератора гаммирующей последовательности на основе ДЛРР объединены преимущества алгоритмов «AUGUST-2» и «AUGUST-3».

Параметры рекурренты (номера отводов «n») ДЛРР на основе сдвигающего регистра RG1 изменяются в псевдослучайном порядке.

Для этого на адресные входы мультиплексора MS подаются логические уровни с выходов параллельного регистра RG3, который запоминает псевдослучайные сигналы с произвольных выходов сдвигающего регистра RG1.

Псевдослучайные временные интервалы смены параметров рекурренты задаются делителем с про-

граммируемым коэффициентом деления СТ.

Такое техническое решение с нелинейным характером изменения параметров рекурренты еще более усложняет криптоанализ, осуществить который в разумные сроки – физически невозможно.

В алгоритме потокового шифрования «AUGUST-5» [8] (рис.2) для увеличения криптостойкости генератора гаммирующей последовательности на основе ДЛРР введено несколько мультиплексоров, изменяющих параметры рекурренты. Например, один мультиплексор коммутирует отводы рекуррентного регистра RG1, которые определяют длину ДЛРР, а остальные мультиплексоры изменяют номера отводов рекуррентного регистра.

Возможна также ситуация, при которой, отвод регистра RG1, определяющий длину ДЛРР, в следующем такте может стать промежуточным отводом, а длина ДЛРР формируется другим мультиплексором.

На рис. 2 приведен случай, когда номера отводов ДЛРР коммутируются в постоянном порядке и через фиксированные временные интервалы. Возможно также изменить порядок коммутации отводов на псевдослучайный (как в алгоритме «AUGUST-3»), или через псевдослучайные временные интервалы (как в алгоритме «AUGUST-2»), или одновременно использовать псевдослучайное управление (как в алгоритме «AUGUST-4»).

ЗАКЛЮЧЕНИЕ

Предложенные и запатентованные алгоритмы потокового шифрования «AUGUST-1», «AUGUST-2», «AUGUST-3», «AUGUST-4» и «AUGUST-5» разрушают линейные свойства ЛРР и тем самым делают

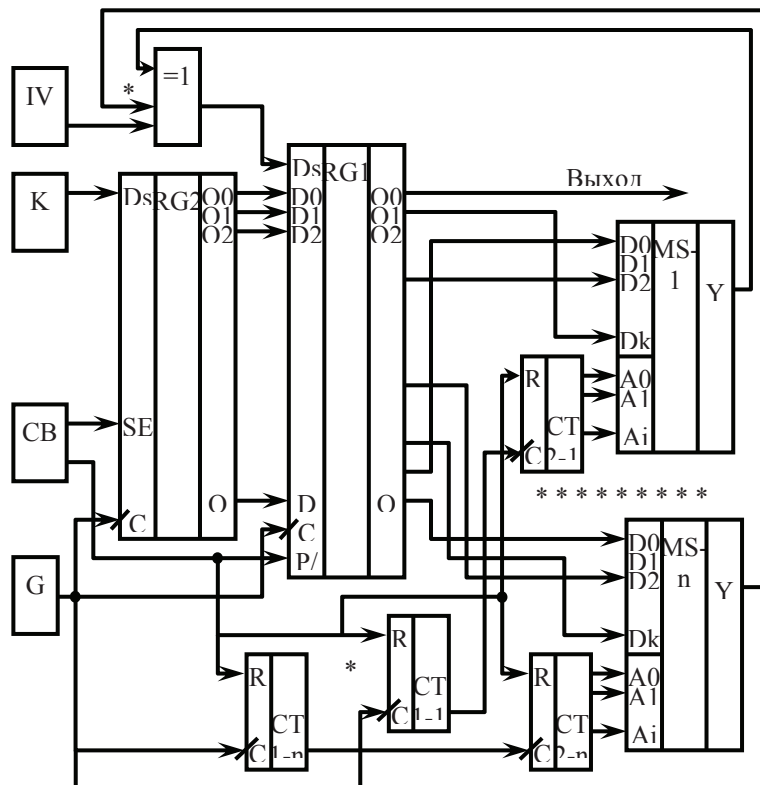


Рис. 2. Алгоритм потокового шифрования «AUGUST-5»

такие системы криптографически более стойкими за счет динамического изменения параметров рекуррентности в процессе формирования псевдослучайной гаммирующей последовательности.

Криптостойкость предложенных алгоритмов потокового шифрования определяется разрядностью кратковременного секретного ключа K_s , которая может составлять от 100 до нескольких тысяч бит. Причем секретным является не только значение ключа K_s , но и его длина.

Скорость формирования псевдослучайной гаммирующей последовательности ограничивается быстродействием используемых логических микросхем и может достигать 1000 МГц

В отличие от известных криптоалгоритмов (DES, AES и др.), в которых полностью известен математический аппарат криптопреобразований, а неизвестным является только единственный секретный параметр – кратковременный ключ, – в предложенных алгоритмах на основе ДЛРР присутствует очень большое количество долговременных секретных параметров (полный перебор которых может занять миллиарды лет).

Поэтому криптоанализ таких алгоритмов с перебором всех долговременных секретных параметров и для каждого такого параметра перебор всех значений секретного кратковременного (сеансового) ключа является физически невозможным в разумные сроки.

Литература.

- [1] Патент Украины на полезную модель № 85039, опубл. Бюл. № 21, 2013 г.
- [2] *Торба А.А.* Быстродействующий детерминированный генератор псевдослучайных последовательностей для потокового шифрования // [Текст]. А.А. Торба, В.А. Бобух, А.А. Бобкова. – Прикладная радиоэлектроника: науч.-техн. журнал. – 2014.– Том 13.– №3.– С. 316 – 318.
- [3] Патент Украины на полезную модель № 93477, опубл. Бюл. № 19, 2014 г.
- [4] *Торба А.А.* Методы повышения криптостойкости алгоритмов потокового шифрования // [Текст]. А.А. Торба, В.А. Бобух, М.О.Торба, А.О.Торба.– // Радиотехника : Всеукр. межвед. науч.-техн. сб. – 2016. – Вып. 184. – С. 178 – 183.
- [5] *Торба А.А.* Методы и средства генерации случайных битовых последовательностей // [Текст]. А.А. Торба, А.А. Бобкова, Ю.И. Горбенко, В.А. Бобух.– Под ред. д.т.н., профессора Горбенко И.Д. – Харьков: Изд-во «Форт», 2012.– 232 с.
- [6] Патент Украины на полезную модель № 93117, опубл. Бюл. № 18, 2014 г.
- [7] Патент Украины на полезную модель № 99194, опубл. Бюл. № 10, 2015 г.
- [8] Патент Украины на полезную модель № 97734, опубл. Бюл. № 7, 2015 г.



Торба Александр Алексеевич, кандидат технических наук, профессор кафедры ЭВМ, ХНУРЭ. Область научных интересов: аппаратные средства криптографических систем.



Бобух Всеволод Анатольевич, кандидат технических наук, начальник отдела аппаратных средств. Область научных интересов: аппаратные средства криптографических систем.



Торба Максим Олегович, студент ХНУРЭ. Область научных интересов: программирование баз данных, аппаратные средства криптографических систем.



Торба Александр Олегович, студент, ХНУРЭ. Область научных интересов: компьютерная анимация, аппаратные средства криптографических систем.

УДК 681.324.067

Детерміновані генератори псевдовипадкових послідовностей для потокового шифрування на основі ДЛРР / О.О. Торба, В.А. Бобух, М.О. Торба, О.О. Торба // Прикладна радіоелектроніка: наук.-техн. журнал. – 2016. – Том 15, № 3.– С. 191 – 194.

В роботі проаналізовано алгоритми потокового шифрування на основі динамічних лінійних рекурентних регістрів (ДЛРР) і методи підвищення їх криптостійкості. Ці алгоритми використовують рандомізований підхід, заснований на створенні об'ємної задачі; криптограф тим самим намагається зробити розв'язання завдання дешифрування фізично неможливим.

Ключові слова: потоковий шифр, динамічний лінійний рекурентний регістр, гамуюча послідовність, рандомізований підхід.

Лл.: 02. Бібліогр.: 08 найм.

UDC 681.324.067

Deterministic pseudorandom sequence generators for stream-based encryption D L R R / A.A. Torba, V.A. Bobuch, M.O. Torba, A.O. Torba // Applied Radio Electronics: Sci. Journ.– 2016.– Vol. 15, № 3.– P. 191 – 194.

The algorithms of streaming encryption based on dynamic linear recurrent registers (DLRR) and methods to improve their reliability are analyzed in the paper. These algorithms use a randomized approach based on a voluminous task; thus, the cryptographer tries to make the solution of a decrypting problem physically impossible.

Keywords: stream cipher, linear dynamic recurrent register, gamma sequence, randomized approach.

Fig.: 02. Ref.: 08 items.

ПОСТКВАНТОВЫЕ ЭЛЕКТРОННЫЕ ПОДПИСИ

УДК 003.026:004.056

АНАЛІЗ ПОСТКВАНТОВИХ МЕХАНІЗМІВ ЕЛЕКТРОННИХ ПІДПИСІВ НА ОСНОВІ ГЕШ-ФУНКЦІЙ

Н.В.КОВАЛЬОВА, Ю.І.ГОРБЕНКО

Розглядається сутність криптоперетворень, що ґрунтуються на використанні функцій гешування при побудованні постквантових електронних підписів. Наводяться результати аналізу криптографічної стійкості вказаних електронних підписів та даються оцінки існуючих алгоритмів та рекомендації з їх застосування.

Ключові слова: алгоритми постквантового електронного підпису, функції гешування, криптографічна стійкість, застосування у постквантовий період.

ВСТУП

У 2016 році відбулось ряд значущих подій, які уже суттєво вплинули на інтенсивний розвиток постквантової криптографії. До них необхідно віднести заяву у вигляді Інтернет – статті Менезеса (ALFRED J. MENEZES) та Кобліца (NEAL KOBLITZ) [4], організацію та проведення NSA та NIST США VII міжнародної конференції з постквантової криптографії, що проходила у лютому 2016 року у Японії [4]. Надзвичайно важливою подією стало опублікування в США звіту «Report on Post – Quantum Cryptography. NISTIR 8105 (DRAFT) [5]», в якому повністю підтверджено можливості успішного квантового криптоаналізу асиметричних криптосистем електронного підпису (ЕП), а також визначені основні проблеми та можливості і етапи їх вирішення. Серед можливих методів та алгоритмів побудови постквантових асиметричних криптосистем ЕП виділяють метод, що ґрунтується на використанні криптографічних перетворень на функціях гешування (НВ криптографія)[2,3]. Можливість впровадження вказаного методу пов'язано з доведенням криптографічної стійкості, забезпеченням необхідної складності (швидкодії) перетворень та обґрунтуванням і побудуванням загальних параметрів та ключів. Зважаючи на стан досліджень в указаному напрямку, на наш погляд, актуальними є задачі відбору та аналізу НВ криптоперетворень по критеріях криптографічної стійкості у постквантовий період, а також дослідження інших властивостей, що визначені NIST США та провідним інститутом ЄС ETSI[4,5].

В [4,5] наведено класи вимог, основними з них є: вимоги з безпеки; техніко-економічні вимоги та техніко-експлуатаційні вимоги. Також запропоновано критерії та показники відбору серед запропонованих кандидатів. Основними серед них є:

- модель безпеки для електронного (цифрового) підпису;

- вимоги до стійкості та додаткові властивості безпеки;
- обґрунтованість та довіра до методів, їх прозорість;
- розміри загальних параметрів та ключів;
- часова та просторова складності тощо.

Також важливим є обґрунтування та виконання відносно об'єктивного порівняльного аналізу ЕП, побудованих на основі різних методів і механізмів.

В ході розв'язання цієї задачі, в першу чергу бажано скористатись методиками, що, наприклад, подані в [5].

Метою цієї статті є відбір кандидатів на постквантові ЕП на основі НВ криптоперетворень, аналіз їх властивостей, обґрунтування та формулювання основних задач аналізу складності побудування параметрів та ключів, а також прямих та зворотних асиметричних криптоперетворень. Зрозуміло, що суттєвою актуальною є проблема доведення криптографічної стійкості, на наш погляд вона розв'язуватиметься на світовому рівні ще декілька років.

1. КРИПТОСИСТЕМИ ЕП, ЩО ЗАСНОВАНІ НА ФУНКЦІЯХ ГЕШУВАННЯ

В [2,5] запропоновані криптографічні перетворення типу електронний підпис (ЕП) для постквантового періоду. Вони ґрунтуються на одноразових схемах ЕП з використанням геш - функцій. Попередній аналіз дозволив виділити серед них ЕП на основі геш - функцій, що відомі як Лампорт - Діффі або Вінтерніц підписи. Особливістю таких підписів є те, що їх криптографічна стійкість ґрунтується на колізійній стійкості геш-функцій, які застосовуються в механізмах (схемах) ЕП. Оскільки сьогодні уже розроблено та прийнято як геш-функції ряд, по суті постквантових геш-функцій, запропоновані механізми, на наш погляд, є перспективними.

Оскільки підписи Вінтерніц і Лампорт - Діффі не можуть бути використані надійно більш, ніж один раз, вони об'єднані з такими структурами, як бінарні де-

рева так, що замість використання ключа підпису для одноразового використання підпису, ключ може бути використаний для ряду підписів, кількість яких обмежена розміром двійкового дерева. Основною концепцією, що лежить у використанні двійкового дерева зі схемами геш - підпису є те, що кожна позиція на дереві розраховується як геш - конкатенація своїх дочірніх вузлів. Вузли, таким чином, обчислюються послідовно з коренем дерева, що є відкритим ключем глобальної схеми підпису. Листя дерева побудовані з одноразових ключів перевірки підпису.

Ця ідея була введена Merkle в 1979 році, але мала ряд недоліків – великі ключі, великий розмір підпису та достатньо повільний підпис.

Значна перевага ЕП на основі геш – це їх гнучкість, оскільки вони можуть бути використані з будь-якою захищеною функцією гешування, і тому, якщо дефект виявлений в захищеній геш - функції, ЕП просто необхідно перевести на нову – стійку геш - функцію, щоби ЕП був стійким.

Значним недоліком схем, пов'язаних з Merkle, є те, що підписувач повинен стежити, які одноразові ключі підпису вже використовувалися. Це може бути складним у великомасштабних середовищах. Варіанти без збереження стану є предметом поточних досліджень. З точки зору ефективності, наступні ітерації доробки значно поліпшили схеми ЕП на основі геш - функцій, але деякі недоліки залишаються. Так для отримання порівняння бітового рівня безпеки, екземпляр XMSS з AES-128 виробляє підписів в десять разів більше, ніж RSA-2048, тобто ЕП здійснює швидше.

На сьогодні існують декілька схем ЕП, заснованих на геш - функціях, серед яких схема Merkle, XMSS, SPHINCS.

2. ОСНОВНІ СХЕМИ ЕЛЕКТРОННИХ ПІДПИСІВ ПОСТКВАНТОВОГО ПЕРІОДУ НА ОСНОВІ ГЕШ-ФУНКЦІЙ

2.1 Схема підпису Merkle

Найбільша проблема одноразових схем ЕП – це управління ключами [10]. Заміна відкритого ключа є дуже складною. Має бути гарантія того, що відкритий ключ належить передбачуваному партнеру з обміну даними і, що відкритий ключ не був змінений. Таким чином, мають використовуватися кілька відкритих ключів, і відкриті ключі мають бути досить короткими. Але в одноразових схемах для кожного підпису використовується новий відкритий ключ, який є досить великим порівняно з іншими схемами. Для того, щоб зробити одноразові схеми підпису можливими, необхідно застосовувати ефективне управління ключами, що зменшить кількість відкритих ключів та їх розміри. Merkle представив схему підпису Merkle (MSS), в якій один відкритий ключ використовується для підпису багатьох повідомлень.

2.1.1 Генерація ключових даних

Схема ЕП Merkle може використовуватися для

підписування обмеженого числа повідомлень з одним відкритим ключем K_{pub} . Число можливих повідомлень має бути ступенем двійки, оскільки можливу кількість повідомлень можна позначити, як $N=2^n$. Перший етап формування відкритого ключа – це створення відкритих ключів X_i і секретних ключів Y_i для 2^n разових підписів. Для кожного відкритого ключа Y_i з $1 \leq i \leq 2^n$ обчислюється геш - значення $h_i = H(Y_i)$. З цими геш - значеннями будується дерево Merkle (так зване геш - дерево). Вузол дерева позначається $a_{i,j}$, де i – це рівень вузла. Рівень вузла визначається відстанню від вузла до листа. Отже, лист дерева має рівень $i = 0$ і корінь має рівень $i = n$. Всі вузли одного рівня нумеруються зліва направо, так що $a_{i,0}$ є крайнім лівим вузлом i -го рівня. У дереві Merkle геш - значення h_i є листям бінарного дерева, таким, що $h_i = a_{0,i}$. Кожен внутрішній вузол дерева є геш - значенням конкатенації двох своїх дочірніх вузлів. Так, $a_{1,0} = H(a_{0,0}||a_{0,1})$ та $a_{2,0} = H(a_{1,0}||a_{1,1})$. Приклад дерева Merkle показано на рис. 1. Таким чином будується дерево із 2^n листям та $2^{n+1} - 1$ вузлами. Корінь дерева $a_{n,0}$ є відкритим ключем схеми підпису Merkle.

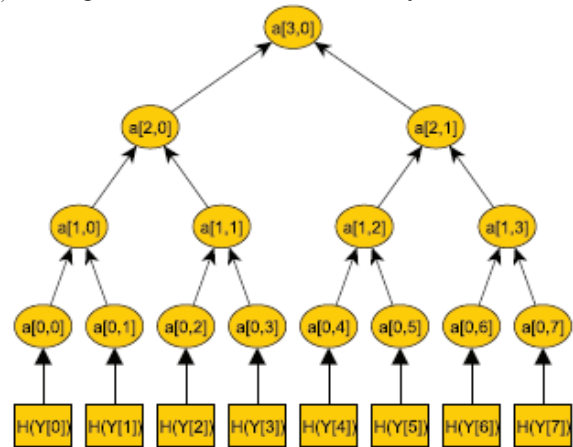


Рис.1. Дерево Merkle з 8 листями

2.1.2 Генерація підпису

Повідомлення M підписується одноразовою схемою підпису, утворюючи в результаті підпис sig' . Це робиться за допомогою однієї пари ключів (X_i, Y_i) . Відповідний до відкритого одноразового ключа лист геш-дерева Y_i - це $a_{0,i} = H(Y_i)$. Шлях геш - дерева є від $a_{0,i}$ до кореня A . Шлях A складається з $n + 1$ вузлів, A_0, \dots, A_n , з листям $A_0 = a_{0,i}$, та коренем дерева $A_n = a_{n,0} = pub$. Для обчислення шляху A потрібен кожний з дочірніх вузлів A_1, \dots, A_n . Таким чином відомо, що A_i є дочірнім вузлом A_{i+1} . Для обчислення наступного вузла A_{i+1} шляху A необхідно знати обидва дочірніх вузла A_{i+1} . Тому потрібен вузол – «родич» A_i . Він позначається $auth_i$, такий, що $A_{i+1} = H(A_i||auth_i)$. Отже, для того, щоб обчислити кожен вузол шляху A потрібно n вузлів $auth_0, \dots, auth_{n-1}$. Далі дані вузлів $auth_0, \dots, auth_{n-1}$ зберігаються. Ці вузли з одноразовим підписом sig' повідомлення M є ЕП $sig = (sig' || auth_2 || auth_3 || \dots || auth_{n-1})$ схеми Merkle. Приклад

шляху автентифікації проілюстрований на рис. 2.

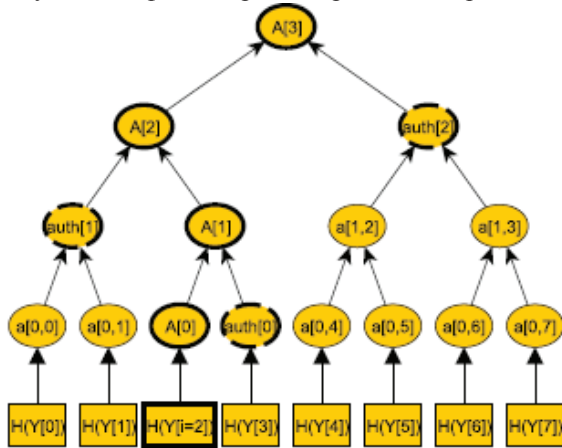


Рис. 2. Дерево Merkle з шляхом А та автентифікацією шляху для $i=2$

2.1.3 Перевірка підпису

Приймач знає відкритий ключ K_{pub} , повідомлення M , і підпис $sig = (sig' || auth_0 || auth_1 || \dots || auth_{n-1})$. По-перше, приймач перевіряє одноразовий ЕП sig' повідомлення M . Якщо sig' є дійсним ЕП M , то приймач обчислює $A_0 = H(Y_i)$ шляхом гешування відкритого ключа одноразового підпису. Для $j = 1, \dots, n-1$, вузли A_j шляху A обчислюються так: $A_j = H(a_{j-1} || b_{j-1})$.

Якщо A_n співпадає з відкритим ключем K_{pub} схеми підпису Merkle, то ЕП є дійсний.

2.1.4 Аналіз властивостей ЕП

Велика перевага схеми підпису Merkle в тому, що багато ЕП можуть генеруватися з використанням тільки одного відкритого ключа. Проте, ця перевага приходить зі збільшенням часу обчислень і довжини підпису.

Для того, щоб генерувати відкритий ключ K_{pub} , мають бути генеровані 2^n одноразових ключів підпису. Також має бути обчислений кожен вузол геш-дерева. Дерево складається з $2^{n+1}-1$ вузлів. Для розрахунку вузла необхідна одна геш-операція, так що $2^{n+1}-1$ геш-операцій необхідно для створення відкритого ключа. Очевидно, що розмір такого дерева обмежений. Так обчислення 2^{40} вузлів є дуже дорогим, обчислення 2^{80} вузлів – неможливим. Для створення підпису необхідні вузли $auth_0, \dots, auth_{n-1}$. Якщо вузли дерева не збережені, вони мають бути генеровані знову для кожного підпису. Створення дерева – є надто складним, так що генерування всього дерева для кожного підпису нездійсненно для великих дерев. Але збереження всіх $2^{n+1}-1$ вузлів призведе до суттєвих вимог до зберігання даних. Отже, необхідна обґрунтована стратегія, щоб обчислювати ЕП швидко без збереження занадто багатої кількості вузлів. Ця проблема називається проблемою обходу дерева Merkle.

Час перевірки ЕП порівняно з часом підпису значно менший. Спочатку має бути перевірений одноразовий підпис. Після цього, обов'язково має бути обчислений шлях $A = A_1, \dots, A_n$. Щоб зробити це, необхідно тільки n геш-операцій, по одній для кожного

вузла. Підпис схеми Merkle складається з одноразового підпису sig' та n вузлів $auth_0, \dots, auth_{n-1}$. Якщо використовується геш-функція 160 біт, розмір підпису буде $|sig| = |sig'| + n * 160$ біт.

2.2 Протокол Лейтона і Мікалі

Протокол ключової угоди (або розподілу) є набором правил зв'язку, за яких два користувачі можуть встановити загальний ключ. Загальний ключ може використовуватися користувачами в майбутньому для забезпечення захищеного зв'язку, наприклад, засобом симетричного шифрування.

На конференції CRYPTO'93 Лейтон і Мікалі запропонували два основні протоколи угоди [6], які були спрямовані на такі сценарії зв'язку, як засновані на чипі Clipper Chip. Потім пропозиції були додатково розширені у вигляді [7]. Перший протокол представлений в роботі [7] є новим і не описаний в [6]. Другий протокол в міститься в [7], по суті він такий же, як перший протокол, що міститься в [6]. Третій протокол також наведений в [7]. В ньому наведено механізм оптимізації другого протоколу [6]. Всі ці протоколи в [7] позначаються LM_1, LM_2 і LM_3 відповідно.

ЕП LM_1 концептуально дуже простий. Проте, протокол не практичний з точки зору кількості секретних ключів, які мають зберігатися користувачем. У LM_1 кількість секретних ключів (кожен з k бітів, для кожного окремого користувача) знаходиться в діапазоні від $O(B^2 \log N)$ до $O(B^3 \log N)$, де N – загальне число користувачів і B – максимальне число нечесних користувачів у системі. Як правило, $k \geq 64$. Тепер припустимо, що LM_1 застосовується в країні з десятьма мільйонами ($N = 2^{23}$) користувачів, серед яких тисячі ($B \approx 2^{10}$) є зловмисниками. Тоді число секретних ключів, які кожен користувач повинен підтримати, дорівнює щонайменше 2^{24} , що ще гірше, ніж просте рішення, в якому кожен користувач тримає $N - 1$ секретних ключів.

LM_3 у першу чергу є варіантом без пам'яті з протоколу на основі узгодження ключа сервера автентифікації (наприклад, протокол Нідхам-Шредера). Секретний ключ бази даних сервера автентифікації видаляється за допомогою методики, яка в даний час стала класичним способом зниження пам'яті, а саме - використання криптографічно «сильної» псевдовипадкової функції. На практиці криптостійкі псевдовипадкові функції зазвичай реалізуються за допомогою секретного ключа алгоритму шифрування в ході застосування стійкого симетричного блокового шифру.

Протокол LM_2 заснований на захищеному VLSI чипі, який містить центральний процесор разом з внутрішньою пам'яттю. Він також припускає існування довіреного агента (або групи агентів з щонайменше одним надійним).

2.3 Схема підпису SPHINCS

Двома основними проблемами, що пов'язані зі схемами підпису на основі геш-функцій, є необхідність підтримувати стан і розмір підписів [8,11]. Цим

можна запобігти тому, щоб рішення, які засновані на геш - функціях, були відповідною заміною для схем ЕП, які використовуються в даний час. SPHINCS вирішує цю проблему шляхом об'єднання підходу Голдрайху з конструкцією традиційного дерева Merkle. Ця вбудована конструкція дерев становить основу SPHINCS.

Повна структура SPHINCS складається в цілому з h шарів, поділених над d шарами суб - дерев. Це можна розглядати як гіпердерево двох рівнів абстракцій, де кожен вузол в глобальному дереві є суб-деревом. Кожне з цих суб-дерев складається з h/d шарів самих вузлів. Позначимо дерева, як τ_i , де $i \in \{1, \dots, d\}$ представляє їх шар у глобальному дереві, і вузли в суб-дереві як $v_{i,j}$, де $j \in \{1, \dots, h/d\}$ – це їх рівень в суб-дереві.

Дерева τ_i є бінарними геш - деревами, лише незначно змінюючись від вихідного дерева Merkle. Кожен з їх вузлів $v_{i,j}$ для $j \in \{1, \dots, h/d - 1\}$ містить геш - значення його дочірніх вузлів, в той час як кожен з вузлів листа на шарі h/d містить ключ до OTS. Припустимо, що ми маємо деяку геш-функцію H , з використанням якої обчислюються ці геш-значення. Як і у випадку з деревами Merkle, дайджест в корені дерева може використовуватися для автентифікації всієї структури за допомогою побудови шляхів автентифікації. Всі ці суб-дерева потім з'єднуються один з одним, як у системі Голдрейх. Використовуючи OTS-ключі у вузлах листа $v_{i,h/d}$ дерев τ_i , підписуються кореневі вузли дерев τ_{i+1} ; нове суб-дерево прикуте до кожного з вузлів листа. Примітки H містять геш своїх дочірніх вузлів, в той час як вузли OTS включають в себе пару ключів для автентифікації їх дочірніх вузлів.

Ключові пари OTS в листі дерев на нижньому шарі не використовуються для перевірки автентичності більшої кількості однакових суб-дерев [8,11]. Замість цього вони використовуються для перевірки справжності відкритого ключа схеми багаторазового підпису (FTS). FTS застосовується аналогічно до OTS, але може використовуватися кілька разів, перш ніж розкриє занадто багато даних секретного ключа. В ході використання FTS, а не OTS, SPHINCS не вимагає стільки вузлів листа для підтримки того ж рівня безпеки; необхідна максимальна ймовірність вибору повторно одного і того ж вузла може бути набагато вище, не порушуючи систему. Весь шлях на нижньому шарі дерева для підпису повідомлень використовуються ці FTS-ключі.

Рисунок 3 описує основну схему SPHINCS.

2.3.1 Генерація ключових даних

Генерація ключів для SPHINCS є досить дешевою операцією завдяки структурі Голдрайху. Виділяються випадкові значення $SK_1 \in \{0, 1\}^n$ і $SK_2 \in \{0, 1\}^n$. Перше з них використовується для генерації ключів, але зручно мати довгострокове випадкове число,

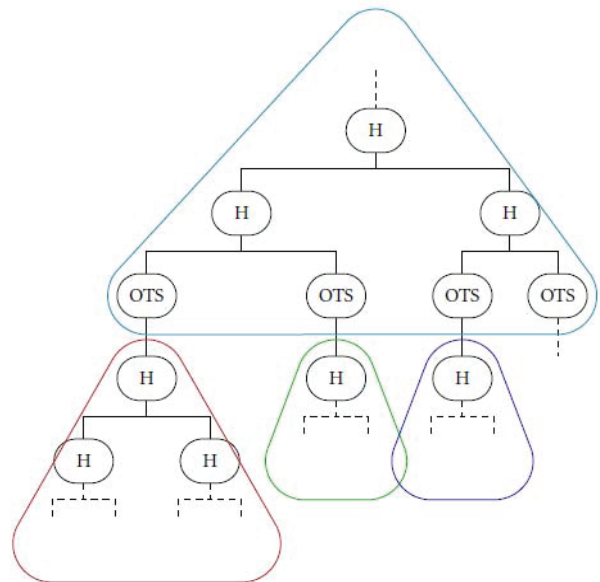


Рис. 3. Зв'язування суб-дерев разом

також доступне для підписання. Крім того, генерується безліч Q випадкових бітових масок, також у межах $\{0, 1\}^n$, які використовуватимуться в різних місцях. Ці маски використовуються у всіх геш - деревах, також як і в OTS, і FTS – на даний момент. Отже, $SK = (SK_1, SK_2; Q)$.

Для того, щоб генерувати відкритий ключ, необхідно сформувати принаймні одне дерево – τ_i -дерево у верхній частині конструкції. Це вимагає генерації OTS-ключів вздовж нижнього шару цього дерева. Важливо, що ці ключі мають бути генеровані детермінованим чином; використовуючи адрес і SK_1 як вхідні дані в деякій односторонній функції можна отримати випадкові значення (seed) для цього ключа. Потім бінарне геш - дерево може бути побудовано на відкритих ключах ключової пари OTS, а також кореневий вузол цього дерева є частиною відкритого ключа SPHINCS (PK_1). Оскільки бітові маски необхідні для перевірки, то вони також мають бути включені з відкритим ключем: $PK = (PK_1; Q)$.

Слід відзначити, що, в той час як SK_1, SK_2 і PK_1 мають розмір n бітів, Q – значно більше. Бітові маски, таким чином, складають найбільшу частину ключів. В SPHINCS-256 конфігурації, наприклад, PK_1 містить тільки 32 байти, в той час, як Q відповідає за $2 * 32 * 16 = 1024$ байт. У загальному випадку, кількістю бітових масок визначається частина схеми, яка вимагає найбільшого числа FTS, OTS або геш - дерев.

2.3.2 Електронний підпис

Як правило, в схемах ЕП з відкритим ключем спочатку обчислюється геш-значення повідомлення, яке має бути підписане, а потім підписують отримане геш - значення. Це гарантує, що вхідна константа буде досить невеликої довжини. У схемі без збереження стану (для схеми Голдрейх), потім обирається випадкова пара ключів в нижньому шарі дерева, щоб підписати геш-значення. Але у схемі SPHINCS, однак,

пара ключів вибирається на основі самого геш-значення повідомлення. Для того, щоб запобігти атакам на конкретні ключові пари, має бути включений деякий випадковий або невідомий фактор – це ключ SK_2 . Спочатку потрібно обчислити значення R за допомогою псевдовипадкової функції, яка приймає SK_2 і повідомлення як вхідні дані, а потім використовує частину цього повідомлення, випадкової величини R , щоб вибрати пару ключів FTS. Крім того, інша частина R використовується для обчислення випадкового геш-значення D повідомлення. Цей дайджест належить підписуванню. Як практичний результат всього цього є вибір пари ключів FTS, що є повністю детермінованим щодо секретного ключа SK_2 і повідомлення M .

Після вибору конкретної пари ключів FTS, вона має бути сформована на основі випадкових значень, отриманих від його місця розташування і SK_1 , а потім використовуватися для підпису D для отримання підпису σ_{FTS} . Разом з повідомленням, отриманим вище і індексом idx обраної пари ключів, цей підпис формує першу частину загального підпису SPHINCS. Позначимо цей ЕП SPHINCS як Σ . Потім генерується пара ключів OTS для батьківського вузла $v_{d,h/d}$ у відповідному суб-дереві у τ_d (знову використовуючи його позицію і SK_1), вона і використовується, щоб підписати відкритий ключ FTS. Позначимо вироблений підпис, як $\sigma_{OTS,d}$. Цей підпис також додається до Σ . Відкритий ключ цього OTS має пройти перевірку справжності, тому обчислюються всі вузли вздовж його шляху автентифікації по всьому дереву у τ_i і включаються в Σ . Позначимо вузли шляху автентифікації в обраному дереві на шарі d як $Auth_d$. При досягненні кореня дерева генерується ключова пара OTS, яка належить до батьківського вузла у $v_{d-1,h/d}$ і використовується для підписання. Ця процедура триває весь шлях до кореня одного дерева в τ_i , який включений в РК. На шляху всі підписи OTS і вузли поряд зі шляхом автентифікації мають бути додані до Σ .

У цілому підпис SPHINCS Σ тепер містить випадкове R , індекс обраної пари ключів FTS, підпис σ_{FTS} та d пар підписів OTS і вузлів вздовж шляху автентифікації $(\sigma_{OTS,i}, Auth_i)$. Все разом: $\Sigma = (idx, R, \sigma_{FTS}, (\sigma_{OTS,1}, Auth_1), \dots, (\sigma_{OTS,d}, Auth_d))$.

2.3.3 Перевірка підпису

Процедура для перевірки ЕП повідомлення M дуже схожа на сам ЕП. Після обчислення D (з використанням R і M), підпис FTS є верифікованим. Як було згадано вище, функції перевірки OTS і FTS, що використовуються в SPHINCS, виводить відкритий ключ. Підписи OTS на цьому відкритому ключі тепер можуть бути верифіковані, в результаті чого отримується відповідний відкритий ключа OTS. Оскільки шлях автентифікації також наводиться в Σ , то тепер може бути обчислений кореневий вузол дерева τ_D . Подібно до того, як створювався підпис, далі продовжуємо вгору по дереву вздовж шляхів автентифікації з перевіркою підпису на кореневих вузлах кожного суб-дерева. З рештою, верифікація переходить до кореневого вузла одного дерева в τ_i . Цей кореневий вузол має бути рівний PK_1 , включеного в РК. Якщо це так, то ЕП дійсний.

2.4 Схема підпису XMSS

Розширена схема підпису Merkle (XMSS - The eXtended Merkle Signature Scheme) була введена в 2011 році і стала проектом в 2015 році [12].

Основна конструкція (Рисунок 4) [12] виглядає як дерево Merkle, за виключенням деяких речей. Дерево XMSS має маску для операції XOR дочірніх вузлів перед тим, як вони гешуються у вузлі їх «батьків». Це інша маска для кожного вузла.

Друга особливість полягає в тому, що лист дерева XMSS не є гешем одноразового відкритого ключа підпису. Корінь іншого дерева називається L-деревом. L-дерево має ту саму ідею масок, застосованих до його вузлів геш-значень, на відмінну від основних

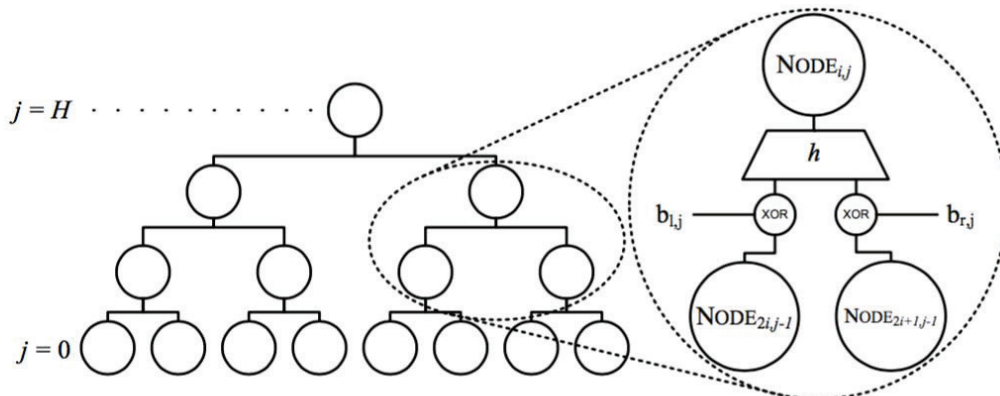


Рис. 4. Конструкція дерева XMSS

XMSS дерев, але спільну для всіх L-дерев. Всередині листя будь-якого L-дерева зберігаються елементи WOTS + відкритого ключа. Дерево не використовується для зберігання відкритого ключа WOTS, але використовується для гешування його таким чином, що можна довести, що геш-функція другого прообразу досить стійка (замість стійкості до колізії). Крім того, основний відкритий ключ складається з кореневого вузла дерева XMSS, а також бітових масок, що використовуються в дереві XMSS і L-дереві.

XMSS є більш актуальною схемою, що знаходиться в процесі стандартизації. Вона заснована на Деревях Merkle з такими поліпшеннями:

- більша ефективність при обході дерева, тобто обчислення шляху вузлів, пов'язаних з заданою одноразовою схемою підпису до кореня дерева і загального відкритого ключа.

- зменшені розміри приватного ключа та краща секретність за рахунок використання генератора псевдовипадкових чисел для створення ключів одноразового підпису.

3. БЕЗПЕКА СХЕМ ПІДПISУ НА ОСНОВІ ГЕШ-ФУНКЦІЙ

Традиційно, безпеку HB схем підпису було пов'язано зі стійкістю до колізій використовуваної геш-функції. В останні роки кілька робіт зосереджені на основі безпеки на більш м'яких припущеннях, таких як стійкість другого прообразу. Є дві основні причини слідування цієї тенденції. З одного боку, атаки проти стійких до колізій SHA1 і MD5 мотивували дослідників розробляти схеми підпису, стійкі до колізій [9]. З іншого боку, стійкість до колізій породжує атаки в той час, як стійкості (другого) прообразу немає. Отже, щоб досягти рівня безпеки λ бітів, необхідна геш-функція з $n = 2\lambda$ біт геш-значень якщо потрібна стійкість до колізії, хоча стійкість прообразу (або другого прообразу) лише для $n = \lambda$ дайджестів. При чому скорочення вихідного розміру геш-функції, що використовується, одразу вдвічі скорочує підписи і розміри ключів схем ЕП, що засновані на геш-функціях.

Дане судження є лише половиною правди, бо воно тримається на неявному припущенні, що геш-функція використовується тільки один раз. Очевидно, що для багатьох криптографічних конструкцій це не так. Розглянемо це на прикладі стійкості прообразу (one-wayness). Для багатьох криптографічних конструкцій при противник зможе взнати величину значень функції, і відбудеться компрометація, як тільки він знайде прообраз для одного з них. Більш конкретно, припустимо, що геш-функція з n бітовими виходами використовується d раз у криптографічній конструкції. Якщо успішно інвертувати геш-функцію на будь-якому з виходів, то це негативно вплине на безпечність схеми. В цьому випадку складність атаки знижується до $O(2^n/d)$ замість $O(2^n)$. Інтуїтивно зро-

зуміло, що це відбувається тому, що кожне вхідне значення, яке противник намагається дістати, має ймовірність бути правильним $d/2^n$ замість $1/2^n$. Це справедливо, якщо геш-функцію розглядати як випадкову функцію.

4. УДОСКОНАЛЕНІ СХЕМИ ПІДПISУ

4.1 XMSS-T

У 2016 році запропоновано схему підпису XMSS-T [9], яка не є уразливою для багатьох цільових атак. З цією метою були запропоновані нові поняття (концепції) стійкості для прообразу, другого прообразу і стійкості до колізій. Була проаналізована загальна безпека геш-функцій в зв'язку з цими новими властивостями проти класичних і квантових атак. Це дозволило отримати верхні і нижні оцінки складності загальних атак.

Більш конкретно, перший тип понять моделює концепцію (з однією функцією, багатоцільовою), яка неявно використовується в недавніх схемах підпису, стійких до колізій, таких, як XMSS, XMSSMT і SPHINCS [8,9,11,12]. У цій концепції, противник A отримує цільові значення r і випадкову функцію з сімейства геш-функцій. Потім A намагається знайти прообраз (або другий прообраз, відповідно) для одного з цільових значень в рамках даної геш-функції. Доведено, що порівняно зі стандартною стійкістю (до другого прообразу), складність запити загальних атак знижується на коефіцієнт r для класичних і на \sqrt{r} для квантових. Потім була введена інша концепція з багатьма функціями (багатоцільова) зі стійкістю до прообразу і стійкості другого прообразу. Для такої концепції, A дає кілька пар функцій і цільові значення, отримані незалежно один від одного в випадковому порядку. Тепер мета A - знайти прообраз (або другий прообраз, відповідно) для одного з цільових значень за відповідною функцією. Доведено, що в цьому випадку складність запити загальних атак така ж, як і для стандартних (з однією функцією, тобто одноцільовою) понять. З огляду на те, що багатофункціональні і багатоцільові поняття настільки ж важкі, як стандартні уявлення про стійкість прообразу і другого прообразу, побудована нова схема підпису з безпекою на основі цих нових концепцій. Оскільки основна конструкція наслідуює це з XMSS, то нову схему позначають XMSS-T, що вказує на XMSS з посиленням заходів безпеки. У той час як XMSS втрачає багато в бітвій безпеці за кількома параметрами, включаючи загальну висоту дерева, XMSS-T втрачає тільки два біти, незалежно від будь-яких параметрів. Відмінністю між XMSSMT і XMSS-T є різні геш-дерева і така одноразова схема підпису, що безпека може бути заснована на багатофункціональних багатоцільових властивостях. Основна зміна полягає в тому, що для кожного виклику геш-функції в геш-дереві або геш-ланцюзі використовуються різні ключі геш-функцій і різні бітові маски. XMSS-T-схема зі збереженням

стану, тому вона може бути не придатна в деяких практичних випадках використання. Але позитивним є те, що можна легко зробити аналогічні зміни в схемах підпису SPHINCS без збереження стану. Грубо кажучи, це зводиться до заміни геш – дерев, що використовуються, а також одноразових підписів.

4.2 SPHINCS-256

Параметри схеми SPHINCS-256 [8,11] обрані з двома цілями: довгострокова безпека 2^{128} від зломисників, що мають доступ до квантових комп'ютерів, та достатній компроміс між швидкістю і розміром підпису. Перша мета визначається параметром безпеки $n=256$, який в свою чергу визначив назву SPHINCS-256. Оптимізуючи інші параметри, потрібно прийняти рішення про відносну важливість швидкості і розміру підпису. Після пошуку у великому просторі параметрів були вибрані такі [12]: $m = 512, h = 60, d = 12, w = 16, t = 216$;

$$k = 32, l = 67, x = 6, a = 64.$$

Для ряду додатків такий вибір досить простий і досить швидкий. Звичайно, можна також визначити різні реалізації SPHINCS, змінюючи інші параметри на користь тієї чи іншої вимоги, наприклад складності чи розміру підпису.

В SPHINCS-256 як параметри використовуються такі значення:

$$n = 256; m = 512; h = 60; d = 12; w = 16; t = 2^{16}; k=32.$$

Тому, беручи до уваги порушників, які мають доступ до потужних квантових комп'ютерів, можна зробити такий висновок. Якщо припустити, що кращі атаки проти геш-функції, що використовується, є загальними атаками, то $H_{k,t}$ забезпечує безпеку вище 2^{128} щодо стійких підмножин, а F і H забезпечують 2^{128} захист від атаки до знаходження прообразу, другого прообразу, а також атак невиявлення F . Аналогічним чином, значення PRFs і PRGS забезпечують безпеку 2^{128} . Таким чином, SPHINCS-256 при вказаних припущеннях забезпечує захист від постквантових порушників 2^{128} .

ВИСНОВКИ

НВ схеми підпису вважаються найбільш перспективною постквантовою альтернативою існуючим схемам, таким як RSA і ECDSA, уразливим для квантових атак. Це особливо актуально, тому що безпека криптографічних геш - функцій добре вивчена. Крім того, є точні докази, що встановлюють зв'язок між складністю порушення схеми та складністю порушення властивостей безпеки геш-функцій, які використовуються в схемах. Це дозволяє точну оцінку безпеки конкретних наборів параметрів.

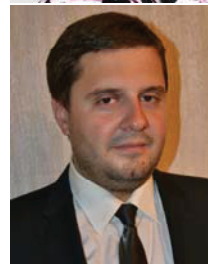
Важливим є те, що на сьогодні в Україні розроблені та стандартизовані по суті постквантові алгоритми гешування – геш-функції ДСТУ 7564:2014. «Інформаційні технології. Криптографічний захист інформації. Функція гешування».

Література

- [1] Горбенко І.Д., Горбенко Ю.І. «Прикладна криптологія». Теорія. Практика. Застосування. Монографія. Видання 2-ге, перероблене й виправлене. Харків. Видавництво «ФОРТ». 2012. – 878с.
- [2] Горбенко Ю.І. Методи побудовання та аналізу, стандартизація та застосування криптографічних систем: Монографія / За загальною редакцією Професора Горбенко Івана Дмитровича. – Харків: Форт, 2015. – 959 с.
- [3] Горбенко Ю. І. Аналіз можливостей квантових комп'ютерів та квантових обчислень для криптоаналізу сучасних криптосистем / Ю. І Горбенко, Р. С. Ганзя. // Східно-європейський журнал передових технологій. – 2014, № 1/9 (67). – С. 8–15.
- [4] A RIDDLE WRAPPED IN AN ENIGMA. NEAL KOBLITZ AND ALFRED J. MENEZES Department of Mathematics, Box 354350, University of Washington, Seattle, WA 98195 U.S.A
- [5] Lili Chen, Stephen Jordan, Yi-Kai-Liu, Dustin Moody, Rene Peralta, Ray Perlner, Daniel Smith-Tone. Report on Post – Quantum Cryptography. NISTIR 8105 (DRAFT). <https://www.google.com.ua/search?>
- [6] T. Leighton and S. Micali, New approaches to secret-key exchange, Presented at Crypto'93.
- [7] T. Leighton and S. Micali, Secret-key agreement without public-key cryptography (extended abstract), in: Advances in Cryptology - CRYPTO'93, Lecture Notes in Computer Science 773 (Springer, Berlin, 1994). 456-479.
- [8] J.Rijneveld. Implementing SPHINCS with restricted memory.2015
- [9] A.Hulsing, J.Rijneveld, F.Song. Mitigating multi-target attacks in hash-based signatures.
- [10] G.Becker. Merkle signature schemes, Merkle Trees and their cryptanalysis.2008.
- [11] D.Bernstein, D.Hopwood, A.Hulsing, T.Lange, R.Niederhagen, L.Papachristodoulou, M.Shneider, P.Schwabe, Z.Wilcox-O'Hearn.SPHINCS: practical stateless hash-based signatures.2015.
- [12] J.Buchmann, E.Dahmen, A.Hulsing. XMSS – A practical forward secure signature scheme based on minimal security assumptions.2011



Ковальова Наталія Вікторівна, студентка кафедри БІКС ХНУ ім.В.Н.Каразіна. Область наукових інтересів: дослідження міжнародних стандартів, аналіз стійкості цифрових підписів, постквантова криптографія.



