Этап 8 – формирование результата. Этот этап направлен на получение выходных данных, характеризующих аномальное состояние. На основе сформированных множеств возможных атак (см. этап 3) и наборов логико-лингвистических связок (см. этап 6), формируется множество пар – "атака→набор логико-лингвистических связок" $\boldsymbol{AT \rightarrow LC = (} \bigcup_{i=1}^{n} AT_i \rightarrow \bigcup_{j=1}^{c_i} LC_{ij} \boldsymbol{)}$ [5]. Посредством этого множества, сформированных ЭП и множества ***LI*** (см. этап 6), с помощью процедуры логического вывода (функционирующей на основе выбранных по решению эксперта МНА и МСФП) определяются конкретные значения лингвистических идентификаторов, характеризующих уровень аномального состояния, который может быть порожден конкретной кибератакой. Другими словами каждому AT_i присваивается один из LI_i. Так, например, атакам $AT_1=SN$, $AT_2=DS$ и $AT_3=SP$ соответственно будет определен уровень Н, БНВ и В. После определения этих результатов осуществляется их визуализация в виде эталонных лингвистических термов, на фоне которых идентифицируется значение переменной, характеризующей текущее состояние системы относительно аномалий.

Предложенный в работе метод базируется на математических моделях и методах нечеткой логики, и содержит восемь базовых этапов, раскрывающих процесс выявления аномального состояния, порождаемого определенным типом кибератак в ИС. На основе этого метода можно создавать или усовершенствовать реальные системы выявления аномалий, порожденных атакующими действиями в компьютерных сетях.

ЛИТЕРАТУРА

1. Корченко О. Г. Построение систем защиты информации на нечетких множествах [Текст] : Теория и практические решения / О. Г. Корченко. — К. : МК-Пресс, 2006. — 320 с.
2. Волянська В. В. Система виявлення аномалій на основі нечітких моделей [Текст] / В. В. Волянська, А. О. Корченко, Є. В. Паціра // Зб. наук. пр. Інституту проблем моделювання в енергетиці НАН України ім. Г. Є Пухова. — Львів : ПП «Системи, технології, інформаційні послуги», 2007. — [Спец. випуск]. — Т.2. — С. 56–60.
3. Корченко О. Г. Системи захисту інформації [Текст] : Монографія / О. Г. Корченко. — К. : НАУ, 2004. — 264 с.
4. Горніцька Д.А. Визначення коефіцієнтів важливості для експертного оцінювання у галузі інформаційної безпеки / Д.А. Горніцька, В.В. Волянська, А.О. Корченко // Захист інформації. — 2012. — №1 (54) . — С. 108-121.
5. Стасюк А.И. Базовая модель параметров для построения систем выявления атак / А.И. Стасюк, А.А. Корченко // Захист інформації. — 2012. — №2 (55). — С. 47-51.
6. Модели эталонов лингвистических переменных для систем выявления атак / М.Г. Луцкий, А.А. Корченко, А.В. Гавриленко, А.А Охрименко // Захист інформації. — 2012. — №2 (55). — С. 71-78
7. Корченко А.А. Модель эвристических правил на логико-лингвистических связках для обнаружения аномалий в компьютерных системах / А.А. Корченко // Захист інформації. — 2012. — №4 (57). — С. 109 -115 .

Надійшла: 24.10.2012 р.

Рецензент: д.т.н., професор Дудикевич В.Б.

УДК 003.26:004.056.55

Кінзерявий В.М., Гнатюк С.О., Кінзерявий О.М.

НОВІ ЕФЕКТИВНІ АЛГОРИТМИ ШИФРУВАННЯ ІНФОРМАЦІЇ

Для підвищення ефективності захисту електронних інформаційних ресурсів були розроблені два алгоритми шифрування на основі фіксованої таблиці підстановок з розширеною розрядністю і динамічних ключезалежних таблиць підстановок. Розроблені алгоритми мінімум у два рази швидші за вітчизняний стандарт шифрування ДСТУ ГОСТ 28147-2009 та практично стійкі до лінійного та диференційного криптоаналізу. Властивості псевдовипадкових послідовностей утворених за допомогою запропонованих алгоритмів шифрування (у режимі лічильника) були досліджені у середовищі статистичних тестів NIST STS, згідно яких вони пройшли комплексний контроль за методикою NIST STS і мають кращі результати за інші генератори.

Ключові слова: криптографія, алгоритми шифрування інформації, криптостійкість, лінійний та диференційний криптоаналіз.