Горбенко Юрій Іванович, канд. техн. наук, технічний директор проектів АТ «ІТ», лауреат державної премії. Наукові інтереси: проектування та застосування криптографічних комплексів, систем та засобів, інфраструктур відкритого ключа.

УДК 003.026:004.056

Анализ постквантовых механизмов цифровой подписи на основе хеш-функций / Н.В.Ковалёва, Ю.И. Горбенко // Прикладная радиоэлектроника: науч.-техн. журнал - 2016. – Том 15. №3. – С. 195 – 202.

Рассматривается суть криптопреобразований, которые основываются на использовании хеш-функций при построении постквантовых электронных подписей. Приведены результаты анализа криптографической стойкости указанных электронных подписей и оценки существующих алгоритмов, а также рекомендации по их применению.

Ключевые слова: алгоритмы постквантовой электронной цифровой подписи, функции хеширования, криптографическая стойкость, применение в постквантовый период.

Ил.: 04. Библиогр.:12 назв

UDC 003.026:004.056

Analysis of postquantum digital signature schemes based on hash functions / N.V.Kovaleva, Yu.I. Gorbenko // Applied Radio Electronics: Sci. Journ. – 2016. Vol. 15. №3. – P. 195 – 202.

The paper considers the essence of cryptotransformations based on hash functions for building postquantum digital signature schemes. The results of analyzing the cryptographic strength of these electronic signatures and evaluation of existing algorithms and recommendations for their use are given.

Keywords: postquantum digital signature algorithms, hash-functions, cryptographic strength, use in postquantum period.

Fig.: 04. Ref.: 12 items.

ШВИДКІ АЛГОРИТМИ ДЛЯ ОБЧИСЛЕННЯ ІЗОГЕНІЙ НА ЕЛІПТИЧНИХ КРИВИХ

В. А. ПОНОМАР, О. Г. БЕРЕЖНИЙ

Розглядаються та аналізуються алгоритми обчислення ізогеній еліптичних кривих над скінченним полем. Аналізується алгоритм, обчислення ізогенії ступеня L , що ґрунтується на швидких алгоритмах розкладання β -функції і пов'язані з ними функції в ряд Вейерштрасса. Наводяться рекомендації та пропозиції відносно оцінки складності алгоритмів обчислення ізогеній ступеню L .

Ключові слова: алгоритми обчислення ізогеній, ізогенії еліптичних кривих, електронні підписи, постквантовий період, швидкі алгоритми обчислення ізогеній еліптичних кривих.

ВСТУП

У 2014 – 2016 роках отримані суттєві результати в побудованні квантового комп'ютера [1,4]. Ще раніше розроблені та практичні готові до використання методи квантового криптоаналізу. Їх реалізація на квантовому комп'ютері дозволить успішно атакувати більшість асиметричних криптосистем. Підтвердженням цьому є спочатку поява і Internet статі «A RIDDLE WRAPPED IN AN ENIGMA [1], в якій зазначається, що в серпні 2015 року агентство національної безпеки (АНБ) уряду США виступило з заявою про слабкість існуючих асиметричних криптосистем відносно квантового криптоаналізу. В 2016 році опубліковано звіт «Report on Post – Quantum Cryptography. NISTIR 8105 (DRAFT) [2]», в якому повністю підтверджено можливості успішного квантового криптоаналізу асиметричних криптосистем, а також визначені основні проблеми та можливості і етапи їх вирішення.

Серед можливих методів та алгоритмів побудови асиметричних криптосистем називається метод, що ґрунтується на використанні криптографічних перетворень на ізогеніях еліптичних кривих. В [3,4] наведено ряд даних, що підтверджують перспективність криптографічних перетворень з використанням математичного апарату ізогеній еліптичних кривих. При цьому, в першу чергу ставляться задачі побудовання криптографічних механізмів електронного підпису(ЕП). Вирішення вказаної, на наш погляд суттєво проблемної задачі, пов'язане з доведенням криптографічної стійкості, забезпеченням необхідної швидкодії (складності) перетворень та обґрунтуванням і побудованням загальних параметрів та ключів. Зважаючи на стан досліджень в указаному напрямку, на наш погляд, актуальними є задачі аналізу та оптимізації криптографічних перетворень на ізогеніях еліптичних кривих. Тому метою цієї статі є обґрунтування та формулювання основних задач аналізу складності побудовання параметрів і ключів, а також прямих та зворотних асиметричних криптоперетворень. Зрозуміло, що суттєво актуальною є проблема доведення криптографічної стійкості, на наш погляд вона

розв'язуватиметься на світовому рівні ще декілька років.

1. Сутність криптографічних перетворень на ізогеніях еліптичних кривих

Спочатку розглянемо основні складові механізму та алгоритмів криптографічних перетворень на ізогеніях еліптичних кривих [3 – 5].

Ізогенія – це раціональне відображення $\varphi : E_1(K) \rightarrow E_2(K)$, де $E_1(K)$ та $E_2(K)$ є еліптичними кривими, а $\varphi(P_\infty) = P_\infty$. Нульова ізогенія – це ізогенія, що відображає усі точки однієї кривої, у точку на нескінченності іншої. Ядром ізогенії є

$$\text{Ker}(\varphi) = \{K_i \in E_1\}; \varphi(K_i) = P_\infty.$$

У цілому ізогенії ініціюють відображення полів функцій на кривих. Степінь розширення $(K(E_1) : \varphi^*K(E_2))$ називається степінню ізогенії.

Відносно певної ізогенії $\varphi : E_1(K) \rightarrow E_2(K)$ існує дуальна ізогенія $\hat{\varphi} : E_2(K) \rightarrow E_1(K)$, така, що $\hat{\varphi} \circ \varphi = [1]$, де l – множення точки кривої E_1 на число l , аналогічно $\varphi \circ \hat{\varphi} = [1]$, де l – множення точки кривої E_2 на число l . При цьому дуальні ізогенії мають однакову степінь.

Для випадку алгебраїчно замкненого поля операція множення точки на число l задає ендоморфізм еліптичної кривої з ядром l^2 точок. Оскільки ізогенія відповідає квадратному кореню з операції множення на l , то ядро ізогенії складається з l точок порядку l , що створюють циклічну групу (однією з них є точка P_∞).

Ізогенії складних степенів можуть використовуватися, як композиція ізогеній простих степеней.

Властивості ізогеній, що використовуються в ході створення криптосистем:

- 1) $\gamma(\varphi(A)) = \varphi(\gamma(A)) = \gamma\varphi(A)$;
- 2) $k^*A_\varphi = \varphi(k^*A)$.

Знаходження ізогеній по ядру. Для знаходження ізогенії $\varphi: E_1(K) \rightarrow E_2(K)$, з заданим ядром використовується формула Велу [6,7,14-15].

Для еліптичної кривої, що задана формулою Вейерштраса:

$$E_1: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6,$$

узагальнений алгоритм знаходження ізогенії по ядру наведено в [9]. З урахуванням того, що в криптографії еліптичних кривих прийнято використовувати спрощену формулу Вейерштраса, то розглянемо алгоритм знаходження ізогенії по ядру й зведемо до прийнятого вигляду, використовуючи формулу

$$E_1: y^2 = x^3 + a_4x + a_6.$$

Нехай S – група точок еліптичної кривої, що буде ядром ізогенії. Тоді:

1. Формуємо набір точок S :

- а) Виключаємо з S точку на нескінченності.
- б) Нехай C_2 – усі точки C , в яких координата у дорівнює нулю, а R – усі інші точки C .
- в) Розділимо точки набору R на R_+ та R_- , але так, що для кожної точки P , що належить R_+ , $-P$ належить R_- .
- г) $S = R_+ \cup C_2$.

2. Для кожної точки $Q \in S$ виконуємо такі обчислення:

- а) $g_Q^x = 3x_Q^2 + a_4$;
- б) $g_Q^y = -2y_Q$;
- в) $v_Q = \begin{cases} g_Q^x, & 2Q = \infty \\ 2g_Q^x, & 2Q \neq \infty \end{cases}$;
- г) $u_Q = (g_Q^y)^2$;
- д) $v = \sum_{Q \in S} v_Q$;

$$д) w = \sum_{Q \in S} (u_Q + x_Q v_Q).$$

3. Розраховуємо формулу еліптичної кривої $E_2(K)$:

- а) $A_4 = a_4 - 5v$;
- б) $A_6 = a_6 - 7w$;
- в) $E_2: y^2 = x^3 + A_4x + A_6$.

4. Розраховуємо координати точки

$$(x_\varphi, y_\varphi) = \varphi(x, y):$$

$$а) x_\varphi = x + \sum_{Q \in S} \left(\frac{v_Q}{x - x_Q} + \frac{u_Q}{(x - x_Q)^2} \right);$$

$$y_\varphi = y - \sum_{Q \in S} \left(\frac{v_Q y}{x - x_Q} + \frac{u_Q y}{(x - x_Q)^2} - \frac{g_Q^x g_Q^y}{(x - x_Q)^2} \right)$$

$$б) \left(u_Q \frac{2y}{(x - x_Q)^3} + v_Q \frac{y - y_Q}{(x - x_Q)^2} - \frac{g_Q^x g_Q^y}{(x - x_Q)^2} \right)$$

2.Вимоги до загальних параметрів та можливості і методи їх побудування

Важливим будівельним блоком у роботі Elkies є алгоритм, який обчислює криві, які є ізогеніями на заданій кривій E . Цей блок використовує модульні поліноми, щоб отримати список ізогеній кривих і формул Velu, щоб отримати явний вигляд ізогеній

$I: E \rightarrow \tilde{E}$, де \tilde{E} знаходиться у відповідній формі Вейерштраса. У цій роботі ми концентруємося на алгоритмах, які будують ступінь L ізогенії I від E і

\tilde{E} . Ми могли б обмежити далі до випадку, коли L непарне просте, оскільки ізогенії можна записати у вигляді композицій ізогеній простого ступеня. До того ж, непарний простий випадок є найбільш важливим в SEA. Проте, наші результати стоять для довільного L . Ми вимагаємо, щоб характеристика p основного поля K було 0 або $p \gg L$. Це обмеження задовольняється в разі зацікавленості в застосуванні до алгоритму SEA, оскільки в іншому випадку p -адичні методи набагато швидше і простіше у використанні.

З нашого припущення про p випливає, що рівняння наших кривих можна записати у формі Вейерштраса

$$y^2 = x^3 + A*x + B. \quad (1)$$

У нульовій характеристиці, крива (1) може бути параметризована $(x, y) = \left(\rho(z), \frac{\rho'(z)}{2} \right)$ зважаючи на те, що класичне диференціальне рівняння

$$\rho'(z)^2 = 4(\rho(z)^3 + A*\rho(z) + B) \quad (2)$$

відповідає ρ -функції Вейерштрасса. Це є основою для нашого обчислення ізогеній. Таким чином, ми доведемо два результати, спочатку на обчисленні Вейерштрасса ρ -функції, а потім на обчисленні самих ізогеній.

Тут особливість полягає у використанні при розрахунках класичних швидких алгоритмів для степеневих рядів і демонструється, як їх можна застосувати до обчислення ізогеній. Позначимо $M : N \rightarrow N$ функцію таку, що многочлени степеня менше n можуть бути мультиплікативні в операціях, де $M(n)$ основне поле. За допомогою швидкого перетворення Фур'є [8, 16], можна знайти

$$M(n) \in O(n \log n \log \log n)$$

над полями, що містять примітивні об'єднанні корені, причому $M(n) \in O(n \log n)$.

3.Вимоги до алгоритмів генерування асиметричних пар ключів та оцінка їх властивостей.

Квадратичний алгоритм Elkies. Далі розглядаватимемо дві криві E і \tilde{E} , через які вейерштрассове рівняння допускати нормалізування ізогенія $I: E \rightarrow \tilde{E}$ степеня 1. Подамо ці криві у вигляді

$$E: y^2 = x^3 + Ax + B \quad \tilde{E}: y^2 = x^3 + \tilde{A}x + \tilde{B}. \quad (3)$$

Визначимо залежно від вхідних даних ізогенії I , які подаватимемо у вигляді

$$I(x, y) = \left(\frac{N(x)}{D(x)}, y \left(\frac{N(x)}{D(x)} \right)' \right). \quad (4)$$

Спочатку розглянемо алгоритм Elkies [11], чия складність квадратична за ступенем 1. Далі з метою порівняння розглянемо два швидких варіанти алгоритму Elkies1998, так звані fastElkies і fastElkies', складність яких відповідно $O(M(1))$ і $O(M(1) \log 1)$.

Як показав аналіз алгоритм Elkies1998 був введений для простого випадку ступеня в роботі [11], але він може застосовуватися для будь-яких великих 1. Перша частина алгоритму спрямована на обчислювальні розкладання $\frac{N(x)}{D(x)}$ на нескінченності; друга частина становить відновлення сум коренів $D(x)$ з цього розкладу.

Для того, щоб провести аналіз, розглянемо раціональні функції $N(x) / D(x)$, які задовольняють нелінійне диференціальне рівняння

$$\begin{aligned} (x^3 + Ax + B) + \left(\frac{N(x)}{D(x)} \right)^2 = \\ \left(\frac{N(x)}{D(x)} \right)^3 + \tilde{A} \left(\frac{N(x)}{D(x)} \right) + \tilde{B}. \end{aligned} \quad (5)$$

Це випливає з того, що I відображає E в \tilde{E} . Диференціальне рівняння (5) перебудовується до наступного рівняння другого порядку

$$\begin{aligned} (3x^3 + A) \left(\frac{N(x)}{D(x)} \right)' + 2(x^3 + Ax + B) \\ \left(\frac{N(x)}{D(x)} \right)' = 3 \left(\frac{N(x)}{D(x)} \right)^2 + \tilde{A}. \end{aligned} \quad (6)$$

Записуючи розкладання раціональної функції $N(x) / D(x)$ на нескінченності

$$\frac{N(x)}{D(x)} = x + \sum_{i \geq 1} \left(\frac{h_i}{x^i} \right)$$

і визначення коефіцієнтів x^{-i} з обох сторін рівняння

(6) дає можливість подати

$$\begin{aligned} h_k = \frac{3}{(k-2)(2k+3)} \sum_{i=1}^{k-2} h_i h_{k-1-i} - \\ \frac{2k-3}{2k+3} A h_{k-2} - \frac{2(k-3)}{2k+3} B h_{k-3}, \end{aligned} \quad (7)$$

для всіх $k \geq 3$, з початковими умовами

$$h_1 = \frac{A - \tilde{A}}{5} \quad \text{і} \quad h_2 = \frac{B - \tilde{B}}{7}.$$

Подання (7) є основою алгоритму Elkies1998; використовуючи його, можна обчислити h_3, \dots, h_{1-2}

з використанням $O\left(1^2\right)$ операцій в K .

Також в алгоритмі 'Elkie 1998 передбачається, що $p_1 = \sigma$. Подання коефіцієнтів в рівнянні дає, що

$$h_i = (2i+1)p_{i+1} + (2i-1)Ap_{i-1} + (2i+2)Bp_{i-2}, \text{ для всіх } i \geq 1. \quad (8)$$

Оскільки h_1, \dots, h_{1-2} відомі, p_2, \dots, p_{1-1} можна вивести, використовуючи $O(1)$ операцій. Потім отримати поліном $D(x)$, або шляхом квадратичного алгоритму або з використанням швидкого алгоритму [2], а $N(x)$ можна отримати за допомогою формули (4), в $O(M(1))$ операцій.

У такому алгоритмі необхідно, щоб в K було одиниць $2, \dots, 2l-1$. Його складність $O(l^2)$, причому вузьким місцем є обчислення коефіцієнтів h_1, \dots, h_{1-2} . Для цього можна застосовувати паралельні обчислення згідно з [7], де диференціальні рівняння Вейерштрасса дають (7), якщо взяти $A=B=0$ в останньому).

4. Швидкі алгоритми.

Важливим є те, що необхідно покращити в алгоритмі Elkies1998 обчислення коефіцієнтів h_i , а решта є незмінною. Але, як показав аналіз, розкладання $N(X) / D(x)$ на нескінченності можна зробити використовуючи диференціальне рівняння (5), за умови, що рівняння, отримано заміною змінних $x \rightarrow 1/x$. Для того, щоб уникнути ускладнення, ми переважно можна застосовувати ступеневі ряди вигляду

$$S(x) = x + \frac{\tilde{A}-A}{10}x^5 + \frac{B-\tilde{B}}{14}x^7 + O(x^9) \in x + x^3K[[x^2]]$$

за умови, що

$$\frac{N(x)}{D(x)} = \frac{1}{S\left(\frac{1}{\sqrt{x}}\right)^2}$$

Також необхідно враховувати, що для S справедливе співвідношення $\tilde{R}=SoR$, де

$$R(z) = 1/\sqrt{p(z)} \text{ і } \tilde{R} = 1/\sqrt{\tilde{p}(z)}.$$

У подальшому, застосовуючи правило ланцюга, можна отримати диференціальне рівняння першого порядку, яке якому задовольняє $S(x)$, тобто

$$\left(Bx^6 + Ax^4 + 1\right)S'(x)^2 = 1 + \tilde{A}S(x)^4 + \tilde{B}S(x)^6. \quad (9)$$

За допомогою (9) для обчислення $N(X) / D(X)$ можна застосувати два алгоритми, така можливість залежить від того, чи відомо коефіцієнт σ чи ні. Для цих алгоритмів маємо

$$S(x) = xT(x^2) \text{ і } U(x) = \frac{1}{T(x)^2} \in 1 + x^2K[[x]] \text{ так}$$

$$\text{що } \frac{N(x)}{D(x)} = xU\left(\frac{1}{x}\right).$$

У першому алгоритмі, що називається як fastElkies, передбачається, що σ відоме, тому

1) Обчислюється

$$C(x) = (Bx^6 + Ax^4 + 1)^{-1} \text{ mod } x^{2l-1} \in K[[x]];$$

2) Обчислюється $S(x) \text{ mod } x^{2l}$ з використанням того, що $G(x, t) = C(x)\left(1 + \tilde{A}t^4 + \tilde{B}t^6\right)$, і далі знаходиться $T(x) \text{ mod } x^{1-1}$;

3) Обчислюється $U(x) = 1/T(x)^2 \text{ mod } x^{1-1}$;

4) Знаходяться коефіцієнти h_1, \dots, h_{1-2} $N(x) / D(x)$, використовуючи $N(x) / D(x) = xU(1/x)$;

5) Робиться підрахунок сум p_2, \dots, p_{1-1} з $D(x)$, з використанням лінійного повторення(8);

6) Відновлюється $D(x)$ з використанням сум, як описано в п.2.

7) Обчислюється $N(x)$ з використанням рівняння також (4).

Етапи 1) і 5) мають складність порядку $O(1)$. Етапи 2), 3), 6) і 7) можуть бути виконані за $O(M(1))$ операцій, а етап (4) не вимагає складних операцій.

У другому алгоритмі, який називають як fastElkies', не ставиться вимога знання σ . Його кроки

1') – 3') є лише результат невеликої зміни кроків (1) – (3). Їх складність має з точністю до констант порядок $O(M(1))$.

Алгоритм вимагає виконання таких етапів:

- 1) Обчислюється
- 2)

$$C(x) = (Bx^6 + Ax^4 + 1)^{-1} \bmod x^{8l-5} \diamond K[[x]];$$

обчислюється $S(x) \bmod x^{8l-4}$ з використанням алгоритму розділу 4, де

$$G(x, t) = C(x) \left(1 + \tilde{A}t^4 + \tilde{B}t^6 \right), \quad i \text{ залишається}$$

вивести $T(x) \bmod x^{4l-2}$;

- 3) Обчислюється

$U(x) = 1/T(x)^2 \bmod x^{4l-2}$ з використанням алгоритму §2.1;

- 4) Реконструюється раціональна функція $U(x)$;

- 5) Знаходиться $N(x)/D(x) = xU(1/x)$.

Виконання швидкої раціональної реконструкції на кроці 4') може бути виконана зі складністю $O(M(1)\log l)$ операції в K . Також можна перевірити, що алгоритм *fastElkies* вимагає що

$2, \dots, 2l-1$ будуть одиниці в K , в той час як алго-

ритм *fastElkies'* вимагає, що $2, \dots, 8l-5$ будуть одиниці в K .

У разі непарного l , необхідно замість $D(x)$ обчислити $g(x)$.

Нехай q_1, q_2, \dots ступеневі суми $g(x)$ так, що

$$q_i = p_i / 2. \text{ Тоді коефіцієнти } h_i \text{ і степені суми } q_i$$

пов'язані співвідношенням

$$h_i = (4+2)q_{i+1} + (4i-2)Aq_{i-1} + (4i-4)Bq_{i-2}. \quad (10)$$

Для того, щоб обчислити $g(x)$ з використанням алгоритму *fastElkies*, достатньо обчислити $S(x) \bmod x^{l+1}$; тоді $T(x)$ і $U(x)$ обчислюються за модулем $x^{(l+1)/2}$. Аналогічним чином, в алгоритмі *fastElkies'* достатньо обчислити $S(x) \bmod x^{4l}$, і $T(x)$ і $U(x)$ за модулем x^{2l} .

5. Метод Старка.

Як показує аналіз, перший subcubic метод для знайдених N і D пов'язано зі Stark [21] і становить розширення \tilde{p} , як і раніше фракцію в (19). Фракція

N/D апроксимується через $\frac{p_n}{q_n}$ і алгоритм зупиня-

ється, коли ступінь $n-l-1$ дає D . Зокрема, це можна застосовувати для будь-якого ступеня ізогенії. Оскільки p і \tilde{p} в $\frac{1}{z^2} + K[[z^2]]$, то достатньо працювати з значеннями в $Z = z^2$.

В результаті маємо

- 1) $T = \tilde{p}(Z) + O(Z^l)$;

- 2) $n=l$;

- 3) $q_0 = 1$;

- 4) $q_1 = 0$;

Оскільки $\deg(q_n) < l-1$, обчислюємо

- a) $n=n+1$;

- b) $a_n = 0$;

- c) Поки $r \geq 1$ знаходимо

$$a_n = a_n + t_{-r} z^r;$$

$$T = T - t_{-r} p^r = t_{-s} z^s + \dots;$$

$r=s$

- d) $q_n = a_n q_{n-1} + q_{n-2}$;

- e) $T := 1/T$;

5) Отримуємо $D := q_n$.

Вказаний алгоритм називається як Stark1972, що має $O(1)$ складність і проходить через стадії 5); а

оцінка досягається в загальному випадку, при $r = 1$ на кожному кроці. Крок, який визначає складність, зводиться до обчислення зворотних значень на стадії (5) з точністю $2l-1-2\deg q_n - 2r$. загальна складність

цих операцій $O(lM(1))$. Множення на стадії (5.d)

може бути зроблено за час $O(lM(1))$, при чому ці

множення можна зробити швидше, якщо це необхідно). Оскільки найбільша ступінь многочленів a_n обмежена $l-1$, p на кроці (5.c) також

зводиться до складності $O(lM(1))$ і. На останок, знаючи, $D(X)$, чисельник $N(x)$ може бути відновлений

зі складністю $O(M(1))$ з використанням.

У разі, коли l непарне, то і як у алгоритмів

SEA обчислюється зі складністю $O(M(1))$ операцій, наприклад, шляхом обчислення $\exp(\log D / 2)$.

Таким чином, загальна складність алгоритму Stark1972 оцінюється як в $O(lM(1))$. Зауважимо, що порівняно з методами, наведеними нижче, алгоритм Stark1972 не вимагає знання σ . p^f на стадії (5.с) може бути амортизоване в контексті алгоритму SEA.

6. Метод Elkies1992.

Розглянемо метод як Elkies1992, що наведений в [10]. Покладемо, що l непарне так, що

$$D(x) = g(x)^2,$$

хоча незначні зміни призведуть до спільного розв'язання.

Диференціюючи двічі вираз (2), отримаємо

$$\frac{d^4 p(z)}{dz^4} = 120p^3 + 72Ap + 48B.$$

Для більш загального випадку можна застосовувати рівності вигляду

$$\frac{d^{2k} p(z)}{dz^{2k}} = \mu_{k,k+1} p^{k+1} + \dots + \mu_{k,0},$$

а для деяких констант $\mu_{k,j}$, які задовольняють рекурентне співвідношення, отримаємо

$$\begin{aligned} \mu^{k+1,j} &= (2j-2)(2j-1)\mu_{k,j-1} + \\ & (2j+1)(2j+2)A\mu_{k,j+1} + (2j+2)(2j+4)B\mu_{k,j+2}; \\ \mu_{k,k+1} &= (2k+1)!. \end{aligned} \quad [8]$$

Використовуючи це рекурентне співвідношення, коефіцієнти $\mu_{k,j}$, для $k \leq d-1$ і $j \leq k+1$, можуть бути обчисленими зі складністю $O(l^2)$ операції в K .

Elkies показав [9], як використовувати ці коефіцієнти для відновлення статечних сум q_2, \dots, q_d в g , за допомогою таких рівностей, отримуючи при $k \geq 1$:

$$\begin{aligned} (2k)!(c_k - c_k) &= \\ 2(\mu_{k,0}q_0 + \dots + \mu_{k,k+1}q_{k+1}). \end{aligned}$$

Використовуючи ці рівності, за умови, що $q_1 = \frac{\sigma}{2}$ і коефіцієнти c_k, \dots, c_k і $\mu_{k,j}$ відомі, можна відновити q_2, \dots, q_d шляхом вирішення системи, в

якої складність $O(l^2)$. Це дає можливість відновити g , використовуючи відповідний алгоритм.

Діагональна система q_2, \dots, q_d може бути зведена до квазілінійної відносно просторової складності. Для цього слід використовувати структуру з діагональної системи, з тим, щоб уникнути явного обчислення зі складністю $O(l^2)$ константи $\mu_{k,j}$.

7. Метод Аткина.

В роботі [2], Аткин запропонував формулу, що дозволяє обчислення $D(x)$ [16, 18] в разі, коли l непарне. Продовжимо це так, щоб охопити випадок довільної l , йогозначаення повертається $D(X)$. Використаємо

$$D(p(z)) = z^{2-2l} \exp(F(z)),$$

де

$$\begin{aligned} F(z) &= -\sigma z^2 + \\ & 2 \left(\sum_{k=1}^{\infty} (lc_k - c_k) (z^{2k+2}) \right) \quad (11) \\ & / ((2k+1)(2k+2)). \end{aligned}$$

Оскільки l і коефіцієнти c_k, c_k передбачаються відомими, можна обчислити $F(z) \bmod z^l$, за умови, що σ відоме. Для цього можна скористатись прямим методом визначення $D(X)$, а потім обчислити експоненти $F(z)$, а також відновити коефіцієнти $D(x)$, але по одному за кожен раз. Це викладено в алгоритмі, що названий як Atkin1992. Тоді, використовуючи серію значень в $Z = z^2$, маємо алгоритм.

1. Підраховуються серії $P_i(Z) = p(Z)^i$ порядку l , для $1 \leq i \leq l-1$;

2. Обчислюються $G(Z) = \exp_1(F(Z))$;

$$\begin{aligned} T &= G; \\ D &= 0; \end{aligned}$$

3. $D = D + tz^l$;

$$T = T - tP_1.$$

На кроці 1) складність оцінюється як $O(lM(1))$; на 2) можна знехтувати, використовуючи або класичну або швидку експоненцію. На кроці 5) складність оцінюється як $O(1)$ більше, операції, а в

загальному випадку як $O(I^2)$. Таким чином, загальна складність алгоритму оцінюється як $O(IM(1))$.

Якщо цей алгоритм використовується в контексті алгоритму SEA, то крок (1) може бути модифікований, оскільки вона залежить тільки від кривої E.

Таким чином, всі p мають обчислюватися для максимального значення I , які використовуватимуться і зберігатимуться. Тому складність цього алгоритму визначатиметься на кроці (5), тобто як $O(I^2)$.

Аналіз показує, що для обчислення $D(X)$, за умови уникнення обчислення всіх $p(Z)$, краще застосовувати рівняння (20), подане у вигляді

$$D\left(\frac{1}{x}\right) = I^{2-2l}((\exp \circ F) \circ I),$$

$$I(x) = p^{-1}\left(\frac{1}{x}\right), \quad (12)$$

де p^{-1} є функціонально зворотним p . Розширення з $I(x)$ близьке $\diamond(1)$, може бути обчислене в $O(1)$ операцій з використанням диференційного рівняння

$$I'(x)^2 = \frac{1}{4x(1+Ax^2+Bx^3)} \quad \text{або} \quad (13)$$

$$I'(x) = \frac{1}{2\sqrt{x}} \frac{1}{\sqrt{1+Ax^2+Bx^3}}.$$

Таким чином, маємо лінійне диференційне рівняння:

$$\frac{I'(x)}{I(x)} = -\frac{1+3Ax^2+4Bx^3}{2x(1+Ax^2+Bx^3)}.$$

З викладеного випливає, що лінійне диференціальне рівняння може бути поданим як

$$J(x) = x^{-2}I(x) = \sum_{i \geq 0} a_i x^i. \quad (14)$$

Розпакування коефіцієнтів в цьому рівнянні, дає лінійну можливість

$$A_{i+1} = -\frac{2i-1}{2(i+1)(2i+3)}((2i-3)Ba_{i-2} + 2Aia_{i-1})$$

для $i \geq 2$, з початковими умовами

$$a_0 = 1, a_1 = 0, a_2 = -\frac{A}{10}.$$

Вказане призводить до наступного алгоритму, що називається `AtkinModComp.e`

Його сутність у тому, що:

1. Обчислюється

$$G(Z) = \exp_1(F(Z));$$

2. Обчислюється $I(x)$, використовуючи рівняння (23);

3. Обчислюється $G(Z)$ як модульна композиція;

4. Визначається D , використовуючи рівняння (13).

Складність алгоритму порядку

$$O\left(M(1)\sqrt{l+1}^{\frac{w+1}{2}}\right) \text{ або } O\left(M(1)\sqrt{l \log l}\right)$$

операцій в K .

Для того, щоб зробити простіше потрібно переглянути $G(I) = (\exp \circ F) \circ I$, що використовується вище.

Рівняння Аткина (11) можна також переписати у вигляді

$$D(p(x)) = \exp\left(-\sigma^2 + 2 \int \int \int \int \ln p(x) - p^{\sim}(x)\right).$$

Потім можна отримати $D(1/x)$ в наступному експонентному вигляді:

$$D\left(\frac{1}{x}\right) = \exp\left(-\sigma I^2 + 2 \int \int \int \int \Gamma\left(\frac{1}{x} - (p^{\sim} \circ I)(x)\right)\right) \quad (15)$$

$$= \exp\left(-\sigma I^2 + 2 \int \int \int \int \Gamma\left(\frac{1}{x} - \frac{N\left(\frac{1}{x}\right)}{D\left(\frac{1}{x}\right)}\right)\right) \quad (16)$$

$$= \exp\left(-\sigma I^2 + 2 \int \int \int \int \Gamma\left(\frac{1}{x} - \frac{1}{S(\sqrt{x})^2}\right)\right) \quad (17)$$

Далі, уточнюючи деталі та послідовність операцій, отримаємо, що складність алгоритму буде такою ж як і для алгоритму `fastElkies`. Це зрозуміло, оскільки

$$\frac{N(x)}{D(x)} = lx - \sigma - 2\sqrt{x^3 + Ax + B} \left(\sqrt{x^3 + Ax + B} \frac{D'(x)}{D(x)} \right)$$

Тоді рівняння (16) не що інше, як інтегральне подання останнього рівняння.

8. Аналіз алгоритмів ЕП за критеріями криптографічної стійкості та складності.

Дослідження алгоритмів проведено, використовуючи бібліотеку NTL C++ [19, 20] та AMD 64 з процесором 3400 (2,4 ГГц). На рис. 1 наведені результати порівняння складності для розширення p , отриманих над кінцевим полем $F_{10}^{2004} + 4683$.

Вигляд обох кривих вказує на те, що теоретичні складності майже квадратичні і майже лінійні, отже добре дотримуються в нашій реалізації. При цьому стрибки за ступенями 2 відображають особливість реалізації FFT NTL арифметику, що використовується в бібліотеці.

algorithm	complexity	need of σ
linear algebra	$O(\ell^3)$	no
Stark1972	$O(\ell M(\ell))$	no
Atkin1992	$O(\ell M(\ell))$	yes
AtkinModComp	$O(M(\ell)\sqrt{\ell} + \ell^{\frac{w+1}{2}})$ or $O(M(\ell)\sqrt{\ell \log \ell})$	yes
Elkies1992	$O(\ell^2)$	yes
Elkies1998	$O(\ell^2)$	yes
fastElkies	$O(M(\ell))$	yes
fastElkies'	$O(M(\ell) \log \ell)$	no

Рис. 1. Порівняння алгоритмів

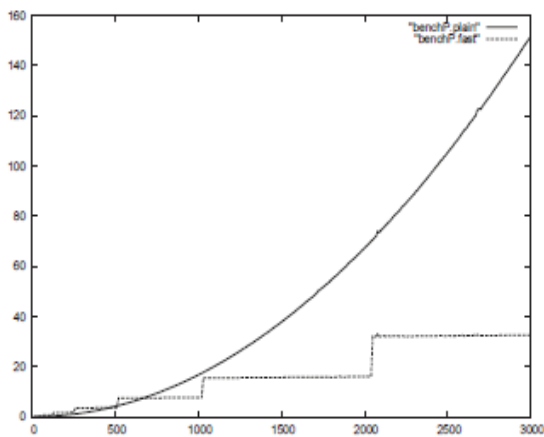


Рис. 2. Затримки для обчислень p на

$$E: y^2 = x^3 + 4589x + 91128 \text{ над } F_{10}^{2004} + 4683$$

Крім того, існує поріг, за яким алгоритм стає ефективним (корисним), це робить його цікавим на практиці.

Тепер звернемо увагу на частини ізогеній, концентруючись на виразі де l первинна, в контексті алгоритму SEA. Отже, в цьому випадку, достатньо обчислити поліном $g(x)$, що $D(x) = g(x)^2$. Всі алгоритми можуть бути адаптовані, як скористатися цим спрощенням показано в пункті 4.3 для наших алгоритмів fastElkies і fastElkies'.

Перша серія таймінгів належить обчисленню ізогенії над невеликим полем,

$$K = F_{10}^{19} + 51,$$

для кривої $E: y^2 = x^3 + 4589x + 91128$. Ми порівнюємо в рис. 2 виступи алгоритмів Elkies1992 з §6.3 і Elkies1998 з § 4.2 для ізогеніїв помірною ступеня $1 \leq 400$. Рис. 3 порівнює тимчасові діаграми, отримані з алгоритмом Elkies1998 і нашої швидкої версії fastElkies з §4.3, для ізогенії ступеня до 6000.

Далі ми порівнюємо на рис. 4 таймінги, отримані $O(M(1))$ алгоритмом fastElkies, що вимагає знання σ , до результатів, отриманих за допомогою його $O(M(1) \log l)$ аналогу fastElkies', який не вимагає цієї інформації.

У всіх фігурах, градуси l з ізогеніями наведені на горизонтальній осі, а таймінги (в секундах) – на вертикальній осі. Знову ж таки, форма обидвох кривих на рис. 3 показує, що теоретичні складності добре дотримуються в нашій реалізації. Криві на рис. 4, показують, що теоретичне відношення $\log l$ між алгоритмами fastElkies і fastElkies' має подальший практичний вплив.

Далі, в таблицях 2 до 8, ми даємо докладні таймінги на обчисленні l -ізогенії для кривої

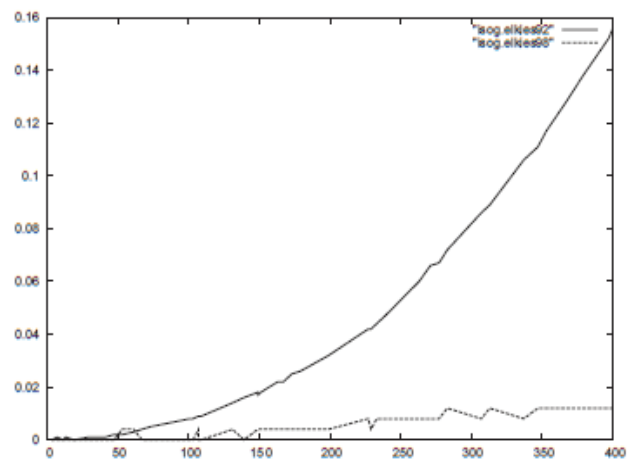


Рис. 3. Elkies1992 проти Elkies1998.

$$E : y^2 = x^3 + Ax + B$$

де

$$A = [10^{1990} \pi] = 31415926\dots58133904,$$

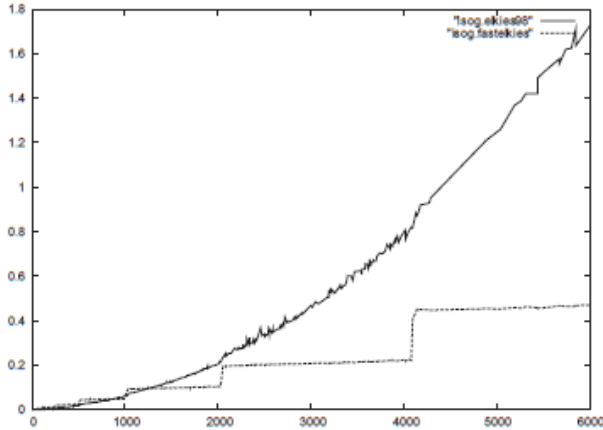


Рис. 4. Elkies1998 проти fastElkies

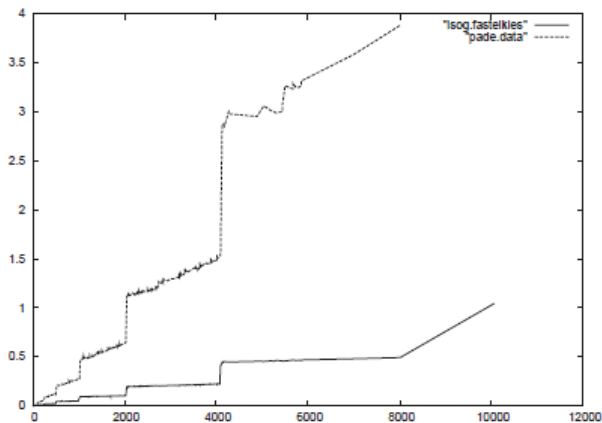


Рис. 5. FastElkies проти FastElkies'

для декількох значень l , над великим кінцевим полем $F_{10}^{2004} + 4683$, а також з використанням різних методів: алгоритми Elkies1992, Elkies1998, а також швидкий варіант fastElkies, алгоритм Stark, Stark1972 і дві версії Atkin1992 і AtkinModComp з алгоритмом Аткин. Отже

$$B = [10^{1990} e] = 27182818\dots94787610,$$

На рисунках 6 і 7 наведено таблиці значень складності для основних підпрограм бібліотеки. У таблиці 2 наведено тимчасові значення, що необхідні для обчислення розкладання p і p' , використовуючи або класичний алгоритм або розглянутий вище швид-

кий варіант. Зазначене, використовується в усіх алгоритмах, за винятком алгоритму fastElkies. У таблиці рис. 8 також наведені значення часової складності для відновлення g відносно сум, спочатку з використанням класичного квадратичного алгоритму, а потім за допомогою швидкого експоненціювання. Вказана можливість реалізована в алгоритмах Elkies1992, і Elkies1998 і його модифікаціях.

l	Computing φ and $\tilde{\varphi}$		
	order	quadratic	fast
1013	511	8.6	7.0
2039	1024	34.6	29.9
3019	1514	75.7	30.3
4001	2005	132.7	31
5021	2515	209.3	64.4

Рис. 6. Таблиця складності обчислення p і p'

l	Computing φ and $\tilde{\varphi}$		
	order	quadratic	fast
1013	511	8.6	7.0
2039	1024	34.6	29.9
3019	1514	75.7	30.3
4001	2005	132.7	31
5021	2515	209.3	64.4

Рис. 7. Таблиця відновлення g від своїх степеневих сум

У таблицях рисунків 8 і 9 наведені тимчасові значення складності для алгоритмів Elkies1992 і Elkies1998 та варіації fastElkies.. У таблиці рис. 8, стовпці μ і p_i вказують на час, що необхідний для обчислення коефіцієнтів μ_{ij} і сум p_i . У таблиці рис. 9, стовпець h_i вказує час, що трититься для обчислення коефіцієнтів h_i з раціональною функцією N/D , та з використанням квадратичного алгоритму Elkies1998 та алгоритму fastElkies. Наступна колонка дає час, який використовується для обчислення статечні суми p_i від h_i з використанням повторення (10).

В таблицях рис. 10 і 11 наведено значення часової для реалізації оригінального алгоритму Atkin в Atkin1992, а також більш швидку версію AtkinModComp з використанням модульного складання.

У таблиці рис. 11, стовпець "експонентний" сто-сується обчислення $\exp(F)$, з використанням швидко-

го алгоритму; стовпець p^k містить значення часу для обчислення всіх серій $p(z)^k$, а стовпець g – для відновлення коефіцієнтів g на основі сум.

У таблиці рис. 11 наведено дані, що отримані з використанням ModComp1 і ModComp2; передостанній стовпець дає час для обчислення $\exp(F)$ і

ℓ	Elkies1992			
	$\varphi, \tilde{\varphi}$	μ	p_i	g
1013		10.4	4.4	
2039	See	49.1	17.9	See
3019	Table 2	130.6	38.9	Table 3
4001		263	68.4	
5021		496.5	106.6	

Рис. 8. Таблица складності алгоритму Elkies1992

ℓ	Elkies1998 and fastElkies			g
	h_i		p_i	
	quadratic	fast		
1013	4.4	4.5	0.05	
2039	17.3	9.6	0.1	See
3019	38.0	19.5	0.16	Table 3
4001	67.2	20.0	0.21	
5021	105.0	40.7	0.27	

Рис. 9. Таблица складності алгоритмів Elkies1998 і fastElkies

для обчислення ступенів I ; останній стовпець містить значення часу виконання остаточного множення.

Аналіз показав, що асимптотично алгоритм ModComp2 є більш швидким, ніж алгоритм ModComp1, так що значення часу в таблиці рис. 12, можуть стати несподіванкою. Цей факт можна пояснити тим, що для проблемних розмірів, що нас цікавлять, переважають стадії алгоритму Mod-Comp1 засновані на поліноміальних операціях. Водночас крок заснований на лінійній алгебрі займає лише близько 10% операцій від усього часу обчислень. Таким чином, практична складність цього алгоритму в розглянутому діапазоні ($1000 < l < 6000$) пропорційно $M(1)\sqrt{l}$, в той час як алгоритму ModComp2 пропорційна $M(1)\sqrt{l \log l}$. Крім того, коефіцієнт пропорційності менше у вбудованій в NTL функції виконання ModComp1, ніж в реалізації ModComp2.

Також необхідно враховувати, що в таблицях рис. 6 – 11, таймінги відображають вже згадувану поведінку FFT, коли поліном множення в діапазоні 1024 – 2047 приблизно вдвічі швидший, ніж в діапазоні 2047 – 4095 і приблизно в чотири рази швидший, ніж в діапазоні 4096 – 8191.

У таблиці рис. 12 наведені значення часу для алгоритму Stark1972; окремо від загальних обчислень p і p^k , виділено час, необхідний для обчислення всіх зворотних (квадратичний алгоритм та швидкі інверсії), а також визначення многочленів q_n .

ℓ	$\varphi, \tilde{\varphi}$	Algorithm Aktin1992		φ^k	g
		exponential naive	fast		
1013		88.4	1.2	72.3	4.4
2039	See	370.1	4.9	304.9	17.7
3019	Table 2	955.9	5.1	755.8	38.9
4001		1503	5.2	1218.9	67.6
5021		3180	10.8	2506.4	108.7

Рис. 10. Таблица оригінального алгоритму Atkins, варіація для $\exp(F)$

ℓ	$\varphi, \tilde{\varphi}$	Algorithm AtkinModComp				g
		$\exp(F)$	$T^{1-\ell}$	modular composition		
				ModComp1	ModComp2	
1013		1.2	2.7	14.3	35.6	0.2
2039	See	2.5	6.6	45.8	111.9	0.4
3019	Table 2	5.1	10.4	95.3	241	0.7
4001		5.2	11.6	143.2	338	0.9
5021		10.9	20.9	240	642	1.4

Рис. 11. Таблица складності алгоритму Atkins з модульним складом

ℓ	$\varphi, \tilde{\varphi}$	Inverses		q_n
		quadratic	fast	
1013		23542	1222.7	28.0
2039	See	> 100000	5113.4	116.9
3019	Table 2		12182	258
4001			20388	418.6
5021			38910	663.1

Рис. 12. Алгоритм Stark1972

У перспективі важливим є порівняння ЕП з іншими кандидатами на постквантові алгоритми ЕП.

ВИСНОВКИ

Складність аналізу алгоритмів, що розглянуті, для випадків великих простих характеристик і для

досить великої і ізогенії, новий $O(M(1))$ алгоритм кращий, ніж раніше відомі.

Поточну реалізацію алгоритму можна додатково оптимізувати, щоб зробити цей алгоритм більш швидким для менших значень ступеня. Справді, відомо, що алгоритми, засновані на ітераційних операціях Ньютона представляють певні скорочення штатів (коефіцієнти що можна передбачити заздалегідь, повторних множників). Видалення цих залишків здійснено в [4, 12], що дозволяє досягти постійного фактора прискорень. На сьогодні, існуючі реалізації спираються лише частково на ці методи; вважається, що подальші зусилля програмування принесє практичні поліпшення.

Інший напрямок для майбутньої роботи полягає в адаптації наведених методів у разі невеликої характеристики. У зв'язку з цим, зміна останньої фази алгоритму Joux and Lercier [13], є багатообіцяючим шляхом пошуку.

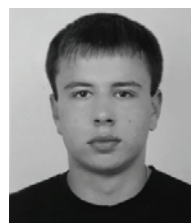
Література

- [1] C. Alonso, J. Gutierrez, and T. Recio. A rational function decomposition algorithm by near-separated polynomials. *Journal of Symbolic Computation*, 19(6):527–544, 1995.
- [2] A. O. L. Atkin. The number of points on an elliptic curve modulo a prime (II). Available at <http://listserv.nodak.edu/archives/nmbrthry.html>.
- [3] D. J. Bernstein. Composing power series over a finite ring in essentially linear time. *Journal of Symbolic Computation*, 26(3):339–341, 1998.
- [4] D. J. Bernstein. Removing redundancy in high-precision Newton iteration, 2000. Available on-line at <http://cr.yp.to/fastnewton.html>.
- [5] I. Blake, G. Seroussi, and N. Smart. Elliptic curves in cryptography, volume 265 of London Mathematical Society Lecture Notes Series. Cambridge University Press, 1999.
- [6] R. P. Brent. Multiple-precision zero-finding methods and the complexity of elementary function evaluation. In *Analytic computational complexity*, pages 151–176. Academic Press, New York, 1976. Proceedings of a Symposium held at Carnegie-Mellon University, Pittsburgh, Pa., 1975.
- [7] R. P. Brent, F. G. Gustavson, and D. Y. Y. Yun. Fast solution of Toeplitz systems of equations and computation of Padé approximants. *Journal of Algorithms*, 1(3):259–295, 1980.
- [8] D. G. Cantor and E. Kaltofen. On fast multiplication of polynomials over arbitrary algebras. *Acta Informatica*, 28(7):693–701, 1991.
- [9] J.-M. Couveignes, L. Dewaghe, and F. Morain. Isogeny cycles and the Schoof-Elkies-Atkin algorithm. Research Report LIX/RR/96/03, LIX, April 1996. Available at <http://www.lix.polytechnique.fr/Labo/Francois.Morain/>.
- [10] N. D. Elkies. Explicit isogenies. Draft, 1992.
- [11] N. D. Elkies. Elliptic and modular curves over finite fields and related computational issues. In D. A. Buell and J. T. Teitelbaum, editors, *Computational Perspectives on Number Theory: Proceedings of a Conference in Honor of A. O. L. Atkin*, volume 7 of AMS/IP Stud-

- ies in *Advanced Mathematics*, pages 21–76. American Mathematical Society, International Press, 1998.
- [12] G. Hanrot, M. Quercia, and P. Zimmermann. The middle product algorithm, I. Speeding up the division and square root of power series. *Applicable Algebra in Engineering, Communication and Computing*, 14(6):415–438, 2004.
- [13] A. Joux and R. Lercier. Counting points on elliptic curves in medium characteristic. *Cryptology ePrint Archive*, Report 2006/176, 2006. <http://eprint.iacr.org/>.
- [14] E. Kaltofen and V. Shoup. Subquadratic-time factoring of polynomials over finite fields. *Mathematics of Computation*, 67(223):1179–1197, 1998.
- [15] K. S. Kedlaya. Counting points on hyperelliptic curves using Monsky-Washnitzer cohomology. *Journal of the Ramanujan Mathematical Society*, 16(4):323–338, 2001.
- [16] V. Müller. Ein Algorithmus zur Bestimmung der Punktzahl elliptischer Kurven über endlichen Körpern der Charakteristik größer drei. PhD thesis, Technische Fakultät der Universität des Saarlandes, 1995.
- [17] A. Schönhage and V. Strassen. Schnelle Multiplikation großer Zahlen. *Computing*, 7:281–292, 1971.
- [18] R. Schoof. Counting points on elliptic curves over finite fields. *Journal de Théorie des Nombres de Bordeaux*, 7(1):219–254, 1995.
- [19] V. Shoup. A new polynomial factorization algorithm and its implementation. *Journal of Symbolic Computation*, 20(4):363–397, 1995.
- [20] V. Shoup. The Number Theory Library. 1996–2005. <http://www.shoup.net/ntl>.
- [21] H. M. Stark. Class-numbers of complex quadratic fields. In W. Kuyk, editor, *Modular functions of one variable I*, volume 320 of *Lecture Notes in Mathematics*, pages 155–174. Springer Verlag, 1973. Proceedings International Summer School University of Antwerp, RUCA, July 17-August 3, 1972.



Пономар Володимир Андрійович, аспірант факультету комп'ютерних наук, кафедри безпеки інформаційних систем і технологій Харківського національного університету імені В.Н. Каразіна. Область наукових інтересів: криптографічні перетворення, безпечне програмування, методи багатфакторної автентифікації та їх застосування з метою захисту інформації, захист криптографічних засобів інформації



Бережний Олександр Григорович, студент факультету комп'ютерних наук, кафедри безпеки інформаційних систем і технологій Харківського національного університету імені В.Н. Каразіна. Область наукових інтересів: криптографічні перетворення, швидкі алгоритми для обчислення ізогеній на еліптичних кривих.

УДК 681.3.06

Быстрые алгоритмы для вычисления изогений эллиптических кривых / В.А. Пономарь, А.Г. Бережной // Прикладная радиоэлектроника: науч.-техн. журнал. – 2016. – Том 15, №. 3 – С. 203 – 214.

В работе рассматриваются и анализируются алгоритмы вычисления изогений эллиптических кривых над конечным полем. Анализируется алгоритм вычисления изогений выбранной степени, который базируется на быстрых алгоритмах разложения β -функции в ряд Вейерштрасса.

Ключевые слова: эллиптические кривые, изогении эллиптических кривых, быстрые алгоритмы вычисления изогений.

Ил. 12. Библиогр.: 21 назв.

UDC 681.3.06

Fast algorithms for calculating isogeny of elliptic curves / V.A. Ponomar, O. G. Berezhnyi // Applied Radio Electronics: Sci. Journ.. – 2016. – Vol. 15, №. 3. – P. 203 – 214.

The paper deals with algorithms for calculating isogeny of elliptic curves over finite fields. The fast algorithm for calculating isogeny with a chosen degree is analyzed, which is based on the fast algorithms of β - function decomposition into Weierstrass row.

Keywords: elliptic curves, isogeny of elliptic curves, fast algorithms for calculating isogeny.

Fig. 12. Ref.: 21 items.

КВАНТОВІ КРИПТОГРАФІЧНІ АЛГОРИТМИ ЕЛЕКТРОННОГО ПІДПISУ НА ОСНОВІ МУЛЬТИВАРІАТИВНИХ КВАДРАТИЧНИХ ПЕРЕТВОРЕНЬ

Д. В. ГАРМАШ, О. О. БАКЛИКОВ, Н. В. ФІЛАТОВА, І. Д. ГОРБЕНКО

Наводяться вимоги до постквантових алгоритмів асиметричних криптоперетворень. Вказується на актуальність та необхідність пошуку, дослідження, стандартизації та застосування криптографічного примітиву типу електронний підпис (ЕП). Розглядається сутність та можливості застосування мультіваріативних квадратичних перетворень в ході реалізації ЕП, робиться попередній аналіз їх властивостей та наводиться практичний приклад.

Ключові слова: вимоги до постквантових електронних підписів, електронний підпис, квантове криптоперетворення, математичні основи мультіваріативних перетворень, мультіваріативне квадратичне перетворення для електронного підпису.

ВСТУП

У 2016 році у США та ЄС розпочалися активні роботи щодо підготовки до проведення конкурсів відносно методів та на їх основі майбутніх кандидатів квантово-захищених алгоритмів криптографічних перетворень. Підтвердженням цьому є заяви АНБ США та технічний звіт NIST США [1,2], розпочаті широкі попередні дослідження та підтримка провідних криптографів проблем постквантової криптографії[1]. Так АНБ та NIST ініціювали роботи щодо організації конкурсу на нові стандарти квантово-захищених криптографічних алгоритмів. Необхідно відмітити значне число досліджень в ЄС[3] та публікацій на міжнародному рівні і в Україні [4,5].

У лютому 2016 року свої плани NIST анонсував на VII Міжнародній конференції з постквантової криптографії, що проходила у Японії [3,4]. У подальшому планується підготувати вимоги та оголосити у 2017 конкурс на кращі криптопримітиви для застосування у постквантовий період. Також планується протягом 3 – 5 років провести їх аналіз та порівняння, а потім у 2020 – 2022 роках прийняти нові криптографічні постквантові стандарти.

Серед множини криптографічних примітивів важливе значення мають електронні підписи (ЕП), що пояснюється їх широким застосуванням та можливостями великих втрат у фінансовій сфері та економіці, зрозуміло у випадку компрометації сьогоденні існуючих стандартизованих ЕП [1 – 5].

Зважаючи на актуальність та необхідність створення постквантових алгоритмів ЕП у цьому напрямі уже розпочаті дослідження, певною мірою визначено математичні основи, на яких можуть бути побудовані постквантові алгоритми ЕП та асиметричного шифрування. За результатами вказаної VII міжнародній конференції [4] в основному визначені шляхи створення постквантових алгоритмів ЕП. Основними з них є дослідження, що ґрунтуються на використанні [1 – 10]:

- мультіваріативного квадратичного криптоперетворення (multivariate-quadratic-equations cryptography) на основі запропонованого Matsumoto та Imai підходу[6];

- перетворення на основі геш-дерева Меркеля (Merkle), у свою чергу побудованого з використанням ідеї Lamport та Die про підпис одного повідомлення [7];

- криптоперетворення у фактор кільці, стійкість якого доводиться на основі використання математичного апарату алгебраїчних решіток (Lattice-based cryptography). Найбільший інтерес у цьому класі є схема асиметричного шифрування Hoffstein-Pipher-Silverman “NTRU”, згідно з якою прийнято та застосовується стандарт США X9.98[8];

- криптоперетворення на основі застосування кодів (Code-based cryptography), класичним прикладом якого є схема асиметричного шифрування та ЕП Mc Eliece з кодами Гоппи (Goppa) [9];

- криптоперетворення та основі використання математичного апарату ізогеній еліптичних кривих[10].

Наш попередній аналіз дозволив зробити висновок, що усі названі криптоперетворення можуть бути використані для побудови постквантових ЕП, тому вони заслуговують відповідної уваги.

Метою цієї статті є узагальнення вимог до постквантових алгоритмів ЕП та попередній аналіз сутності, можливостей та властивостей криптографічного перетворення типу ЕП на основі використання мультіваріативного квадратичного перетворення.

1. ВИМОГИ ДО ПОСТКВАНТОВИХ ЕП

Аналіз показав, що уже сьогодні США на рівні NIST, Європейський Союз на рівні ETSI, Японія та Німеччина розпочали активну роботу з формування вимог до квантово-захищених криптоалгоритмів. Так, до кінця 2026 року планується в основному розробити та розпочати відкрите обговорення квантово-захищених алгоритмів [1 – 4]. За результатами вказаних робіт усі вимоги можна поділити на безумовні

або цільові, тобто відносно криптографічної стійкості, техніко-економічні та техніко-експлуатаційні.

Як мінімальні вимоги до можливих кандидатів можна віднести [5]:

- безумовне доведення криптостійкості проти квантового криптоаналізу;
- забезпечення однієї з функцій криптоперетворення, наприклад ЕП;
- відкритість алгоритму ЕП для криптографічного аналізу криптографічною спільнотою;
- можливість реалізації та застосування ЕП у широкому діапазоні платформ.

Продовжується подальший розгляд вимог до постквантових примітивів, їх конкретизація здійснюється, як уже вказувалось вище, у трьох напрямках:

- вимоги відносно стійкості до криптографічного аналізу, причому вони визнаються безумовними, тобто повинні кандидатом виконуватись безумовно;
- техніко-економічні вимоги в основному в частині часової та просторової складностей (складність обчислень та витрати на пам'ять); технічні характеристики реалізації алгоритмів;
- техніко-експлуатаційні вимоги в частині простоти реалізації та використання.

Сьогодні вказані вимоги уточнюються та деталізуються, ми в подальшому покладемо в ході аналізу за основу та посилатимемося на необхідні в подальшому.

2. МАТЕМАТИЧНІ ОСНОВИ МУЛЬТИВАРІАТИВНИХ КВАДРАТИЧНИХ ПЕРЕТВОРЕНЬ

Механізм (схема) мультиваріативних квадратичних перетворень Т. Мацумото і Х. Імаї представили на конференції Eurocrypt у 1988 р [6]. Пізніше в цьому напрямі французом Жаком Патарином були досліджені "мономіальні криптосистеми приховування інформації» [4,6,11]. Вони засновані на розширенні поля $i(t)$ (у поліноміальній формі). Пізніше Ж. Патарин розробив інші механізми (схеми) перетворень. У співпраці з Евіадам Кипнісом і Луї Губіном в 1997 році він представив перетворення "Balanced Oil and Vinegar", а у 1999 "Unbalanced Oil and Vinegar". У подальшому запропоновані перетворення практичної реалізації не отримали. Але з часом, коли з'явилися постквантові загрози, увага була звернута і на мультиваріативні квадратичні перетворення. Сьогодні ці перетворення розглядаються криптологами як реальні кандидати на постквантовий ЕП.

Мультиваріативні перетворення ґрунтуються на використанні кінцевого поля [6,11 – 13]

$$k = \frac{GF[2]}{x^2 + x + 1}, \quad (1)$$

в якому 2^2 елементів. Для спрощення вони позначаються множиною чисел $\{0,1,2,3\}$. Причому 0 є ну-

лем у полі k , 1 є одиницею, 2 є поліномом x , а 3 є поліномом $1+x$. По суті, коефіцієнти квадратичних поліномів приймають значення над полем $4=2^2$. Як квадратичні поліноми вибираються такі:

$$G_0 = (x_1, x_2, x_3) = 1 + x_2 + 2x_0x_2 + 3x_1^2 + 3x_1x_2 + x_2^2;$$

$$G_1 = (x_1, x_2, x_3) = 1 + 3x_0 + 2x_1 + x_2 + x_0^2 + x_0x_1 + 3x_0x_2 + x_1^2;$$

$$G_2(x_1, x_2, x_3) = 3x_2 + x_0^2 + 3x_1^2 + x_1x_2 + 3x_2^2.$$

Вважається, що наведені поліноми є багато вимірними поліномами над кінцевим полем та вони можуть застосовуватися у багатофакторній криптографія під час розробки асиметричних криптографічних примітивів.

Механізм (схема) Мацумото та Імаї (МІА). Для цієї схеми використовується центральне рівняння над полем розширення E ступеня n . Воно має вигляд

$$P(x') := x'^{q^{\lambda}+1} \quad (2)$$

для $q := |F|$ та деякого $\lambda \in \mathbb{N}$.

Також між F^n та E , в процесі перетворення використовується коефіцієнт бієкції. Нехай вектор $a \in F^n$ подається у вигляді (a_1, \dots, a_n) , причому $a_i \in F$, нехай також $b \in E$ та має вигляд $b_{n-1}t^{n-1} + \dots + b_1t + b_0$ з $b_i \in F$, а $i(t)$ є визначальним многочленом E . Далі, нехай $x'^{q^{\lambda}}$ є лінійним рівнянням над F для будь-якого $\lambda \in \mathbb{N}$, а x' призводить до квадратних рівнянь над F . Причому вказана бієкція справджується, якщо $\gcd(q^n - 1, q^{\lambda} + 1) = 1$.

Рівняння приховування в полі. Використовується та сама ідея, як у МІА, але для влаштування люку використовується інша ідея. Як і для МІА, використовуються центральні рівняння над полем розширення E зі ступенем n . Вони мають вигляд:

$$P'(x') := \sum_{\substack{0 \leq i, j \leq d \\ q^i + q^j \leq d}} C'_{i,j} x'^{q^i + q^j} + \sum_{\substack{0 \leq k \leq d \\ q^k \leq d}} B'_k x'^{q^k} + A', \quad (3)$$

$$\text{де } \begin{cases} C'_{i,j} x'^{q^i + q^j} \text{ для } C'_{i,j} \in E - \text{квадратичні члени} \\ B'_k x'^{q^k} \text{ для } B'_k \in E - \text{лінійні члени} \\ A' \text{ для } A' \in E - \text{константа} \end{cases}$$

для $i, j \in \mathbb{N}$ та деякого $d \in \mathbb{N}$.

Базові класи. ґрунтуються на незбалансованій схемі Оіла та Вінежера (UOV). Використовуються значення vinegar та oil (v та o). Мається $n = v + o$ та потрібно

$v = 20 \dots 30$ для безпечної схеми. Крім того, $\epsilon \in \mathbb{m}$. При цьому центральні поліноми мають вигляд:

$$p_i'(x_1', \dots, x_n') := \sum_{j=1}^v \sum_{k=1}^n \gamma_{i,j,k}' x_j' x_k' + \sum_{j=1}^n \beta_{i,j}' x_j' + a_i' \quad (4)$$

для $1 \leq i \leq m$ та коефіцієнтів

$$\alpha_i', \beta_{i,j}', \gamma_{i,j,k}' \in F$$

Примітка: ці рівняння стають лінійними, якщо значення присвоюються до значень Віженера x_1', \dots, x_v' .

Далі застосовуються поетапні трикутні системи (STS). В них система P має такий вигляд:

$$\text{Крок 1} \left\{ \begin{array}{l} p_1' \quad (x_1', \dots, x_r') \\ \cdot \\ \cdot \\ p_r' \quad (x_1', \dots, x_r') \end{array} \right. \quad (5)$$

$$\text{Крок } l \left\{ \begin{array}{l} p_{(l-1)r+1}' \quad (x_1', \dots, x_r', \dots, x_{(l-1)r+1}', \dots, x_{lr}') \\ \cdot \\ \cdot \\ p_{lr}' \quad (x_1', \dots, x_r', \dots, x_{(l-1)r+1}', \dots, x_{lr}') \end{array} \right.$$

3. КЛЮЧОВІ ДАНІ ТА БАЗОВІ ПОЗНАЧЕННЯ

Мультіваріативні квадратичні перетворення є криптографічними, якщо під час його виконання застосовуються спеціальні дані – асиметричний ключ. Він складається з відкритого ключа K_b та особистого (таємного) K_o . Основною вимогою до асиметричної пари (K_b, K_o) є вимога, щоб

$$K_b \neq K_o \quad (6)$$

та щоб при знанні одного із них – наприклад, як на практиці – відкритого, визначення таємного було експоненційно складним. Як мінімум у деяких випадках – субекспоненційно складним.

Відкритий ключ K_b будується із поліномів скінченного поля P . На практиці це завжди сукупність коефіцієнтів $p_i' s$, що складаються (розміщаються) у певному порядку. Це робиться з метою зменшення складності обчислень. Оскільки K_b є відкритий ключ, то $P(0)$ завжди дорівнюється нулю.

Таємний ключ складається з інформації, що міститься в S, T і Q . Тобто, складається з $(M_S^{-1}, c_S), (M_T^{-1}, c_T)$ та усіх параметрів, які існують

в Q . Теоретично, один з c_S та c_T може бути зайвим, але він зберігається у будь-якому випадку [4,6].

Для того, щоб перевірити підпис або зашифрувати інформацію [5,6], застосовується відкритий ключ у вигляді

$$z = P(w). \quad (7)$$

Для того, щоб підписати або розшифрувати, застосовується таємний ключ у вигляді

$$y = T^{(-1)}(z), \quad x = Q^{-1}(y) \quad \text{і} \quad w = S^{-1}(x), \quad (8)$$

але потрібно зважити на те, що це може бути тільки один з багатьох прообразів, який не обов'язково є зворотною функцією у змісті криптографічного перетворення.

Необхідно відмітити, що навіть якщо ми обмежимося криптосистемами, для яких відкритий ключ є набором поліномів $P = (p_1, \dots, p_m)$ у змінних $w = (w_1, \dots, w_n)$, де всі змінні і коефіцієнти знаходяться в $K = F_q$, шлях, який приховує перетворення (можливо лазівку) не є унікальним.

Проте, повідомлення завжди можна захистити за допомогою афінних перетворень S, T . Тобто, $P = T \circ Q \circ S : K^n \rightarrow K^m$, або

$$P: w = (w_1, \dots, w_n) \rightarrow x = M_S w + c_S \rightarrow y = z = M_T y + c_T = (z_1, \dots, z_m) \quad (9)$$

Також необхідно відмітити, що у будь-якому перетворенні головне перетворення Q належить до певного класу квадратичних відображень, при чому для нього зворотне перетворення з точки зору складності виконання є поліноміально складним. При цьому таємні відображення S та T є афінними (можливо навіть лінійними) та повного рангу. В цьому випадку x_j називається центральною змінною, а поліноми y_j та x називаються центральними поліномами. Далі, коли необхідно знайти різницю між змінною та значенням, то її позначають як $y_i = q_i(x)$. При цьому ключ K_c є основою такого механізму.

Також наведемо позначення, які використовуються далі. Розмір блоку шифру або набору повідомлень – m елементів F_q . Блок відкритого тексту або розмір підпису – n елементів у скінченному полі F_q . Розмір відкритого ключа $\frac{mn(n+3)}{2}$,

F_q – елементи, які зберігаються.

Розмір таємного ключа оцінюється як

$$(n^2 + m^2 + [\# \text{параметри в } Q]), \quad (10)$$

де F_q – елементи, які зберігаються у відповідному форматі.

Складність таємного перетворення оцінюється як

$$(n^2 + m^2). \quad (11)$$

Складність відкритого перетворення з наближенням можна оцінити як

$$mn^2 / 2F^q. \quad (12)$$

Трудомісткість (складність) генерації ключа - n^2 в полі F оцінюється в інтервалі

$$(O(n^4) - O(n^5)). \quad (13)$$

Попередній аналіз дозволив виявити основний недолік мультіваріативних перетворень – суттєве збільшення, порівняно з традиційними криптосистемами RSA, DSA або ECC, довжини ключів. Але вказане, як і інші питання властивостей мультіваріативного перетворення та умов його застосування, вимагають непростих досліджень.

4. КРИПТОСИТЕМА ЕП НА ОСНОВІ МУЛЬТИВАРІАТИВНОГО КВАДРАТИЧНОГО ПЕРЕТВОРЕННЯ

Відкритий ключ у такій системі є послідовністю [4,6]

$$P_1, P_2, \dots, P_{2b} \hat{F}_2[w_1, \dots, w_{4b}]. \quad (14)$$

Із $2b$ поліномів з $4b$ змінних. w_1, \dots, w_{4b} з коефіцієнтами у полі $F_2 \in \{0,1\}$. Кожний поліном має мати ступінь не більше 2 та без квадратичних термів, та поданий як послідовність

$$1, w_1, \dots, w_{4b}, w_1w_2, w_1w_3, \dots, w_{4b-1}w_{4b}. \quad (15)$$

У цілому, відкритий ключ має довжину $16b^3 + 4b^2 + 2b$ бітів. Наприклад, для $b=128$, розмір відкритого ключа складатиме 4 Mbyte. Для інших b дані наведено в таблиці 1.

Таблиця 1

Параметри відкритого ключа залежно від b

Значення b	$16b^3 + 4b^2 + 2b$ бітів	Відкритий ключ
128	$16 * 128^3 + 4 * 128^2 + 2 * 128$	33620224 бітів
256	$16 * 256^3 + 4 * 256^2 + 2 * 256$	268698112 бітів
512	$16 * 512^3 + 4 * 512^2 + 2 * 512$	2148533248 бітів

Значною перевагою підпису на основі MQ-криптографії можна вважати те, що підпис є коротким. Інші MQ-підписи з більш коротшими відкритими ключами мають підписи ще у більшості випадків ще коротші. Для здійснення атаки зловмиснику необхідно знайти послідовність з $4b$ w_1, \dots, w_{4b} бітів, що породжує $2b$ зазначених вище вихідних бітів, це і є головною проблемою для нього.

У таблиці 2 наведені значення ймовірностей вгадування послідовності із $4b$ бітів

$$(P_1(w_1, \dots, w_{4b}), \dots, P_{2b}(w_1, \dots, w_{4b}))$$

В таблиці 3 наведені значення складності вгадування послідовності із $4b$ бітів для більш досконалих атак, таких як «XL» -атаки.

Таблиця 2

Ймовірності вгадування послідовності із $4b$ бітів

Значення b	2^{-2b}	Ймовірність вгадування
128	$2^{-2 \times 128}$	$8.636168555094445e - 78$
256	$2^{-2 \times 256}$	$7.458340731200207e - 155$
512	$2^{-2 \times 512}$	$5.562684646268003e - 309$

Таблиця 3

Значення складності вгадування послідовності із $4b$ бітів для «XL» -атаки.

Значення b	2^{2b}	Кількість операцій
128	$2^{2 \times 128}$	$1.157920892373162e + 77$
256	$2^{2 \times 256}$	$1.3407807929942597e + 154$
512	$2^{2 \times 512}$	$1,797693134862315907729305190789e + 308$

Але для більшості квадратичних поліномів P_1, \dots, P_{2b} з $4b$ змінними на даний момент можуть бути зроблені невідомі атаки за 2^{2b} операцій. Ця проблема вирішується вже тривалий час.

Важливою перевагою підпису на основі MQ-криптографії над НВ - підписом є те, що підпис є коротким. Інші MQ-системи мають ще коротші підписи і, у більшості випадків, більш короткий відкритий ключ.

Алгоритм електронного підпису. Розглянемо як здійснюється електронний підпис під час застосування мультіваріативного перетворення [4,6].

Підписувач генерує відкритий ключ P_1, \dots, P_{2b} з таємною структурою, більш конкретно, з HFE^{v-} структурою, яка дозволяє підписувачу вирішити вищевказану проблему за прийнятний час. Можливо, що зловмисник зможе розкрити HFE^{v-} структуру у відкритому ключі або у відкритому ключі разом із послідовністю легітимних підписів. Але така атака поки ще не відома.

Нехай зафіксовано стандартний незвідний поліном $\phi \in F_2(t)$ ступеню $3b$. Визначимо L як поле $F_2(t)/\phi$ розмірністю 2^{3b} . Критичним кроком під час формування підпису є пошук кореня таємного одномірного (univariate) поліному малого ступеня над L , а саме, поліному в $L[x]$ зі ступенем не більше ніж $2b$. Існує декілька стандартних алгоритмів для вирішення цієї задачі за час $b^{O(1)}$.

Таємний поліном обирається так, щоб мати всі ненульові експоненти вигляду $2^i + 2^j$ або 2^i .

Якщо елементи $x \in L$ продані у вигляді

$$x_0 + x_1 t_1 + \dots + x_{3b-1} t_{3b-1}, \text{ де } x_i \in F_2, \quad (15)$$

тоді

$$x_2 = x_0 + x_1 t^2 + \dots + x_{3b-1} t^{6b-2}$$

$$x_4 = x_0 + x_1 t^4 + \dots + x_{3b-1} t^{12b-4}$$

і т. д.

Таким чином, $x^{2^i + 2^j}$ є квадратичним поліномом зі змінними x_0, \dots, x_{3b-1} .

Деякі прості перетворення приховують структуру цього поліному і породжують відкритий ключ.

Таємний ключ підписувача має три компоненти.

1. Оборотна матриця S розмірністю $4b \times 4b$ з коефіцієнтами в F_2 .
2. Поліном $Q \in L[x, n_1, n_2, \dots, n_b]$, де кожний терм має одну з шести можливих форм:
- 3.

$$lx^{(2^i + 2^j)},$$

де $l \in L, 2^i < 2^j, 2^i + 2^j \leq 2b$;

$$\text{де } lx^{(2^i)} j,$$

де $l \in L, 2^i \leq 2b$ (16)

$$lv_i v_j$$

$$lx^{(2^i)}$$

$$ln_j$$

$$l$$

Якщо $b = 128$, то ми маємо $944b$ можливих термів, кожний з яких має 384-бітний коефіцієнт l , а загальний об'єм складатиме 443 Kbytes.

4. Матрицю T розмірністю $2b \times 3b$ рангу $2b$ з коефіцієнтами у F_2 .

В цьому випадку підписувач обчислює відкритий ключ

$$(x_0, x_1, \dots, x_{3b-1}, v_1, v_2, \dots, v_b) \quad (17)$$

як S раз вектор (w_1, \dots, w_{4b}) всередині фактор-кільця

$$L[\omega_1, \dots, \omega_{4b}] / (\omega_1^2 - \omega_1, \dots, \omega_{1b}^2 - \omega_{1b}).$$

Далі необхідно обчислити

$$x = \sum x_i t^i \text{ та } y = Q(x, V_1, V_2, \dots, V_b)$$

та подати у вигляді

$$Y_0 + Y_1t + \dots + Y_{3b-1}t^{3b-1}$$

де кожний $Y_i \in F_2[\omega_1, \dots, \omega_{4b}]$.

Потім обчислити $(P_1, P_2, \dots, P_{2b})$ як T раз вектор-стовпчик

$$(Y_0, Y_1, \dots, Y_{3b-1}).$$

Безпосередньо підписування здійснюється у зворотному напрямку з використанням тих самих конструкцій.

1. Починаємо з величини P_1, P_2, \dots, P_{2b} . Спочатку для того, щоб отримати значення $(Y_0, Y_1, \dots, Y_{3b-1})$ розв'язується таємне лінійне рівняння

$$T(Y_0, Y_1, \dots, Y_{3b-1}) = (P_1, P_2, \dots, P_{2b}). \quad (17)$$

При цьому існує 2^b можливих варіантів розв'язання $(Y_0, Y_1, \dots, Y_{3b-1})$. Обираємо випадковим чином одне із них.

2. Обираємо випадково значення

$V_1, V_2, \dots, V_{b1} \in F_2$ і підставляємо його у таємний поліном $Q(x, V_1, V_2, \dots, V_b)$, отримуючи поліном

$$Q(x) \in L[x].$$

3. Обчислюємо

$$Y = Y_0 + Y_1t + \dots + Y_{3b-1}t^{3b-1} \in L$$

вирішуємо $Q(x) = Y$, внаслідок чого отримуємо $x \in L$. У випадку, коли існує декілька коренів, процес починається з початку.

4. Запишемо x як

$$x_0 + x_1t + \dots + x_{3b-1}t^{3b-1},$$

де $x_0, x_1, x_{3b-1} \in F_2$.

Розв'язуємо таємне рівняння

$$S(\omega_1, \dots, \omega_{4b}) = (x_0, \dots, x_{3b-1}, v_1, \dots, v_b) \quad (18)$$

та отримуємо підпис.

Цей приклад є прикладом з класу HFE^v – конструкцій, що запропоновані Potanin у 1996 році.

В ньому HFE - приховане рівняння в полі $Q(x) = Y$.

« \leftarrow » означає пропуск декількох бітів. Тобто

$Q(x) = Y$ є еквівалентом $3b$ рівнянь, але публікується тільки $2b$ рівнянь.

« v » – означає «vinegar» змінні v_1, v_2, \dots, v_b .

Безпосереднє (чисте) HFE перетворення, тобто без пропусків бітів та без v -змінної може бути атаковано за $2^{(lg b)^2}$ операцій атакою Grobner але HFE^v -перетворення протистоятиме такій атаці.

В таблиці 4 наведено оцінки складності такої атаки.

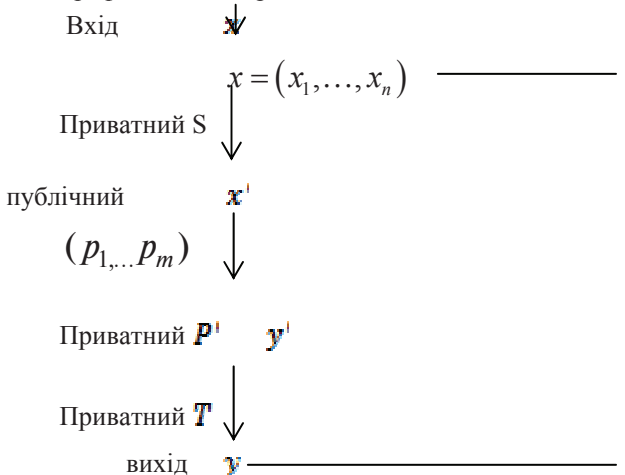
Таблиця 4

Складність атаки на HFE^v - перетворення

Значення b	$2^{(lg b)^2}$	Відкритий ключ
128	$2^{(lg 128)^2}$	216974
256	$2^{(lg 256)^2}$	556559
512	$2^{(lg 512)^2}$	1618688

5. ОСОБЛИВОСТІ ПОЛОЖЕНЬ МУЛЬТИВАРІАТИВНИХ КРИПТОГРАФІЧНИХ ПЕРЕТВОРЕНЬ

Генерування ключів можна подати у вигляді такого графічного відображення



Далі використовуються многочлени над малими кінцевими полями F , наприклад, $GF(2)$, $GF(128)$ або $GF(256)$, що орієнтовано, в тому числі, на 8-бітні мікропроцесори. За деяких умов можуть використовуватися розширення полів E розмірності n над полем F .

У результаті маємо:

- таємний ключ:

$$(S, P', T) \in AGL_n(F) \times MQ_m(F^n) \times AGL_m(F) \quad (19)$$

- відкритий (публічний) ключ:

-

$$P \in MQ_m(F^n), P = T \circ P' \circ S \quad (20)$$

- рівняння для відкритого ключа:

$$P_i(x_1, \dots, x_n) = \sum_{1 \leq j \leq k \leq n} \gamma_{j,k} x_j x_k + \sum_{j=1}^n \beta_{ij} x_j + a_i \text{ для } 1 \leq i \leq m \quad (21)$$

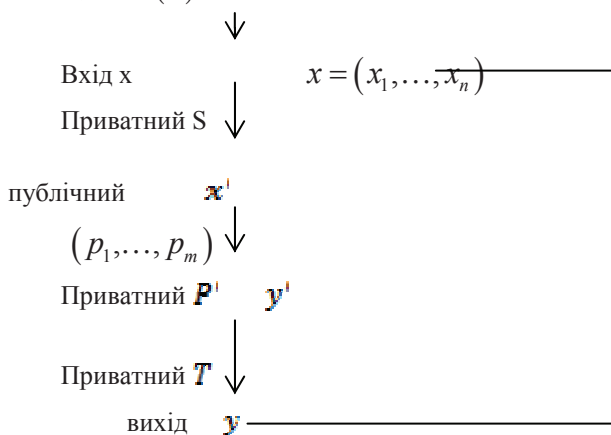
Причому коефіцієнти $a_i, \beta_{ij}, \gamma_{ij}, k \in F^a$

$$P(x) = (p_1(x_1, \dots, x_n), \dots, p_m(x_1, \dots, x_n))$$

У загальному випадку при прямому перетворенні (електронному підписі чи за шифруванні) робиться обчислення $y \in F^m$ для даного $x \in F^n$ шляхом оцінки $y = P(x)$.

Перевірка підпису може здійснюватись таким чином.

Дана пара $(x, y) \in F^n \times F^m$. Перевірити рівняння $y = P(x)$.



Важливою властивістю є можливість інверсії для афінних перетворень. Нехай

$$S(x) = M_s x + v_s,$$

$M_s \in F^{n \times n}, v_s \in F^n$. Потрібно, щоб M_s було зворотним, після цього можна обчислити $S^{-1}(x') = M_s^{-1}(x' - v_s)$. Далі можна інвертувати $T(y') = u$ для даного y' .

Генерування підпису. Необхідно для даного $y \in F^m$ зробити інверсію на кожному кроці та опублікувати відповідний x як підпис y .

Аналіз показує, що унікальна інверсія P' не може бути можливою, якщо є надмірність $H = h(x)$, навіть якщо використовується з $h(\bullet)$ криптографічно безпечною геш-функція.

Наведемо приклад та зробимо аналіз ключів. У криптосистемі мультіваріативного перетворення існують таємні та відкриті ключі. Таємний ключ складається з двох афінних перетворень, S і T , а також дозволяє легко інвертувати квадратичне відображення $P' : F^m \rightarrow F^n$. Позначимо через матрицю n афінних

ендоморфізмів. $S : F^m \rightarrow F^n$ через M_s і вектор зсуву $v_s \in F^n$ і аналогічно для $T : F^m \rightarrow F^n$. В результаті отримуємо

$$S(x) = M_s x + v_s$$

$$T(y) = M_T y' + v_T. \quad (22)$$

В (22) параметри $(S^{-1}, P'^{-1}, T^{-1})$ – це особисті ключі (ще їх називають люками). Публічний ключ є поєднанням $P = S \circ P' \circ T$ і його важко інвертувати, якщо люк є невідомим.

6. ПРИКЛАД МУЛЬТИВАРІАТИВНОГО КВАДРАТИЧНОГО ПЕРЕТВОРЕННЯ

Нехай x_1 і x_2 мають спільний PDF [15]

$$f_{x_1, x_2}(x_1, x_2) = 2, \quad 0 < x_1 < x_2 < 1,$$

і нуль в іншому випадку. Потрібно підрахувати спільний PDF випадкових величин

$$y_1 = \frac{x_1}{x_2}, \quad y_2 = x_2.$$

Розв'язок у загальному випадку.

1. Зважаючи на те, що
- 2.

$$x(2) = \{(x_1, x_2) : 0 < x_1 < x_2 < 1\}$$

$$g_1(t_1, t_2) = \frac{t_1}{t_2}, \quad g_2(t_1, t_2) = t_2;$$

3. Зворотні перетворення

$$Y_1 = \frac{X_1}{X_2}, Y_2 = X_2 \leftrightarrow X_1 = Y_1 Y_2, X_2 = Y_2 \quad \text{тому}$$

$$g_1^{-1}(t_1, t_2) = t_1, t_2, g_2^{-1}(t_1, t_2) = t_2$$

4. Діапазон: знайти $Y(2)$. Розглянемо точку перетворення з $X^{(2)}$ в $Y^{(2)}$. Для пари точок $(x_1, x_2) \in X^{(2)}$ і $(y_1, y_2) \in Y^{(2)}$, пов'язаних між собою перетворенням, маємо

$0 < x_1 < x_2 < 1 \leftrightarrow 0 < y_1 y_2 < y_2 < 1$ і, отже, можна отримати нерівності:

$$0 < y_2 < 1 \quad \text{і} \quad 0 < y_1 < 1$$

$$Y^{(2)} = (0, 1) \times (0, 1)$$

5. Якобіан для точок $(y_1, y_2) \in Y^{(2)}$ знаходиться

3

$$D_y = \begin{bmatrix} \frac{\partial x_1}{\partial y_1} & \frac{\partial x_1}{\partial y_2} \\ \frac{\partial x_2}{\partial y_1} & \frac{\partial x_2}{\partial y_2} \end{bmatrix} = \begin{bmatrix} y_2 & y_1 \\ 0 & 1 \end{bmatrix} \rightarrow$$

$$\rightarrow |J(y_1, y_2)| = |\det D_y| = |y_2| = y_2$$

Запишемо це для $(x_1, x_2) \in X^{(2)}$

$$D_x = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ x_2 & x_2^2 \\ 0 & 1 \end{bmatrix} \rightarrow$$

$$\rightarrow |J(x_1, x_2)| = |\det D_x| = \left| \frac{1}{x_2} \right| = \frac{1}{x_2}$$

$$\text{Перевіримо, що } |J(y_1, y_2)| = \frac{1}{|J(x_1, x_2)|}.$$

Нарешті, ми маємо

$$f_{Y_1, Y_2}(y_1, y_2) = 3y_2,$$

$0 < y_1 < 1, 0 < y_2 < 1$ і нуль в іншому випадку

7. АНАЛІЗ ВЛАСТИВОСТЕЙ МУЛЬТИВАРІАТИВНИХ КВАДРАТИЧНИХ ПЕРЕТВОРЕНЬ

Вважається, що мультиваріативні квадратичні перетворення з точки зору знаходження таємного ключа є експоненційно складними, що і є їх важливою перевагою. Приклад параметрів захищеності наведено в таблиці 5 [4, 11 – 13].

Таблиця 5

Параметри захищеності мультиваріативного перетворення

Розмір [біт]	Параметри	MQ-система [кілобайти]	Оцінка
259	$q = 128, m = n = 37$	23	< 1
569	$q = 128, m = n = 67$	134	< 1

Однією з основних вимог до ЕП є вимога, щоб процедури генерування та перевірки підпису були не складними – не вище за поліноміально складні. Тобто генерувати та перевіряти підпис потрібно швидко. Також достатньо великі відкриті ключі можна не змінювати часто, тому що це не є проблемою. Але треба звернути увагу на суттєве розширення ЕП,

навіть, якщо пропускну здатність повідомлень є високою.

Наведені в таблиці 6 параметри були взяті з ЕП Quartz (– схема підпису в європейському проєкті NESSIE). Потрібно звернути увагу на низьку швидкість розширення підпису. Проте, час генерації йде до 5 секунд (екстраполяція з кварцу).

Таблиця 6

Параметри ЕП Quartz

Геш [біти]	Параметри	Секретний ключ [кілобайти]	Публічний ключ [кілобайти]	Підпис	Перевірка	Розширення [біти]
160	$q=128$ $n=67$ $r=11$	7.8	112.3	<1	<1	237
Повідомлення [біти]	Параметри		Публічний ключ [кілобайти]	Підпис	Перевірка	Розширення
173	$q=2$ $n=173$ $r=10$		310.2	5,000	<5	10

Таблиця 7

Параметри для безпечної зміни ЕП Quartz.

Параметри	Секретний ключ [кілобайти]	Публічний ключ [кілобайти]	Підпис	Перевірка	Підпис
q=2 n=107 r=7	3	71	10,000	<1	128

У цілому наведені в таблицях 1 – 7 дані дозволяють зробити попередні висновки, що мультіваріативні криптоперетворення можуть розглядатися як кандидати на постквантові криптографічні перетворення типу ЕП.

8. ЗАГАЛЬНІ ВИМОГИ ДО ЕП

Однією із важливих проблемних задач, які

потрібно вирішити на першому етапі розробки методів побудовання постквантових алгоритмів ЕП, є обґрунтування загальних та спеціальних вимог до них. У цьому напрямі отримано ряд результатів. Серед них необхідно відмітити [4,11 – 14]. В таблицях 8 та 9 наведено вимоги, що висунуті NIST США [14] та ETSI “ЕС [12-13].

Таблиця 8

Вимоги з безпеки NIST США до постквантових ЕП

Модель безпеки для цифрового підпису	Модель безпеки EUF-СМА. Умови безпеки: доступ зломисника менше, ніж до 2^{64} обраних повідомлень.
Вимоги до стійкості	1) 128 біт класичної безпеки / 64 біт квантової захищеності (запас стійкості AES-128) 2) 128 біт класичної безпеки / 80 біт квантової захищеності (запас стійкості SHA-256/ SHA3-256) 3) 192 біт класичної безпеки / 96 біт квантової захищеності (запас стійкості AES-192) 4) 192 біт класичної безпеки / 128 біт квантової захищеності (запас стійкості SHA-384/ SHA3-384) 5) 256 біт класичної безпеки / 128 біт квантової захищеності (запас стійкості AES-256)
Додаткові властивості безпеки	«perfect forward secrecy». (удосконалена випереджаюча безпека). Стійкість до атак сторонніми каналами. Стійкість до мультиключових атак. Стійкість до відмов.
Інші вимоги	Прозорі математичні розв’язання. Обґрунтованість стійкості

Таблиця 9

Вимоги з безпеки ETSI ЄС до постквантових ЕП

Вимоги безпеки:	
–	Проходження громадського контролю та визнання науковим співтовариством. <ul style="list-style-type: none"> – Надійне підтвердження стійкості. – Актуальність моделі безпеки. – Висока складність можливих атак.
–	Можливість використання в безпечному протоколі розподілу ключів.
–	Можливість поєднання кількох функцій безпеки (наприклад, встановлення ключів і схеми автентифікації).
–	Зручність кількісної оцінки заявлених класичних і квантових рівнів безпеки.
–	Визначеність рекомендованих ключових розмірів для заданого рівня безпеки (наприклад, 80-біт, 112 біт, 128 біт або 256 біт).
Класична безпека	Стійкість проти класичних атак.
Квантова безпека	Стійкість проти «квантових» атак. Зокрема, стійкість до алгоритму Гровера (подвоєння розміру ключа).

Доказова безпека	Базування на задачах, які мають високу складність обчислення. Можливе ігнорування зниження рівня складності, за умови, що практична стійкість не зміниться.
Довгострокова безпека	Можливість використання у протоколів типу TLS 1.3 з підтримкою forward secure cipher suites.
Активна безпека	Стійкість проти атак з адаптивним підбором.
Ефективність	Використання рекомендованих параметрів розмірів для заданого рівня безпеки. Незалежність швидкодії та кількості раундів перетворень від платформи реалізації. Швидкість генерації ключів і часу, необхідного для поширення нового ключа. Інші практичні вимоги (наприклад, стійкість до відмов).

Крім наведених вимог з криптографічної стійкості висунуті також техніко-економічні та техніко-експлуатаційні вимоги [14]. Очевидно в найближчі 3 – 4 роки вони будуть покладені в основу під час розробки постквантових стандартів ЕП.

ВИСНОВКИ

1. У зв'язку з можливістю появи квантового комп'ютера актуальними є завдання створення постквантових алгоритмів ЕП. У цьому напрямі уже розпочаті дослідження, певною мірою визначено математичні основи, на яких можуть бути побудовані постквантові алгоритми ЕП.

2. Реалізація квантово-захищених алгоритмів вимагає великих матеріально-технічних ресурсів. Вказане пов'язане з великими довжинами ключів і загальних параметрів. Сучасний рівень розвитку техніки дозволяє оптимістично ставитися до можливості ефективної реалізації квантово-захищених алгоритмів.

3. Мультиваріативні квадратичні перетворення можуть бути застосованими у стандарті ЕП. Вони були використані для побудови схем підпису, але всі спроби побудувати надійну схему шифрування не увінчалися успіхом.

4. Попередній аналіз показав, що мультиваріативні квадратичні перетворення можуть вирішити проблему захищеності від атак на основі квантових комп'ютерів, але для цього ще потрібно провести величезний обсяг досліджень та робіт, а також вкласти значні ресурси.

5. Попередній аналіз показує, що розміри загальних параметрів та ключів не викликають сумнівів відносно криптографічної стійкості стандарту, розробленого на основі мультиваріативного квадратичного перетворення. Але залишається проблема просторової складності, яка пов'язана зі значними довжинами загальних параметрів та ключів.