Вступ. Відповідно до сучасного вітчизняного законодавства в галузі інформаційної безпеки держави, постійно зростають вимоги до захисту державних інформаційних ресурсів (ДІР), які передаються та зберігаються в державних інформаційно-комунікаційних системах. Особливої уваги потребує галузь цивільної авіації (ЦА), де циркулює значна кількість ДІР, а внутрішнє середовище швидко і суттєво змінюється із впровадженням сучасних інформаційних і комунікаційних технологій. Найбільшого захисту потребують ресурси критичних авіаційних інформаційних систем, наприклад, системи управління повітряним рухом, системи дистанційного технічного обслуговування, диспетчерські системи тощо [1-3]. Основними міжнародними документами, що регламентують процеси захисту ЦА від кіберзагроз є [1-3], вони містять вичерпний перелік заходів, яких необхідно вжити для мінімізації впливу кіберзагроз на ресурси критичних авіаційних інформаційних систем. Організація криптографічного захисту інформаційних ресурсів є одним із найбільш вагомих заходів. Криптографічний захист вважається одним із найбільш надійних та ефективних методів захисту інформації, його основною та беззаперечною перевагою є забезпечення захисту безпосередньо самих даних, а не доступу до них. Основним критерієм при виборі криптосистем є стійкість, проте для деяких завдань (наприклад, для шифрування великого об'єму даних, захисту онлайнових банківських платіжних систем тощо) ключову роль відіграє швидкість криптографічної обробки даних. Не зважаючи на велику різноманітність сучасних методів (алгоритмів) шифрування, далеко не всі володіють необхідним рівнем ефективності (швидкодії та стійкості). Крім того, стрімкий розвиток обчислювальних засобів та їх одночасне здешевлення формулюють нові вимоги як до стійкості, так і до швидкодії криптосистем – старі криптоалгоритми відходять у історію, а їх місце займають нові, які пройшли певний конкурсний відбір і довели свою спроможність забезпечити захист на певний період часу у майбутньому [4-10]. На даний момент, національним стандартом шифрування є ДСТУ ГОСТ 28147:2009. Проте, враховуючи значний прогрес у сфері методів і засобів криптоаналізу, він переходить в клас «морально застарілих» [9]. Так, відповідно до вимог проекту NESSIE, алгоритм ГОСТ 28147-2009 відноситься тільки до третього (найменшого) класу стійкості [8-9]. Крім того, ГОСТ 28147-2009 не відповідає, як мінімум, сучасним вимогам зі швидкості шифрування даних [7]. Також, його недоліками є складність апаратної реалізації та використання секретних довготривалих ключових даних, які постачаються в певному встановленому порядку [7]. Тому розробка нових алгоритмів шифрування для підвищення ефективності захисту інформації є актуальною науковою задачею. **Метою** роботи є підвищення ефективності захисту електронних інформаційних ресурсів за рахунок розробки нових блочних алгоритмів шифрування.

1. Спосіб підвищення швидкодії блочних шифрів. Розглянемо клас блочних шифрів з множиною відкритих (шифрованих) повідомлень $V_n = \{0,1\}^n$ ($n = 128 \cdot p$, $p \in N$), множиною раундових ключів $K = V_n$ та сімейством криптографічних перетворень: $F_k = f_r k_r \circ ... \circ f_1 k_1$, $k = (k_1, ..., k_r) \in K^r$, r – кількість раундів шифрування. Раундове перетворення для будь-яких $x \in V_n$, $k \in K$, $i \in \overline{1,r}$ описується так: $f_{i,k} = \begin{cases} \varphi(x+k), & i < r \\ s_r(x+k), & i = r \end{cases}$, де $+$ – це окремо визначена для кожного раунду операція додавання за модулем 2 або за модулем $2^{32 \cdot l}$ ($l \in N, l \le n/32$). Підстановки $\varphi(x)$ та $s(x)$ визначаються за формулами: $\varphi(x) = L(s_i(x)), x \in V_n$, $s_i(x) = (s_i'(x_{c-1}), ..., s_i'(x_0))$, де $x = (x_{c-1}, ..., x_0)$, $x_j \in V_t$, $t \ge 4$, $c = n/t$, $j \in \overline{0,c-1}$, s_i' – m таблиць підстановок на множині V_t, що використовуються в i-му раунді ($m \in \overline{1,c}$), а $L(x)$ - лінійне перетворення, яке використовують в блочному шифрі.

Для вищеописаного класу блочних шифрів справедливі відомі аналітичні верхні оцінки параметрів [11, 12], що характеризують практичну стійкість відносно методів диференційного [13, 14] та лінійного криптоаналізу [13, 15]:

$$EDP(\Omega) \le \Delta^{(r-1)B_L/2+1}, \quad (1)$$

$$ELP(\Omega) \le \Lambda^{(r-1)B_L/2+1}, \quad (2)$$

де $\Delta = \max\left\{d_+^{s'}(\alpha,\beta): \alpha,\beta \in V_t \setminus \{0\}\right\}$ – максимальна імовірність проходження різниці таблиці замін s' […], $\Lambda = \max\left\{l_+^{s'}(\alpha,\beta): \alpha,\beta \in V_t \setminus \{0\}\right\}$ – максимальна імовірність лінійної апроксимації таблиці замін s' [11], B_L – число галузей активізації лінійного перетворення $L(x)$ ($B_L = \min\left\{wt(x) + wt(xL^{-1})\right\}$) [11], $EDP(\Omega)$ – середня імовірність диференційної характеристики Ω [11], а $ELP(\Omega)$ – середня імовірність лінійної характеристики Ω [11].