Література

[1] A riddle wrapped in an enigma Neal koblitz and Alfred j.menezes. <https://www.google.com.ua/search?q=a+riddle+wrapped+in+an+enigma+neal+koblitz+and+alfred+j.+menezes>

- [2] *Lili Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, Daniel Smith-Tone*. Report on Post – Quantum Cryptography. Nistir 8105 (draft). <https://www.google.com.ua/search?>
- [3] Інтернет-ресурс. Режим доступу <http://www.nkj.ru/archive/articles/5309/>
- [4] Інтернет-ресурс. Режим доступу <http://www.win.tue.nl/diamant/symposium05/abstracts/wolf.pdf>
- [5] *Горбенко І.Д.* Аналіз проблем криптографічного захисту інформації у пост-квантовий період та можливі шляхи їх вирішення/ Горбенко І.Д. Кузнецов О.О., Олійников Р.В., Потій О.В, Горбенко Ю.І., Ганзя Р.С., Пономар В.І. // Матеріали V-ої міжнародної науково-технічної конференції «Захист інформації і безпеки інформаційних систем». – Львів, 2016 (02-06 – 03.06). – С. 52.
- [6] *Reinier Brooker*. Constructing supersingular elliptic curves. *J. Comb. Number Theory*, (3): pp. 269–273, 2009.
- [7] *McGrew D., Curcio M.* Hash-Based Signatures draft-mcgrew-hash-sigs-00 [Електронний ресурс] / D. McGrew, M. Curcio - Режим доступа: <https://tools.ietf.org/html/draft-mcgrew-hash-sigs-00>
- [8] *Daniel J. Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, and Christine van Vredendaal*. NTRU Prime, <https://ntruprime.cr.yt.to/ntruprime-20160511.pdf>.
- [9] *D. J. Bernstein*. Grover vs. McEliece. In N. Sendrier, editor, Post-Quantum Cryptography, Third International Workshop, PQCrypto 2010, Darmstadt, Germany, May 25-28, 2010. Proceedings, volume 6061 of Lecture Notes in Computer Science, pages 73–80. Springer, 2010.
- [10] *Steven D. Galbraith*. Constructing isogenies between elliptic curves over Finite Fields. *LMS J. Comput. Math*, 2: pp. 118–138 (electronic), 1999.
- [11] *Moody D.* Post-Quantum Cryptography: NIST's Plan for the Future. The Seventh International Conference on Post-Quantum Cryptography, Japan, 2016. Режим доступу: [\[https://pqcrypto2016.jp/data/pqc2016_nist_announcement.pdf\]](https://pqcrypto2016.jp/data/pqc2016_nist_announcement.pdf).
- [12] ETSI GR QSC 001 V.1.1.1 (2016-07). Quantum-Safe Cryptography (QSC); Quantum-safe algorithmic framework
- [13] ETSI White Paper №8: Quantum safe cryptography and security. – 2015
- [14] Proposed Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process <http://csrc.nist.gov/groups/ST/post-quantum-crypto/documents/call-for-proposals-draft-aug-2016.pdf>



Гармаш Дмитро Васильович, студент факультету комп'ютерних наук, кафедри безпеки інформаційних систем і технологій Харківського національного університету імені В.Н. Каразіна. Наукові інтереси: електронний цифровий підпис, криптографічний захист інформації.



Бакликов Олександр Олександрович, інженер зі створення комплексних систем захисту інформації. Наукові інтереси: криптографічний захист інформації.



Філатова Наталія Вадимівна, студентка факультету комп'ютерних наук, кафедри безпеки інформаційних систем і технологій Харківського національного університету імені В.Н. Каразіна. Наукові інтереси: електронний цифровий підпис, криптографічний захист інформації.



Горбенко Іван Дмитрович, доктор технічних наук, професор, Харківський національний університет ім. В.Н.Каразіна, професор кафедри безпеки інформаційних систем і технологій.

УДК 003.26:004/056

Квантовые криптографические алгоритмы электронной подписи на основе мультивариативных квадратичных преобразований / Д.В. Гармаш, О.О. Бакликов, Н.В. Филатова, И.Д. Горбенко // Прикладная радиоэлектроника: науч.-техн. журнал. – 2016. – Том 15, № 3. – С. 215 – 225.

Приводятся требования к постквантовым алгоритмам асимметрических криптопреобразований. Указывается актуальность и необходимость поиска, исследования, стандартизации и применения криптографического примитива типа электронной подписи (ЭП). Рассматривается сущность и возможности применения мультивариативных квадратичных преобразований при реализации ЭП, делается предварительный анализ их свойств и приводится практический пример.

Ключевые слова: требования к постквантовым электронным подписям, электронные подписи, квантовое криптопреобразование, математические основы мультивариативных преобразований, мультивариативное квадратичное преобразование для электронной подписи.

Табл.: 09. Библиогр.: 14 назв.

UDC 003.26:004/056

Quantum cryptographic algorithms of electronic signature based on multivariate quadratic transformations / D.V. Garmash, O.O. Baklykov, N.V. Filatova, I.D.Gorbenko // Applied Radio Electronics: Sci. Journ. – 2016. – Vol. 15, № 3. – P. 215 – 225.

The paper provides the requirements for postquantum algorithms of asymmetric cryptotransformations. The urgency and need to search for, research, standardize and use the cryptographic primitive of the type of electronic signatures (ES) are indicated. The essence and possibilities of applying multivariate quadratic transformations in implementing the ES are considered, a preliminary analysis of their properties is performed and a practical example is provided.

Keywords: requirements for postquantum electronic signatures, electronic signatures, quantum cryptotransformation, mathematical foundations of multivariate transformations, multivariate quadratic transformation for electronic signature.

Tab.: 09. Ref.: 14 items.

СТАТИСТИЧЕСКАЯ МОДЕЛЬ ФУНКЦИОНИРОВАНИЯ СИСТЕМЫ ПЕРЕДАЧИ ИНФОРМАЦИИ ПРИ ИСПОЛЬЗОВАНИИ АЛГЕБРАИЧЕСКИХ МЕТОДОВ ОБРАБОТКИ ПСЕВДОСЛУЧАЙНЫХ КОДОВ

С.Г. ВЕКЛИЧ, Т.В. ЛАВРОВСКАЯ, С.Г. РАССОМАХИН

Рассматривается возможность практического применения псевдослучайных кодов в современных системах передачи информации с оценкой их возможного улучшения по параметрам удельной частотной и энергетической эффективности. Построена статистическая модель функционирования системы передачи информации, включающая этапы выбора сообщения источника, генерации псевдослучайного кода, обработки на основе алгебраического метода и декодирования по правилу максимального правдоподобия.

Ключевые слова: псевдослучайный помехоустойчивый код, алгебраические методы обработки, амплитудная модуляция, линейная конгруэнтная генерация, кодовое расстояние.

ВВЕДЕНИЕ

Постановка проблемы. Научно-технический прогресс в области телекоммуникаций в современном обществе предоставляет широкие возможности для информационного обмена. Это является мощным стимулом для развития различных информационных технологий, которые прочно входят в повседневную жизнь. Ведение бизнеса с помощью электронной коммерции, использование Blockchain, применение облачных технологий – вот далеко не полный перечень достижений ИТ области. При этом из года в год возрастает номенклатура и количество технических средств обработки и передачи информации, которые работают в сетях беспроводной связи. Это делает актуальным поиск новых решений рационального использования частотно-энергетического ресурса каналов передачи данных для построения технологий, позволяющих одновременно повысить скорость передачи данных и снизить требуемую мощность передатчиков.

В теории передачи информации известна историческая роль методологии, основанной на использовании случайно выбираемых кодов, в доказательстве фундаментальных теорем для зашумленных каналов [1, 2]. Однако, доказательства на основе случайного выбора кода обычно называются неконструктивными, поскольку до сегодняшнего дня случайные (псевдослучайные) коды (ПСК) для обеспечения помехоустойчивости и конфиденциальности процесса передачи информации не используются. Это является следствием отсутствия приемлемых по вычислительной сложности методов построения и декодирования ПСК, обеспечивающих корректирующую способность, близкую к максимальному правдоподобию. Реализация конструктивных алгоритмов построения и обработки ПСК может быть получена только при

использовании детерминированных алгоритмов генерации псевдослучайных символов кодовых слов.

Привлекательность технологий ПСК заключается в возможности создания сигнально-кодовых конструкций, которые позволяют одновременно повысить как частотную, так и энергетическую эффективность СПИ. Однако основной преградой для широкого использования ПСК в настоящее время является отсутствие не переборных методов декодирования.

Вычислительная сложность алгоритмов, основанных на вычислении евклидовых расстояний возрастает экспоненциально с увеличением длины блока кода и при практически требуемых значениях длины блока является неприемлемой. Получение простых линейных алгебраических методов декодирования наталкивается на трудности, вытекающие из нелинейности детерминированных алгоритмов генерации ПСП. Таким образом, можно утверждать, что идеи применения ПСК могут найти конструктивное воплощение в случае, если будут найдены линейные (линеаризованные) методы декодирования таких кодов.

Цель работы: иллюстрация потенциальных возможностей ПСК на основе исследования статистической модели процесса передачи данных по зашумленным каналам с использованием методов демодуляции, основанных на алгебраических методах.

ОСНОВНАЯ ЧАСТЬ

Рассмотрим обобщенную модель построения блокового псевдослучайного кода. Для получения помехоустойчивого кода поток символов источника, который необходимо передать, разбивается на блоки фиксированной длины по k - бит. При этом каждая комбинация из k двоичных символов источника трактуется, как десятичная количественная величина x_0 – порядковый номер, который определяет даль-

нейшую последовательность псевдослучайных чисел x_1, x_2, \dots, x_{n-1} , таким образом x_0 – является числом, порождающим кодовое слово z соответствующим номером. Каждому блоку из k двоичных символов источника ставится в соответствие блок из n недвоичных чисел кода, при этом величина

$$R = \frac{k}{n} \quad (1),$$

является скоростью ПСК и показывает отношение количества информационных двоичных символов сообщения к количеству недвоичных символов кода, предназначенных для передачи по каналу связи. Поскольку длина кодового слова ПСК n , фактически, может выбираться независимо от длины исходного блока двоичных символов k , то скорость кода (1) может быть как больше, так и меньше единицы.

Процесс кодирования состоит в вычислении последовательности символов кодового слова на основе известного порождающего числа x_0 . Получение $(n-1)$ чисел кодового слова на основе числа x_0 осуществляется с применением детерминированной технологии линейной конгруэнтной генерации (ЛКГ) [3]. Свойство линейности этой технологии является принципиальным, поскольку без его выполнения реализация вычислительно простых способов декодирования будет невозможной. Значения числовых символов кодовых слов вычисляются в соответствии со следующими правилами:

– $x_0 \in [0 \dots (2^k - 1)]$ – начальный символ – условный порядковый номер сообщения источника, который порождает кодовое слово;

– $x_i = (a \cdot x_{i-1} + b) \bmod m$, $i \in [1, \dots, (n-1)]$ – числа кодового слова псевдослучайного кода, которые вычисляются по рекуррентному способу, где операция $(A) \bmod m$ – означает вычисление A по модулю m ;

– a, b – константы, m – модуль вычислений;

– a, b, m – целые позитивные числа, которые удовлетворяют условия: $m = 2^k$; b и m – взаимно простые числа, при этом величины $(a-1)$ и m выбираются кратными 4-м.

При выполнении указанных выше условий полученные псевдослучайные целые числа, распределенные равномерно в диапазоне $[0 \dots (2^k - 1)]$, а их последовательность обладает максимальным значением периода повторения. Стоит отметить, что произвольное i -е число последовательности кодового слова связано с начальным порождающим числом x_0 зависимостью:

$$x_i = \left(a^i \cdot x_0 + \frac{a^i - 1}{a - 1} b \right) \bmod m, \quad i \in [1, \dots, n-1]. \quad (2)$$

Рассмотрим наиболее простой способ формирования канальной формы сигналов ПСК на физическом

уровне, основанный на методах амплитудно-фазовой модуляции. Перед процедурой модуляции кодовые слова, являющиеся векторами с десятичными целочисленными координатами, подвергаются преобразованию – центрированию и масштабированию числовых символов. Операция центрирования чисел кодовых слов относительно нуля производится для минимизации необходимой мощности передатчика, при этом исходная последовательность $X = \{x_0, x_1, \dots, x_{m-1}\}$ преобразуется в центрированную последовательность $X' = \{x'_0, x'_1, \dots, x'_{m-1}\}$, где

$$x'_i = x_i - \frac{m-1}{2}, \quad i = 0, 1, \dots, n-1. \quad (3)$$

Числа последовательности X' уже не являются целыми, представляют дискретный ряд с шагом 1 и равномерно распределены в диапазоне $\left[-\frac{m-1}{2}, \frac{m-1}{2} \right]$, абсолютная величина которого равна $\Delta = (m-1)$.

После операции центрирования, координаты полученных векторов X' нормируются в соответствии с выделенным бюджетом энергии на передачу кодового слова. Обозначим E_b – значение средней энергии, которая тратится на передачу одного двоичного символа источника. Тогда для передачи одного символа кода в канале допустимо затратить энергию передатчика, которая определяется выделенным бюджетом и скоростью кода:

$$E_{\text{симв}} = R \cdot E_b, \quad (4)$$

где R – скорость кода, определенная выражением (1).

Числа кодового слова распределены по дискретному равномерному закону, симметрично относительно нуля внутри диапазона Δ . При этом дисперсия D (средняя мощность) сигнала, передающего значение произвольного символа кодового слова определяется дисперсией равномерного распределения и может быть вычислена из выражения

$$D \approx \frac{\Delta^2}{12}. \quad (5)$$

На основании вычисления дисперсии равномерного распределения и введенного ограничения на энергию символа, появляется возможность для определения дополнительных требований к величине нормированного значения Δ^* . Использование выражений (4) и (5) дает:

$$\frac{(\Delta^*)^2}{12} \cdot \tau = R \cdot E_b, \quad \Rightarrow \quad \Delta^* = 2\sqrt{\frac{3}{\tau} \cdot R \cdot E_b}, \quad (6)$$

где τ – продолжительность передачи одного символа кода в канале, которая определяется скоростью модуляции.

Величина Δ^* используется в дальнейшем для нормирования значений псевдослучайных чисел в соответствии с бюджетом выделенной энергии. Процесс нормирования чисел центрированной относи-

тельно нуля последовательности X' состоит в вычислении элементов новой последовательности

$$Z = \{z_0, z_1, \dots, z_{m-1}\}$$

по следующей формуле:

$$z_i = x'_i \cdot \frac{\Delta^*}{\Delta}, \quad i = 0, 1, \dots, n-1. \quad (7)$$

Полученная последовательность чисел

$$Z = \{z_0, z_1, \dots, z_{m-1}\}$$

используется для построения канальной формы составного сигнала, передающего значение кодового слова ПСК. Данный сигнал может быть получен на основе гармонического ортогонального разложения Фурье длительности $n \cdot \tau$:

$$S(t) = \sqrt{\frac{2}{n}} \left\{ \sum_{i=0}^{n-1} z_i \cdot \cos \left[2\pi \left(f_n + \frac{i}{n \cdot \tau} \right) t \right] + \sum_{j=\frac{n}{2}}^{n-1} z_j \cdot \sin \left[2\pi \left(f_n + \frac{j - \frac{n}{2}}{n \cdot \tau} \right) t \right] \right\}, \quad t \in [0, n\tau]. \quad (8)$$

Здесь величина f_n обозначает низшую поднесущую частоту в спектре сигнала для передачи кодового слова; величины $\left(f_n + \frac{i}{n \cdot \tau} \right)$, $\left(f_n + \frac{j - \frac{n}{2}}{n \cdot \tau} \right)$, соответ-

ственно, $i(j)$ -е поднесущие частоты; множитель $\sqrt{\frac{2}{n}}$ введен для обеспечения сохранения выделенного бюджета энергии на передачу кодового символа (4), поскольку z_i используется, как амплитуды гармонических колебаний квадратурных компонент, продолжительностью $n \cdot \tau$.

Для примера на рис. 1 показаны отрезки комплексных огибающих сигналов при следующих параметрах псевдослучайного помехоустойчивого кода: $k=10$, $n=5$, $R=2$, $\Delta=31$, $E_b=1$ [Вт·с], $f_n = \frac{1}{5}$ [Гц],

$\tau=1$ [с]. При этом, для генерации кодовых слов по технологии ЛКГ использованы параметры:

$$m = 2^k = 1024, a = 5, b = 19.$$

Внешний вид сигналов ПСК напоминает отрезки реализаций случайных процессов. При увеличении длины блока n , спектр огибающих будет расширяться, а их вид приближаться к виду теплового шума.

Стоит отметить, что общий сигнал кодового слова псевдослучайного помехоустойчивого кода предусматривает параллельное кодирование недвоичных символов (чисел) кода, как амплитуд ортогональных на $T = n \cdot \tau$ гармонических поднесущих колебаний.

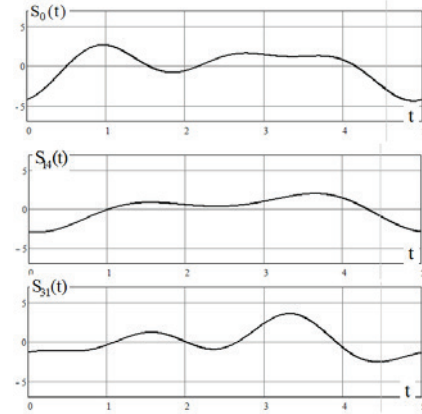


Рис. 1. Примеры комплексных огибающих составных сигналов для разных кодовых слов ПСК

При этом за время T предусматривается параллельная передача n символов кодового слова на квадратурах поднесущих частот.

Рассмотрим процесс цифровой обработки (демодуляции) сигналов кодовых слов ПСК, задаваемых моделью (8) и рис. 1. Данный процесс может быть реализован на основе алгебраических методов. Для этого необходимо произвести дискретизацию сигнала по времени. Минимально необходимое количество независимых измерений сигнала на длительности $n \cdot \tau$ совпадает с длиной кодового слова ПСК n . Моменты проведения измерений распределяются равномерно на отрезке сигнала кодового слова ПСК и определяются величиной T/v , где $v > 1$ – константа, значение которой находится из условия достижения наилучшей определенности решаемой в дальнейшем системы линейных алгебраических уравнений (СЛАУ). Моменты дискретных измерений задаются вектором

$$Td = \{td_0, \dots, td_{(n-1)q}\}, \quad (9)$$

где $q \geq 1$ – степень переопределения системы, определяемая отношением числа фактически составленных уравнений СЛАУ к их минимально необходимому количеству n . Компоненты вектора (9) вычисляются из следующего выражения:

$$td_i = \tau \cdot \frac{i}{q \cdot n} \cdot \frac{n}{v}, \quad i = 0, \dots, (n-1)q. \quad (10)$$

Тогда значение сигнала (8) в заданные моменты времени (9) можно записать в виде последовательности дискретных значений

$$S(Td) = \{s(td_0), s(td_1), \dots, s(td_{(n-1)q})\}. \quad (11)$$

Для реализации процесса демодуляции сигнала ПСК используется метод линейной алгебраической обработки сложных сигнальных конструкций. Идея данного метода заключается в определении амплитуд квадратур поднесущих частот сигнала при его известной структуре.

Для этого составляется система линейных алгебраических уравнений вида:

$$A \cdot G = B, \quad (12)$$

где A – матрица коэффициентов квадратурных компонент на интервале модуляции; B – вектор значений сигнала в цифровом представлении в каждый момент времени; G – вектор искомых значений амплитуд для заданного интервала модуляции.

В общем случае для решения системы (12) наиболее выгодным с точки зрения максимального учета информации о сигнале является решение переопределенной СЛАУ $N > n$, где $N = n \cdot q$. Для формирования переопределенной СЛАУ используются дополнительные измерения сигнала из выборки (11), содержащей большее количество уравнений при том же самом количестве неизвестных.

Размерность прямоугольной матрицы A определяется как $(N \times n)$. Система (12) при наилучшем значении v является совместной, хорошо определенной, что обеспечивает существование единственного решения. Для решения СЛАУ на произвольном i -м интервале модуляции при передаче n -символьного кодового слова ПСК следует выбрать $(n \cdot q)$ равномерно расположенных отсчетов массива измерений выборки, начиная с позиции начала наблюдения полного тактового интервала сигнала. Матрица коэффициентов при искомых значениях амплитуд квадратурных колебаний A и матрица-столбец свободных членов уравнений B формируется, с использованием максимального количества измерений на интервале модуляции длительностью T :

$$A = \| \| a_{i,j} \| \|, \quad i = 0, \dots, (n-1) \cdot q, \quad j = 0, \dots, (n-1);$$

$$a_{i,j} = \sqrt{\frac{2}{n}} \cdot \cos \left[2\pi \left(f_n + \frac{j}{n \cdot \tau} \right) t_i \right], \quad 0 \leq j \leq \left(\frac{n}{2} - 1 \right); \quad (13)$$

$$a_{i,j} = \sqrt{\frac{2}{n}} \cdot \sin \left[2\pi \left(f_n + \frac{j - \frac{n}{2}}{n \cdot \tau} \right) t_i \right], \quad \frac{n}{2} \leq j \leq (n-1).$$

Матрица-столбец свободных членов формируется в виде вектора измерений сигнала на длительности одного интервала модуляции:

$$B = S(Td), \quad b_i = s(td_i), \quad (14)$$

где $i = 0, \dots, (n-1) \cdot q$.

Система (12) имеет множество способов получения точных или приближенных решений. На практике наиболее часто используют метод на основе правила наименьших квадратов, приводящий к оценке вида:

$$G = (A^T \cdot A)^{-1} A^T \cdot B. \quad (15)$$

Решение системы (15) является приближенным, но, в условиях помеховых искажений и шума квантования при измерениях, результат получается более точным, чем при решении строгой (не переопределенной) системы (12). Помехоустойчивость решения достигается путем усреднения действия помех при

числе измерений сигнала, превышающим минимальное необходимое [4].

Дальнейшая обработка вектора G позволяет определить номер полученного сигнала. Определение номера принятого сигнала заключается в нахождении номера минимального по абсолютной величине неотрицательного элемента вектора

$$Z' = \{z'_0, z'_1, \dots, z'_{m-1}\}.$$

Элементы данного вектора поочередно вычисляются, как евклидово расстояние между наблюдаемым на выходе канала вектором и всеми возможными векторами кодовой книги ПСК:

$$z'_c = |G - z_c|, \quad c = 0, \dots, (m-1). \quad (13)$$

Критерий максимального правдоподобия при приеме кодового слова с номером i имеет вид:

$$X^* = X_i \text{ if } z'_i = \min_{i \in 0 \dots (m-1)} \{Z'\}. \quad (14)$$

Для статистической оценки помехоустойчивости систем передачи информации, использующих псевдослучайные коды рассмотренной выше конструкции, разработаны статистические модели приема сигналов в условиях отсутствия помех и в условиях зашумленного канала. Схема алгоритма работы статистической модели системы с псевдослучайным кодированием в условиях отсутствия помех (действует только шум квантования) представлена на рисунке 2.

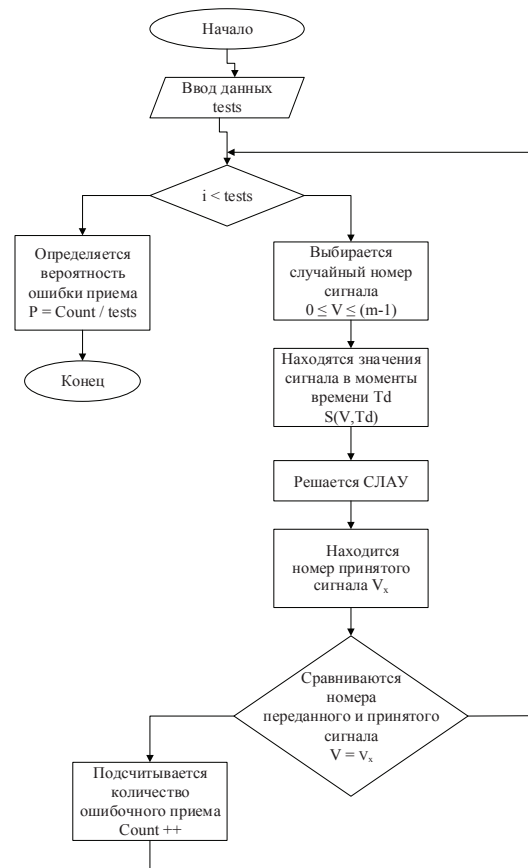


Рис. 2. Блок-схема алгоритма статистической модели обработки ПСК в условиях отсутствия канальных помех

В модели проводится серия из tests испытаний, каждое из которых предполагает случайный выбор произвольного кодового слова, формирование канальной формы составного сигнала ПСК в соответствии с правилом (8), решение СЛАУ демодуляции (13) – (15) и принятие решения по критерию (14) на основе вычисления элементов кортежа взаимных расстояний (13). По результатам испытаний производится оценка вероятности возникновения ошибок при декодировании. Данная модель предназначена для проверки робастности алгоритмов формирования и алгебраической обработки отрезков псевдослучайных последовательностей, которые соответствуют кодовым словам ПСК.

Исследование помехоустойчивости разработанных конструкций ПСК в условиях действия аддитивного гауссова шума с плоским спектром проведено на основе статистической модели, схема алгоритма которой представлена на рисунке 3.

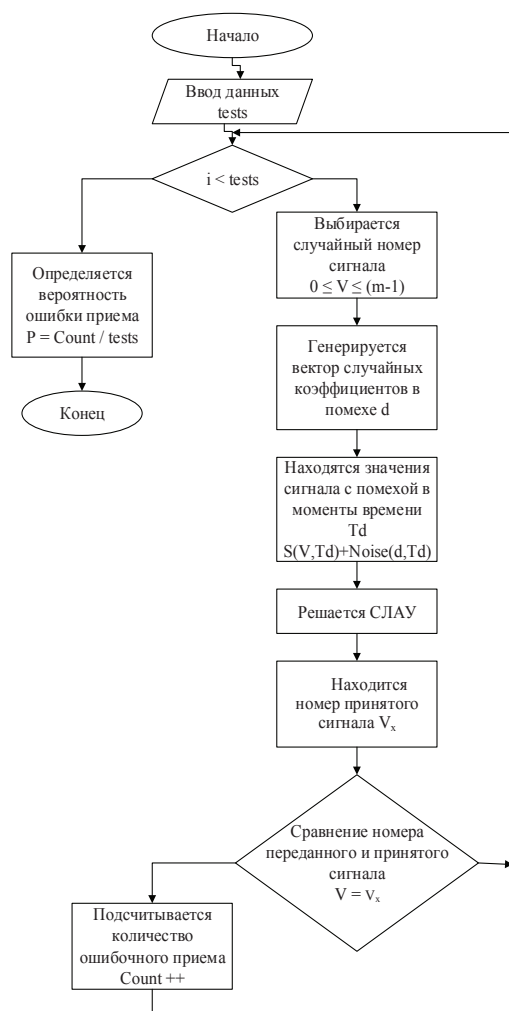


Рис. 3. Блок-схема алгоритма статистической модели для испытаний ПСК в условиях действия гауссовской аддитивной помехи

Аналитическая модель реализации помехи в данной модели имеет следующий вид:

$$\text{Noise}(t) = \sqrt{\frac{1}{n}} \left\{ \sum_{i=0}^{\frac{n-1}{2}} d_i \cdot \text{Cos} \left[2\pi \left(f_n + \frac{i}{n \cdot \tau} \right) t \right] + \sum_{j=\frac{n}{2}}^{n-1} d_j \cdot \text{Sin} \left[2\pi \left(f_n + \frac{j - \frac{n}{2}}{n \cdot \tau} \right) t \right] \right\}, \quad t \in [0, T], \quad (15)$$

где номенклатура поднесущих квадратур совпадает с соответствующей номенклатурой, использованной при построении канальной формы составного сигнала ПСК (8); d – n -мерный вектор с нормально распределенными координатами, обладающими нулевым математическим ожиданием и дисперсией N_0/n . Величина N_0 равна спектральной плотности мощности аддитивной гауссовской помехи. Использованная методика нормировки мощности сигнала (4) – (7) позволяет определить величину отношения сигнал/шум, приведенную к одному передаваемому двоичному символу сообщения источника, в виде:

$$S/N = (N_0)^{-1}. \quad (16)$$

Принимаемый сигнал в данной модели является продуктом аддитивного взаимодействия:

$$S_N(t) = S(t) + \text{Noise}(t). \quad (17)$$

Отличие алгоритма статистических испытаний в зашумленном канале (рис. 3) от рассмотренного ранее заключается в добавлении реализации помехи со структурой, определяемой (15) перед осуществлением алгебраической обработки аддитивной смеси, наблюдаемой на выходе гауссова канала. При этом в модели, кроме задания параметров ПСК скорости R и длины блока n , предусмотрена возможность регулирования величины отношения сигнал/шум путем изменения величины спектральной плотности мощности помехи N_0 .

Статистическая модель позволяет исследовать помехоустойчивость конструкций ПСК при действии помех в канале, а также оценить эффективность алгебраических методов обработки при демодуляции. Результаты статистических испытаний моделей формирования и алгебраической обработки ПСК при наличии аддитивного гауссова шума представлены на рисунках 4 – 6.

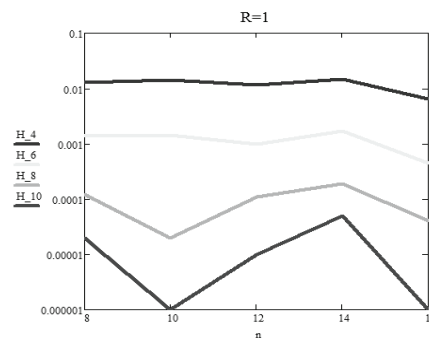


Рис. 4. Зависимости вероятности ошибки приема кодового слова ПСК от длины блока ПСК n при различных значениях отношения сигнал/шум N и R = 1

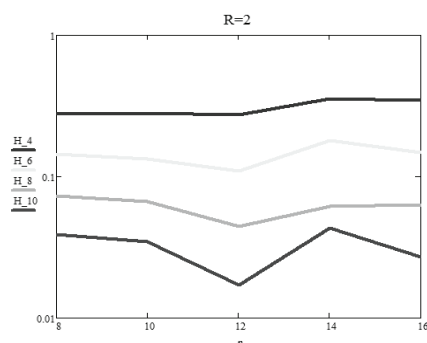


Рис. 5. Зависимости вероятности ошибки приема кодового слова ПСК от длины блока ПСК n при различных значениях отношения сигнал/шум H и $R = 2$

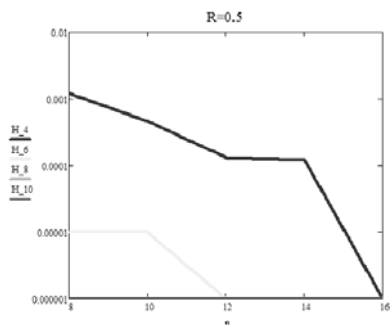


Рис. 6. Зависимости вероятности ошибки приема кодового слова ПСК от длины блока ПСК n при различных значениях отношения сигнал/шум H и $R = 0,5$

ЗАКЛЮЧЕНИЕ

Полученные результаты статистического исследования моделей функционирования СПИ при использовании алгебраических методов демодуляции и применении декодирования псевдослучайных кодов ЛКГ по правилу максимального правдоподобия показывают, что, практически, при любых скоростях кода R , не превышающих пропускную способность гауссова канала, простое увеличение длины блока кодовых слов позволяет добиться сколь угодно малой величины вероятности декодирования с ошибкой. Применение ПСК в сочетании с алгебраической демодуляцией псевдошумовых последовательностей снижает вычислительную сложность задачи обработки рассмотренных сигнально-кодовых конструкций. Это позволяет констатировать перспективность рассмотренных технологий для реализации в современных высокоскоростных системах передачи информации.

Литература

- [1] Shannon C. E., A Mathematical Theory of Communication / Shannon C. E. // Bell Syst. Tech. J., Julay-Oct. 1948. – Vol. 27. – P. 379 – 423, 623 – 656.
- [2] Shannon C.E. Communication in the presence of noise / Shannon C.E. //Proc. IRE.,Jan. 1949.– Vol. 37. – P. 10 – 21.
- [3] Лавровская, Т.В. Физическая модель псевдослучайных кодов в многомерном Евклидовом пространстве/ Т.В. Лавровская, С.Г. Рассомахин // Системи озброєння і військова техніка. – 2016. – Вып. 3 (47). – С. 79-84

- [4] Веклич С.Г. Линейная алгебраическая обработка сложных сигнальных конструкций / С.Г. Веклич, С.Г. Рассомахин // Системи обробки інформації. – 2016. – Вып. 8 (145). – С. 40 – 43



Веклич Сергей Геннадиевич, аспирант кафедры БИСТ ХНУ имени В. Н. Каразина. Научные интересы: Информационная безопасность, теория передачи информации.



Лавровская Тамила Валериевна, аспирант кафедры БИСТ ХНУ имени В. Н. Каразина. Научные интересы: Информационная безопасность, теория передачи информации.



Рассомахин Сергей Геннадиевич, доктор технического наук, доцент, заведующий кафедрой БИСТ ХНУ им. В.Н. Каразина. Научные интересы: Информационная безопасность, теория передачи информации.

УДК 621.37:621.391

Статистична модель функціонування системи передачі інформації при використанні алгебраїчних методів обробки псевдовипадкових кодів / С.Г. Веклич, Т.В. Лавровська, С.Г. Рассомахин // Прикладна радіоелектроніка: наук.-техн. журнал.– 2016. – Том 15, № 3. – С. 226 – 231.

Розглядається можливість практичного застосування псевдовипадкових кодів у сучасних системах передачі інформації з оцінкою їх, можливого, поліпшення за параметрами питомої частотної та енергетичної ефективності. Побудована статистична модель функціонування системи передачі інформації, що включає етапи вибору повідомлення джерела, генерації псевдовипадкового коду, обробки на основі алгебраїчного методу і декодування за правилом максимальної правдоподібності.

Ключові слова: псевдовипадковий завадостійкий код, алгебраїчні методи обробки, амплітудна модуляція, лінійна конгруентна генерація, кодова відстань.

Ил.: 06. Библиогр.: 04 назв.

UDC 621.37:621.391

Statistical model of functioning an information transmission system using algebraic methods of processing pseudorandom codes / S.G. Veklych, T.V. Lavrovskaya, S.G. Rassomakhin // Applied Radio Electronics: Sci. Journ. – 2016. – Vol. 15, № 3. – P. 226 – 231.

The possibility of practical application of pseudorandom codes in modern information transmission systems with assessment of their possible improving in parameters of specific frequency and energetic efficiency is considered. A statistical model of functioning an information transmission system is constructed which includes stages of choosing of the message of a source, pseudorandom code generations, processing on the basis of an algebraic method and decoding by the rule of maximum likelihood.

Keywords: interference pseudorandom code, algebraic processing methods, amplitude modulation, linear congruent generation, code distance.

Fig.: 06. Ref.: 04 items.

МЕТОД ПРОТИДІЇ АТАКАМ НА ТАБЛИЦІ МАРШРУТИЗАЦІЇ НА ОСНОВІ АРХІТЕКТУР БОТНЕТІВ ДЛЯ ОДНОРАНГОВОЇ ПІРИНГОВОЇ МЕРЕЖІ BITCOIN

П.І. СТЕЦЕНКО, Г.З. ХАЛІМОВ

Представлений метод протидії атакам на таблиці маршрутизації для однорангової пірингової мережі Bitcoin. Метод базується на архітектурах ботнетів. Клас атак на таблиці маршрутизації включає в себе атаки затемнення, ботнет-атаки та атаки інфраструктури. Метод включає в себе послідовність із десяти дій. Дії 1 – 4 рекомендується застосовувати в зв'язці, дія 5 є самостійною, дії 6 – 10 є опціональними. Розглянуто вразливості механізму адресації пірингової мережі Bitcoin. Розглянуто 4 сценарії зайняття зловмисником таблиці перевірених адрес вузла, розраховано порівняльні оцінки очікуваної кількості перезаписаних адрес. Метод дозволяє значно підвищити складність реалізації атак на таблиці маршрутизації, окремі дії підвищують необхідну кількість ресурсів для атаки вдвічі.

Ключові слова: криптовалюта Bitcoin, однорангова пірингова мережа, таблиця маршрутизації, ботнет-атака, одноранговий вузол, таблиця перевірених адрес, таблиця нових адрес.

ВСТУП

Сьогодні активно розвиваються різні додатки, в основі яких лежить концепція криптовалюти Bitcoin. Ключовими особливостями цих додатків є відкритість, децентралізація і концепція однорангової пірингової мережі. Останні кілька років проводилися дослідження безпеки протоколу Bitcoin, в той час як безпека однорангової пірингової мережі залишалася недостатньо дослідженою. Таким чином, дослідження безпеки даних мереж і розробка методу протидії, що підвищує складність реалізації атак на таблиці маршрутизації і не порушує принципи відкритості і децентралізації, є актуальною проблемою щодо широкого ряду додатків.

Аналіз публікацій

Криптовалюта Bitcoin використовує неструктуровану однорангову мережу. Замість аналізу специфічних особливостей існуючої мережі Bitcoin, ряд робіт присвячений розробці нових неструктурованих мереж, які є стійкими до візантійських атак [1, 2, 3, 4]. Внесення в чорний список некоректно функціонуючих тимчасових вузлів описується в роботі [4]. Централізоване рішення на основі інфраструктури відкритих ключів, яке не підходить для Bitcoin, представлено в роботі [2]. Проект Brahms є повністю децентралізованим, і обмежує швидкість обміну інформацією між одноранговими вузлами, що значно відрізняється від концепції Bitcoin [3]. Багато ботнетів, у тому числі і Bitcoin, використовують неструктуровані однорангові пірингові мережі і миттєвий обмін повідомленнями [5].

Визначення проблеми

У проаналізованих роботах не пропонувалися рішення щодо підвищення складності реалізації атак на таблиці маршрутизації в рамках однорангової пірингової мережі Bitcoin. Проблема розробки комплексного методу протидії такому класу атак є актуальним

завданням не тільки для криптовалюти Bitcoin, а й для ряду додатків, побудованих на аналогічній платформі.

Мета роботи

Метою даної роботи є розробка методу протидії на основі архітектур ботнетів щодо класу атак на таблиці маршрутизації, який включає в себе атаки затемнення, ботнет-атаки і атаки інфраструктури.

Для досягнення поставленої мети необхідно вирішити такі задачі:

- провести аналіз механізму зберігання мережної інформації в одноранговій піринговій мережі Bitcoin;
- навести структурне подання ботнет-атаки на однорангову пірингову мережу Bitcoin;
- розробити метод протидії щодо атак на таблиці маршрутизації.

1. МЕХАНІЗМ ЗБЕРІГАННЯ МЕРЕЖНОЇ ІНФОРМАЦІЇ

Зовнішні IP-адреси зберігаються в таблицях перевірених і нових адрес вузла. Таблиці знаходяться на диску і зберігаються, коли вузол перезавантажується.

Таблиця перевірених адрес складається з 64 блоків, кожен з яких може зберігати до 64 унікальних адрес для однорангових вузлів, з якими вузол успішно встановив вхідне чи вихідне з'єднання. Поряд з кожною збереженою адресою однорангового вузла, вузол зберігає мітку часу останнього успішного підключення до цього однорангового вузла.

Кожна адреса однорангового вузла відображається в блок у таблиці перевірених адрес, шляхом взяття геша від IP-адреси однорангового вузла і групи однорангового вузла, де група є /16 IPv4-префіксом, що містить IP-адресу однорангового вузла. Блок таблиці перевірених адрес вибирається у такий спосіб:

```
i = Hash( SK, IP ) % 4
Bucket = Hash( SK, Group, i ) % 64
return Bucket,
```


де SK – випадкове число, вибране з появою вузла;
IP – IP-адреса однорангового вузла і номер порту;
Group – група однорангового вузла.

Таким чином, кожна IP-адреса відображається в окремому блоці таблиці перевірених адрес і кожна група відображається в не більше ніж чотирьох блоках.

Адреса однорангового вузла вставляється у відповідний блок таблиці перевірених адрес, коли вузол успішно підключається до однорангового вузла. Механізм Bitcoin витискання застосовується, якщо блок заповнений (тобто містить 64 адреси). Даний механізм складається з таких етапів:

1) з блоку випадковим чином вибираються чотири адреси;

2) адреса, що має найстаршу мітку часу, замінюється адресою нового однорангового вузла в таблиці перевірених адрес;

3) вставляється до таблиці нових адрес.

Мітка часу, пов'язана з адресою однорангового вузла, оновлюється, якщо адреса однорангового вузла вже присутня в блоці. Мітка часу також оновлюється, коли активно підключений одноранговий вузол передає VERSION, ADDR, INVENTORY, GETDATA або PING повідомлення і пройшло більше 20 хвилин з моменту останнього оновлення.

Таблиця нових адрес складається з 256 блоків, кожен з яких може містити до 64 адрес для однорангових вузлів, з якими вузол ще не ініціював успішного з'єднання. Вузол наповнює таблицю нових адрес інформацією, яка отримана від DNS сідерів, або з ADDR повідомлень.

Визначення 1. ADDR повідомлення – це повідомлення адресації для отримання мережної інформації від однорангових вузлів, містить в собі до 1000 IP-адрес з їх мітками часу.

Визначення 2. DNS-сідер – це сервер, який відповідає на DNS-запити від Bitcoin-вузлів криптографічно неавтентифікованим списком IP-адрес для Bitcoin-вузлів.

Адреси в таблиці нових адрес також мають пов'язані з ними мітки часу. На адреси, отримані від DNS-сідерів, ставиться випадкова мітка часу давністю від 3 до 7 днів, у той час як на адреси, витягнуті з ADDR повідомлень, ставиться їх мітка часу з ADDR повідомлення плюс дві години.

Кожна адреса a вставляється в таблицю нових адрес, що відноситься до групи, і групи джерел, що містить IP-адреси підключеного однорангового вузла або DNS-сідера, від якого вузол дізнався адресу a . Блок вибирається в такий спосіб:

```
i = Hash( SK, Src_Group, Group ) %
32
Bucket = Hash( SK, Src_Group, i ) %
256
return Bucket,
```

де SK – випадкове число, вибране з появою вузла;

Group = / 16, що містить IP для вставки;

Src_Group = / 16, що містить IP передавального однорангового вузла.

Кожна пара (група, група джерел) гешується в один блок таблиці нових адрес. Кожен блок містить унікальні адреси. Якщо блок сповнений, то над усіма 64 адресами в блоці виконується механізм Bitcoin витискання. Якщо який-небудь з адрес має термін більше 30 днів або має занадто багато невдалих спроб підключення, то така адреса витискується на користь нової адреси. В іншому випадку Bitcoin витискання використовується з невеликою зміною – адреса, що витискається, відкидається. Окрему адресу можна відображати в кілька блоків, якщо вона оголошується декількома одноранговими вузлами.

2. БОТНЕТ-АТАКА

Визначення 3. Ботнет – взаємопов'язана мережа комп'ютерів, заражених шкідливою програмою без відома користувача і контрольованих зловмисником. Як засіб передачі керуючих команд комп'ютерам-учасникам ботнету використовується протокол прикладного рівня IRC.

Bitcoin гарантує, що вузол не зберігає занадто багато IP-адрес з однієї і тієї ж групи (тобто /16 адрес IPv4 в блоці адрес) [6].

Ботнет, атакуючий, утримує t адрес в різних групах. У роботі кожна адреса береться як гешування у рівномірно випадковий блок у таблиці перевірених адрес, тому кількість адрес, що геруються, для кожного блоку біноміально розподіляється як $B\left(t, \frac{1}{64}\right)$, де

$B(n, p)$ – біноміальний розподіл, що підраховує успіхи в послідовності з n незалежних так/ні спроб, кожна з яких приносить "так" з ймовірністю p .

Сценарії, які відображають скільки з 64×64 записів в таблиці перевірених адрес може бути зайнято зловмисником, наведені на рис. 1.

1) Від початку порожній. У кращому випадку для атакуючого всі 64 блоки з самого початку порожні. Очікувана кількість злочинних адрес, збережених у таблиці перевірених адрес, дорівнює:

$$64E \left[\min \left(64, B \left(t, \frac{1}{64} \right) \right) \right]. \quad (1)$$

2) Bitcoin витискання. Розглянемо найгірший випадок для зловмисника, коли кожен блок i заповнений 64 законними адресами. Ці адреси матимуть більш старі мітки часу, ніж всі адреси зловмисника A_i , які він намагається вставити в блок i . Слід зазначити, що адреси вузлів, які активно підключені до атакованого вузла, не обов'язково мають більш старі мітки часу.

Механізм Bitcoin витискання вимагає, щоб для кожної адреси, яка знову додається, вибиралися чоти-

ри випадкових адреси, що зберігаються в блоці, і витискувалась адреса з найстаршою міткою часу. Якщо одна з чотирьох вибраних адрес є легітимною (і вона буде старше всіх адрес зловмисника), тоді легітимна адреса буде перезаписана адресою атакуючого.

Нехай Y_a для $a = 0 \dots A_i$ – кількість злочинних адрес, які зберігаються в блоці i , враховуючи, що зловмисник вставив a унікальних адрес у блок i . Нехай $X_a = 1$, якщо a -й вставлена адреса успішно перезаписала легітимну адресу та $X_a = 0$ в іншому випадку. Тоді

$$E[X_a | Y_{a-1}] = 1 - \left(\frac{Y_{a-1}}{64}\right)^4$$

З цього маємо:

$$E[X_a | Y_{a-1}] = Y_{a-1} + 1 - \left(\frac{Y_{a-1}}{64}\right)^4 \quad (2)$$

$$E[Y_i] = 1 \quad (3)$$

де вираз (3) випливає з того, що блок від початку заповнений легітимними адресами. Тепер маємо рекурентне співвідношення для $E[Y_a]$, яке можна вирішити чисельно. Можна знайти, що $E[Y_a] > 63$ при $a \geq 101$. Таким чином, зловмисник може перезаписувати 63 з 64 легітимних адрес в блоці після вставки 101 унікальної адреси. Таким чином, очікувана кількість злочинних адрес у всіх блоках обчислюється таким чином:

$$64 \sum_{a=1}^t E[Y_a] \Pr\left[B\left(t, \frac{1}{64}\right) = a\right]. \quad (4)$$

3) Випадкове витискання. Розглядається найгірший випадок для атакуючого, коли кожен блок заповнений легітимними адресами, але тепер передбачається, що кожна вставлена адреса витискає випадково вибрану адресу. Слід зазначити, що механізм, закладений у Bitcoin, працює за іншим принципом, а даний варіант аналізується для порівняння.

Лема. Якщо k елементів випадковим чином та незалежно один від одного вставлені в n блоків таблиці, а X є випадковою змінною, що підраховує кількість непустих блоків, то маємо:

$$E[X] = n \left(1 - \left(\frac{n-1}{n}\right)^k\right) \approx n \left(1 - e^{-\frac{k}{n}}\right) \quad (5)$$

Таким чином, вираз (5) випливає з $(n-1)/n \approx e^{-1/n}$ для $n \gg 1$.

Застосовуючи лему, знаходимо очікувану кількість злочинних адрес в усіх блоках:

$$4096 \left(1 - \left(\frac{4095}{4096}\right)^t\right), \quad (6)$$

де t – кількість адрес, що утримує атакуючий в різних групах;

4096 – це кількість всіх записів у таблиці перевірених адрес 64 блоки \times 64 адреси в кожному.

4) Використання декількох раундів. Атака на таблицю маршрутизації проходить в кілька раундів; в кожному раунді атакуючий неодноразово вставляє кожну зі своїх t адрес в таблицю перевірених адрес. Водночас як кожна адреса завжди відображається в тому ж блоці в таблиці перевірених адрес у кожному раунді, Bitcoin витискання відображає кожну адресу в інший слот у цьому ж блоці в кожному раунді. Таким чином, зловмисна адреса, яка не зберігається в таблиці перевірених адрес в кінці одного раунду, все ще може бути успішно збережена в цьому блоці в наступному раунді.

До сих пір розглядався тільки один раунд. Однак проведення атаки в кілька раундів дозволить зберегти в таблицю перевірених адрес більшу кількість адрес. Після достатньої кількості раундів, очікуване число адрес задається виразом (1), тобто атака виконується як при кращому випадку для атакуючого.

Аналіз ресурсів для запуску ботнет-атаки. Ботнет-атака, проведена в кілька раундів, може дозволити зловмиснику повністю заповнити таблицю перевірених адрес, що складається з близько 6000 адрес («від початку порожня» лінія на рис. 1).

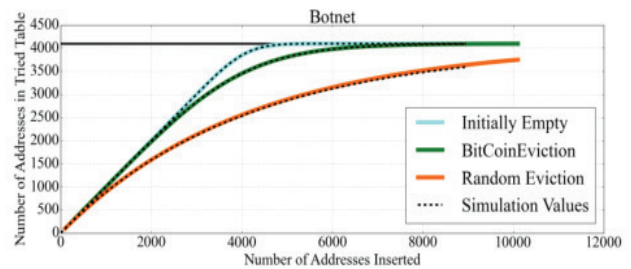


Рис. 1. Графік відношення очікуваної кількості адрес, збережених в таблиці перевірених адрес, для різних сценаріїв до кількості адрес (ботів) t

В ході побудови графіка значення обчислювалися з виразів (1), (4), (6). З графіка видно, що з кількістю ботів до 1000, ступінь заповнення таблиці перевірених адрес однакова для всіх трьох сценаріїв. Далі у випадку сценарію, при якому таблиця перевірених адрес від початку не заповнена, в ході використання менше 5000 ботів зловмисник повністю заповнює таблицю. Очевидно, що це найбільш сприятливий сценарій для зловмисника. При сценарії Bitcoin витискання повне заповнення таблиці перевірених адрес досягається зловмисником в ході використання близько 7000 ботів, а застосування випадкового витискання не дозволяє повністю заповнити таблиці в ході використання 10000 ботів. Таким чином, використання випадкового витискання підвищує кількість ресурсів, необхідних для перезапису таблиці перевірених адрес зловмисними адресами. Це підвищує складність реалізації атаки в цілому.

Така атака не може бути з легкістю запущена з законного хмарного сервісу. Як правило, такі сервіси виділяють <20 адрес на орендаря [7, 8, 9]. Проте,

Bitcoin був атакований ботнетами такого і навіть більшого розмірів [10, 11, 12]. Наприклад, ботнет Miner володів 29000 хостами з зовнішніми IP-адресами [13]. Водночас як деякі ботнет-напади концентруються в кількох діапазонах IP-адрес [14], важливо пам'ятати, що ботнет-атака, описувана в цій роботі, вимагає не більше ≈ 6000 груп. Багато інших ботнетів вимагають набагато більше ресурсів [5]. Наприклад, ботнет Walowdas був в основному в діапазонах 58.x-100.x і 188.x-233.x [14], що створює $42 \times 2^8 + 55 \times 2^8 = 24832$ груп.

Перезапис таблиці нових адрес. Для повного розуміння дій зловмисника слід розглянути як він повинен відправляти ADDR повідомлення, які перезапишуть таблицю нових адрес "сміттям" із IP-адрес. Нехай використовуватимуть "сміття" з нерозподіленого блоку адрес класу А IPv4 252.0.0/8, призначеного IANA як "зарезервовані для майбутнього використання" [15]; будь-які підключення до цих адрес терпітимуть невдачу, змушуючи атакований вузол вибирати адресу з таблиці перевірених адрес. Слід нагадати, що пара (група, група джерел) визначає блок, у якому зберігається адреса в ADDR повідомленні. Таким чином, якщо атакуючий контролює вузли в s різних групах, то s – це кількість груп джерел. Передбачається, що вузли в кожній групі джерел можуть висувати ADDR повідомлення, що містять адреси з g різних груп. "Сміття", тобто 252.0.0/8 блок адрес, задає верхню межу g на рівні $2^8 = 256$. Кожна група містить a різних адрес.

Оцінка ботнет-атаки. У ботнет-атаці кожен з вузлів атакуючого t знаходиться в окремій групі джерел. Для $s = t > 200$, що має місце для всіх атак ботнетів, застосовуючи лему, отримуємо, що очікувана кількість заповнених блоків таблиці за допомогою s груп джерел становить:

$$E[N] = 256 \left(2 - \left(\frac{255}{256} \right)^{32s} \right). \quad (7)$$

Вираз (7) показує, що кількість груп джерел $s = t$ необмежена. Таким чином, потрібно, щоб кожен одноранговий вузол відправляв одне ADDR повідомлення, що містить 1000 адрес з 250 окремими групами з чотирьох адрес кожна. Оскільки $s = t$, таке велике, можна змодельовати це, припускаючи, що кожна пара (група, група джерел) вибирає блок у таблиці нових адрес рівномірно випадковим чином і вставляє 4 адреси в той блок. Таким чином, очікувана кількість адрес, введених у блок, приблизно складатиме:

$$4 \times E \left[B \left(250t, \frac{1}{256} \right) \right] = 3.9t.$$

При $t > 200$ очікується, принаймні, 780 адрес,

вставлених у кожен блок. З виразів (2) і (3), отримуємо $E[Y_{780}] \approx 64$, таким чином, кожен блок таблиці нових адрес, ймовірно, буде повним.

3. МЕТОД ПРОТИДІЇ НА ОСНОВІ АРХІТЕКТУР БОТНЕТІВ

Узагальнюючи вище сказане, атакуючий з достатньою кількістю IP-адрес і часу може успішно провести атаку на таблиці маршрутизації будь-якого цільового вузла, незалежно від стану таблиць перевірених і нових адрес вузла, що атакується. В даному розділі представлено метод протидії, який робить атаки на таблиці маршрутизації більш складними в реалізації. Даний метод протидії заснований на архітектурах ботнетів, і розроблений так, щоб не конфліктувати з архітектурою мережі Bitcoin і не порушувати принципи відкритості та децентралізації.

Метод включає в себе наступну послідовність дій. Дії 1 – 4 рекомендується застосовувати в зв'язці, дії 5 є самостійною, дії 6 – 10 є опціональними.

Перші п'ять дій гарантують, що:

- якщо атакований вузол має h легітимних адрес у таблиці перевірених адрес до початку атаки, та p -частка з них приймає вхідні з'єднання під час атаки, коли атакований вузол перезавантажується, то навіть зловмисник з необмеженою кількістю адрес не зможе успішно провести атаку на таблиці маршрутизації по відношенню до такого вузла з ймовірністю, що перевищує:

$$Pr = f^8 < \left(1 - \frac{p \times h}{64 \times 64} \right)^8; \quad (8)$$

- якщо вихідне з'єднання атакованого вузла з найстаршою міткою часу спрямоване до легітимного однорангового вузла до початку проведення атаки, то вона завершується невдачею, якщо той одноранговий вузол приймає вхідні з'єднання, коли атакований вузол перезавантажується.

1) Детерміноване випадкове витискання. Замінити Bitcoin витискання таким чином: аналогічно тому, як кожна адреса детерміновано гешується в окремий блок у таблицях перевірених і нових адрес, додатково детерміновано гешувати адресу в окремий слот в цьому блоці. Таким чином, зловмисник не зможе збільшити кількість адрес, що зберігаються, шляхом повторної вставки однієї і тієї самої адреси в декількох раундах (п. 4, розд. 2). Завдяки цьому зменшується кількість адрес для атаки, що зберігаються в таблиці перевірених адрес.

2) Випадковий вибір адрес. Розглянутий клас атак також використовує значний ухил у бік формування вихідних з'єднань до адрес зі свіжими мітками часу. Таким чином зловмисник, якому належить лише невелика частка адрес $f = 30\%$ в таблиці перевірених адрес атакованого вузла, може збільшити ймовірність успіху за рахунок збільшення τ_t (часу, інвестованого в атаку). Така перевага

атакуючого може бути усунена, якщо адреси вибиратимуться випадковим чином з таблиць перевірених і нових адрес. Таким чином, показник успіху 50% завжди вимагає, щоб зловмисник заповнював $\sqrt[3]{0.5} = 91.7\%$ таблиці перевірених адрес, що вимагає близько 3680 тимчасових вузлів у ботнет-атаці. Поєднуючи це з детермінованим випадковим витісненням, цифра підвищиться до 10194 ботів для ймовірності успіху 50%.

3) Тестування адреси перед витисканням.

Перед збереженням адреси в його детерміновано підбраному слоті в блоці в таблиці перевірених адрес, спочатку пропонується перевіряти наявність адреси з більш старою міткою часу, що зберігається в цьому ж слоті. Якщо така адреса є, то короткочасно спробувати підключитися до неї, і якщо з'єднання виявилось успішним, то не витискати стару адресу з таблиці перевірених адрес, і зберігати нову адресу в таблицю перевірених адрес тільки у разі збою з'єднання.

Припустимо, що існує h легітимних адрес у таблиці перевірених адрес до початку проведення атаки. Також передбачається, що кожна з h легітимних адрес у таблиці перевірених адрес діюча, тобто приймає вхідні з'єднання незалежно один від одного з ймовірністю p . За допомогою тестування перед витисканням зловмисник не зможе витиснути $p \times h$ легітимних адрес (в очікуванні) з таблиці перевірених адрес, незалежно від кількості різних адрес, які він контролює. Таким чином, навіть якщо решта таблиці перевірених адрес заповнена зловмисними адресами, ймовірність успішного проведення атак на таблиці маршрутизації по відношенню до цільового вузла обмежена виразом (8).

Це значно відрізняється від функціонуючого на даний момент протоколу, в якому зловмисники з достатньою кількістю адрес мають необмежену можливість успіху, навіть якщо таблиця перевірених адрес заповнена легітимними адресами.

4) Використання короткочасних вихідних тестових підключень. Дана дія передбачає додавання вихідного з'єднання, яке б встановлювало короткочасне тестове з'єднання з випадково обраними адресами в таблиці нових адрес. Якщо з'єднання є успішним, то адреса витискається з таблиці нових адрес і вставляється в таблицю перевірених, а в іншому випадку, адреса витискається з таблиці нових адрес.

Подібні з'єднання допоможуть вичистити сміття з таблиці нових адрес при збільшенні кількості адрес з новими мітками часу в таблиці перевірених адрес, які найімовірніше будуть онлайн, коли вузол перезавантажиться.

5) Резервні з'єднання при перезавантаженні атакованого вузла. Дана дія є самостійною по відношенню до розглянутих вище. Ґрунтуючись на результатах роботи [16], додаються два з'єднання, які зберігаються між перезавантаженнями. Таким чином,

додається резервна таблиця, що записує адреси поточних вихідних з'єднань і час першого підключення до кожної адреси. Після перезавантаження, вузол присвячує два додаткових вихідних з'єднання найстарішим резервним адресам, які приймають вхідні з'єднання. Тепер, на додаток до подолання інших дій методу, успішний зловмисник повинен також розірвати резервні з'єднання. Атаки на таблиці маршрутизації зазнають невдачі, якщо атакований вузол підключається до резервної адреси, яка не контролюється атакуючим.

Крім цих п'яти дій, можна використовувати додаткові механізми збільшення складності реалізації атак на таблиці маршрутизації.

б) Збільшення кількості блоків у таблицях адрес. Одним з найбільш очевидних способів збільшення складності реалізації атак на таблиці маршрутизації є збільшення розміру таблиць перевірених і нових адрес. Припустимо, що кількість блоків у таблиці перевірених адрес подвоєно. Отже, застосовуючи лему, можемо обчислити очікувану кількість заповнених блоків під час використання заданої кількості груп в атаці:

$$E[\Gamma] = 64 \left(1 - \left(\frac{63}{64} \right)^{4s} \right) \approx \left(1 - e^{-\frac{4s}{64}} \right), \quad (9)$$

де s – кількість груп, які використовуються зловмисником в атаці;

Γ – кількість непустих блоків у таблиці перевірених адрес, $\Gamma \in [0; 64]$.

Якщо розглядати атаки інфраструктури і ботнету, то з виразу (9) випливає, що блоки, заповнені s групами змінюються від $\left(1 - e^{-\frac{4s}{64}} \right)$ до $\left(1 - e^{-\frac{4s}{128}} \right)$.

Значення ступенів заповнення таблиці перевірених адрес для різної кількості непустих блоків і кількості груп, які використовуються зловмисником в атаці, наведені в табл. 1.

Таблиця 1

Ступінь заповнення таблиці перевірених адрес зловмисними адресами залежно від кількості груп в атаці і кількості непустих блоків

$\Gamma \backslash S$	64	128	256
1	0,061	0,031	0,016
5	0,268	0,145	0,075
10	0,465	0,268	0,145
20	0,713	0,465	0,268
50	0,956	0,790	0,542
100	0,998	0,956	0,790
500	1	1	0,999
1000	1	1	1

Таким чином, зловмиснику для атаки інфраструктури необхідно подвоїти кількість груп для того, щоб очікувано заповнити ту ж частину таблиці перевірених адрес. Аналогічно і для ботнет-атаки необхідно подвоїти кількість ботів. Важливо відзначити, однак, що цей захід протидії корисний тільки тоді, коли таблиця перевірених адрес вже містить достатню кількість легітимних адрес, щоб зловмисник мав меншу частину адрес в цій таблиці. Однак, якщо таблиця перевірених адрес у цілому порожня (або містить в більшості своїй застарілі адреси для вузлів, які більше не є діючими), то зловмисник як і раніше матиме більшу частину адрес в таблиці перевірених адрес, навіть якщо кількість блоків у таблиці перевірених адрес збільшилася. Таким чином, ця дія має супроводжуватися іншою дією (наприклад, використанням короточасних вихідних тестових підключень), що збільшує кількість легітимних адрес, що зберігаються в таблиці перевірених адрес.

7) Збільшення кількості вихідних з'єднань. 80% однорангових вузлів Bitcoin дозволяють як мінімум 40 вхідних з'єднань [17]. Таким чином, можна зробити запит вузлів, щоб зробити кілька додаткових вихідних з'єднань не ризикуючи, що мережа витратить ємність з'єднання. Деякі вузли (наприклад, шлюзи об'єднань майнінгу), як показують виміри, вже це роблять [18]. Наприклад, використовуючи дванадцять вихідних з'єднань замість восьми (на додаток до короточасних вихідних і двом резервним з'єднанням), можна зменшити ймовірність успішного проведення атаки з f^8 до f^{12} . Таким чином, для досягнення ймовірності успіху в 50% атакуючому для атаки інфраструктури тепер потрібно 46 груп, а для ботнет-атаки – 11796 ботів. Хоча це поліпшення не таке значне як дії 1 – 5, це простий спосіб підвищити складність реалізації атак на таблиці маршрутизації.

8) Заборона незапрошених ADDR повідомлень. Вузол може не приймати великі незапрошені ADDR повідомлення (з кількістю адрес більше заданої) від вхідних однорангових вузлів, і запитувати ADDR повідомлення від вихідних з'єднань тільки коли його таблиця нових адрес містить занадто мало адрес. Це запобігає флудингу таблиці нових адрес атакowanego вузла сміттям шляхом злочинних вхідних з'єднань. Ця зміна не є шкідливою, оскільки навіть в поточній мережі достатньо адрес в таблиці нових адрес [18]. Слід звернути увагу на те, що вузол запитує ADDR повідомлення при встановленні вихідного з'єднання. Одноранговий вузол відповідає n випадково вибраними адресами зі своїх таблиць перевірених і нових адрес, де $n \in [x; 2500]$, а x – це 23% адрес, що зберігаються одноранговим вузлом. Якщо кожен одноранговий вузол посилає порядку $n = 1700$ адрес, то таблиця нових адрес вже на $8n/16384 = 83\%$ заповнена до моменту, коли Bitcoin вузол завершує

установку вихідних з'єднань.

9) Підвищення різноманітності вхідних з'єднань. На даний момент вузол Bitcoin може мати всі вхідні з'єднання від однієї IP-адреси, що робить занадто легким процес монополізації вхідних з'єднань атакowanego вузла під час атаки затемнення або атак нестачі сполук [19]. Дана дія передбачає обмеження прийняття з'єднань від однієї IP-адреси.

10) Моніторинг та виявлення аномалій. Розглянутий в роботі клас атак на таблиці маршрутизації має кілька специфічних "відбитків", які роблять його помітним. Такі "відбитки" включають в себе:

- сплеск короточасних вхідних TCP-з'єднань від різних IP-адрес;
- великі ADDR повідомлення, що посилаються по таким з'єднанням;
- вміст у ADDR повідомленнях IP-адрес, що є сміттям.

Зловмисник, який раптово підключає велику кількість вузлів до мережі Bitcoin, також може бути виявлений. Таким чином, системи моніторингу та виявлення аномалій, що вишукують таку поведінку, також можуть бути корисні. Такі системи змушуватимуть зловмисника проводити атаки на низькій швидкості або витратити ресурси на перезапис таблиці нових адрес (замість того, щоб використовувати "сміття").

ЗАКЛЮЧЕННЯ

У роботі проведений аналіз і описаний механізм зберігання мережної інформації однорангової пірингової мережі Bitcoin. У ході проведеного аналізу встановлено, що найбільш уразливим місцем у механізмі адресації є зберігання мережної інформації, а саме таблиці перевірених адрес і механізм Bitcoin витискання. Таблиця перевірених адрес складається з 64 блоків, кожен з яких може зберігати до 64 унікальних адрес для однорангових вузлів, з якими вузол успішно встановив вхідне чи вихідне з'єднання. Поряд з кожною збереженою адресою однорангового вузла, вузол зберігає мітку часу останнього успішного підключення до цього однорангового вузлу.

Коли вузол успішно підключається до однорангового вузлу, адреса однорангового вузла вставляється у відповідний блок таблиці перевірених адрес. Якщо блок заповнений (тобто містить 64 адреси), то використовується Bitcoin витискання. Дане витискання дозволяє зловмисникові помістити свої адреси в таблицю перевірених адрес атакowanego вузла, що, в свою чергу, дозволяє йому отримати контроль над усіма з'єднаннями атакowanego вузла і здійснювати широкий спектр атак на протокол криптовалюти.

Поданий у роботі метод протидії атакам на таблиці маршрутизації містить у собі ряд дій, спрямованих на підвищення складності реалізації атаки для зловмисника. Цей метод базується на архітектурах

ботнетів. У разі сценарію, при якому таблиця перевірених адрес від початку не заповнена, в ході використання менше 5000 ботів зловмисник повністю заповнює таблицю. При сценарії Bitcoin витискання повне заповнення таблиці перевірених адрес досягається зловмисником в ході використання близько 7000 ботів. Показник успіху 50% завжди вимагає, щоб зловмисник заповнював таблиці перевірених адрес, що вимагає близько 3680 тимчасових вузлів у ботнет-атаці. Поєднуючи це з детермінованим випадковим витисненням, цифра підвищиться до 10194 ботів для ймовірності успіху 50%. Отримуємо підвищення складності більше, ніж на 40%. Метод, наведений у роботі, передбачає опціональне використання окремих дій. Наприклад, шоста дія змушує зловмисника для атаки інфраструктури подвоювати кількість груп для того, щоб очікувано заповнити ту ж частину таблиці перевірених адрес. Аналогічно і для атаки ботнету необхідно подвоїти кількість ботів.

Таким чином, під час розробки децентралізованих систем на базі децентралізованої платформи, яка є аналогічною платформі Bitcoin, слід звертати увагу на перший рівень архітектури – на однорангову мережу. Вразливості першого рівня архітектури збільшують кількість вразливостей і простоту їх реалізації на останньому рівні. Стосовно атак на таблиці маршрутизації слід зазначити, що основним вразливим місцем є механізм зберігання і управління мережевою інформацією.

Література:

- [1] *Anceaume E., Busnel Y., Gambs S.* On the power of the adversary to solve the node sampling problem. In Transactions on Large Scale Data and Knowledge Centered Systems XI. Springer, 2013, pp. 102-126.
- [2] *Bakker A., Van Steen M.* Puppetcast: A secure peer sampling protocol. In European Conference on Computer Network Defense (EC2ND) (2008), IEEE, pp. 3-10.
- [3] *Bortnikov E., Gurevich M., Keidar I.* Brahms: Byzantine resilient random membership sampling. Computer Networks 53, 13 (2009), pp. 2340-2359.
- [4] *Jesi G. P., Montresor A., Van Steen M.* Secure peer sampling. Computer Networks 54, 12 (2010), pp. 2086-2098.
- [5] *Roscow C., Andriess D., Werner T.* Sok: P2pwned-modeling and evaluating the resilience of peer-to-peer botnets. In IEEE Symposium on Security and Privacy (2013), IEEE, pp. 97-111.
- [6] *Nakamoto S.*: Bitcoin: A peer-to-peer electronic cash system // <https://bitcoin.org/bitcoin.pdf>. – 2008. – 9 p. – 09.09.2016.
- [7] Amazon web services elastic ip. <http://aws.amazon.com/ec2/faqs/#elastic-ip>. Accessed: 2016-06-18.
- [8] Microsoft azure ip address pricing <http://azure.microsoft.com/en-us/pricing/details/ip-addresses/>. Accessed: 2016-11-18.
- [9] Rackspace: Requesting additional ipv4 addresses for cloud servers. http://www.rackspace.com/knowledge_center/article/requesting-additional-ipv4-addresses-for-cloud-servers. Accessed: 2016-11-18.
- [10] *Johnson B., Laszka, A., Grossklags, J.* Game-theoretic analysis of ddos attacks against Bitcoin mining pools. In Financial Cryptography and Data Security. Springer, 2014, pp. 72-86.
- [11] *King L.* Bitcoin hit by 'massive' ddos attack as tensions rise. Forbes <http://www.forbes.com/sites/leoking/2014/02/12/bitcoin-hit-by-massive-ddos-attack-as-tensions-rise/>. (December 2 2015).
- [12] *Vasek M., Thornton M., Moore T.* Empirical analysis of denial-of-service attacks in the Bitcoin ecosystem. In Financial Cryptography and Data Security. Springer, 2014, P. 57 – 71.
- [13] *Plohmman D., Gerhards-Padilla E.* Case study of the miner botnet. In Cyber Conflict (CYCON), 2012 4th International Conference (2012), IEEE, pp. 1-16.
- [14] *STOCK, B., GOBEL, J., ENGELBERTH, M., FREILING, F. C., AND HOLZ, T.* Walowdac: Analysis of a peer-to-peer botnet. In European Conference on Computer Network Defense (EC2ND) (2009), IEEE, P. 13 – 20.
- [15] IANA. Iana ipv4 address space registry. <http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xhtml>, January 2015.
- [16] *Dingledine R., Hopper N., Kadianakis G.* One fast guard for life (or 9 months). In 7th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs 2014) (2014).
- [17] *Biryukov A., Pustogarov I.* Bitcoin over Tor isn't a good idea. arXiv preprint arXiv:1410.6079 (2014).
- [18] *Miller A., Litton J., Pachulski A.* Discovering Bitcoin's network topology and influential nodes. Tech. rep., University of Maryland, 2015.
- [19] *Dillon J.* Bitcoin-development mailinglist: Protecting bitcoin against network-wide dos attack. <http://sourceforge.net/bitcoin/mailman/message/31168096>, 2013. Accessed: 2016-10-11.



Стеценко Павло Ігорович, аспірант кафедри БІТ ХНУРЕ. Область наукових інтересів: захист інформації у децентралізованих системах.



Халімов Геннадій Зайдулович, доктор технічних наук, професор кафедри БІТ ХНУРЕ. Область наукових інтересів: криптографія на групах.

УДК 004.056.5

Метод противодействия атакам на таблицы маршрутизации на основе архитектур ботнетов для одноранговой пиринговой сети Bitcoin / П.И. Стеценко, Г.З. Халимов // Прикладная радиоэлектроника науч.-техн. журнал. – 2016. – Том 15, № 3. – С. 232 – 239.

Представлен метод противодействия атакам на таблицы маршрутизации для одноранговой пиринговой сети Bitcoin. Метод основан на архитектурах ботнетов. Класс атак на таблицы маршрутизации включает в себя атаки затмения, ботнет-атаки и атаки инфраструктуры. Метод включает в себя последовательность из десяти действий. Действия 1 – 4 рекомендуется применять в связке, действие 5 является самостоятельным, действия 6 – 10 являются опциональными. Рассмотрены уязвимости механизма адресации пиринговой сети Bitcoin. Рассмотрены 4 сценария занятия злоумышленником таблицы проверенных адресов узла, рассчитаны сравнительные оценки ожидаемого количества перезаписанных адресов. Метод позволяет значительно повысить сложность реализации атак на таблицы маршрутизации, отдельные действия повышают необходимое количество ресурсов для атаки вдвое.

Ключевые слова: криптовалюта Bitcoin, одноранговая пиринговая сеть, таблица маршрутизации, ботнет-атака, одноранговый узел, таблица проверенных адресов, таблица новых адресов.

Табл.: 01. Ил.: 01. Библиогр.: 19 наим.

UDC 004.056.5

Method of countering attacks on routing tables based on the botnet architectures for Bitcoin peer-to-peer network / P.I. Stetsenko, G.Z. Khalimov // Applied Radio Electronics: Sci. Journ. – 2016. – Vol. 15, №.3 – P. 232 – 239.

A method of countering attacks on routing tables for the Bitcoin peer-to-peer network is presented. The method is based on the botnet architectures. A class of attacks on the routing tables includes eclipse attacks, botnet-attacks, and infrastructure attacks. The method includes a sequence of ten operations. Operations 1-4 are recommended to be used in conjunction, operation 5 is independent, and operations 6-10 are optional. The vulnerability of Bitcoin peer-to-peer network addressing mechanism has been considered. Four various scenarios of the attacker's occupation of a table with checked addresses have been considered, the comparative assessment of the expected number of the rewritten addresses has been calculated. The method can significantly increase the complexity of the implementation of the attacks on the routing tables, certain operations raise the required number of resources to an attack by a factor of 2.

Keywords: Bitcoin cryptocurrency, peer-to-peer network, routing table, botnet-attack, peer, table with checked addresses, table with new addresses.

Tab.: 01. Fig.: 01. Ref.: 19 items.

АТАКА ІНФРАСТРУКТУРИ НА ОДНОРАНГОВУ ПІРИНГОВУ МЕРЕЖУ BITCOIN

П.І. СТЕЦЕНКО, О.О. ПЕРЕКОПСЬКИЙ, Г.З. ХАЛІМОВ

Представлена атака інфраструктури на однорангову пірингову мережу Bitcoin. Ця атака відноситься до класу атак на таблиці маршрутизації. Дозволяє зловмиснику контролювати трафік атакованого вузла. Проаналізовано три можливі теоретичні сценарії, які можуть мати місце при атаці інфраструктури: блок таблиці перевірених адрес від початку пустий, використання Bitcoin витискання та випадкового витискання. Отримані кількісні оцінки очікуваної кількості вставлених зловмисником адрес для кожного розглянутого сценарію. Встановлено, що найбільш сприятливим для зловмисника є сценарій від початку порожнього блоку. А найбільш ефективним з погляду безпеки є випадкове витискання.

Ключові слова: атака інфраструктури, криптовалюта Bitcoin, однорангова пірингова мережа, таблиця перевірених адрес, мережна інформація, одноранговий вузол.

ВСТУП

Сьогодні впроваджується широке коло децентралізованих систем у різні сфери економіки та бізнесу. У своїй більшості ці децентралізовані системи мають архітектуру, яка є аналогічною архітектурі криптовалюти Bitcoin. Як основа передачі інформації Bitcoin використовує коцепцію однорангової пірингової мережі. Взагалі архітектуру даної криптовалюти можна подати у вигляді трьох рівней: однорангова пірингова мережа, технологія Blockchain та протокол безпосередньо криптовалюти Bitcoin.

Безпека однорангової пірингової мережі, зокрема, аналіз уразливостей окремих компонентів механізму зберігання мережної інформації, залишається маловивченою, в той час як безпека самого протоколу криптовалюти Bitcoin вивчена досить широко [1 – 5]. Атака інфраструктури належить до класу атак на таблиці маршрутизації. Успішна реалізація даної атаки дозволить зловмиснику реалізовувати атаки на вищі рівні архітектури з меншою складністю. Таким чином, аналіз проведення атаки інфраструктури в одноранговій піринговій мережі Bitcoin є актуальним завданням.

1. МЕХАНІЗМ РОЗПОВСЮДЖЕННЯ ТА ЗБЕРІГАННЯ МЕРЕЖНОЇ ІНФОРМАЦІЇ

Механізм розповсюдження та зберігання мережної інформації докладно описаний у роботах [6, 7].

Розповсюдження мережної інформації. Мережна інформація поширюється по мережі Bitcoin за допомогою DNS-сідерів і ADDR повідомлень.

Розмір списку обмежений рамками DNS, тому, максимально можлива кількість IP-адрес, які можуть бути повернуті за допомогою одного запиту DNS, становить близько 4000. DNS-сідер отримує адреси шляхом періодичного збору даних мережі Bitcoin. Мережа Bitcoin має шість DNS-сідерів, які запитуються лише у двох випадках. Перший випадок – коли новий вузол вперше приєднується до мережі. Вузол намагається підключитися до DNS-сідерів для отримання списку активних IP-адрес. Другий – коли існуючий вузол перезавантажується і перепідключається

до нових однорангових вузлів. У цьому випадку DNS-сідер запитується тільки після 11 секунд з того моменту як вузол почав намагатися встановити з'єднання і має менше двох вихідних з'єднань.

Одноранговий вузол заноситься в чорний список, якщо з нього було відправлено ADDR повідомлення, що містить більше 1000 адрес. Вузли приймають незапрошені ADDR повідомлення. ADDR повідомлення запитується тільки при встановленні вихідного з'єднання з одноранговим вузлом. Одноранговий вузол відповідає на 1 – 3 ADDR повідомлень, кожне з яких містить до 1000 адрес, випадковим чином вибраних зі своїх таблиць. Одноранговий вузол відправляє в цілому n випадково вибраних адрес з таблиць перевірених і нових адрес, де n випадкове число в інтервалі $[x; 2500]$, а x – 23% від кількості адрес, що зберігаються одноранговим вузлом.

Вузли направляють ADDR повідомлення одноранговим вузлам у двох випадках. Перший випадок – кожен день вузол відправляє свою власну IP-адресу в ADDR повідомленні кожному одноранговому вузлу. Другий – коли вузол приймає ADDR повідомлення з не більше, ніж 10 адресами, повідомлення пересилається на два підключених однорангових вузла, обраних випадковим чином. Особливо, якщо ADDR повідомлення містить адреси, які не можуть бути маршрутизовані для однорангового вузла. ADDR повідомлення пересилатимуть тільки одному одноранговому вузлу, якщо, наприклад, одноранговий вузол з IPv4-адресою отримав IPv6-адресу. Для вибору цих однорангових вузлів, вузол приймає геш IP-адреси кожного підключеного однорангового вузла і секретне випадкове слово (nonce), пов'язане з днем. Вузол вибирає однорангові вузли з лексично першим і другим геш-значеннями. Кожен вузол зберігає відомий список адрес, які він відправив або дізнався від кожного з підключених до нього однорангових вузлів. Слід зазначити, що вузол ніколи не розсилає адреси за відомим списком своєму одноранговому вузлу. Це необхідно для запобігання нескінченного поширення

застарілих ADDR повідомлень. Відомі списки скидаються щодня.

Зберігання мережної інформації. Під мережною інформацією розуміють зовнішні IP-адреси. Адреси зберігаються в таблицях перевірених і нових адрес вузла. У таблиці перевірених адрес міститься 64 блоки, які, в свою чергу, можуть зберігати до 64 унікальних адрес для однорангових вузлів, з якими вузол успішно встановив вхідне чи вихідне з'єднання. Також вузол зберігає мітку часу, яка ставиться на останнє успішне підключення до даного однорангового вузла.

Слід зазначити, що кожна IP-адреса відображається в окремому блоці в таблиці перевірених адрес. В свою чергу, кожна група відображатиметься не більше, ніж в чотири блоки. Після успішного підключення вузла до однорангового вузла, адреса однорангового вузла вставлятиметься до відповідного блоку в таблиці перевірених адрес. У разі, якщо блок повністю заповнений адресами, тобто містить 64 адреси, використовуватиметься механізм Bitcoin витискання. Даний механізм передбачає такі етапи:

- вибір чотирьох адрес з блоку таблиці перевірених адрес випадковим чином;
- заміна адреси з найстаршою міткою часу адресою нового однорангового вузла в таблиці перевірених адрес;
- вставка адреси в таблицю нових адрес.

Кожна адреса a , яка вставляється в таблицю нових адрес, відноситься до групи і групи джерел. Група містить IP-адресу підключеного однорангового вузла або DNS-сідера, від якого вузол дізнався адресу a .

Кожна пара (група, група джерел) гешуватиметься в один блок в таблиці нових адрес. Кожен блок містить унікальні адреси. Механізм Bitcoin витискання застосовуватиметься над усіма 64 адресами блоку, коли блок повністю заповнений. Витискання адреси на користь нової адреси буде здійснюватись у разі, коли:

- будь-яка з адрес має мітку часу, що перевищує 30 днів;
- будь-яка з адрес має занадто багато невдалих спроб підключення.

В іншому випадку механізм Bitcoin витискання використовується з невеликою зміною – адреса, яка витискатиметься, відкинеться. У разі, якщо адреса оголошується відразу декількома одноранговими вузлами, її можна відображати в кілька блоків.

2. АТАКА ІНФРАСТРУКТУРИ

Визначення. Атака інфраструктури – атака, що відноситься до класу атак на таблиці маршрутизації, в якій атакуючий контролює кілька блоків IP-адрес і може перехопити Bitcoin трафік, який надсилається на будь-яку IP-адресу в блоці, тобто він утримує множинні набори адрес в одній і тій же групі.

Зловмисник має адреси в s різних групах джерел. Група джерел містить IP-адреси передавальних одно-

рангових вузлів. Визначається, наскільки може бути заповнена таблиця перевірених адрес зловмисником, контролюючим s груп джерел, які містять t IP-адрес/груп.

Оцінка атаки інфраструктури. В атаці інфраструктури, кількість груп джерел s обмежена, а кількість груп g необмежена. Можемо розрахувати кількість блоків, заповнених групами джерел s , з використанням леми [8]:

$$E[N] = 256 \left(1 - \left(\frac{255}{256} \right)^{32s} \right). \quad (1)$$

Таким чином, очікується заповнення ≈ 251 з 256 нових блоків із $s = 32$ групи.

Кожна пара (група, група джерел) відображає унікальний блок у таблиці нових адрес, і кожен з блоків у таблиці нових адрес може містити 64 адреси. Використовується Bitcoin витискання і передбачається, що кожний блок у таблиці нових адрес повністю заповнений легітимними адресами, які мають більш старі мітки часу, ніж усі адреси, що вставлені зловмисником за допомогою ADDR повідомлень. Оскільки всі a адрес у конкретній парі (група, група джерел) відображаються в одному блоці, то це означає, що число адрес, які насправді зберігаються в цьому блоці, задається $E[Y_a]$ за допомогою рекурентного співвідношення таких виразів:

$$E[Y_a | Y_{a-1}] = Y_{a-1} + 1 - \left(\frac{Y_{a-1}}{64} \right)^4, \quad (2)$$

$$E[Y_1] = 1. \quad (3)$$

При $a = 125$ адрес, зловмисник очікує перезапис $E[Y_a] = 63,8$ з 64 легітимних адрес у блоці. Таким чином, необхідно, щоб кожна група джерел мала 32 однорангових вузла, а кожен одноранговий вузол має відправляти ADDR повідомлення з вісьмома різними групами, що складаються з $a = 125$ адрес. Таким чином, існує $g = 32 \times 8 = 256$ груп у групі джерел, що є максимальним числом груп, доступних у блоці зі сміттям IP-адрес. Кожен одноранговий вузол передає рівно одне ADDR повідомлення з $8 \times 125 = 1000$ адресами, в цілому $256 \times 125 \times s$ різних адрес, які були надіслані усіма одноранговими вузлами. Слід зазначити, що існує 2^{24} адреси в $252.0.0.0/8$ блоці, отже всі ці адреси різні, якщо $s < 524$.

Ресурси для початку атаки інфраструктури.

Для того, щоб визначити які організації мають достатню кількість ресурсів (IP-адрес) для запуску атак інфраструктури, використовуються дані CAIDA's та AS [9, 10] з липня 2015 року, а також інформацію з бази даних RIPE [11]. Таким чином, отримані результати свідчать, що вже існувало 448 організацій з більш ніж $s = 32$ групами джерел і з, щонайменше, $t = 256$ адресами в кожній групі. Отже, якщо ці організації виділяють $\tau_t = 5$ годин на атаку з часом раунду $\tau_\alpha = 27$

хвилин, то ймовірність успіху атаки інфраструктури перевищує 80%.

Інтернет-провайдери в різних країнах утримують достатню кількість груп джерел ($s \geq 32$) для цієї мети; наприклад, Судан (Sudanese Mobile), Колумбія (ETB), ОАЕ (Etisalat), Гватемала (Telgua), Туніс (Tunisia Telecom), Саудівська Аравія (Saudi Telecom Company) і Домініка (Cable and Wireless). В США Міністерство внутрішніх справ має достатню кількість груп джерел ($s = 35$), так само як і в Південній Кореї Міністерство інформації і зв'язку має ($s = 41$) груп джерел тощо [9].

Оцінка кількості непустих блоків при заданій кількості груп джерел в атаці. Моделюється процес наповнення таблиці перевірених адрес (відповідно до розділу 1), ґрунтуючись на тому, що чотири незалежні геш-функції відображають кожну з s груп джерел в один із 64 блоків у таблиці перевірених адрес. Нехай $\Gamma \in [0, 64]$ підраховує кількість непустих блоків у таблиці перевірених адрес. Знайдемо із використанням леми очікувану кількість непустих блоків при заданій кількості груп джерел s :

$$E[\Gamma] = 64 \left(1 - \left(\frac{63}{64} \right)^{4s} \right) \approx \left(1 - e^{-\frac{4s}{16}} \right). \quad (4)$$

Виходячи з виразу (4) очевидним є заповнення 55,5 з 64 блоків з $s = 32$, і всіх інших блоків, крім блоку з $s > 67$ груп джерел. Графік залежності очікуваного числа непустих блоків у таблиці перевірених адрес і кількості груп наведено на рис. 1.

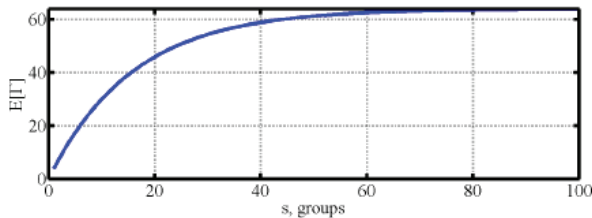


Рис. 1. Графік залежності очікуваного числа непустих блоків $E[\Gamma]$ в таблиці перевірених адрес і кількості груп джерел s

Успіх атаки інфраструктури залежить у великій мірі від τ_t (часу, витраченого на реалізацію атаки) і f (частини, що складають адреси зловмисника в таблиці перевірених адрес). Щоб визначити скільки адрес повинен контролювати зловмисник для заданого значення f , в роботі використовується ймовірнісний аналіз. Слід зазначити, що навіть якщо значення f мале, атакуючий все ще може досягти успіху шляхом збільшення τ_t . Як було сказано в розділі 1, Bitcoin гарантує, що вузол не зберігає занадто багато IP-адрес з однієї і тієї ж групи (тобто /16 адрес IPv4 в блоці адрес).

Оцінка кількості адрес зловмисника у кожному блоці таблиці перевірених адрес. Визначимо тепер у в кожному блоці таблиці перевірених адрес, за умови, що t адрес міститься в кожній групі. Для цього спочатку необхідно знайти скільки різних адрес ге-

шуються в даний блок, а потім знайти, скільки з цих адрес насправді зберігатиметься в блоці.

Кількість адрес, що гешуються в блок. Кожна група робить 4 рівномірних випадкових вибори одного з 64 можливих блоків у таблиці перевірених адрес. З огляду на окремих блок i , ймовірність того, що одна група гешується в блок i , становить:

$$P_{1/\alpha} = 1 - \left(\frac{63}{64} \right)^4 \approx \frac{1}{16}, \quad (5)$$

де α означає, що вибраний один із 64 можливих блоків.

Якщо G_i підраховує кількість різних груп, гешувальних в блок i , то G_i біноміально розподілено як $G_i \sim B(s, P_{1/\alpha})$. Кількість адрес, що гешуються в блок i , є випадковою змінною $A_i \sim B(gt, P_{1/4})$ за умови, якщо $G_i = g$ груп гешуються в блок i та кожна група містить t адрес, що гешуються в окремі блоки (кількість цих блоків може досягати чотирьох). Ця кількість адрес матиме такий розподіл:

$$\Pr[A_i = a] = \sum_{g=0}^s \Pr[B(gt, P_{1/4}) = a] \Pr[B(s, P_{1/\alpha}) = g], \quad (6)$$

де A_i – кількість адрес, що гешуються в блок;

a – кількість адрес у групі;

s – кількість груп джерел;

g – кількість груп;

t – кількість адрес зловмисника у групі.

Очікуване значення кількості адрес, що гешуються в блок i дорівнює:

$$E[A_i] = \frac{t s}{4 \alpha}.$$

Ця оцінка є заниженою. Передбачається, що кожна група гешується рівно в 4 блоки. На практиці група може відобразитися в Z блоків, де Z – випадкова величина в інтервалі $\{1, 2, 3, 4\}$. Випадкова величина

$Z - 1 \approx B\left(3, \frac{63}{64}\right)$ має біноміальний розподіл і

$E[Z] = 1 + 3 \frac{63}{64} = 3.95$. Слід зазначити, що така заниже-

на оцінка є доречною, тому що необхідно визначити скільки потрібно адрес атакуючому для заповнення блоку.

Розрахунок розподілу A_i проводився з виразу (6) та наведений на рис. 2 [7].

Розподіл має загострену форму, де перший пік відповідає $G_i = 1$, тобто, одна група гешується в блок i , другий пік відповідає $G_i = 2$ тощо. Більш того, в той час як важко побачити на графіку, існує також мала ймовірність того, що $A_i = 0$. Така ймовірність виникає, коли $G_i = 0$.

Кількість адрес, що зберігаються в блоці. Тепер, коли відомо, що A_i адрес гешуються в блок i , необхідно з'ясувати, скільки з цих адрес насправді зберігатиметься в блоці. Розглянемо можливі теоретичні сценарії, у яких: блок від початку пустий, використо-

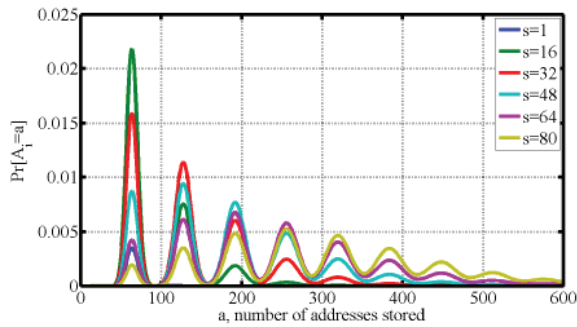


Рис. 2. Розподіл A_i для різної кількості груп s та з $t = 256$ адресами в групі

ується Bitcoin витискання та випадкове витискання.

1) Блок від початку пустий. У кращому випадку для атакуючого блок i спочатку пустий. Очікувана кількість адрес, які в кінцевому випадку зберігатимуться в блоці i , враховуючи, що, щонайменше, одна група гешується в блок i :

$$E[\min(64, A_i) | G_i > 0]. \quad (7)$$

Це величина, яка становить інтерес, тому що, якщо жодна з груп не відображається в блок i , блок не може бути заповнений за рахунок збільшення кількості адрес зловмисника в кожній групі t ; замість цього необхідно збільшувати кількість груп джерел s .

2) Bitcoin витискання. У гіршому випадку для атакуючого, припускаємо, що блок i повністю заповнений 64 легітимними адресами. Нехай Y_a – кількість адрес зловмисника, котрі дійсно зберігаються в блоці i , враховуючи, що зловмисник вставив a унікальних адрес у блок i . Якщо використовується механізм Bitcoin витискання, $E[Y_a]$ задається рекурентним співвідношенням з виразів (2), (3). Очікувана кількість адрес, збережених у блоці i , за умови, що, щонайменше, одна група гешується в блок i , дорівнює:

$$\sum_{a=0}^{64} E[Y_a] \Pr[A_i = a | G_i > 0]. \quad (8)$$

Дану кількість адрес можна обчислити кількісно шляхом об'єднання рекурсії для $E[Y_a]$ і розподілу A_i з виразу (6).

3) Випадкове витискання. Передбачається, що блок i повністю заповнений легітимними адресами, але тепер щоразу, коли адреса вставляється, її витискає випадково вибраний адрес. Якщо Y_a визначається, як зазначено вище, то в силу леми і підставляючи вираз (9) в (8), отримуємо очікуване число адрес зловмисника, збережених у блоці i , за умови, що, щонайменше, одна група гешується в блок i :

$$E[Y_a] = 64 \left(1 - \left(\frac{63}{64} \right)^a \right). \quad (9)$$

Ступінь заповнення таблиці перевірених адрес. Очікувана кількість адрес зловмисника, вставлених у таблицю перевірених адрес при атаці інфраструктури з 32 групами, для трьох теоретичних сце-

наріїв у співвідношенні з різною кількістю адрес в кожній групі наведено на рис. 3.

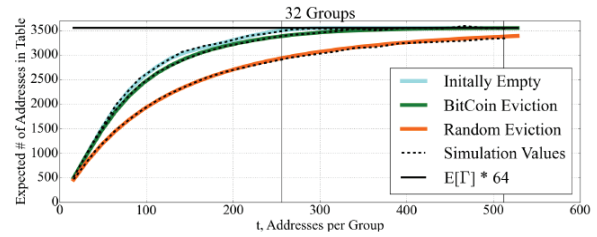


Рис. 3. Очікувані кількості вставлених адрес зловмисника для різних сценаріїв

Результати обчислені з виразів (4), (7) та (8). Горизонтальна лінія становить собою всі $E[G]$ блоки з виразу (4).

Отже, найпростішим теоретичним сценарієм для зловмисника сценарій від початку пустих блоків або коли проводиться достатня кількість раундів, одного раунду /24 блоку при $t = 256$ адрес. Цього достатньо, щоб заповнити кожен блок, використовуючи $s = 32$ груп. З 32 групами по 256 адрес кожна (8192 адреси в цілому) зловмисник може заповнити таблицю перевірених адрес на приблизно $f = 86\%$ після достатньої кількості раундів. Зловмисник майже так само ефективний в сценарії Bitcoin витискання, використовуючи тільки один раунд, але один раунд набагато менш ефективний порівняно зі сценарієм випадкового витискання.

Розмір таблиць перевірених адрес і нових адрес. У гіршому випадку для атаки таблиці перевірених і нових адрес мають бути повністю заповненими новими адресами. Таблиці нових адрес Bitcoin вузлів заповнюються досить швидко – ступінь заповнення 99% досягається протягом 48 годин. У таблиці перевірених адрес міститься невелика кількість нових адрес. Навіть після закінчення 43 днів, таблиця перевірених адрес була заповнена не більше, ніж на $300/4096 \approx 8\%$ [7]. Це пояснюється тим, що до вузлів приходять занадто мало вхідних з'єднань від публічних IP-адрес. Таким чином, більшість записів у таблиці перевірених адрес є результатом успішних вхідних з'єднань від публічних IP-адрес, узятих з таблиці нових адрес.

ВИСНОВКИ

Представлена атака інфраструктури на однорангову пірингову мережу Bitcoin. Атака відноситься до класу атак на таблиці маршрутизації, дозволяє зловмиснику контролювати трафік атакованого вузла. Головною вразливістю, що дозволяє реалізувати цю атаку, є механізм Bitcoin витискання. Механізм призначений для оновлення адрес у таблиці маршрутизації у випадку, коли вона повністю заповнена й існують нові адреси для підключень. Використовуючи механізм витискання, зловмисник має можливість вставити свої адреси у таблицю перевірених адрес цільового вузла.

У роботі розглянуто три можливі теоретичні сце-

нарії, які можуть мати місце при атаці інфраструктури: блок таблиці перевірених адрес від початку пус- тий, використання Bitcoin витискання та випадкового витискання.

Для зловмисника найпростішим сценарієм буде, коли всі блоки від початку порожні або коли атака проводиться в достатню кількість раундів. Цього до- статньо, щоб заповнити кожен блок, використовуючи $s = 32$ груп. З 32 групами по 256 адрес кожна (8192 адреси в цілому) зловмисник може заповнити табли- цю перевірених адрес на приблизно $f = 86\%$, після достатньої кількості раундів.

Сценарій випадкового витискання є більш ефек- тивним з точки зору складності реалізації атаки ін- фраструктури.

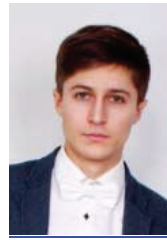
Підвищення складності реалізації атаки інфра- структури фактично полягає лише у збільшенні кіль- кості необхідних зловмиснику ресурсів для досягнен- ня прийняттого ступеня наповненості таблиці переві- рених адрес.

Література:

- [1] Courtois N. T. Bahack L. On subversive miner strategies and block with holding attack in Bitcoin digital currency / N. T. Courtois, L. Bahack. – arXiv preprint: 1402.1718. – 2014. – 29 p.
- [2] Eyal I., Sirer E. G. Majority is not enough: Bitcoin mining is vulnerable / I. Eyal, E. G. Sirer // In Financial Cryptography and Data Security. – Springer. – 2014. – P. 436 – 454.
- [3] Johnson B. Game-theoretic analysis of ddos attacks against Bitcoin mining pools / Johnson B., Laszka A., Grossklags J., Vasek M. and Moore T. // In Financial Cryptography and Data Security. – Springer. – 2014. – P. 72 – 86.
- [4] Kroll J. A., Davey I. C., and Felten E. W. The economics of Bitcoin mining, or Bitcoin in the presence of adversaries. In Proceedings of WEIS (2013). – 2013. – 32 p.
- [5] Shomer A. On the phase space of block-hiding strategies // IACR Cryptology ePrint Archive. – 2014. – 139 p.
- [6] Decker C., Wattenhofer R. Information propagation in the Bitcoin network // In IEEE Thirteenth International Conference on Peer-to-Peer Computing (P2P). – IEEE. – 2013. – P. 1 – 10.
- [7] Mille, A., Litton J., Pachulski A. Discovering bitcoin's network topology and influential nodes. Tech. rep., University of Maryland. – 2015. – 73 p.
- [8] Singh A., Ngan T.-W. J., Druschel P., Eclipse attacks on overlay networks: Threats and defenses. In IEEE INFOCOM. – 2006. – 68 p.
- [9] CAIDA. AS to Organization Mapping Dataset. – (Да- та звернення 27.10.2016).
- [10] CAIDA. Routeviews prefix to AS Mappings Dataset for IPv4 and IPv6. – (Дата звернення 27.10.2016).
- [11] RIPE. Ripestat // [Electronic resource]: <https://stat.ripe.net/data/announced-prefixes>.



Стеценко Павло Ігорович аспірант ка- федри БІТ ХНУРЕ. Область наукових інтересів: захист інформації в децентралі- зованих системах.



Перекопський Олександр Олександрович аспірант кафедри БІТ ХНУРЕ. Об- ласть наукових інтересів: методи захисту інформації.



Халімов Геннадій Зайдулович, доктор технічних наук, професор кафедри БІТ ХНУРЕ. Область наукових інтересів: криптографія на групах.

УДК 004.056.5

Атака інфраструктури на однорангову пиринго- вую сеть Bitcoin. / П.И. Стеценко, А.А. Перекопский, Г.З. Халимов // Прикладная радиоэлектроника: науч.-техн. журнал. – 2016. – Том 15, № 3 – С. 240 – 244.

Представлена атака инфраструктуры на одноранговую пиринговую сеть Bitcoin. Эта атака относится к классу атак на таблицы маршрутизации. Позволяет злоумышленнику контролировать трафик атакованного узла. Проанализиро- ваны три возможных теоретических сценария, которые мо- гут иметь место при атаке инфраструктуры: блок таблицы проверенных адресов изначально пустой, применение Bitcoin вытеснения и случайного вытеснения. Получены количественные оценки ожидаемого количества вставлен- ных злоумышленником адресов для каждого рассматри- ваемого сценария. Установлено, что наиболее благоприятным для злоумышленника сценарий изначально пустого блока. А наиболее эффективным с точки зрения безопасности явля- ется случайное вытеснение.

Ключевые слова: атака инфраструктуры, криптовалюта Bitcoin, одноранговая пиринговая сеть, таблица проверен- ных адресов, сетевая информация, одноранговый узел.

Ил.: 03. Библиогр. 11 наим.

UDC 004.056.5

Infrastructure attack on a Bitcoin peer-to-peer network. / P.I. Stetsenko, A.A. Perekopskiy, G.Z. Khalimov // Applied Radio Electronics: Sci. Journ. – 2016. – Vol. 15. № 3 – P. 240 – 244.

An infrastructure attack on a Bitcoin peer-to-peer network is presented. This attack belongs to a class of attacks on routing tables. This attack allows an attacker to control the traffic of the attacked node. Three theoretical scenarios have been considered which could take place in an infrastructure attack: a block in the table with checked addresses is initially empty, using of Bitcoin eviction and random eviction. Quantitative assessments of the expected number of embedded malicious addresses for each considered scenario have been obtained. It is found that the most favorable scenario for the attacker is the initially empty block. And the most effective in terms of security is random eviction.

Keywords: infrastructure attack, Bitcoin cryptocurrency, peer-to-peer network, table with checked addresses, network information, peer.

Fig.:03. Ref.: 11 items.

ПОИСК РЕГИСТРОВ СДВИГА С НЕЛИНЕЙНОЙ ОБРАТНОЙ СВЯЗЬЮ, ФОРМИРУЮЩИХ ПОСЛЕДОВАТЕЛЬНОСТЬ МАКСИМАЛЬНОГО ПЕРИОДА

Н.А. ПОЛУЯНЕНКО

В статье рассматривается один из важных элементов генератора поточных шифров – регистры сдвига с нелинейной обратной связью (РСНОС). Рассмотрена проблема построения РСНОС, генерирующих последовательность максимального периода (М-последовательность). Дополняется ранее предложенный подход для поиска таких регистров. Показано, что с помощью предложенного подхода возможно исключить более 99% РСНОС, которые гарантированно не будут генерировать М-последовательность и тем самым значительно повысить скорость поиска РСНОС, генерирующих М-последовательность.

Ключевые слова: регистры сдвига с нелинейной обратной связью, М-последовательность, нелинейные полиномы, поточные шифры, псевдослучайные последовательности.

ВВЕДЕНИЕ

Информационная безопасность имеет первостепенное значение в современном мире для сферы управления и защиты государства, защиты коммерческой тайны и т.д. В настоящее время, большинство информации научного, финансового, юридического характера обрабатывается и хранится на компьютерах, а также взаимодействуют с другими компьютерами через открытую или незащищенную инфраструктуру. Многие из этих данных носят конфиденциальный характер.

Для того, чтобы защитить конфиденциальную информацию от несанкционированного или случайного доступа, как правило, применяют криптографические методы. Одним из наиболее распространенных подходов является использование псевдослучайной последовательности (ПСП), с помощью которой производят шифрование информации. Зашифрованная таким образом информация может быть восстановлена в первоначальное состояние только авторизованным пользователем.

В большинстве случаев, биты ПСП генерируются с помощью регистров сдвига с линейной обратной связью (РСЛОС). Преимуществом РСЛОС является простота реализации, высокая скорость и способность генерировать последовательность со статистическими характеристиками, как и случайная последовательность [1]. Кроме того, РСЛОС часто применяют для обнаружения и коррекции ошибок [2], сжатия данных [3], тестирования [4], а также в криптографии [5].

Распространенными криптографическими алгоритмами, которые построены с использованием РСЛОС, являются: поточный шифр A5/1, который используется для обеспечения конфиденциальности в телефонной сотовой связи стандарта GSM [6], поточный шифр E0, который используется в протоколе Bluetooth [7], и сжимающий генератор [8]. Основным недостатком РСЛОС является его линейность, которая приводит к относительно простому криптоанализу [9].

В качестве альтернативы РСЛОС для генерации ПСП в поточных шифрах были предложены регистры сдвига с нелинейной обратной связью (РСНОС). РСНОС на основе поточных шифров включаются в Achterbahn [10], Dragon [11], Grain [12], Trivium [13], VEST [14]. В работах [15, 16] показано, что РСНОС более устойчивы к криптоаналитическим атакам, чем РСЛОС. Вместе с тем, построение РСНОС большого размера с гарантированным периодом, остается нерешенной проблемой [17]. Только некоторые частные случаи были рассмотрены [18, 19, 20].

На сегодняшний день, наиболее полные и подробные работы по синтезу РСНОС, которые генерируют М-последовательности, представлены в работах [21, 22, 23, 24].

В данной статье продолжается работа по ранее предложенному подходу поиска РСНОС, гарантированно генерирующих ПСП максимальной длины (М-последовательность). В основе предложенного метода лежит анализ вида обратных связей в РСНОС и их взаимного расположения. Выдвигаются требования, невыполнение которых однозначно говорит о невозможности исследуемым РСНОС генерировать М-последовательность.

Общая модель РСНОС

Общая конструкция РСНОС для регистра, состоящего из $L = 4$ ячеек, приведена на рис. 1. В регистрах используется произведение только двух ячеек, и такие РСНОС назовем РСНОС второго порядка. В дальнейшем, под РСНОС понимаем РСНОС второго порядка в $GF(2)$.

На рис. 1 введены следующие обозначения: $a_{ij} \in \{0,1\}$ – коэффициент обратной связи, соответствует наличию или отсутствию обратной связи от произведения i -й и j -й ячейки регистра; $q_i(t) \in \{0,1\}$ – значение i -го регистра в момент времени t ; Q – генерируемая последовательность бит.

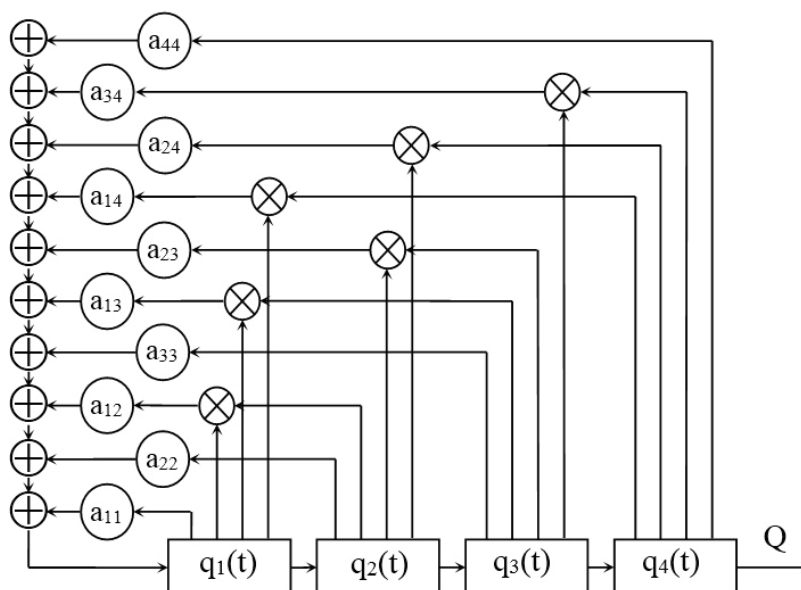


Рис. 1. Общая конструкция РСНОС

Знаком \otimes обозначена нелинейная функция умножения, а \oplus – линейная функция сложения.

Для удобства рассмотрения и восприятия обозначений коэффициентов обратной связи a_{ij} в РСНОС, отобразим их в виде матрицы:

$$\begin{matrix} a_{11} & a_{12} & a_{13} & \dots & a_{1L} \\ & a_{22} & a_{23} & \dots & a_{2L} \\ & & a_{33} & \dots & a_{3L} \\ & & & \dots & \dots \\ & & & & a_{LL} \end{matrix}$$

При этом диагональные элементы, т. е. a_{ii} , соответствуют частному случаю – РСЛОС.

В дополнение к восьми ранее изложенным Требованиям, описанным в [25, 26, 27], можно сформулировать следующее требование.

Описание Требования При рассмотрении всего множества последовательностей, которые могут генерировать РСНОС, обязательно будут присутствовать кольца с периодами меньше максимально возможного. Если исключить все кольца с периодом меньше $T_{\max} = 2^L - 1$, исключая соответствующие им комбинации a_{ij} , то останется только множество последовательностей (с соответствующими a_{ij}), которое будет являться М-последовательностью. Таким образом, для нахождения множества комбинаций a_{ij} генерирующих М-последовательность необходимо и достаточно определить и исключить все множество

комбинаций a_{ij} , генерирующих последовательность с периодами T принадлежащим интервалу $1 \leq T < T_{\max}$.

Для определенности скажем, что каждая последовательность в периоде будет начинаться с 1 и соответственно заканчиваться 0. Следующая за финальным нулем единица будет началом очередного периода. Это определение имеет одно исключение, когда $T = 1$, в этом случае последовательность в периоде будет состоять из одной 1 (и соответственно заканчиваться тоже 1).

В общем случае, при введенной системе обозначений (см. рис. 1), обратную связь для РСНОС, в момент времени t , можно задать в следующем виде:

$$q_1(t+1) = \sum_{i=1}^L a_{ii}q_i(t) + \sum_{i=1}^{L-1} \sum_{j=i+1}^L a_{ij}q_i(t)q_j(t),$$

а генерируемую при этом последовательность:

$$Q = \{q_1(t), q_1(t+1), q_1(t+2), \dots, q_1(t+i)\}.$$

Очевидно, что состояние i -й ячейки регистра в момент времени t соответствует генерируемому РСНОС значению в момент времени $t-i$. Следовательно, все состояния ячеек регистра определяется $q_i(t) = q_1(t+1-i)$.

Сгенерированная последовательность с периодом T записывается в виде:

$$Q_T = q_{i+1}, q_{i+2}, q_{i+3}, \dots, q_{i+T}, q_{i+1}, \dots$$

При использовании приведенных обозначений, состояние регистра из L ячеек, генерирующего последовательность с периодом T , можно записать как:

$$[q_{i+1}, q_{i+2}, q_{i+3}, \dots, q_{i+T}, q_{i+1}, \dots, q_{i+L}]$$

при $T < L$;

$$[q_{i+1}, q_{i+2}, q_{i+3}, \dots, q_L] \text{ при } T \geq L.$$

Таким образом, множество комбинаций состояний ячеек в регистре, или, что эквивалентно этому – генерируемой последовательности, можно разбить на подмножества состояний, соответствующих определенным периодам. В качестве примера запишем все возможные последовательности с $T < T_{\max}$ для $L = 3$ ($T_{\max} = 2^3 - 1 = 7$), сгруппировав их в соответствии с периодами:

$T = 1$	$T = 4$	$T = 5$	$T = 6$
$Q_{T=1}^1 = \underline{1111111}$	$Q_{T=4}^1 = \underline{1000100}$	$Q_{T=5}^1 = \underline{1000010}$	$Q_{T=6}^1 = 1000001$
$T = 2$ $Q_{T=2}^1 = \underline{1010101}$	$Q_{T=4}^2 = \underline{1100110}$	$Q_{T=5}^2 = \underline{1100011}$	$Q_{T=6}^2 = 1100001$
$T = 3$ $Q_{T=3}^1 = \underline{1001001}$ $Q_{T=3}^2 = \underline{1101101}$	$Q_{T=4}^3 = \underline{1010101}$ $Q_{T=4}^4 = \underline{1110111}$	$Q_{T=5}^3 = \underline{1010010}$ $Q_{T=5}^4 = \underline{1110011}$ $Q_{T=5}^5 = \underline{1001010}$ $Q_{T=5}^6 = \underline{1101011}$ $Q_{T=5}^7 = \underline{1011010}$ $Q_{T=5}^8 = \underline{1111011}$	$Q_{T=6}^3 = 1010001$ $Q_{T=6}^4 = 1110001$ $Q_{T=6}^5 = 1001001$ $Q_{T=6}^6 = 1101001$ $Q_{T=6}^7 = 1011001$ $Q_{T=6}^8 = 1111001$ $Q_{T=6}^9 = 1000101$ $Q_{T=6}^{10} = 1100101$ $Q_{T=6}^{11} = 1010101$ $Q_{T=6}^{12} = 1110101$ $Q_{T=6}^{13} = 1001101$ $Q_{T=6}^{14} = 1101101$ $Q_{T=6}^{15} = 1011101$ $Q_{T=6}^{16} = 1111101$

Условием, при котором какой-либо из периодов будет повторяться, является генерация регистром значения $q_i = q_{i+T}$ для всех $i = 1, \dots, T$. Если регистр, взятый с произвольными коэффициентами обратных связей a_{ij} , удовлетворяет вышеприведенному условию, то можно однозначно утверждать, что такой регистр генерирует кольцо с периодом T .

Задавая определенные состояния ячеек, проверяем, какое значение генерирует регистр при выбранных коэффициентах обратных связей. Проверку выполняем с помощью соответствующих шаблонов.

Обозначим шаблон как $S_T^{r/k}$, где k порядковый

номер шаблона в r -й последовательности для заданного проверяемого периода T .

Составим шаблоны каждой из итераций в последовательностях приведенного выше примера.

Для $T = 1$:

$$S_{T=1}^{1/1} = \begin{matrix} a_{11} = 1 & a_{12} = 1 & a_{13} = 1 \\ a_{22} = 1 & a_{23} = 1 \Rightarrow 1 \\ a_{33} = 1 \end{matrix}$$

Под $a_{ij} = 1$ понимаем значимый коэффициент a_{ij} , т. е. такой коэффициент, значение которого влия-

ет на формирование выходного бита. Если $a_{ij} = 0$, то указанный коэффициент не является значимым, т. е. не оказывает влияние на формирование выходного значения. Под обозначением « $\Rightarrow 1$ » понимаем, что шаблон должен сгенерировать 1 или, что эквивалентно этому, сумма всех значимых коэффициентов a_{ij} должно быть число нечетное. Если « $\Rightarrow 0$ », то шаблон должен сгенерировать 1, что эквивалентно четной сумме всех значимых коэффициентов a_{ij} .

Учитывая введенные обозначения, приведем шаблоны для $T = 2$:

$$S_{T=2}^{1/1} = \begin{matrix} 1 & 0 & 1 \\ 0 & 0 & \Rightarrow 0 \\ 1 & & \end{matrix} \quad S_{T=2}^{1/2} = \begin{matrix} 0 & 0 & 0 \\ 1 & 0 & \Rightarrow 1 \\ 0 & & \end{matrix}$$

Таким образом, если в РСНОС одновременно выполняется условие для шаблона $S_{T=2}^{1/1}$ и $S_{T=2}^{1/2}$ (т. е., четность суммы коэффициентов $a_{11} + a_{13} + a_{33}$ и нечетность коэффициента a_{22} , а учитывая, что он единственный, то, следовательно $a_{22} = 1$) является необходимым и достаточным условием для формирования РСНОС длины $L = 3$ последовательности $Q = 10101010\dots$

Шаблоны для $T = 3$ имеют вид:

$$S_{T=3}^{1/1} = \begin{matrix} 1 & 0 & 0 \\ 0 & 0 & \Rightarrow 0 \\ 0 & & \end{matrix} \quad S_{T=3}^{1/2} = \begin{matrix} 0 & 0 & 0 \\ 1 & 0 & \Rightarrow 0 \\ 0 & & \end{matrix}$$

$$S_{T=3}^{1/3} = \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & \Rightarrow 1 \\ 1 & & \end{matrix}$$

$$S_{T=3}^{2/1} = \begin{matrix} 1 & 1 & 0 \\ 1 & 0 & \Rightarrow 0 \\ 0 & & \end{matrix} \quad S_{T=3}^{2/2} = \begin{matrix} 0 & 0 & 0 \\ 1 & 1 & \Rightarrow 1 \\ 0 & & 1 \end{matrix}$$

$$S_{T=3}^{2/3} = \begin{matrix} 1 & 0 & 1 \\ 0 & 0 & \Rightarrow 1 \\ 1 & & \end{matrix}$$

Шаблоны для $T = 4$ имеют вид:

$$S_{T=4}^{1/1} = \begin{matrix} 1 & 0 & 0 \\ 0 & 0 & \Rightarrow 0 \\ 0 & & \end{matrix} \quad S_{T=4}^{1/2} = \begin{matrix} 0 & 0 & 0 \\ 1 & 0 & \Rightarrow 0 \\ 0 & & \end{matrix}$$

$$S_{T=4}^{1/3} = \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & \Rightarrow 0 \\ 1 & & \end{matrix} \quad S_{T=4}^{1/4} = \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & \Rightarrow 1 \\ 0 & & \end{matrix}$$

$$S_{T=4}^{2/1} = \begin{matrix} 1 & 1 & 0 \\ 1 & 0 & \Rightarrow 0 \\ 0 & & \end{matrix} \quad S_{T=4}^{2/2} = \begin{matrix} 0 & 0 & 0 \\ 1 & 1 & \Rightarrow 0 \\ 0 & & 1 \end{matrix}$$

$$S_{T=4}^{2/3} = \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & \Rightarrow 1 \\ 1 & & \end{matrix} \quad S_{T=4}^{2/4} = \begin{matrix} 1 & 0 & 0 \\ 0 & 0 & \Rightarrow 1 \\ 0 & & \end{matrix}$$

$$S_{T=4}^{3/1} = \begin{matrix} 1 & 0 & 1 \\ 0 & 0 & \Rightarrow 0 \\ 1 & & \end{matrix} \quad S_{T=4}^{3/2} = \begin{matrix} 0 & 0 & 0 \\ 1 & 0 & \Rightarrow 1 \\ 0 & & \end{matrix}$$

$$S_{T=4}^{3/3} = \begin{matrix} 1 & 0 & 1 \\ 0 & 0 & \Rightarrow 0 \\ 1 & & \end{matrix} \quad S_{T=4}^{3/4} = \begin{matrix} 0 & 0 & 0 \\ 1 & 0 & \Rightarrow 1 \\ 0 & & \end{matrix}$$

$$S_{T=4}^{4/1} = \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & \Rightarrow 0 \\ 1 & & \end{matrix} \quad S_{T=4}^{4/2} = \begin{matrix} 0 & 0 & 0 \\ 1 & 1 & \Rightarrow 1 \\ 0 & & 1 \end{matrix}$$

$$S_{T=4}^{4/3} = \begin{matrix} 1 & 0 & 1 \\ 0 & 0 & \Rightarrow 1 \\ 1 & & \end{matrix} \quad S_{T=4}^{4/4} = \begin{matrix} 1 & 1 & 0 \\ 1 & 0 & \Rightarrow 1 \\ 0 & & \end{matrix}$$

Заметим, что шаблоны $S_{T=4}^{1/k}$ и соответствующая им последовательность $Q = 10001000\dots$ физически не может быть реализована согласно конструкции РСНОС. Так как заполнение нулями всех регистров есть состояние запрещенное, и не может ни при каких коэффициентах a_{ij} образовать на выходе 1.

Также обратим внимание, что последовательность $Q_{T=4}^3$ и соответствующие ей шаблоны $S_{T=4}^{3/k}$ на самом деле являются суммой двух подпериодов $Q_{T=2}^1$ и соответствуют шаблонам $S_{T=2}^{1/1}$ и $S_{T=2}^{1/2}$.

Шаблоны для остальных периодов строятся аналогично вышеприведенным. Среди остальных шаблонов также присутствуют шаблоны и соответствующие им последовательности, которые либо не могут быть реализованы из-за особенности конструкции РСНОС, либо их можно представить в виде других периодов.

Можно воспользоваться альтернативным вариантом. Составить шаблоны только для последователь-

ностей максимального периода и проверим РСНОС на соответствие этим шаблонам. Однако, данный подход эквивалентен построению последовательности де Брейна и на практике, при больших значениях L , трудно реализуем.

Обобщая вышеизложенный пример, видим, что для того чтобы отсеять все возможные комбинации коэффициентов обратных связей для $L = 3$, которые будут генерировать последовательности с периодами меньшими максимально возможного, необходимо проанализировать вид РСНОС на соответствие девяти шаблонам. Или же составить и проанализировать два набора шаблонов для максимального периода.

Для больших значений L , начиная примерно с $L = 15$, работа с массивом шаблонов для $T = T_{\max}$ является задачей, трудно реализуемой для персональных компьютеров, использующих только данный метод. Причем, затрачиваемое время на проверку шаблонов, значительно превосходит время, затрачиваемое на проверку периода самой сгенерированной последовательности.

Количественная оценка применение Требования 9

Для РСНОС $L = 7$ общее количество возможных комбинаций a_{ij} составляет 268 435 455. Общее количество комбинаций a_{ij} , которое не соответствует Требованию 9 при $T = 1, \dots, L - 231\,569\,191$ (86%). Оставшиеся комбинации (то есть те, которые удовлетворяют Требованию 9 при введенных ограничениях на размер тестируемого периода) – 36 866 264 (13,7%). Дополнительно применяя Требования 1, 3, 5 и 7, описанные в [25, 26, 27], сокращает оставшееся множество до 297 454 комбинаций, что соответствует 0,11% от общего множества.

С увеличением размера тестируемого периода, число полиномов, которые не прошли Требование 9, значительно уменьшается. С увеличением размера тестируемого периода также пропорционально увеличивается число шаблонов, которое необходимо протестировать, что увеличивает время на проверку тестируемого периода. Это позволяет нам обосновать очередность проверок шаблонов: от меньшего к большему значению T .

Прочерченные шаблоны Требования 9 достаточно определить один раз для заданного L , после чего их можно применять при проверках для всех комбинаций коэффициентов a_{ij} .

Оценка затрачиваемых ресурсов

Количество возможных периодов для заданного L (обозначим через i_L) состоит из суммы всевозможных наборов периодов $T = 1, 2, \dots, L$ (обозначим число таких вариантов для отдельно взятого T через

i_T). Соответствующее для каждого периода количество шаблонов (обозначим как i_{TS}) возрастает с ростом L . Оценим верхнюю границу этого количества.

По определению, первым значением в периоде должна быть 1, а последним 0. Следовательно, эти элементы будут фиксированы, а все остальные могут принимать любые значения. Откуда получаем, что максимальное возможное число для заданного T , будет определяться соотношением:

$$i_T \leq 2^{T-2}.$$

Заметим, что период $T = 1, 2$ являются исключением. В обоих случаях возможен лишь один период, это последовательность равная $q_1 = 1$ (для $T = 1$) и $q_1 = 1, q_2 = 0$ (для $T = 2$). В результате чего, количество возможных периодов, для заданного L , может быть подсчитано следующим соотношением:

$$i_L = \sum_{k=1}^L i_{T=k}.$$

При проверке на возможность РСНОС генерировать какой-либо из тестируемых периодов, следует проверить все возможные комбинации, которые будут обеспечивать создание заданного кольца. Количество шаблонов соответствует значению проверяемого периода, то есть:

$$i_{TS} = T.$$

Таким образом, полное количество шаблонов (обозначим как S_i), которые необходимо проверять при проверке РСНОС на соответствие Требованию 9, можно определить по формуле:

$$S_i = 1_{(k=1)} + 2_{(k=2)} + \sum_{k=3}^L (k \cdot 2^{k-2})$$

При программной реализации, размещая все шаблоны в одном массиве, размерность массива будет $2^{L-2} \cdot L \cdot (n_L + 1)$, где $n_L = L \cdot (L + 1) / 2$ число различных коэффициентов a_{ij} . К значению n_L добавлена 1, т. к. для каждого шаблона необходимо запоминать какое число он должен генерировать, 1 или 0.

Как видим, размерность данного массива возрастает с ростом L по степенной зависимости и уже при $L = 20$ измеряется в Гбайтах, что является пробле-

мой при реализации на персональных компьютерах из-за ограничения оперативной памяти.

Введение различного рода оптимизаций в алгоритм позволяет существенно уменьшить объем затраченной памяти для проверки каждого периода, но не общую тенденцию роста затрачиваемых ресурсов.

Оценка временных затрат

В таблице 1 приведено время, затрачиваемое на проверку всего множества различных комбинаций a_{ij} в зависимости от числа взятых для проверки периодов и, соответственно, числа шаблонов, для различных значений L . Результат приведен без проверки самих комбинаций на генерацию М-последовательности. В таблице 2 указано время, затраченное на аналогичные тесты, но с проверкой на генерацию М-последовательностей.

Из результатов, приведенных в таблицах 1 и 2 можно сделать следующие выводы:

1. Подтверждается ранее приведенный результат, что с увеличением размера регистра увеличивается время на проверку в соответствии с Требуемым 9 по степенному закону.

2. Многие из комбинаций коэффициентов a_{ij} , для отдельно взятого периода проверяемого кольца, дают также кольца, но с меньшим периодом. Это объясняет то, что время, затраченное на проверку шаблонов только для $T = 8$ (при $L = 8$) превосходит в 1.2 раза время, затраченное на проверку шаблонов с $T = 2 - 8$ и говорит в пользу очередности проведения проверок, начиная с меньших значений T .

3. Увеличение объема тестируемых периодов не приводит к сокращению затраченного времени на поиск М-ПСНОС. При тестировании, для каждого L , существует оптимальное значение T . Для $L = 7$ оптимальный период находится в пределах $T = 2 - 5$, для $L = 8$ в пределах $T = 2 - 6$ и для $L = 9$ – при $T = 2 - 7$. С ростом числа проверяемых периодов с одной стороны уменьшается число комбинаций, которое необходимо проверить на генерацию М-последовательности. С другой стороны увеличивается число шаблонов и, соответственно, время на их проверку, что и приводит к увеличению общего затраченного времени.

Таблица 1

Тестируемые T	Затраченное время (сек.)					
	$L = 4$	$L = 5$	$L = 6$	$L = 7$	$L = 8$	$L = 9^1$
0	<0.01	<0.01	<0.01	0.2813	38.7969	10 375
2	<0.01	<0.01	<0.01	0.3750	47.5313	12 750
3	<0.01	<0.01	<0.01	0.4063	59.4531	16 600
2-3	<0.01	<0.01	<0.01	0.4375	60.1719	16 559
4	<0.01	<0.01	<0.01	0.5000	68.4844	18 150
2-4	<0.01	<0.01	<0.01	0.5469	77.3125	21 949
5		<0.01	<0.01	0.7343	103.313	28 243
2-5		<0.01	<0.01	0.8438	112.375	31 942
6			<0.01	0.9219	137.063	38 658
2-6			0.0156	1.0781	156.906	44 493
7				1.4375	202.031	61 683
2-7				1.5469	219.219	62 335
8					361.078	102 751
2-8					296.984	95 796
9						186 261
2-9						143 883

¹⁾ оценочное время, полученное прогнозированием.

Таблица 2

Тестирование вместе с генерацией гаммы и определением ее периода											
Тестируемые T	$L = 4$		$L = 5$		$L = 6$		$L = 7$		$L = 8$		$L = 9^1$
	Протестировано комбинаций	Затраченное время (сек.)	Протестировано комбинаций	Затраченное время (сек.)	Протестировано комбинаций	Затраченное время (сек.)	Протестировано комбинаций	Затраченное время (сек.)	Протестировано комбинаций	Затраченное время (сек.)	Затраченное время (сек.)
2-4	16	<0.01	208	<0.01	7 216	0.047	463 680	3.438			
2-5			184	0.016	6 039	0.047	386 444	3.250	49 915 956	701.84	343 301
2-6					984	0.047	337 093	3.484	43 618 366	675.47	322 810

Продолжение таблицы 2

2-7							297 454	3.609	38 011 514	694.98	309 319
2-8									34 169 520	763.52	320 032
2-9											345 337

¹⁾ оценочное время, полученное прогнозированием.

ЗАКЛЮЧЕНИЕ

Применение Требования 9 при анализе вида и взаимного расположения коэффициентов обратных связей a_{ij} в РСНОС позволяет, теоретически, исключить все регистры, не генерирующие M-последовательность. Объем исключаемых РСНОС ограничен объемом используемой памяти и затрачиваемым временем.

Совместное использование Требования 9 с другими Требованиями позволяет значительно сократить, затрачиваемые на вычисления ресурсы и увеличить процент отсекаемого множества РСНОС не генерирующий M-последовательность. Для $L = 7$ проверка вида a_{ij} на соответствие Требованиям 1, 2, 5, 7 и Требование 9, при ограничении тестируемого периода $T = 1, \dots, L$, позволила исключить из рассмотрения 99,89% РСНОС которые гарантированно не будут генерировать M-последовательность.

Практическая значимость представленной методики состоит в том, что она делает возможным сокращение исследуемого множества РСНОС, исключив регистры, не генерирующие M-последовательность, и тем самым повысить скорость поиска M-РСНОС.

Литература

[1] S. Golomb, Shift Register Sequences. Aegean Park Press, 1982.

[2] J. McCluskey, "High speed calculation of cyclic redundancy codes," in Proceedings of the 1999 ACM/SIGDA seventh international symposium on Field programmable gate arrays, FPGA '99, (New York, NY, USA), pp. 250–256, ACM, 1999.

[3] G. Mrugalski, J. Rajski, and J. Tyszer, "Ring generators - New devices for embedded test applications," Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 23, no. 9, pp. 1306–1320, 2004.

[4] R. David, Random Testing of Digital Circuits. New York: Marcel Dekker, 1998.

[5] S. Mukhopadhyay and P. Sarkar, "Application of LFSRs for parallel sequence generation in cryptologic algorithms," in Computational Science and Its Applications - ICCSA 2006, vol. 3982 of Lecture Notes in Computer Science, pp. 436–445, Springer Berlin / Heidelberg, 2006.

[6] E. Biham and O. Dunkelman, "Cryptanalysis of the A5/1 GSM stream cipher," in INDOCRYPT '00: Proceedings of the First International Conference on Progress in Cryptology, (London, UK), pp. 43–51, Springer-Verlag, 2000.

[7] O. Y. Shaked, "Cryptanalysis of the Bluetooth E0 cipher," citeseer.ist.psu.edu/744254.html.

[8] D. Coppersmith, H. Krawczyk, and Y. Mansour, "The shrinking generator," in CRYPTO '93: Proceedings of the

13th annual international cryptology conference on Advances in cryptology, (New York, NY, USA), pp. 22–39, Springer-Verlag New York, Inc., 1994.

[9] B. Schneier, "A self-study course in block-cipher cryptanalysis," Cryptologia, vol. XXIV, no. 1, pp. 18–33, 2000.

[10] B. Gammel, R. Gottfert, and O. Kniffner, "Achterbahn-128/80: Design and analysis," in SASC'2007: Workshop Record of The State of the Art of Stream Ciphers, pp. 152–165, 2007.

[11] K. Chen, M. Henricken, W. Millan, J. Fuller, L. Simpson, E. Dawson, H. Lee, and S. Moon, "Dragon: A fast word based stream cipher," in eSTREM, ECRYPT Stream Cipher Project, 2005. Report 2005/006.

[12] M. Hell, T. Johansson, and W. Meier, "Grain - a stream cipher for constrained environments," citeseer.ist.psu.edu/732342.html.

[13] C. D. Canniere and B. Preneel, "TRIVIUM specifications," citeseer.ist.psu.edu/734144.html.

[14] B. Gittins, H. A. Landman, S. O'Neil, and R. Kelson, "A presentation on VEST hardware performance, chip area measurements, power consumption estimates and benchmarking in relation to the aes, sha-256 and sha-512." Cryptology ePrint Archive, Report 2005/415, 2005. <http://eprint.iacr.org/>.

[15] B. Preneel, "A survey of recent developments in cryptographic algorithms for smart cards," Comput. Networks, vol. 51, no. 9, pp. 2223–2233, 2007.

[16] A. Canteaut, "Open problems related to algebraic attacks on stream ciphers," in WCC, pp. 120–134, 2005.

[17] E. Dubrova, A scalable method for constructing Galois NLFSRs with period 2^n-1 using cross-join pairs. IEEE Transactions on Information Theory. https://www.researchgate.net/profile/Elena_Dubrova/publication/267264884_A_Scalable_Method_for_Constructing_Galois_NLFSRs_with_Period_2_n_-_1_using_Cross-Join_Pairs/links/5584137808ae89172b88a75d.pdf. 2013.

[18] C. J. Jansen, Investigations On Nonlinear Streamcipher Systems: Construction and Evaluation Methods. Ph.D. Thesis, Technical University of Delft, 1989.

[19] D. Linardatos and N. Kalouptsidis, "Synthesis of minimal cost nonlinear feedback shift registers," Signal Process., vol. 82, no. 2, pp. 157–176. 2002.

[20] J. S. I. Janicka-Lipska, "Boolean feedback functions for full-length nonlinear shift registers," Telecommunications and Information Technology, vol. 5, pp. 28–29, 2004.

[21] E. Dubrova, A Method for Generating Full Cycles by a Composition of NLFSRs. Designs, Codes and Cryptography, ISSN 0925 – 1022, E – ISSN 1573 – 7586, November, Vol. 73, № 2, 469 – 486 p. 2014.

[22] E. Dubrova, A list of maximum – period NLFSRs. Cryptology ePrint Archive, Report 2012/166, 2012. <http://eprint.iacr.org/2012/166>. 2012.

[23] E. Dubrova, M. Teslenko, H. Tenhunen, On analysis and synthesis of (n,k) – non – linear feedback shift registers. in Design and Test in Europe, pp. 133–137. 2008.

- [24] T. Rachwalik, J. Szmidi, R. Wicik, J. Zablocki, Generation of Nonlinear Feedback Shift Registers with special – purpose hardware. Cryptology ePrint Archive: Report 2012/314, <http://eprint.iacr.org/2012/314>. 2012.
- [25] Потий А.В., Полуяненко Н.А. Анализ свойств регистров сдвига с нелинейной обратной связью второго порядка генерирующих, последовательность с максимальным периодом // Прикладная радиоэлектроника. – 2008, № 3. – С. 282 – 290
- [26] Потий О.В., Полуяненко М.О. Вибір утворюючих поліномів для регістра зсуву з нелінійним зворотним зв'язком другого порядку, що генерують послідовність з максимальним періодом. // COMPUTER SCIENCE AND CYBERSECURITY. – Харківський національний університет імені В.Н. Каразіна, Випуск 2(2), 2016. Електронний ресурс. Режим доступу: <http://periodicals.karazin.ua/cscs/article/view/6209/5747>
- [27] Аналіз, розробка та дослідження постквантових криптографічних примітивів та обґрунтування умов їхнього застосування в Україні: звіт про НДР (проміжний). Том 1. – Аналіз та порівняльні дослідження симетричних криптографічних перетворень на постквантовий період / ХНУ ім. В.Н. Каразіна; кер. Кузнецов О.О.; вик.: Сватовський І.І.



Полуяненко Николай Александрович, аспирант кафедри безпеки інформаційних систем і технологій Харківського національного університету ім. В. Н. Каразіна. Область наукових інтересів: криптографічна захист інформації, поточні шифри, аналіз функціонування і безпеки систем захисту інформації.

УДК 004.056.55

Пошук регістрів зсуву з нелінійним зворотним зв'язком, що формує послідовність максимального періоду / М.О. Полуяненко // Прикладна радиоелектроніка: наук.-техн. журнал. – 2016. – Том 15, №3. – С 245– 252.

У статті розглянуто один з важливих елементів генератора поточних шифрів – регістри зсуву з нелінійним зворотним зв'язком (РЗНЗЗ). Розглянуто проблему побудови РЗНЗЗ, що генерують послідовність максимального періоду (М-послідовність). Доповнюється раніше запропонований підхід до пошуку вказаних регістрів. Показано, що за допомогою запропонованого підходу можливо виключити більш ніж 99% РЗНЗЗ, які гарантовано не генерують М-послідовність, за допомогою чого значно зростає швидкість пошуку РЗНЗЗ, що генерують М-послідовність.

Ключові слова: регістри зсуву з нелінійним зворотним зв'язком, М-послідовність, нелінійні поліноми, поточні шифри, псевдо випадкові послідовності.

Табл.: 02. Іл.: 01. Біблогр.: 27 найм.

UDC 004.056.55

The searching of non-linear feedback shift registers forming a maximal length sequence / N.A. Poluyanenko // Applied Radio Electronics: Sci. Journ. – 2016. – Vol. 15, № 3. P. 245– 252.

In this paper one of the most important elements of a generator of stream ciphers – non-linear feedback shift registers (NLFSR) – is considered. The problem of constructing NLFSRs that generate a maximal length sequence (M-sequence) is considered. The previously proposed approach for searching such kind of registers is complemented. The approach, which can exclude 99% NLFSR that are not appropriate for the M-sequence, is shown. That approach greatly decreases time of searching NLFSRs that generate the M-sequence.

Keywords: non-linear feedback shift registers, M-sequence, non-linear polynoms, stream ciphers, pseudo-random sequences.

Tab.: 02. Fig.: 01. Ref.: 27 items.

МЕТОДЫ ОПЕРАТИВНОГО КОНТРОЛЯ ДАННЫХ В СИСТЕМЕ ОСТАТОЧНЫХ КЛАССОВ, ОСНОВАННЫЕ НА ПРИНЦИПЕ ПАРАЛЛЕЛЬНОЙ НУЛЕВИЗАЦИИ

В.А. КРАСНОБАЕВ, С.А. КОШМАН, А.С. ЯНКО

В статье рассмотрены методы оперативного контроля данных в системе остаточных классов (СОК), основанные на принципе параллельной нулевизации. Сущность первого предлагаемого метода контроля состоит в том, что процедура нулевизации осуществляется параллельно во времени одновременно по двум основаниям СОК. С целью уменьшения времени контроля данных в СОК в статье предложен второй метод контроля, основанный на использовании процедуры параллельной нулевизации с определением последующих остатков непозиционной кодовой структуры. В статье представлены данные расчета и сравнительного анализа основных характеристик методов контроля данных в СОК.

Ключевые слова: система остаточных классов (СОК), метод контроля данных в СОК, принцип и процедура параллельной нулевизации, непозиционная кодовая структура.

ВВЕДЕНИЕ

В [1, 2] были рассмотрены методы контроля данных (Н1 и Н2) в системе остаточных классов (СОК). Эти методы обладают существенным недостатком – значительное время контроля данных в СОК [1, 3 – 5].

В статье представлены методы (Н3 и Н4), позволяющие повысить оперативность контроля данных в СОК. Данные методы контроля данных основаны на принципе параллельной нулевизации непозиционных кодовых структур (НКС) в СОК [1, 6].

1. ПЕРВЫЙ МЕТОД ОПЕРАТИВНОГО КОНТРОЛЯ ДАННЫХ В СОК

Первый (Н3), рассматриваемый в статье, метод оперативного контроля данных в СОК (метод параллельной нулевизации (ПНН)) отличается, от описанных в [1], следующим образом. Сущность предлагаемого метода контроля состоит в том, что процедура нулевизации осуществляется парал-

лельно во времени по двум основаниям. Для n -четного числа имеем $a_i^{(i-1)}, a_{n-i+1}^{(i-1)}$ ($i = \overline{1, n/2}$), а именно $a_1^{(0)}, a_n^{(0)}, a_2^{(1)}, a_{n-1}^{(1)}, a_3^{(2)}, a_{n-2}^{(2)}, \dots, a_{n/2}^{(n/2)}, a_{n/2+1}^{(n/2)}$ (рис. 1). Для n - нечетного числа имеем, что $a_1^{(0)}, a_n^{(0)}, a_2^{(1)}, a_{n-1}^{(1)}, a_3^{(2)}, a_{n-2}^{(2)}, \dots, a_{(n+1)/2}^{((n+1)/2-1)}$ (рис. 2). В этом случае для произвольного значения i КН для соответствующего числа имеют следующий вид

$$A^{(i)} = [\overbrace{0\|0\|\dots\|0}^{i-\text{нулей}}\|a_{i+1}^{(i)}\| \dots \|a_{n-i+1}^{(i)}\| \overbrace{0\|0\|\dots\|0}^{i-\text{нулей}}\|a_{n+1}^{(i)}\|, KH^{(i)} = [0\|0\|\dots\|0\|t_{i+1}^{(i)}\|t_{i+2}^{(i)}\|\dots\|t_{n-i-1}^{(i)}\|t_{n-i}^{(i)}\|0\|\dots\|0\|0\|t_{n+1}^{(i)}\|];$$

$$t_{i+1}^{(i)} = \overline{0, m_{i+1}}, t_{n-i}^{(i)} = \overline{0, m_{n-i}}; t_{i+1}^{(i)} = a_{i+1}^{(i)}, t_{n-i}^{(i)} = a_{n-i}^{(i)}.$$

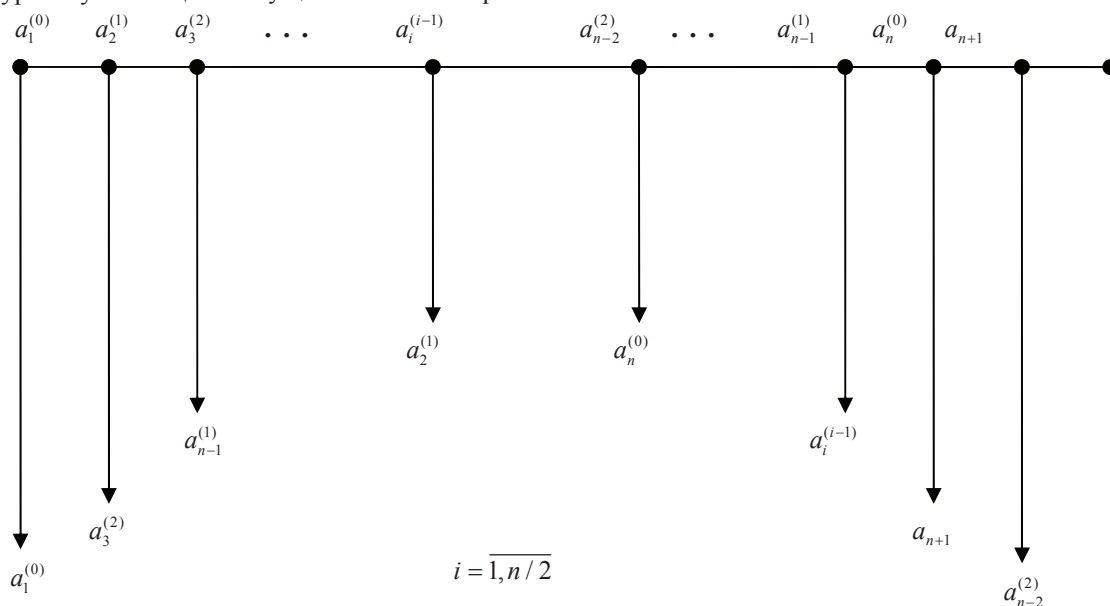


Рис. 1. Схема выборки констант нулевизации для метода ПНН (n – четное число)

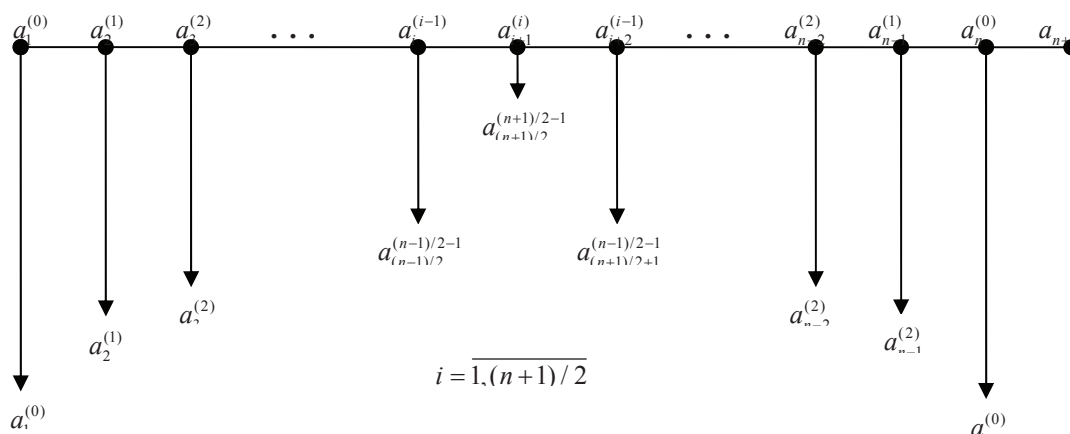


Рис. 2. Схема выборки констант нулевизации для метода ПНН (n – нечетное число)

Для произвольного значения i имеем, что

$$\begin{aligned}
 A^{(i+1)} &= A^{(i)} - KH^{(i)} = [0 \parallel 0 \parallel \dots \parallel 0 \parallel a_{i+2}^{(i)} \parallel a_{i+3}^{(i)} \parallel \dots \\
 &\dots \parallel a_{n-i-2}^{(i)} \parallel a_{n-i-1}^{(i)} \parallel a_{n-i}^{(i)} \parallel 0 \parallel \dots \parallel 0 \parallel a_{n+1}^{(i)}] - [0 \parallel 0 \parallel \dots \\
 &\dots \parallel 0 \parallel t_{i+1}^{(i)} \parallel t_{i+2}^{(i)} \parallel t_{i+3}^{(i)} \parallel \dots \parallel t_{n-i-2}^{(i)} \parallel t_{n-i-1}^{(i)} \parallel t_{n+1}^{(i)} \parallel 0 \parallel \dots \\
 &\dots \parallel 0 \parallel t_{n+1}^{(i)}] = \{0 \parallel 0 \parallel \dots \parallel 0 \parallel [a_{i+1}^{(i)} - t_{i+1}^{(i)}] \bmod m_{i+1} \parallel \\
 &\parallel [a_{i+2}^{(i)} - t_{i+2}^{(i)}] \bmod m_{i+2} \parallel [a_{i+3}^{(i)} - t_{i+3}^{(i)}] \bmod m_{i+3} \parallel \dots \\
 &\dots \parallel [a_{n-i-2}^{(i)} - t_{n-i-2}^{(i)}] \bmod m_{n-i-2} \parallel [a_{n-i-1}^{(i)} - t_{n-i-1}^{(i)}] \bmod m_{n-i-1} \parallel \\
 &\parallel [a_{n-i}^{(i)} - t_{n-i}^{(i)}] \bmod m_{n-i} \parallel 0 \parallel \dots \parallel 0 \parallel [a_{n+1}^{(i)} - t_{n+1}^{(i)}] \bmod m_{n+1}\} = \\
 &= [0 \parallel 0 \parallel \dots \parallel 0 \parallel 0 \parallel a_{i+2}^{(i+1)} \parallel a_{i+3}^{(i+1)} \parallel \dots \parallel a_{n-i-2}^{(i+1)} \parallel a_{n-i-1}^{(i+1)} \parallel \\
 &\parallel 0 \parallel 0 \parallel \dots \parallel 0 \parallel 0 \parallel a_{n+1}^{(i+1)}].
 \end{aligned}$$

Алгоритм выполнения процедуры ПНН представлен в табл. 1.

Перед получением значения $\gamma_{n+1} = a_{n+1}^{(n/2)}$ для n - чётного числа, имеем, что

$$\begin{aligned}
 A^{(n/2-1)} &= [0 \parallel 0 \parallel \dots \parallel 0 \parallel a_{n/2}^{(n/2-1)} \parallel a_{n/2+1}^{(n/2-1)} \parallel \\
 &\parallel 0 \parallel \dots \parallel 0 \parallel 0 \parallel a_{n+1}^{(n/2-1)}]. \\
 KH^{(n/2-1)} &= [0 \parallel 0 \parallel \dots \parallel 0 \parallel t_{n/2}^{(n/2-1)} \parallel t_{n/2+1}^{(n/2-1)} \parallel \\
 &\parallel 0 \parallel \dots \parallel 0 \parallel 0 \parallel t_{n+1}^{(n/2-1)}], \\
 t_{n/2}^{(n/2-1)} &= 0, m_{n/2}, t_{n/2+1}^{(n/2-1)} = 0, m_{n/2+1}, t_{n/2}^{(n/2-1)} = a_{n/2}^{(n/2-1)}, \\
 t_{n/2+1}^{(n/2-1)} &= a_{n/2+1}^{(n/2-1)}. \\
 A^{(H)} &= A^{(n/2)} = A^{(n/2-1)} - KH^{(n/2-1)} = \\
 &= \{0 \parallel 0 \parallel \dots \parallel 0 \parallel [a_{n/2}^{(n/2-1)} - t_{n/2}^{(n/2-1)}] \bmod m_{n/2} \parallel \\
 &\parallel [a_{n/2+1}^{(n/2-1)} - t_{n/2+1}^{(n/2-1)}] \bmod m_{n/2+1} \parallel 0 \parallel 0 \parallel \dots \parallel 0 \parallel 0 \parallel \\
 &= [a_{n+1}^{(n/2-1)} - t_{n+1}^{(n/2-1)}] \bmod m_{n+1}\} = \\
 &= [0 \parallel 0 \parallel \dots \parallel 0 \parallel 0 \parallel 0 \parallel \dots \parallel 0 \parallel 0 \parallel a_{n+1}^{(n/2)}], \text{ где } \gamma_{n+1} = a_{n+1}^{(n/2)}.
 \end{aligned}$$

Перед получением значения $\gamma_{n+1} = a_{n+1}^{(n/2)}$ для n - нечётного числа, имеем, что

$$A^{((n+1)/2-1)} = [0 \parallel 0 \parallel \dots \parallel 0 \parallel a_{(n+1)/2}^{((n+1)/2-1)} \parallel 0 \parallel \dots \parallel 0 \parallel a_{n+1}^{((n+1)/2-1)}]$$

Таблица 1

Алгоритм ПНН

№ операции	Содержание операции
1	Обращение по значениям остатков $a_1^{(0)}$ и $a_n^{(0)}$ числа $A^{(0)}$ в BKH_0 за $KH^{(0)}$.
2	Выполнение операции вычитания $A^{(1)} = A^{(0)} - KH^{(0)}$.
3	Обращение по значениям остатков $a_2^{(1)}$ и $a_{n-1}^{(1)}$ числа $A^{(1)}$ в BKH_1 за $KH^{(1)}$.
4	Выполнение операции вычитания $A^{(2)} = A^{(1)} - KH^{(1)}$.
5	Обращение по значениям остатков $a_2^{(2)}$ и $a_{n-2}^{(2)}$ числа $A^{(2)}$ в BKH_2 за $KH^{(2)}$.
6	Выполнение операции вычитания $A^{(3)} = A^{(2)} - KH^{(2)}$.
...	...
i	Выполнение операции вычитания $A^{(i)} = A^{(i-1)} - KH^{(i-1)}$.
$i+1$	Обращение по значениям остатков $a_{i+1}^{(i)}$ и $a_{n-i}^{(i)}$ числа $A^{(i)}$ в BKH_i за $KH^{(i)}$.
$i+2$	Выполнение операции вычитания $A^{(i+1)} = A^{(i)} - KH^{(i)}$.
...	...
$n-3$	Обращение по значениям остатков $a_{n/2-1}^{(n/2-2)}$ и $a_{n/2+2}^{(n/2-2)}$ числа $A^{(n/2-2)}$ в $BKH_{n/2-2}$ за $KH^{(n/2-2)}$.
$n-2$	Выполнение операции вычитания $A^{(n/2-1)} = A^{(n/2-2)} - KH^{(n/2-2)}$.
$n-1$	Обращение по значениям остатков $a_{n/2}^{(n/2-1)}$ и $a_{n/2+1}^{(n/2-1)}$ числа $A^{(n/2-1)}$ в $BKH_{n/2-1}$ за $KH^{(n/2-1)}$.
n	Выполнение операции вычитания $A^{(n/2)} = A^{(n/2-1)} - KH^{(n/2-1)}$. Получение нулевизируемого $A^{(H)}$ числа $A^{(H)} = A^{(n/2)} = [0 \parallel 0 \parallel \dots \parallel 0 \parallel \dots \parallel 0 \parallel 0 \parallel 0 = a_{n+1}^{(n/2)}]$.

$$KH^{((n+1)/2-1)} = [0 \parallel 0 \parallel \dots \parallel 0 \parallel t_{(n+1)/2}^{((n+1)/2-1)} \parallel 0 \parallel \dots \parallel 0 \parallel t_{n+1}^{((n+1)/2-1)}],$$

$$t_{(n+1)/2}^{((n+1)/2-1)} = \overline{0, m_{(n+1)/2}}; t_{(n+1)/2}^{((n+1)/2-1)} = a_{(n+1)/2}^{((n+1)/2-1)}.$$

$$KH^{((n+1)/2-1)} = \left\{ 0 \parallel 0 \parallel \dots \parallel 0 \parallel [a_{(n+1)/2}^{((n+1)/2-1)} - t_{(n+1)/2}^{((n+1)/2-1)}] \bmod m_{(n+1)/2} \parallel 0 \parallel \dots \parallel 0 \parallel 0 \parallel \dots \parallel 0 \parallel 0 \parallel [a_{n+1}^{((n+1)/2-1)} - t_{n+1}^{((n+1)/2-1)}] \bmod m_{n+1} \right\} =$$

$$= [0 \parallel 0 \parallel \dots \parallel 0 \parallel \dots \parallel 0 \parallel 0 \parallel a_{n+1}^{(n+1)/2}], \text{ где } \gamma_{n+1} = a_{n+1}^{(n+1)/2}.$$

$$A^{(H)} = A^{(n+1)/2} = A^{((n+1)/2-1)} - KH^{((n+1)/2-1)} =$$

Метод ПНН в СОК представлен на рис. 3.

№ операции (такта)	Содержание операции
1	<p>Обращение по значениям остатков $a_1^{(0)}$ и $a_n^{(0)}$ числа</p> <p>$A = A^{(0)} = [a_1^{(0)} \parallel a_2^{(0)} \parallel a_3^{(0)} \parallel \dots \parallel a_{i-1}^{(0)} \parallel a_i^{(0)} \parallel a_{i+1}^{(0)} \parallel \dots \parallel a_{n-3}^{(0)} \parallel a_{n-2}^{(0)} \parallel a_{n-1}^{(0)} \parallel a_n^{(0)}]$ в BKH_0 за константой нулевизации $KH^{(0)} = [t_1^{(0)} \parallel t_2^{(0)} \parallel t_3^{(0)} \parallel \dots \parallel t_{i-1}^{(0)} \parallel t_i^{(0)} \parallel t_{i+1}^{(0)} \parallel \dots \parallel t_{n-3}^{(0)} \parallel t_{n-2}^{(0)} \parallel t_{n-1}^{(0)} \parallel t_n^{(0)}]$;</p> <p>$t_1^{(0)} = a_1^{(0)}, t_n^{(0)} = a_n^{(0)}; t_1^{(0)} = \overline{0, m_1 - 1}, t_n^{(0)} = \overline{0, m_n - 1}$</p>
2	<p>Выполнение операции вычитания</p> <p>$A^{(1)} = A^{(0)} - KH^{(0)} = [a_1^{(0)} \parallel a_2^{(0)} \parallel a_3^{(0)} \parallel \dots \parallel a_{i-1}^{(0)} \parallel a_i^{(0)} \parallel a_{i+1}^{(0)} \parallel \dots \parallel a_{n-3}^{(0)} \parallel a_{n-2}^{(0)} \parallel a_{n-1}^{(0)} \parallel a_n^{(0)}] - [t_1^{(0)} \parallel t_2^{(0)} \parallel t_3^{(0)} \parallel \dots \parallel t_{i-1}^{(0)} \parallel t_i^{(0)} \parallel t_{i+1}^{(0)} \parallel \dots \parallel t_{n-3}^{(0)} \parallel t_{n-2}^{(0)} \parallel t_{n-1}^{(0)} \parallel t_n^{(0)}] = \{ [a_1^{(0)} - t_1^{(0)}] \bmod m_1 \parallel [a_2^{(0)} - t_2^{(0)}] \bmod m_2 \parallel [a_3^{(0)} - t_3^{(0)}] \bmod m_3 \parallel \dots \parallel [a_{i-1}^{(0)} - t_{i-1}^{(0)}] \bmod m_{i-1} \parallel [a_i^{(0)} - t_i^{(0)}] \bmod m_i \parallel [a_{i+1}^{(0)} - t_{i+1}^{(0)}] \bmod m_{i+1} \parallel \dots \parallel [a_{n-3}^{(0)} - t_{n-3}^{(0)}] \bmod m_{n-3} \parallel [a_{n-2}^{(0)} - t_{n-2}^{(0)}] \bmod m_{n-2} \parallel [a_{n-1}^{(0)} - t_{n-1}^{(0)}] \bmod m_{n-1} \parallel \dots \parallel [a_n^{(0)} - t_n^{(0)}] \bmod m_n \parallel [a_{n+1}^{(0)} - t_{n+1}^{(0)}] \bmod m_{n+1} \} = [0 \parallel a_2^{(1)} \parallel a_3^{(1)} \parallel \dots \parallel a_{i-1}^{(1)} \parallel a_i^{(1)} \parallel a_{i+1}^{(1)} \parallel \dots \parallel a_{n-3}^{(1)} \parallel a_{n-2}^{(1)} \parallel a_{n-1}^{(1)} \parallel 0 \parallel a_{n+1}^{(1)}].$</p>
3	<p>Обращение по значениям остатков $a_2^{(1)}$ и $a_{n-1}^{(1)}$ числа</p> <p>$A^{(1)} = [0 \parallel a_2^{(1)} \parallel a_3^{(1)} \parallel \dots \parallel a_{i-1}^{(1)} \parallel a_i^{(1)} \parallel a_{i+1}^{(1)} \parallel \dots \parallel a_{n-3}^{(1)} \parallel a_{n-2}^{(1)} \parallel a_{n-1}^{(1)} \parallel 0 \parallel a_{n+1}^{(1)}]$ в BKH_1 за константой нулевизации $KH^{(1)} = [0 \parallel t_2^{(1)} \parallel t_3^{(1)} \parallel \dots \parallel t_{i-1}^{(1)} \parallel t_i^{(1)} \parallel t_{i+1}^{(1)} \parallel \dots \parallel t_{n-3}^{(1)} \parallel t_{n-2}^{(1)} \parallel t_{n-1}^{(1)} \parallel 0 \parallel t_{n+1}^{(1)}]$; $t_2^{(1)} = a_2^{(1)}, t_{n-1}^{(1)} = a_{n-1}^{(1)};$ $t_2^{(1)} = \overline{0, m_2 - 1}, t_{n-1}^{(1)} = \overline{0, m_{n-1} - 1}.$</p>
4	<p>Выполнение операции вычитания</p> <p>$A^{(2)} = A^{(1)} - KH^{(1)} = [0 \parallel a_2^{(1)} \parallel a_3^{(1)} \parallel \dots \parallel a_{i-1}^{(1)} \parallel a_i^{(1)} \parallel a_{i+1}^{(1)} \parallel \dots \parallel a_{n-3}^{(1)} \parallel a_{n-2}^{(1)} \parallel a_{n-1}^{(1)} \parallel 0 \parallel a_{n+1}^{(1)}] - [0 \parallel t_2^{(1)} \parallel t_3^{(1)} \parallel \dots \parallel t_{i-1}^{(1)} \parallel t_i^{(1)} \parallel t_{i+1}^{(1)} \parallel \dots \parallel t_{n-3}^{(1)} \parallel t_{n-2}^{(1)} \parallel t_{n-1}^{(1)} \parallel 0 \parallel t_{n+1}^{(1)}] = \{ 0 \parallel [a_2^{(1)} - t_2^{(1)}] \bmod m_2 \parallel [a_3^{(1)} - t_3^{(1)}] \bmod m_3 \parallel [a_4^{(1)} - t_4^{(1)}] \bmod m_4 \parallel \dots \parallel [a_{i-1}^{(1)} - t_{i-1}^{(1)}] \bmod m_{i-1} \parallel [a_i^{(1)} - t_i^{(1)}] \bmod m_i \parallel [a_{i+1}^{(1)} - t_{i+1}^{(1)}] \bmod m_{i+1} \parallel \dots \parallel [a_{n-3}^{(1)} - t_{n-3}^{(1)}] \bmod m_{n-3} \parallel [a_{n-2}^{(1)} - t_{n-2}^{(1)}] \bmod m_{n-2} \parallel [a_{n-1}^{(1)} - t_{n-1}^{(1)}] \bmod m_{n-1} \parallel 0 \parallel [a_{n+1}^{(1)} - t_{n+1}^{(1)}] \bmod m_{n+1} \} = [0 \parallel 0 \parallel a_4^{(2)} \parallel \dots \parallel a_{i-1}^{(2)} \parallel a_i^{(2)} \parallel a_{i+1}^{(2)} \parallel \dots \parallel a_{n-3}^{(2)} \parallel a_{n-2}^{(2)} \parallel 0 \parallel 0 \parallel a_{n+1}^{(2)}].$</p>
5	<p>Обращение по значениям остатков $a_3^{(2)}$ и $a_{n-2}^{(2)}$ числа</p> <p>$A^{(2)} = [0 \parallel 0 \parallel a_3^{(2)} \parallel \dots \parallel a_{i-1}^{(2)} \parallel a_i^{(2)} \parallel a_{i+1}^{(2)} \parallel \dots \parallel a_{n-3}^{(2)} \parallel a_{n-2}^{(2)} \parallel 0 \parallel 0 \parallel a_{n+1}^{(2)}]$ в BKH_2 за константой нулевизации $KH^{(2)} = [0 \parallel 0 \parallel t_3^{(2)} \parallel \dots \parallel t_{i-1}^{(2)} \parallel t_i^{(2)} \parallel t_{i+1}^{(2)} \parallel \dots \parallel t_{n-3}^{(2)} \parallel t_{n-2}^{(2)} \parallel 0 \parallel 0 \parallel t_{n+1}^{(2)}]$, $t_3^{(2)} = a_3^{(2)}, t_{n-2}^{(2)} = a_{n-2}^{(2)};$ $t_3^{(2)} = \overline{0, m_3 - 1}, t_{n-2}^{(2)} = \overline{0, m_{n-2} - 1}.$</p>
6	<p>Выполнение операции вычитания</p> <p>$A^{(3)} = A^{(2)} - KH^{(2)} = [0 \parallel 0 \parallel a_3^{(2)} \parallel \dots \parallel a_{i-1}^{(2)} \parallel a_i^{(2)} \parallel a_{i+1}^{(2)} \parallel \dots \parallel a_{n-3}^{(2)} \parallel a_{n-2}^{(2)} \parallel 0 \parallel 0 \parallel a_{n+1}^{(2)}]$</p>

	$\ 0\ a_{n+1}^{(2)} - [0\ 0\ t_3^{(2)}\ \dots\ t_{i-1}^{(2)}\ t_i^{(2)}\ t_{i+1}^{(2)}\ \dots\ t_{n-3}^{(2)}\ t_{n-2}^{(2)}\ 0\ 0\ t_{n+1}^{(2)}] = \{0\ 0\ [a_3^{(2)} - t_3^{(2)}] \bmod m_3\ $ $\ [a_4^{(2)} - t_4^{(2)}] \bmod m_4\ \dots\ [a_{i-1}^{(2)} - t_{i-1}^{(2)}] \bmod m_{i-1}\ [a_i^{(2)} - t_i^{(2)}] \bmod m_i\ [a_{i+1}^{(2)} - t_{i+1}^{(2)}] \bmod m_{i+1}\ \dots$ $\ [a_{n-3}^{(2)} - t_{n-3}^{(2)}] \bmod m_{n-3}\ [a_{n-2}^{(2)} - t_{n-2}^{(2)}] \bmod m_{n-2}\ 0\ 0\ [a_{n+1}^{(2)} - t_{n+1}^{(2)}] \bmod m_{n+1}\} = [0\ 0\ 0\ a_4^{(3)}\ a_5^{(2)}\ \dots\ a_{i-1}^{(3)}\ $ $a_i^{(3)}\ a_{i+1}^{(3)}\ \dots\ a_{n-4}^{(3)}\ a_{n-3}^{(3)}\ 0\ 0\ 0\ a_{n+1}^{(3)}].$
...	...
Для значения $A^{(i)}$	<p>Обращение по значениям остатков $a_i^{(i-1)}$ и $a_{n-i+1}^{(i-1)}$ числа</p> $A^{(i-1)} = [0\ 0\ \dots\ 0\ a_i^{(i-1)}\ a_{i+1}^{(i-1)}\ a_{i+2}^{(i-1)}\ \dots\ \dots\ a_{n-i-3}^{(i-1)}\ a_{n-i-2}^{(i-1)}\ a_{n-i-1}^{(i-1)}\ 0\ 0\ \dots\ 0\ a_{n+1}^{(i-1)}]$ <p>в БКН$_{i-1}$ за константой нулевизации $KH^{(i-1)} = [0\ 0\ \dots\ 0\]$</p> $\ t_i^{(i-1)}\ t_{i+1}^{(i-1)}\ t_{i+2}^{(i-1)}\ \dots\ t_{n-i-1}^{(i-1)}\ t_{n-i}^{(i-1)}\ t_{n-i+1}^{(i-1)}\ 0\ 0\ \dots\ 0\ t_{n+1}^{(i-1)}]; t_i^{(i-1)} = a_i^{(i-1)}, t_{n-i+1}^{(i-1)} = a_{n-i+1}^{(i-1)};$ $t_i^{(i-1)} = \overline{0, m_i - 1}, t_{n-i+1}^{(i-1)} = \overline{0, m_{n-i+1} - 1}.$
	<p>Выполнение операции вычитания</p> $A^{(i)} = A^{(i-1)} - KH^{(i-1)} = [0\ 0\ 0\ \dots\ 0\ a_i^{(i-1)}\ a_{i+1}^{(i-1)}\ \dots\ 0\ 0\ a_{n+1}^{(i-1)}] -$ $-[0\ 0\ 0\ \dots\ 0\ t_i^{(i-1)}\ t_{i+1}^{(i-1)}\ \dots\ 0\ 0\ t_{n+1}^{(i-1)}] = \{0\ 0\ \dots\ 0\ [a_i^{(i-1)} - t_i^{(i-1)}] \bmod m_i\ [a_{i+1}^{(i-1)} - t_{i+1}^{(i-1)}] \bmod m_{i+1}\ $ $\ [a_{i+2}^{(i-1)} - t_{i+2}^{(i-1)}] \bmod m_{i+2}\ \dots\ [a_{n-i-1}^{(i-1)} - t_{n-i-1}^{(i-1)}] \bmod m_{n-i-1}\ [a_{n-i}^{(i-1)} - t_{n-i}^{(i-1)}] \bmod m_{n-i}\ [a_{n-i+1}^{(i-1)} - t_{n-i+1}^{(i-1)}] \bmod m_{n-i+1}\ $ $\ 0\ \dots\ 0\ [a_{n+1}^{(i-1)} - t_{n+1}^{(i-1)}] \bmod m_{n+1}\} = [0\ 0\ \dots\ 0\ 0\ a_{i+1}^{(i)}\ a_{i+2}^{(i)}\ a_{i+3}^{(i)}\ \dots\ a_{n-i-1}^{(i)}\ a_{n-i}^{(i)}\ 0\ 0\ \dots\ 0\ a_{n+1}^{(i)}]$
Для значения $A^{(i+1)}$	<p>Обращение по значениям остатков $a_{i+1}^{(i)}$ и $a_{n-i}^{(i)}$ числа</p> $A^{(i)} = [0\ 0\ 0\ \dots\ 0\ 0\ a_{i+1}^{(i)}\ \dots\ a_{n-i-1}^{(i)}\ a_{n-i}^{(i)}\ 0\ 0\ 0\ a_{n+1}^{(i)}]$ <p>в БКН$_i$ за константой нулевизации $KH^{(i)} = [0\ 0\ 0\ \dots\ 0\ 0\ t_{i+1}^{(i)}\ \dots\ t_{n-i-1}^{(i)}\ t_{n-i}^{(i)}\ 0\ 0\ t_{n+1}^{(i)}]; t_{i+1}^{(i)} = a_{i+1}^{(i)}, t_{n-i}^{(i)} = a_{n-i}^{(i)}; t_{i+1}^{(i)} = \overline{0, m_{i+1} - 1},$</p> $t_{n-i}^{(i)} = \overline{0, m_{n-i} - 1}.$
	<p>Выполнение операции вычитания</p> $A^{(i+1)} = A^{(i)} - KH^{(i)} = [0\ 0\ \dots\ 0\ a_{i+1}^{(i)}\ a_{i+2}^{(i)}\ a_{i+3}^{(i)}\ \dots\ a_{n-i-2}^{(i)}\ a_{n-i-1}^{(i)}\ $ $\ a_{n-i}^{(i)}\ 0\ \dots\ 0\ a_{n+1}^{(i)}] - [0\ 0\ \dots\ 0\ t_{i+1}^{(i)}\ t_{i+2}^{(i)}\ t_{i+3}^{(i)}\ \dots\ t_{n-i-2}^{(i)}\ t_{n-i-1}^{(i)}\ t_{n-i}^{(i)}\ 0\ \dots\ 0\ t_{n+1}^{(i)}] = \{0\ 0\ \dots\ 0\ $ $\ [a_{i+1}^{(i)} - t_{i+1}^{(i)}] \bmod m_{i+1}\ [a_{i+2}^{(i)} - t_{i+2}^{(i)}] \bmod m_{i+2}\ [a_{i+3}^{(i)} - t_{i+3}^{(i)}] \bmod m_{i+3}\ \dots\ [a_{n-i-2}^{(i)} - t_{n-i-2}^{(i)}] \bmod m_{n-i-2}\ $ $\ [a_{n-i-1}^{(i)} - t_{n-i-1}^{(i)}] \bmod m_{n-i-1}\ [a_{n-i}^{(i)} - t_{n-i}^{(i)}] \bmod m_{n-i}\ 0\ \dots\ 0\ [a_{n+1}^{(i)} - t_{n+1}^{(i)}] \bmod m_{n+1}\} = [0\ 0\ \dots\ 0\ 0\ a_{i+1}^{(i+1)}\ $ $a_{i+2}^{(i+1)}\ \dots\ a_{n-i-2}^{(i+1)}\ a_{n-i-1}^{(i+1)}\ 0\ 0\ \dots\ 0\ 0\ a_{n+1}^{(i+1)}].$
...	...
$n-1$	<p>В дальнейшем для n чётного n нечётного и чисел получим. Для n чётного числа. Обращение по значениям остатков $a_{n/2}^{(n/2-1)}$ и $a_{n/2+1}^{(n/2-1)}$ числа</p> $A^{(n/2-1)} = [0\ 0\ \dots\ 0\ a_{n/2}^{(n/2-1)}\ a_{n/2+1}^{(n/2-1)}\ 0\ \dots\ 0\ 0\ a_{n+1}^{(n/2-1)}]$ <p>в БКН$_{n/2-1}$ за константой нулевизации $KH^{(n/2-1)} = [0\ 0\ \dots\ 0\ t_{n/2}^{(n/2-1)}\ t_{n/2+1}^{(n/2-1)}\ 0\ \dots\ 0\ 0\ t_{n+1}^{(n/2-1)}]; t_{n/2}^{(n/2-1)} = a_{n/2}^{(n/2-1)},$</p> $t_{n/2+1}^{(n/2-1)} = a_{n/2+1}^{(n/2-1)}; t_{n/2}^{(n/2-1)} = \overline{0, m_{n/2} - 1}, t_{n/2+1}^{(n/2-1)} = \overline{0, m_{n/2+1} - 1}.$
	<p>Для n нечётного числа.</p> <p>Обращение по значению остатка $a_{(n+1)/2}^{((n+1)/2-1)}$ числа</p> $A^{((n+1)/2-1)} = [0\ 0\ \dots\ 0\ a_{(n+1)/2}^{((n+1)/2-1)}\ 0\ \dots\ 0\ a_{n+1}^{((n+1)/2-1)}]$ <p>в БКН$_{(n+1)/2-1}$ за константой нулевизации $KH^{((n+1)/2-1)} = [0\ 0\ \dots\ 0\ t_{(n+1)/2}^{((n+1)/2-1)}\ 0\ \dots\ 0\ t_{n+1}^{((n+1)/2-1)}], t_{(n+1)/2}^{((n+1)/2-1)} = a_{(n+1)/2}^{((n+1)/2-1)};$</p>

	$t_{(n+1)/2}^{((n+1)/2-1)} = 0, m_{(n+1)/2} - 1.$
n	<p>Для n чётного и n нечётного чисел получим следующие значения нулевизируемого числа $A^{(H)}$.</p> <p style="text-align: center;">Для n чётного числа.</p> <p>Получение нулевизируемого $A^{(H)}$ числа:</p> $A^{(H)} = A^{(n/2)} = A^{(n/2-1)} - KH^{(n/2-1)} = [0 \ 0 \ \dots \ 0 \ a_{n/2}^{(n/2-1)} \ $ $\ a_{n/2+1}^{(n/2-1)} \ 0 \ \dots \ 0 \ 0 \ a_{n+1}^{(n/2-1)}] - [0 \ 0 \ \dots \ 0 \ t_{n/2}^{(n/2-1)} \ t_{n/2+1}^{(n/2-1)} \ 0 \ \dots \ 0 \ 0 \ t_{n+1}^{(n/2-1)}] = \{0 \ 0 \ \dots \ 0 \ $ $\ [a_{n/2}^{(n/2-1)} - t_{n/2}^{(n/2-1)}] \bmod m_{n/2} \ [a_{n/2+1}^{(n/2-1)} - t_{n/2+1}^{(n/2-1)}] \bmod m_{n/2+1} \ 0 \ \dots \ 0 \ 0 \ [a_{n+1}^{(n/2-1)} - t_{n+1}^{(n/2-1)}] \bmod m_{n+1}\} =$ $= [0 \ 0 \ \dots \ 0 \ 0 \ 0 \ \dots \ 0 \ 0 \ a_{n+1}^{(n/2)}], \text{ где } \gamma_{n+1} = a_{n+1}^{(n/2)}.$ <p style="text-align: center;">Для n нечётного числа.</p> <p>Получение нулевизируемого $A^{(H)}$ числа:</p> $A^{(H)} = A^{(n+1/2)} = A^{((n+1)/2-1)} - KH^{((n+1)/2-1)} = [0 \ 0 \ \dots \ 0 \ $ $\ a_{(n+1)/2}^{((n+1)/2-1)} \ 0 \ \dots \ 0 \ a_{n+1}^{((n+1)/2-1)}] - [0 \ 0 \ \dots \ 0 \ t_{(n+1)/2}^{((n+1)/2-1)} \ 0 \ \dots \ 0 \ t_{n+1}^{((n+1)/2-1)}] = \{0 \ 0 \ \dots \ 0 \ $ $\ [a_{(n+1)/2}^{((n+1)/2-1)} - t_{(n+1)/2}^{((n+1)/2-1)}] \bmod m_{(n+1)/2} \ 0 \ \dots \ 0 \ [a_{n+1}^{((n+1)/2-1)} - t_{n+1}^{((n+1)/2-1)}] \bmod m_{n+1}\} = [0 \ 0 \ \dots \ 0 \ \dots \ 0 \ 0 \ $ $\ a_{n+1}^{((n+1)/2)}], \text{ где } \gamma_{n+1} = a_{n+1}^{((n+1)/2)}.$
	$T_{H3} = n \cdot \tau$

Рис. 3. Метод ПНН в СОК

Время T_{H3} выполнения процедуры нулевизации для первого (Н3) метода ПНН определяется как

$$T_{H3} = n \cdot \tau_{сл}. \quad (1)$$

При реализации процедуры нулевизации для второго (Н3) метода в блоке констант нулевизации (БН) вычислителя в СОК необходимо иметь

$$K_{H3} = \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} (m_i \cdot m_{n-i+1} - 1) \text{ констант нулевизации. При}$$

этом количество N_{H3} двоичных разряда констант нулевизации БН определяется выражением

$$K_{H3} = \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} (m_i \cdot m_{n-i+1} - 1) \cdot (n - 2i + 1).$$

Существенным недостатком известных методов (Н1-Н3) контроля данных в СОК является необходимость значительных временных затрат на контроль, что обуславливает низкую оперативность контроля и значительные непроизводительные вычислительные затраты [1, 7-9].

2. ВТОРОЙ МЕТОД ОПЕРАТИВНОГО КОНТРОЛЯ ДАННЫХ В СОК

С целью повышения оперативности контроля данных, за счет уменьшения времени реализации процедуры нулевизации, в статье предложен второй

(Н4) метод контроля данных в СОК – метод параллельной нулевизации с определением последующих остатков (ПНН ОПО). Предлагаемый метод контроля основывается на процедуре использования парной нулевизации чисел с дополнительной операцией предварительной выборкой остатков (см. метод Н2 [1]). Суть метода контроля состоит в том, что предварительная выборка производится одновременно по двум остаткам $a_{i+1}^{(i)}$ и $a_{n-i}^{(i)}$ числа

$A^{(i)} = [0 \| \overbrace{0 \| \dots \| 0}^{i-\text{нулей}} \| a_{i+1}^{(i)} \| a_{i+2}^{(i)} \| \dots \| a_{n-i-1}^{(i)} \| a_{n-i}^{(i)} \|$
 $\| 0 \| \dots \| 0 \| 0 \| a_{n+1}^{(i)}].$ Таким образом, при реализации процедуры нулевизации совмещаются во времени операция выбора, по остаткам $a_{i+1}^{(i)}$ и $a_{n-i}^{(i)}$ числа $A^{(i)} = [0 \| 0 \| \dots \| 0 \| a_{i+1}^{(i)} \| a_{i+2}^{(i)} \| a_{i+3}^{(i)} \| \dots \| a_{n-i-2}^{(i)} \| a_{n-i-1}^{(i)} \|$
 $\| a_{n-i}^{(i)} \| 0 \| \dots \| 0 \| a_{n+1}^{(i)}],$ константы нулевизации $KH^{(i)} = [0 \| 0 \| \dots \| 0 \| t_{i+1}^{(i)} \| t_{i+2}^{(i)} \| t_{i+3}^{(i)} \| \dots \| t_{n-i-2}^{(i)} \| t_{n-i-1}^{(i)} \|$
 $\| t_{n-i}^{(i)} \| 0 \| \dots \| 0 \| t_{n+1}^{(i)}]$ и операция определения, по значениям остатков $a_{i+1}^{(i)}$ и $a_{n-i}^{(i)}$, последующих значений остатков $a_{i+2}^{(i+1)}$ и $a_{n-i-1}^{(i+1)}$ числа $A^{(i+1)} = [0 \| 0 \| \dots \| 0 \| 0 \|$
 $\| a_{i+2}^{(i+1)} \| a_{i+3}^{(i+1)} \| \dots \| a_{n-i-2}^{(i+1)} \| a_{n-i-1}^{(i+1)} \| 0 \| 0 \| \dots \| 0 \| 0 \| a_{n+1}^{(i+1)}].$ Также совмещаются во времени операция вычитания

$A^{(i+1)} = A^{(i)} - KH^{(i)}$ и операция выбора очередной кон- $\|t_{i+2}^{(i+1)}\| \dots \|t_{n-i-2}^{(i+1)}\| \|t_{n-i-1}^{(i+1)}\| \|0\| \|0\| \|t_{n+1}^{(i+1)}\|$. Алгоритм нулеви-
 станты нулевизации $KH^{(i+1)} = [0 \|0\| \|0\| \dots \|0\| \|0\| \|0\| \|0\|]$ зации представлен в табл. 2.

Таблица 2

Алгоритм ПНН ОПО

№ опера- ции	Содержание операции	
1	2	3
1	Обращение по значениям остатков $a_1^{(0)}$ и $a_n^{(0)}$ числа $A^{(0)}$ в BKH_0 за $KH^{(0)}$.	Образование значений остатков $a_2^{(1)}$ и $a_{n-1}^{(1)}$ числа $A^{(1)}$ в виде $a_2^{(1)} = t_2^{(1)} = [a_2^{(0)} - a_1^{(0)}] \bmod m_2$ и $a_{n-1}^{(1)} = t_{n-1}^{(1)} = [a_{n-1}^{(0)} - a_n^{(0)}] \bmod m_{n-1}$.
2	Выполнение операции вычитания $A^{(1)} = A^{(0)} - KH^{(0)}$.	Обращение по значениям остатков $a_2^{(1)}$ и $a_{n-1}^{(1)}$ числа $A^{(1)}$ в BKH_1 за $KH^{(1)}$.
3	Выполнение операции вычитания $A^{(2)} = A^{(1)} - KH^{(1)}$.	Образование значений остатков $a_3^{(2)}$ и $a_{n-2}^{(2)}$ числа $A^{(2)}$ в виде $a_3^{(2)} = t_3^{(2)} = [a_3^{(1)} - a_2^{(1)}] \bmod m_3$ и $a_{n-2}^{(2)} = t_{n-2}^{(2)} = [a_{n-2}^{(1)} - a_{n-1}^{(1)}] \bmod m_{n-2}$.
⋮	⋮	⋮
i	Выполнение операции вычитания $A^{(i)} = A^{(i-1)} - KH^{(i-1)}$.	Обращение по значениям остатков $a_{i+1}^{(i)}$ и $a_{n-i}^{(i)}$ числа $A^{(i)}$ в BKH_i за $KH^{(i)}$.
$i+1$	Выполнение операции вычитания $A^{(i+1)} = A^{(i)} - KH^{(i)}$.	Образование значений остатков $a_{i+2}^{(i+1)}$ и $a_{n-i-1}^{(i+1)}$ числа $A^{(i+1)}$ в виде $a_{i+2}^{(i+1)} = t_{i+2}^{(i+1)} = [a_{i+2}^{(i)} - a_{i+1}^{(i)}] \bmod m_{i+2}$ и $a_{n-i-1}^{(i+1)} = t_{n-i-1}^{(i+1)} = [a_{n-i-1}^{(i)} - a_{n-i-2}^{(i)}] \bmod m_{n-i-1}$.
$i+2$	Обращение по значениям остатков $a_{i+2}^{(i+1)}$ и $a_{n-i-1}^{(i+1)}$ числа $A^{(i+1)}$ в BKH_{i+1} за $KH^{(i+1)}$.	Образование значений остатков $a_{i+3}^{(i+2)}$ и $a_{n-i-2}^{(i+2)}$ числа $A^{(i+2)}$ в ви- де $a_{i+3}^{(i+2)} = t_{i+3}^{(i+2)} = [a_{i+3}^{(i+1)} - a_{i+2}^{(i+1)}] \bmod m_{i+3}$ и $a_{n-i-2}^{(i+2)} = t_{n-i-2}^{(i+2)} = [a_{n-i-2}^{(i+1)} - a_{n-i-3}^{(i+1)}] \bmod m_{n-i-2}$.
⋮	⋮	⋮
$k-2$	Обращение по значениям остатков $a_{n/2-1}^{(n/2-2)}$ и $a_{n/2+2}^{(n/2-2)}$ числа $A^{(n/2-2)}$ в $BKH_{n/2-2}$ за $KH^{(n/2-2)}$.	Образование значений остатков $a_{n/2}^{(n/2-1)}$ и $a_{n/2+1}^{(n/2-1)}$ числа $A^{(n/2-1)}$ в виде $a_{n/2}^{(n/2-1)} = t_{n/2}^{(n/2-1)} = [a_{n/2}^{(n/2-2)} - a_{n/2-1}^{(n/2-2)}] \bmod m_{n/2}$ и $a_{n/2+1}^{(n/2-1)} = t_{n/2+1}^{(n/2-1)} = [a_{n/2+1}^{(n/2-2)} - a_{n/2}^{(n/2-2)}] \bmod m_{n/2+1}$.
$k-1$	Выполнение операции вычитания $A^{(n/2-1)} = A^{(n/2-2)} - KH^{(n/2-2)}$.	Обращение по значениям остатков $a_{n/2}^{(n/2-1)}$ и $a_{n/2+1}^{(n/2-1)}$ числа $A^{(n/2-1)}$ в $BKH_{n/2-1}$ за $KH^{(n/2-1)}$.
k	Выполнение операции вычитания $A^{(n/2)} = A^{(n/2-1)} - KH^{(n/2-1)}$. Получение нулевизируемого $A^{(H)}$ числа $A^{(H)} = A^{(n/2)} = [0 \ 0\ \ 0\ \dots \ 0\ \ 0\ \dots \ 0\ \ 0\ \ 0\ \ 0\ \gamma_{n+1} = a_{n+1}^{(n/2)}]$.	

Значения величин Δa_{i+2} , Δa_{n-i-1} , которые будут вычтены из соответствующих значений $a_{i+2}^{(i)}$ и $a_{n-i-1}^{(i)}$, чтобы получить значения остатков, $a_{i+2}^{(i+1)}$ и $a_{n-i-1}^{(i+1)}$, определяются только значениями соответствующих остатков числа $a_{i+1}^{(i)}$ и $a_{n-i}^{(i)}$. Аналитически это можно представить в виде следующих двух выражений

$$a_{i+2}^{(i+1)} = [a_{i+2}^{(i)} - \Delta a_{i+2}] \bmod m_{i+2} \text{ и}$$

$$a_{n-i-1}^{(i+1)} = [a_{n-i-1}^{(i)} - \Delta a_{n-i-1}] \bmod m_{n-i-1}.$$

В процессе выборки $KH^{(i)}$ по значениям остатков $a_{i+1}^{(i)}$ и $a_{n-i}^{(i)}$ числа $A^{(i)}$, эти же остатки будут переданы в вычислитель по соответствующим основаниям

m_{i+2} и m_{n-i-1} . Из двухвходовых таблиц $F_1\{a_{i+2}^{(i+1)}\} = [a_{i+1}^{(i)}, a_{i+2}^{(i)}]$ и $F_2\{a_{n-i-1}^{(i+1)}\} = [a_{n-1}^{(i)}, a_{n-i-1}^{(i)}]$ выбираются (определяются) значения $a_{i+2}^{(i+1)}$ и $a_{n-i-1}^{(i+1)}$. В этом случае общее количество тактов, свободных от сложения, во время которых производится обращение в БН и образования очередного адреса равняется значению $[(n+1)/2]$, (где $[x]$ – целое, наиболее ближайшее к x число, но его не превосходящее). При этом нулевизация проводится одновременно по двум информационным основаниям СОК $a_1, a_n; a_2, a_{n-1}$ и т.д. После каждого двух вычитаний требуется еще один дополнительный временной такт для образования очередного адреса и обращения к накопителю констант нулевизации. В связи с этим на каждые два такта сложения ($\tau_{cl} = \tau_0$) приходится один такт, свободный от сложения.

На основе вышеизложенного время выполнения операции нулевизации для второго Н4 метода оперативного контроля определится следующим образом

$$T_{H4} = \left[\frac{n+1}{2} \right] \cdot \tau_{cl} + \left[\frac{\frac{n+1}{2} + 1}{2} \right] \cdot \tau_{выб}. \quad (2)$$

Учитывая, что $\tau_{cl} = \tau_{выб}$ получим:

$$T_{H4} = \left(\left[\frac{n+1}{2} \right] + \left[\frac{\frac{n+1}{2} + 1}{2} \right] \right) \cdot \tau_{cl} \quad (3)$$

При n – четном, выражение (3) принимает вид:

$$T'_{H4} = \left(\frac{n}{2} + \left[\frac{\frac{n}{2} + 1}{2} \right] \right) \cdot \tau_{cl} \quad (4)$$

Если $\frac{n}{2}$ – четное, то

$$T'_{H4} = \frac{3}{4} n \cdot \tau_{cl}. \quad (5)$$

Если $\frac{n}{2}$ – нечетное, то

$$T'_{H4} = \left(\frac{3n+2}{4} \right) \cdot \tau_{cl}. \quad (6)$$

При n нечетном:

$$T''_{H4} = \left(\frac{n+1}{2} + \left[\frac{\frac{n+1}{2} + 1}{2} \right] \right) \cdot \tau_{cl}. \quad (7)$$

Если $\frac{n+1}{2}$ четное, то

$$T''_{H4} = \frac{3}{4} (n+1) \cdot \tau_{cl}. \quad (8)$$

Если $\frac{n+1}{2}$ нечетное, то

$$T''_{H4} = \left(\frac{3n+5}{4} \right) \cdot \tau_{cl}. \quad (9)$$

На рис. 4 приведены временные диаграммы работы БН для метода ПН (диаграмма Н1), для метода ПН ОПО (диаграмма Н2) и также для первого ПНН (диаграмма Н3) и второго ПНН ОПО (диаграмма Н4) методов контроля, рассмотренных в статье. Где: Об $a_i^{(i-1)}, a_{n-i+1}^{(i-1)}$, – обращение по значениям цифр $a_i^{(i-1)}$ и $a_{n-i+1}^{(i-1)}$ числа

$A^{(i-1)} = (0 \parallel \dots \parallel 0 \parallel a_i^{(i-1)} \parallel a_{i+1}^{(i-1)} \parallel \dots \parallel a_{n-i}^{(i-1)} \parallel a_{n-i+1}^{(i-1)} \parallel 0 \parallel \dots \parallel 0 \parallel a_{n+1}^{(i-1)})$ в БН за константой нулевизации вида $KH^{(i)} = (0 \parallel \dots \parallel 0 \parallel t_{i,i} \parallel t_{i+1,i} \parallel \dots \parallel t_{n-i,i} \parallel t_{n-i+1,i} \parallel 0 \parallel \dots \parallel 0, t_{n+1,i})$; $a_{i+1}^{(i+1)}, a_{n-i}^{(i+1)}$ – создание по значениям $a_i^{(i)}$ и $a_{n-i+1}^{(i)}$ числа $A^{(i-1)}$ следующих цифр $a_{i+1}^{(i)}$ и $a_{n-i}^{(i)}$ для числа $A^{(i)} = (0, \dots, 0, a_{i+1}^{(i)}, \dots, a_{n-i}^{(i)}, 0, \dots, 0, a_{n+1}^{(i)})$; $\sum i$ – операция вычитания значения константы $KH^{(i-1)}$ из числа $A^{(i-1)}$, т.е. проведение операции $A^{(i-1)} - KH^{(i-1)}$.

Доказательство выведенного соотношения (3) удобно провести методом математической индукции по n .

Первый этап доказательства. Для минимального значения $n=3$ время нулевизации равно $T_{H4} = 3 \cdot \tau_{cl}$. Это очевидно из рис. 4, диаграмма Н4 (ПНН ОПО).

Второй этап. Допустим, что выражение (3) справедливо и при $n = K$, т.е.

$$T_{H4} = \left(\left[\frac{K+1}{2} \right] + \left[\frac{\left[\frac{K-1}{2} \right] + 1}{2} \right] \right) \cdot \tau_{cl}.$$

Третий этап. Докажем, что выражение (3) справедливо и при $n = K+1$, т.е.

$$T_{H4} = \left(\left[\frac{K+2}{2} \right] + \left[\frac{\left[\frac{K+2}{2} \right] + 1}{2} \right] \right) \cdot \tau_{cl}.$$

При K – четном ($K+1$ – нечетное) имеем:

$$T'_{H4} = \left(\frac{K}{2} + 1 + \left[\frac{\frac{K}{2} + 2}{2} \right] \right) \cdot \tau_{cl}.$$

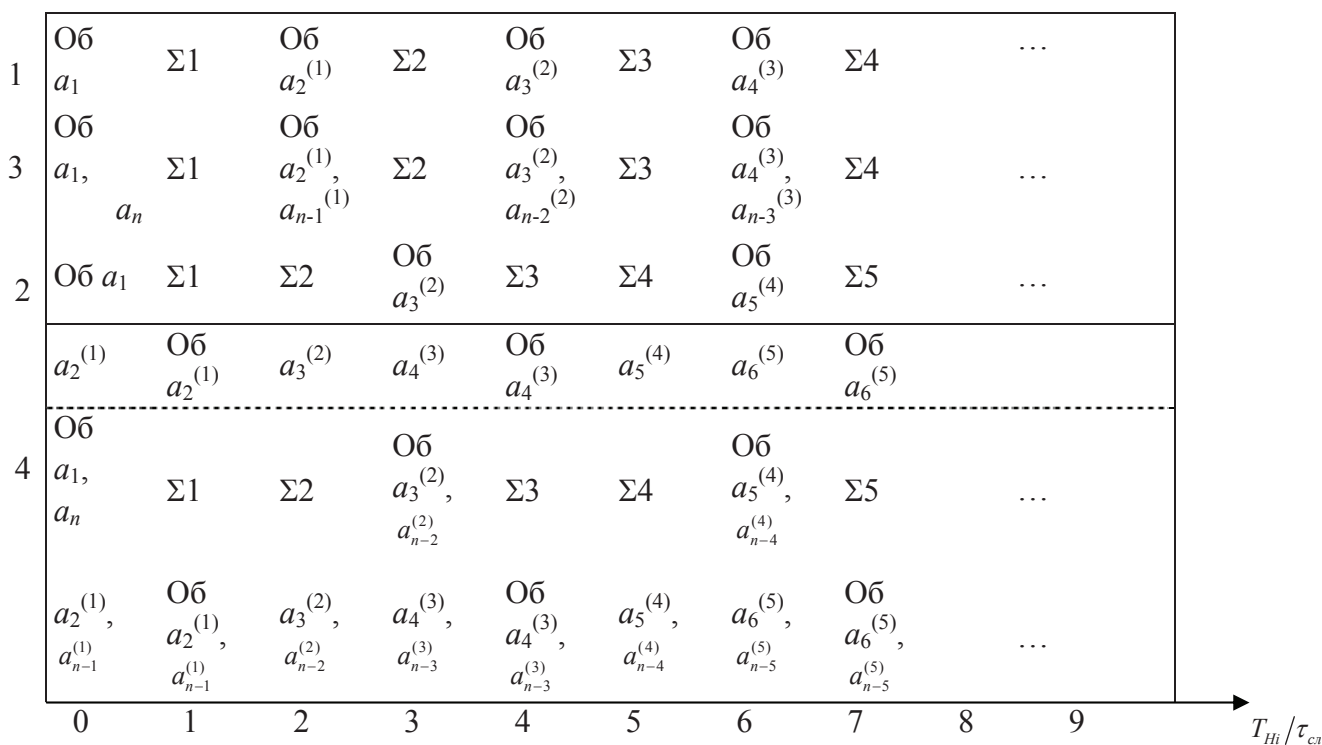


Рис. 4. Временные диаграммы работы БН при различных методах нулевизации

Если $\frac{K}{2}$ четное, то $T'_{H4} = \left(\frac{3K+8}{4}\right) \cdot \tau_{cl}$. Если

$\frac{K}{2}$ – нечетное, то $T'_{H4} = \left(\frac{3K+6}{4}\right) \cdot \tau_{cl}$.

При K нечетном ($K+1$ - четное), имеем:

$$T''_{H4} = \left(\frac{K+1}{2} + \left\lfloor \frac{\left\lfloor \frac{K+1}{2} \right\rfloor + 1}{2} \right\rfloor \right) \cdot \tau_{cl}.$$

Если $\frac{K+1}{2}$ - четное, то $T''_{H4} = \left(\frac{3K+3}{4}\right) \cdot \tau_{cl}$; если

$\frac{K+1}{2}$ – нечетное, то $T''_{H4} = \left(\frac{3K+5}{4}\right) \cdot \tau_{cl}$.

Тогда в соответствии с выражениями (5), (6), (7) и (9) запишем, что:

$$\frac{3K+3}{4} \cdot \tau_{cl} = \frac{3}{4}(K+1) \cdot \tau_{cl},$$

$$\frac{3K+5}{4} \cdot \tau_{cl} = \left\{ \frac{3(K+1)+2}{4} \right\} \cdot \tau_{cl},$$

$$\frac{3K+6}{4} \cdot \tau_{cl} = \frac{3}{4} \{ (K+1)+1 \} \cdot \tau_{cl},$$

$$\frac{3K+8}{4} \cdot \tau_{cl} = \left\{ \frac{3(K+1)+5}{4} \right\} \cdot \tau_{cl}.$$

Таким образом, выражение (3) справедливо и при $n=K+1$, что и требовалось доказать. Метод контроля данных ПНН ОПО представлен на рис. 5.

3. РАСЧЕТ И СРАВНИТЕЛЬНЫЙ АНАЛИЗ ОСНОВНЫХ ХАРАКТЕРИСТИК МЕТОДОВ КОНТРОЛЯ ДАННЫХ В СОК

При выборе метода контроля данных в СОК необходимо учитывать количественные значения показателей (характеристик), характеризующих данный метод. Так, для реализации процедуры нулевизации для второго (Н4) метода в БН необходимо иметь

$$K_{H4} = \sum_{i=1}^{\left\lfloor \frac{n}{2} \right\rfloor} (m_i \cdot m_{n-i+1} - 2)$$

констант нулевизации. При этом количество N_{H3} двоичных разряда констант нулевизации определяется выражением

$$K_{H4} = \sum_{i=1}^{\left\lfloor \frac{n}{2} \right\rfloor} (m_i \cdot m_{n-i+1} - 2) \cdot (n - 2i + 1).$$

№ операции (такта)	Содержание операции	
1	2	3
1	<p>Обращение по значениям остатков $a_1^{(0)}$ и $a_n^{(0)}$ числа $A = A^{(0)} = [a_1^{(0)} \ a_2^{(0)} \ a_3^{(0)} \ \dots \ a_{i-1}^{(0)} \ a_i^{(0)} \ a_{i+1}^{(0)} \ \dots \ a_{n-3}^{(0)} \ a_{n-2}^{(0)} \ a_{n-1}^{(0)} \ a_n^{(0)}]$ в БКН₀ за константой нулевизации $KH^{(0)} = [t_1^{(0)} \ t_2^{(0)} \ t_3^{(0)} \ \dots \ t_{i-1}^{(0)} \ t_i^{(0)} \ t_{i+1}^{(0)} \ \dots \ t_{n-3}^{(0)} \ t_{n-2}^{(0)} \ t_{n-1}^{(0)} \ t_n^{(0)}]$; $t_1^{(0)} = a_1^{(0)}$, $t_n^{(0)} = a_n^{(0)}$; $t_1^{(0)} = \overline{0, m_1 - 1}$, $t_n^{(0)} = \overline{0, m_n - 1}$.</p>	<p>Образование значений остатков $a_2^{(1)}$ и $a_{n-1}^{(1)}$ числа $A^{(1)} = [0 \ a_2^{(1)} \ a_3^{(1)} \ \dots \ a_{i-1}^{(1)} \ a_i^{(1)} \ a_{i+1}^{(1)} \ \dots \ a_{n-3}^{(1)} \ a_{n-2}^{(1)} \ a_{n-1}^{(1)}]$ в виде $a_2^{(1)} = t_2^{(1)} = [a_2^{(0)} - a_1^{(0)}] \bmod m_2$ и $a_{n-1}^{(1)} = t_{n-1}^{(1)} = [a_{n-1}^{(0)} - a_n^{(0)}] \bmod m_{n-1}$.</p>
2	<p>Выполнение операции вычитания $A^{(1)} = A^{(0)} - KH^{(0)} = [a_1^{(0)} \ a_2^{(0)} \ a_3^{(0)} \ \dots \ a_{i-1}^{(0)} \ a_i^{(0)} \ a_{i+1}^{(0)} \ \dots \ a_{n-3}^{(0)} \ a_{n-2}^{(0)} \ a_{n-1}^{(0)} \ a_n^{(0)}] - [t_1^{(0)} \ t_2^{(0)} \ t_3^{(0)} \ \dots \ t_{i-1}^{(0)} \ t_i^{(0)} \ t_{i+1}^{(0)} \ \dots \ t_{n-3}^{(0)} \ t_{n-2}^{(0)} \ t_{n-1}^{(0)} \ t_n^{(0)}] = \{ [a_1^{(0)} - t_1^{(0)}] \bmod m_1 \ [a_2^{(0)} - t_2^{(0)}] \bmod m_2 \ [a_3^{(0)} - t_3^{(0)}] \bmod m_3 \ \dots \ [a_{i-1}^{(0)} - t_{i-1}^{(0)}] \bmod m_{i-1} \ [a_i^{(0)} - t_i^{(0)}] \bmod m_i \ [a_{i+1}^{(0)} - t_{i+1}^{(0)}] \bmod m_{i+1} \ \dots \ [a_{n-3}^{(0)} - t_{n-3}^{(0)}] \bmod m_{n-3} \ [a_{n-2}^{(0)} - t_{n-2}^{(0)}] \bmod m_{n-2} \ [a_{n-1}^{(0)} - t_{n-1}^{(0)}] \bmod m_{n-1} \ [a_n^{(0)} - t_n^{(0)}] \bmod m_n \ [a_{n+1}^{(0)} - t_{n+1}^{(0)}] \bmod m_{n+1} \} = [0 \ a_2^{(1)} \ a_3^{(1)} \ \dots \ a_{i-1}^{(1)} \ a_i^{(1)} \ a_{i+1}^{(1)} \ \dots \ a_{n-3}^{(1)} \ a_{n-2}^{(1)} \ a_{n-1}^{(1)} \ 0 \ a_{n+1}^{(1)}]$.</p>	<p>Обращение по значениям остатков $a_2^{(1)}$ и $a_{n-1}^{(1)}$ числа $A^{(1)} = [0 \ a_2^{(1)} \ a_3^{(1)} \ \dots \ a_{i-1}^{(1)} \ a_i^{(1)} \ a_{i+1}^{(1)} \ \dots \ a_{n-3}^{(1)} \ a_{n-2}^{(1)} \ a_{n-1}^{(1)} \ 0 \ a_{n+1}^{(1)}]$ в БКН₁ за константой нулевизации $KH^{(1)} = [0 \ t_2^{(1)} \ t_3^{(1)} \ \dots \ t_{i-1}^{(1)} \ t_i^{(1)} \ t_{i+1}^{(1)} \ \dots \ t_{n-3}^{(1)} \ t_{n-2}^{(1)} \ t_{n-1}^{(1)} \ 0 \ t_{n+1}^{(1)}]$; $t_2^{(1)} = a_2^{(1)}$, $t_{n-1}^{(1)} = a_{n-1}^{(1)}$; $t_2^{(1)} = \overline{0, m_2 - 1}$, $t_{n-1}^{(1)} = \overline{0, m_{n-1} - 1}$.</p>
3	<p>Выполнение операции вычитания $A^{(2)} = A^{(1)} - KH^{(1)} = \{ 0 \ [a_2^{(1)} - t_2^{(1)}] \bmod m_2 \ [a_3^{(1)} - t_3^{(1)}] \bmod m_3 \ [a_4^{(1)} - t_4^{(1)}] \bmod m_4 \ \dots \ [a_{i-1}^{(1)} - t_{i-1}^{(1)}] \bmod m_{i-1} \ [a_i^{(1)} - t_i^{(1)}] \bmod m_i \ [a_{i+1}^{(1)} - t_{i+1}^{(1)}] \bmod m_{i+1} \ \dots \ [a_{n-3}^{(1)} - t_{n-3}^{(1)}] \bmod m_{n-3} \ [a_{n-2}^{(1)} - t_{n-2}^{(1)}] \bmod m_{n-2} \ [a_{n-1}^{(1)} - t_{n-1}^{(1)}] \bmod m_{n-1} \ 0 \ [a_{n+1}^{(1)} - t_{n+1}^{(1)}] \bmod m_{n+1} \} = [0 \ 0 \ a_3^{(2)} \ a_4^{(2)} \ \dots \ a_{i-1}^{(2)} \ a_i^{(2)} \ a_{i+1}^{(2)} \ \dots \ a_{n-3}^{(2)} \ a_{n-2}^{(2)} \ 0 \ 0 \ a_{n+1}^{(2)}]$.</p>	<p>Образование значений остатков $a_3^{(2)}$ и $a_{n-2}^{(2)}$ числа $A^{(2)} = [0 \ 0 \ a_3^{(2)} \ \dots \ a_{i-1}^{(2)} \ a_i^{(2)} \ a_{i+1}^{(2)} \ \dots \ a_{n-3}^{(2)} \ a_{n-2}^{(2)} \ 0 \ 0 \ a_{n+1}^{(2)}]$ в виде $a_3^{(2)} = t_3^{(2)} = [a_3^{(1)} - a_2^{(1)}] \bmod m_3$ и $a_{n-2}^{(2)} = t_{n-2}^{(2)} = [a_{n-2}^{(1)} - a_{n-1}^{(1)}] \bmod m_{n-2}$.</p>
⋮	⋮	⋮
Для значения $A^{(i)}$	<p>Выполнение операции вычитания $A^{(i)} = A^{(i-1)} - KH^{(i-1)} = [0 \ 0 \ 0 \ \dots \ 0 \ a_i^{(i-1)} \ a_{i+1}^{(i-1)} \ \dots \ 0 \ 0 \ a_{n+1}^{(i-1)}] - [0 \ 0 \ 0 \ \dots \ 0 \ t_i^{(i-1)} \ t_{i+1}^{(i-1)} \ \dots \ 0 \ 0 \ t_{n+1}^{(i-1)}] = \{ 0 \ 0 \ 0 \ \dots \ 0 \ [a_i^{(i-1)} - t_i^{(i-1)}] \bmod m_i \ [a_{i+1}^{(i-1)} - t_{i+1}^{(i-1)}] \bmod m_{i+1} \ \dots \ [a_{i+2}^{(i-1)} - t_{i+2}^{(i-1)}] \bmod m_{i+2} \ \dots \ [a_{n-i-1}^{(i-1)} - t_{n-i-1}^{(i-1)}] \bmod m_{n-i-1} \ [a_{n-i}^{(i-1)} - t_{n-i}^{(i-1)}] \bmod m_{n-i} \ [a_{n-i+1}^{(i-1)} - t_{n-i+1}^{(i-1)}] \bmod m_{n-i+1} \ 0 \ 0 \ \dots \ 0 \ [a_{n+1}^{(i-1)} - t_{n+1}^{(i-1)}] \bmod m_{n+1} \} = [0 \ 0 \ 0 \ \dots \ 0 \ 0 \ a_{i+1}^{(i)} \ a_{i+2}^{(i)} \ \dots \ a_{n-i-1}^{(i)} \ a_{n-i}^{(i)} \ 0 \ 0 \ \dots \ 0 \ 0 \ a_{n+1}^{(i)}]$.</p>	<p>Обращение по значениям остатков $a_{i+1}^{(i)}$ и $a_{n-i}^{(i)}$ числа $A^{(i)} = [0 \ 0 \ 0 \ \dots \ 0 \ 0 \ a_{i+1}^{(i)} \ a_{i+2}^{(i)} \ \dots \ a_{n-i-1}^{(i)} \ a_{n-i}^{(i)} \ 0 \ 0 \ \dots \ 0 \ 0 \ a_{n+1}^{(i)}]$ в БКН_i за константой нулевизации $KH^{(i)} = [0 \ 0 \ 0 \ \dots \ 0 \ t_{i+1}^{(i)} \ t_{i+2}^{(i)} \ \dots \ t_{n-i-1}^{(i)} \ t_{n-i}^{(i)} \ 0 \ 0 \ \dots \ 0 \ 0 \ t_{n+1}^{(i)}]$; $t_{i+1}^{(i)} = a_{i+1}^{(i)}$, $t_{n-i}^{(i)} = a_{n-i}^{(i)}$; $t_{i+1}^{(i)} = \overline{0, m_{i+1} - 1}$; $t_{n-i}^{(i)} = \overline{0, m_{n-i} - 1}$.</p>
Для значения $A^{(i+1)}$	<p>Выполнение операции вычитания $A^{(i+1)} = A^{(i)} - KH^{(i)} = [0 \ 0 \ 0 \ \dots \ 0 \ a_{i+1}^{(i)} \ a_{i+2}^{(i)} \ a_{i+3}^{(i)} \ \dots \ a_{n-i-2}^{(i)} \ a_{n-i-1}^{(i)} \ a_{n-i}^{(i)} \ 0 \ \dots \ 0 \ 0 \ a_{n+1}^{(i)}] - [0 \ 0 \ 0 \ \dots \ 0 \ t_{i+1}^{(i)} \ t_{i+2}^{(i)} \ t_{i+3}^{(i)} \ \dots \ t_{n-i-2}^{(i)} \ t_{n-i-1}^{(i)} \ t_{n-i}^{(i)} \ 0 \ \dots \ 0 \ 0 \ t_{n+1}^{(i)}] = \{ 0 \ 0 \ 0 \ \dots \ 0 \ [a_{i+1}^{(i)} - t_{i+1}^{(i)}] \bmod m_{i+1} \ [a_{i+2}^{(i)} - t_{i+2}^{(i)}] \bmod m_{i+2} \ \dots \ [a_{n-i-2}^{(i)} - t_{n-i-2}^{(i)}] \bmod m_{n-i-2} \ [a_{n-i-1}^{(i)} - t_{n-i-1}^{(i)}] \bmod m_{n-i-1} \ [a_{n-i}^{(i)} - t_{n-i}^{(i)}] \bmod m_{n-i} \ 0 \ \dots \ 0 \ 0 \ [a_{n+1}^{(i)} - t_{n+1}^{(i)}] \bmod m_{n+1} \}$.</p>	<p>Образование значений остатков $a_{i+2}^{(i+1)}$ и $a_{n-i-1}^{(i+1)}$ числа $A^{(i+1)} = [0 \ 0 \ 0 \ \dots \ 0 \ 0 \ 0 \ a_{i+2}^{(i+1)} \ a_{i+3}^{(i+1)} \ \dots \ a_{n-i-2}^{(i+1)} \ a_{n-i-1}^{(i+1)} \ 0 \ 0 \ \dots \ 0 \ 0 \ a_{n+1}^{(i+1)}]$ в виде</p>

	$\ [a_{i+2}^{(i)} - t_{i+2}^{(i)}] \bmod m_{i+2} \ [a_{i+3}^{(i)} - t_{i+3}^{(i)}] \bmod m_{i+3} \ \dots$ $\dots \ [a_{n-i-2}^{(i)} - t_{n-i-2}^{(i)}] \bmod m_{n-i-2} \ [a_{n-i-1}^{(i)} - t_{n-i-1}^{(i)}] \bmod m_{n-i-1} \ $ $\ [a_{n-i}^{(i)} - t_{n-i}^{(i)}] \bmod m_{n-i} \ 0 \ \dots \ 0 \ [a_{n+1}^{(i)} - t_{n+1}^{(i)}] \bmod m_{n+1} \} =$ $= [0 \ 0 \ \dots \ 0 \ 0 \ a_{i+2}^{(i+1)} \ a_{i+3}^{(i+1)} \ \dots \ a_{n-i-2}^{(i+1)} \ a_{n-i-1}^{(i+1)} \ 0 \ 0 \ \dots$ $\dots \ 0 \ 0 \ a_{n+1}^{(i+1)}].$	$a_{i+2}^{(i+1)} = t_{i+2}^{(i+1)} = [a_{i+2}^{(i)} - a_{i+1}^{(i)}] \bmod m_{i+2} \text{ и}$ $a_{n-i-1}^{(i+1)} = t_{n-i-1}^{(i+1)} = [a_{n-i-1}^{(i)} - a_{n-i-2}^{(i)}] \bmod m_{n-i-1}.$
	<p>Обращение по значениям остатков $a_{i+2}^{(i+1)}$ и $a_{n-i-1}^{(i+1)}$ числа</p> $A^{(i+1)} = [0 \ 0 \ \dots \ 0 \ 0 \ a_{i+2}^{(i+1)} \ a_{i+3}^{(i+1)} \ \dots \ a_{n-i-2}^{(i+1)} \ a_{n-i-1}^{(i+1)} \ 0 \ $ $\ 0 \ \dots \ 0 \ 0 \ a_{n+1}^{(i+1)}] \text{ в БКН}_{i+1} \text{ за константой нулевизации}$ $KH^{(i+1)} = [0 \ 0 \ \dots \ 0 \ 0 \ t_{i+2}^{(i+1)} \ t_{i+3}^{(i+1)} \ \dots \ t_{n-i-2}^{(i+1)} \ t_{n-i-1}^{(i+1)} \ 0 \ $ $\ 0 \ \dots \ 0 \ 0 \ t_{n+1}^{(i+1)}]; t_{i+2}^{(i+1)} = a_{i+2}^{(i+1)}, t_{n-i-1}^{(i+1)} = a_{n-i-1}^{(i+1)};$ $t_{i+2}^{(i+1)} = \overline{0, m_{i+2} - 1}, t_{n-i-1}^{(i+1)} = \overline{0, m_{n-i-1} - 1}.$	<p>Образование значений остатков $a_{i+3}^{(i+2)}$ и $a_{n-i-2}^{(i+2)}$ числа</p> $A^{(i+2)} = [0 \ 0 \ \dots \ 0 \ 0 \ $ $\ a_{i+3}^{(i+2)} \ a_{i+4}^{(i+2)} \ \dots \ a_{n-i-3}^{(i+2)} \ a_{n-i-2}^{(i+2)} \ 0 \ 0 \ \dots$ $\dots \ 0 \ 0 \ a_{n+1}^{(i+2)}] \text{ в виде}$ $a_{i+3}^{(i+2)} = t_{i+2}^{(i+2)} = [a_{i+3}^{(i+1)} - a_{i+2}^{(i+1)}] \bmod m_{i+3} \text{ и}$ $a_{n-i-2}^{(i+2)} = t_{n-i-2}^{(i+2)} = [a_{n-i-2}^{(i+1)} - a_{n-i-3}^{(i+1)}] \bmod m_{n-i-2}.$
⋮	⋮	⋮
$k-2$	<p>Обращение по значениям остатков $a_{n/2+2}^{(n/2-2)}$ и $a_{n/2-1}^{(n/2-2)}$ числа</p> $A^{(n/2-2)} = [0 \ 0 \ \dots \ 0 \ 0 \ a_{n/2+2}^{(n/2-2)} \ a_{n/2-1}^{(n/2-2)} \ a_{n/2+1}^{(n/2-2)} \ a_{n/2+2}^{(n/2-2)} \ $ $\ 0 \ 0 \ \dots \ 0 \ 0 \ a_{n+1}^{(n/2-2)}] \text{ в БКН}_{n/2-2} \text{ за константой нулевизации}$ $KH^{(n/2-2)} = [0 \ 0 \ \dots \ 0 \ 0 \ t_{n/2+2}^{(n/2-2)} \ t_{n/2-1}^{(n/2-2)} \ \dots$ $\dots \ t_{n/2+1}^{(n/2-2)} \ t_{n/2+2}^{(n/2-2)} \ 0 \ 0 \ \dots \ 0 \ 0 \ t_{n+1}^{(n/2-2)}];$ $t_{n/2-1}^{(n/2-2)} = a_{n/2-1}^{(n/2-2)}, t_{n/2+2}^{(n/2-2)} = a_{n/2+2}^{(n/2-2)}; t_{n/2-1}^{(n/2-2)} = \overline{0, m_{n/2-1} - 1},$ $t_{n/2+2}^{(n/2-2)} = \overline{0, m_{n/2+2} - 1}.$	<p>Образование значений остатков $a_{n/2}^{(n/2-1)}$ и $a_{n/2+1}^{(n/2-1)}$ числа</p> $A^{(n/2-1)} = [0 \ 0 \ \dots \ 0 \ 0 \ $ $\ a_{n/2}^{(n/2-1)} \ a_{n/2+1}^{(n/2-1)} \ 0 \ 0 \ \dots \ 0 \ 0 \ a_{n+1}^{(n/2-1)}]$ <p>в виде $a_{n/2}^{(n/2-1)} = t_{n/2}^{(n/2-1)} =$</p> $= [a_{n/2}^{(n/2-2)} - a_{n/2-1}^{(n/2-2)}] \bmod m_{n/2} \text{ и } a_{n/2+1}^{(n/2-1)} =$ $= t_{n/2+1}^{(n/2-1)} = [a_{n/2+1}^{(n/2-2)} - a_{n/2}^{(n/2-2)}] \bmod m_{n/2+1}.$
$k-1$	<p>Выполнение операции вычитания</p> $A^{(n/2-1)} = A^{(n/2-2)} - KH^{(n/2-2)} = [0 \ 0 \ \dots \ 0 \ 0 \ a_{n/2-1}^{(n/2-2)} \ $ $\ a_{n/2}^{(n/2-2)} \ a_{n/2+1}^{(n/2-2)} \ a_{n/2+2}^{(n/2-2)} \ 0 \ 0 \ \dots \ 0 \ 0 \ a_{n+1}^{(n/2-2)}] - [0 \ $ $\ 0 \ \dots \ 0 \ 0 \ t_{n/2-1}^{(n/2-2)} \ t_{n/2}^{(n/2-2)} \ t_{n/2+1}^{(n/2-2)} \ t_{n/2+2}^{(n/2-2)} \ 0 \ 0 \ \dots$ $\dots \ t_{n+1}^{(n/2-2)}] = \{0 \ 0 \ \dots \ 0 \ 0 \ [a_{n/2-1}^{(n/2-2)} - t_{n/2-1}^{(n/2-2)}] \bmod m_{n/2-1} \ $ $\ [a_{n/2}^{(n/2-2)} - t_{n/2}^{(n/2-2)}] \bmod m_{n/2} \ \dots$ $\ [a_{n/2+1}^{(n/2-2)} - t_{n/2+1}^{(n/2-2)}] \bmod m_{n/2+1} \ $ $\ [a_{n/2+2}^{(n/2-2)} - t_{n/2+2}^{(n/2-2)}] \bmod m_{n/2+2} \ 0 \ 0 \ \dots \ 0 \ 0 \ $ $\ [a_{n+1}^{(n/2-2)} - t_{n+1}^{(n/2-2)}] \bmod m_{n+1} \} = [0 \ 0 \ \dots \ 0 \ 0 \ 0 \ a_{n/2}^{(n/2-1)} \ $ $\ a_{n/2+1}^{(n/2-1)} \ 0 \ 0 \ \dots \ 0 \ 0 \ a_{n+1}^{(n/2-1)}].$	<p>Обращение по значениям остатков $a_{n/2}^{(n/2-1)}$ и $a_{n/2+1}^{(n/2-1)}$ числа $A^{(n/2-1)} = [0 \ 0 \ \dots$</p> $\dots \ 0 \ 0 \ a_{n/2}^{(n/2-1)} \ a_{n/2+1}^{(n/2-1)} \ 0 \ 0 \ \dots \ 0 \ 0 \ $ $0 \ a_{n+1}^{(n/2-1)}]; \text{ в БКН}_{n/2-1} \text{ за константой нулевизации } KH^{(n/2-1)} = [0 \ 0 \ \dots \ 0 \ 0 \ $ $\ t_{n/2}^{(n/2-1)} \ t_{n/2+1}^{(n/2-1)} \ 0 \ 0 \ \dots \ 0 \ 0 \ t_{n+1}^{(n/2-1)}];$ $t_{n/2}^{(n/2-1)} = a_{n/2}^{(n/2-1)}, t_{n/2+1}^{(n/2-1)} = a_{n/2+1}^{(n/2-1)};$ $t_{n/2}^{(n/2-1)} = \overline{0, m_{n/2} - 1}, t_{n/2+1}^{(n/2-1)} = \overline{0, m_{n/2+1} - 1}.$
k	<p>Получение нулевизируемого $A^{(H)}$ числа. Выполнение операции $A^{(n/2-1)} = A^{(n/2-2)} - KH^{(n/2-2)} =$</p> $= [0 \ 0 \ \dots \ 0 \ a_{n/2}^{(n/2-1)} \ a_{n/2+1}^{(n/2-1)} \ 0 \ \dots \ 0 \ a_{n+1}^{(n/2-1)}] - [0 \ 0 \ \dots \ 0 \ t_{n/2}^{(n/2-1)} \ t_{n/2+1}^{(n/2-1)} \ 0 \ \dots \ 0 \ t_{n+1}^{(n/2-1)}] =$ $= [0 \ 0 \ \dots \ 0 \ [a_{n/2}^{(n/2-1)} - t_{n/2}^{(n/2-1)}] \bmod m_{n/2} \ [a_{n/2+1}^{(n/2-1)} - t_{n/2+1}^{(n/2-1)}] \bmod m_{n/2+1} \ 0 \ \dots \ 0 \ $ $\ [a_{n+1}^{(n/2-1)} - t_{n+1}^{(n/2-1)}] \bmod m_{n+1} = [0 \ 0 \ \dots \ 0 \ \dots \ 0 \ 0 \ (\gamma_{n+1} = a_{n+1}^{(n/2)})].$	
$T_{H4} = \left[\left[\frac{n+1}{2} \right] + \left[\frac{n+1}{2} + 1 \right] \right] \cdot \tau_{ca}$		

Рис. 5. Метод контроля данных ПНН ОПО

Выведенные выражения (3), (4) и (7) являются рабочими формулами для оценки быстродействия реализации процедуры нулевизации в зависимости от величин значений n и $\tau_{сл}$. В практических расчетах, для определения времени контроля данных в СОК рекомендовано пользоваться выражениями (5), (6), (8)

четыре метода контроля данных представлены в таблице 3. На основании данные таблицы 4, в таблице 5 представлены результаты расчета характеристик рассмотренных методов контроля для l -байтовых ($l = \overline{1, 4, 8}$) разрядных сеток вычислителя в СОК.

Таблица 3

Основные характеристики методов контроля данных в СОК

Методы контроля данных H_i		Время нулевизации T_{H_i}	Количество констант нулевизации K_{H_i}	Количество двоичных разрядов констант N_{H_i}
H1	Метод последовательной нулевизации	$T_{H1} = 2 \cdot n \cdot \tau_{сл}$	$K_{H1} = \sum_{i=1}^n (m_i - 1)$	$N_{H1} = \sum_{i=1}^n (m_i - 1) \cdot (n - i + 1)$
H2	Метод последовательной нулевизации с определением последующего остатка	$T_{H2} = \left(\left[\frac{n-1}{2} \right] + n \right) \cdot \tau_{сл}$	$K_{H2} = \sum_{i=1}^{n-1} (m_i - 1)$	$N_{H2} = \sum_{i=1}^{n-1} (m_i - 1) \cdot (n - i)$
H3	Метод параллельной нулевизации	$T_{H3} = n \cdot \tau_{сл}$	$K_{H3} = \sum_{i=1}^{\left[\frac{n}{2} \right]} (m_i \cdot m_{n-i+1} - 1)$	$N_{H3} = \sum_{i=1}^{\left[\frac{n}{2} \right]} (m_i \cdot m_{n-i+1} - 1) \cdot (n - 2 \cdot i + 1)$
H4	Метод параллельной нулевизации с определением последующих остатков	$T_{H4} = \left(\left[\frac{n+1}{2} \right] + \left[\frac{\left[\frac{n+1}{2} \right]}{2} \right] \right) \cdot \tau_{сл}$	$K_{H4} = \sum_{i=1}^{\left[\frac{n}{2} \right]} (m_i \cdot m_{n-i+1} - 2)$	$N_{H4} = \sum_{i=1}^{\left[\frac{n}{2} \right]} (m_i \cdot m_{n-i+1} - 2) \cdot (n - 2 \cdot i + 1)$

Таблица 4

Совокупность оснований СОК для l -байтовых ($l = \overline{1, 4, 8}$) разрядных сеток вычислителя

Величина разрядной сетки $l(n)$	Информационные основания СОК $\{m_i\}, i = \overline{1, n}$	Контрольное основание СОК m_{n+1}
1(4)	$m_1 = 3, m_2 = 4, m_3 = 5, m_4 = 7$	$m_5 = 11$
2(6)	$m_1 = 2, m_2 = 5, m_3 = 7, m_4 = 9, m_5 = 11, m_6 = 13$	$m_7 = 17$
3(8)	$m_1 = 3, m_2 = 4, m_3 = 5, m_4 = 7, m_5 = 11, m_6 = 13, m_7 = 17, m_8 = 19$	$m_9 = 23$
4(10)	$m_1 = 2, m_2 = 3, m_3 = 5, m_4 = 7, m_5 = 11, m_6 = 13, m_7 = 17, m_8 = 19, m_9 = 23, m_{10} = 29$	$m_{11} = 31$
8(16)	$m_1 = 3, m_2 = 4, m_3 = 5, m_4 = 7, m_5 = 11, m_6 = 13, m_7 = 17, m_8 = 19, m_9 = 23, m_{10} = 29, m_{11} = 31, m_{12} = 37, m_{13} = 41, m_{14} = 43, m_{15} = 47, m_{16} = 53$	$m_{17} = 59$

Таблица 5

Расчетные данные характеристик методов контроля данных в СОК для l -байтовых ($l = \overline{1, 4, 8}$) разрядных сеток

$l(n)$	Н1			Н2			Н3			Н4		
	$\frac{T_{H1}}{\tau_{сл}}$	K_{H1}	N_{H1}	$\frac{T_{H2}}{\tau_{сл}}$	K_{H2}	N_{H2}	$\frac{T_{H3}}{\tau_{сл}}$	K_{H3}	N_{H3}	$\frac{T_{H4}}{\tau_{сл}}$	K_{H4}	N_{H4}
1(4)	8	15	31	5	9	16	4	39	79	3	37	75
2(6)	12	41	106	8	29	65	6	141	349	4	138	340
3(8)	16	71	217	11	53	146	8	263	995	6	259	979
4(10)	20	119	412	14	91	293	10	479	1955	7	474	1930
8(16)	32	367	1947	23	315	1580	16	2581	16493	12	2573	16429

Для простоты и удобства проведения сравнительного анализа эффективности использования представленных методов контроля, обобщенные данные таблицы 5 целесообразно разместить в трех (по числу характеристик) отдельных таблицах (табл. 6 – 8).

Таблица 6

Временные характеристики методов контроля

$l(n)$	$\frac{T_{H_i}}{\tau_{сл}}$			
	Н1	Н2	Н3	Н4
1 (4)	8	5	4	3
2 (6)	12	8	6	4
3 (8)	16	11	8	6
4 (10)	20	14	10	7
8 (16)	32	23	16	12

Таблица 7

Характеристики методов контроля

$l(n)$	K_{H_i}			
	Н1	Н2	Н3	Н4
1 (4)	15	9	39	37
2 (6)	41	29	141	138
3 (8)	71	53	263	259
4 (10)	119	91	479	474
8 (16)	367	315	2581	2573

Таблица 8

Характеристики методов контроля

$l(n)$	N_{H_i}			
	Н1	Н2	Н3	Н4
1 (4)	31	16	79	75
2 (6)	106	65	349	340
3 (8)	217	146	995	979
4(10)	412	293	1955	1930
8(16)	1947	1580	16493	16429

На основе табл. 8 составлена таблица 9 данных сравнительного анализа эффективности применения метода оперативного контроля ПНН ОПО в сравне-

нии с существующими методами контроля в СОК по быстрдействию реализации процедуры нулевизации.

Таблица 9

Данные сравнительного анализа времени контроля данных в СОК

n	$T_{H_i} = T/\tau$				Выигрыш в [%]		
	T_{H1}	T_{H2}	T_{H3}	T_{H4}	K_{H1}	K_{H2}	K_{H3}
4	8	5	4	3	62	40	25
6	12	8	6	4	66	55	33
8	16	11	8	6	62	45	25
10	20	14	10	7	65	53	30
16	32	23	16	12	62	47	25

Коэффициент K_{H_i} эффективности использования метода контроля ПНН ОПО, в сравнении с существующими методами нулевизации, определяется соотношением $K_{H_i} = \frac{T_{H1} - T_{H_i}}{T_{H1}} \cdot 100\%$ ($i = \overline{1, 3}$). Из таблицы

9 видна высокая эффективность предложенного в статье метода оперативного контроля (Н4), основанного на реализации процедуры параллельной нулевизации с определением последующих остатков.

ЗАКЛЮЧЕНИЕ

В статье исследованы методы оперативного контроля данных, представленных в СОК. Даны аналитические соотношения для оценки времени контроля данных, а также для расчета количества аппаратных затрат для его реализации. В качестве метода контроля данных, обеспечивающего максимальную оперативность контроля в СОК (минимальное время контроля) рекомендован метод (Н4) параллельной нулевизации с определением последующих остатков. В качестве метода контроля в СОК обеспечивающего минимальное количество аппаратных затрат для реализации процедуры нулевизации чисел рекомендован метод (Н2) последовательной нулевизации с определением последующего остатка.

Литература

[1] Акушкин И. Я., Юдицкий Д. И. Машинная арифмети-

ка в остаточных классах – М.: Сов. Радио, 1968. – 440с.

- [2] Мороз С. А., Краснобаев В. А. Методы контроля, диагностики и коррекции ошибок данных в информационно-телекоммуникационной системе, функционирующей в классе вычетов // Інформаційно-керуючі системи на залізничному транспорті. - 2012. - № 2. - С. 60 – 78.
- [3] Krasnobayev V. A., Koshman S. A., Mavrina M. A. A method for increasing the reliability of verification of data represented in a residue number system // Cybernetics and Systems Analysis. – November 2014. – Volume 50, Issue 6, pp 969-976.
- [4] V. A. Krasnobayev. Method for Realization of Transformations in Public-Key Cryptography // Telecommunications and Radio Engineering (USA), 2007, Vol. 66, Issue 17, pp. 1559-1572.
- [5] S. A. Koshman and V. A. Krasnobayev. Method of Realization of Cryptographic RSA Transformations on the Basis of Application of Modular Number System // Biomedical Soft Computing and Human Sciences (JAPAN), 2011, Vol.17, No.2, pp. 31-36.
- [6] Краснобаев В.А., Кошман С. А., Маврина М. А. Метод исправления однократных ошибок данных, представленных кодом класса вычетов // Электрон. Моделирование. 2013. – Т. 35, № 5. – С. 43–56.
- [7] Сиора А.А., Краснобаев В.А., Харченко В.С. Отказоустойчивые системы с версионно-информационной избыточностью в АСУ ТП: Монография.-Х.: МОН, НАУ им. Н.Е. Жуковского (ХАИ), 2009. – 320с.
- [8] Naumenko, N.I., Stasev, Yu.V., Kuznetsov, A.A. Methods of synthesis of signals with prescribed properties // Cybernetics and Systems Analysis, Volume 43, Issue 3, May 2007, Pages 321 – 326.
- [9] Oliynykov R., Gorbenko I., Dolgov V., Kaidalov D. Improvement for distinguisher efficiency of the 3-round Feistel network and a random permutation // Proceedings of the 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS'2011, 2011. – P. 743 – 746.



Краснобаев Виктор Анатольевич, доктор технических наук, профессор, профессор кафедры электроники и управляющих систем ХНУ им. В.Н. Каразина. Область научных интересов: теория и практика создания компьютерных систем и компонентов в системе остаточных классов.



Кошман Сергей Александрович, кандидат технических наук, доцент, доцент кафедры автоматизации и компьютерно-интегрированных технологий ХНТУСХ им. П. Василенка. Область научных интересов: теория и практика помехоустойчивого кодирования данных в системе остаточных классов.



Янко Алина Сергеевна, кандидат технических наук, ассистент кафедры компьютерной инженерии Полтавского национального технического университета им. Ю. Кондратюка. Область научных интересов: методы быстрой обработки данных в системе остаточных классов.

УДК: 681.142

Методи оперативного контролю даних у системі залишкових класів, що засновані на принципі паралельної нулевізації / В.А. Краснобаєв, С.О. Кошман, А.С. Янко // Прикладна радіоелектроніка: наук.-техн. журнал. – 2016. – Том 15, № 3. – С. 253 – 265.

У статті розглянуто методи оперативного контролю даних у системі залишкових класів (СЗК), що засновані на принципі паралельної нулевізації. Сутність першого запропонованого методу контролю полягає у тому, що процедура нулевізації здійснюється паралельно у часі одночасно за двома основами СЗК. З метою зменшення часу контролю даних у СЗК у статті запропоновано другий метод контролю, що заснований на використанні процедури паралельної нулевізації з визначенням подальших залишків непозиційної кодової структури. У статті представлені дані розрахунку і порівняльного аналізу основних характеристик методів контролю даних у СЗК.

Ключові слова: система залишкових класів (СЗК), метод контролю даних у СЗК, принцип і процедура паралельної нулевізації, непозиційна кодова структура.

Табл.: 09. Іл.: 05. Бібліогр.: 09 найм.

UDC 681.142

Methods of data control in a residual class system that are based on the principle of parallel nulevisation / V.A. Krasnobayev, S.A. Koshman, A.S. Yanko // Applied Radio Electronics: Sci. Journ. – 2016. – Vol. 15, № 3. – P. 253 – 265.

The methods of operational data control in a residual class system (RCS), based on the principle of parallel nulevisation are considered in the paper. The essence of the first proposed method of control is that the procedure of nulevisation is carried out in parallel time simultaneously on two RCS bases. The second control method based on the use of parallel nulevisation procedures with subsequent determination of residues of a nonpositional code structure is proposed in the paper for the minimization of the time of data control in the RCS. The calculation data and comparative analysis of the main characteristics of these control methods in the RCS are discussed in the paper.

Keywords: residual class system (RCS), data control method in RCS, principle and procedure of parallel nulevisation, nonpositional code structure.

Tab.: 09. Fig.: 05. Ref.: 09 items.

ЗЕЛЕНСКИЙ АЛЕКСАНДР АЛЕКСЕЕВИЧ (24.06 1943 – 15. 05. 2016)



Ушел из жизни академик АН ПРЭ, выдающийся специалист в области радиолокации, доктор технических наук, профессор, заведующий кафедрой Приема, передачи и обработки сигналов, директор и научный руководитель Научно-технического центра радиоэлектронных и медицинских приборов и технологий ХАИ.

Им опубликовано 2 монографии, множество учебных пособий, более 300 научных статей, 100 авторских свидетельств и патентов СССР, Украины и Финляндии.

Сотрудникам и друзьям Александр Алексеевич запомнился мудрым Наставником и Учителем, всесторонне образованным, интеллигентным человеком, готовым помочь и советом, и делом в сложных рабочих и житейских ситуациях, руководителем, заботящимся о коллективе, умеющим стимулировать работу без крика и суеты, как бы по твоему же желанию, исключительно доброжелательным и справедливым. На руководимой им кафедре свято соблюдалось правило «приказ Зеленского еще как-то можно проигнорировать, но просьбу – никогда». Многих он вывел в люди, многим помог, не прося о благодарности и искренне радуясь успехам других. Успевал позаботиться о семье, о друзьях, о коллективе, но не всегда – о себе и своем здоровье.

Мы всегда будем помнить о нем, о руководимом им докторском Совете, который теперь единогласно принятым решением носит имя Александра Алексеевича Зеленского.

Друзья, коллеги, ученики.

ПРИКЛАДНАЯ РАДИОЭЛЕКТРОНИКА

Научно-технический журнал

Ответственный секретарь

Я. В. Сашкова

Корректор

Б. П. Косиковская

Перевод на английский язык

К. Т. Умяров

Компьютерный дизайн и верстка

Я. В. Сашкова

Рекомендовано засіданням Бюро Президії Академії наук прикладної радіоелектроніки
(протокол № 3 від 28.09.2016 р.).

Свідоцтво про державну реєстрацію КВ № 6037 від 09.04.2002 р.

Журнал включений до списку фахових видань ВАК України
з технічних наук
(постанова президії ВАК України № 1-05/2 від 10.03.2010),
з фізико-математичних наук (фізика)
(постанова президії ВАК України № 1-05/5 від 1.07.2010)

Підписано до друку 28.09.2016. Формат 60 × 84 ¹/₈.
Папір офсет. Друк офсет. Умов.-друк. арк. 9,8. Облік.-вид. арк. 9,2.
Тираж 90 прим. Ціна договірна.

Віддруковано в ТОВ «ДРУКАРНЯ МАДРИД»
61024, м. Харків, вул. Максимільянівська, 11. Тел.: (057) 756-53-25
www.madrid.in.ua, e-mail: info@madrid.in.ua