Параметри $d_+^{s'}(\alpha,\beta)$ та $l_+^{s'}(\alpha,\beta)$ визначаються за наступними формулами (якщо у якості операції "+" використовується додавання за модулем 2) [11, 12]:

$$d_+^{s'}(\alpha,\beta) = 2^{-t}\sum_{k\in V_t}\delta(s'(k+\alpha)\oplus s'(k),\beta), \quad (3)$$

$$l_+^{s''}(\alpha,\beta) = 2^{-t}\sum_{k\in V_t}\left(2^{-t}\sum_{x\in V_t}(-1)^{\alpha x\oplus\beta s'(x+k)}\right)^2, \quad (4)$$

де δ - символ Кронекера ($\delta(u,v) = \begin{cases}0, u <> v\\ 1, u == v\end{cases}$).

Згідно відомої методології побудови блочних шифрів [16] можна, при використанні лінійних перетворень з більшим параметром B_L чи таблиць замін з меншими показниками Δ та Λ, обґрунтовано зменшити кількість раундів шифрування r блочного шифру, щоб при цьому забезпечувалась його практична стійкість до лінійного та диференційного криптоаналізу. На основі формул (1)-(2) було визначено мінімальну кількість раундів (див. табл. 1) при якій досягається практична стійкість до лінійного та диференційного криптоаналізу блочних шифрів (розглянутого класу) із довжиною секретного ключа та блоку даних 128 біт.

Таблиця 1

Мінімальна кількість раундів для забезпечення практичної стійкості до диференційного та лінійного криптоаналізу

МДР-коди, що покривають	Таблиці підстановок на множині		
	V_8 ($\Delta = \Lambda = 2^{-6}$)	V_{16} ($\Delta = \Lambda = 2^{-14}$)	V_{32} ($\Delta = \Lambda = 2^{-30}$)
4 байти ($B_L = 5$)	$r = 10$	$r = 5$	$r = 3$
8 байт ($B_L = 9$)	$r = 6$	$r = 3$	$r = 2$
16 байт ($B_L = 17$)	$r = 4$	$r = 2$	$r = 2$

У якості лінійного перетворення розглядалися МДР-коди (коди з максимально допустимою відстанню), що покривають w байт ($w = 4, 8, 16$) та дозволяють забезпечити число галузей активізації рівним $w+1$. Досліджувались таблиці замін на множині V_q (де $q = 8, 16, 32$), для яких теоретично досяжний рівень Δ та Λ дорівнює $2^{-(q-2)}$.

Згідно табл. 1, для підвищення швидкодії блочних шифрів (розглянутого класу) можна виконати наступне: 1) Розширити множину підстановок – використовувати таблиці замін на множині V_{16} (у більшості сучасних алгоритмів шифрування використовуються таблиці підстановок на множині V_8). Для використання однієї такої таблиці потрібно лише 128 КБ пам'яті, що зараз цілком допустимо. Таблиця замін на множині V_{32} потребують набагато більше пам'яті, тому на даний момент їх недоцільно використовувати. 2) Замінити одні

МДР-коди на інші, так щоб покривалась більша кількість байт (збільшиться кількість операцій в раунді, проте, не суттєво). 3) Зменшити кількість раундів блочного шифру, що призведе до підвищення його швидкодії. Наприклад, цей спосіб можна застосувати до алгоритмів шифрування Калина ($\Delta, \Lambda \ge 2^{-5}, B_L = 9, r = 11$) [11] та AES ($\Delta = \Lambda = 2^{-6}, B_L = 5, r = 11$) [4], але в такому випадку (при зменшенні r) потрібно досліджувати їх на стійкість вже до інших методів криптоаналізу, що є дуже непростим завданням. Проте така можливість підвищення їх швидкодії гідна уваги.

2. Нові алгоритми шифрування. На базі описаного способу підвищення швидкодії блочних шифрів були розроблені два алгоритми шифрування інформації на основі фіксованої таблиці підстановок з розширеною розрядністю (*Luna*) і динамічних ключезалежних таблиць підстановок (*Neptun*), псевдокод процедури шифрування яких наведений на рис. 1а та рис. 1б, відповідно. Дані алгоритми використовують 128 бітні блоки даних (представлені у вигляді 4×4 байтної матриці) з підтримкою секретного ключа довжиною 128, 256 та 512 бітів, з якого формується необхідна кількість 128-бітних розширених ключів, що представляються у вигляді матриці розміром 4×4 байт. Кількість раундів шифрування r залежить від довжини секретного ключа. При довжині секретного ключа у 128, 256 та 512 бітів у алгоритмі *Luna* $r = 7, 9, 13$, а у алгоритмі *Neptun* $r = 9, 13, 21$ відповідно. Під операцією $AddKeyMod2(state, subkey[i])$ мається на увазі побітове додавання за модулем 2 відповідних бітів розширеного ключа $subkey[i]$ та блоку даних $state$.

Luna

Input: 128-бітний вхідний блок даних $state$,
128-бітні розширені ключі $subkey[i]$, $i = \overline{0, r+2}$.
Output: 128-бітний вихідний блок даних.

1. $ExtraMix(state, subkey[0])$;
2. $AddKeyMod2(state, subkey[1])$;
3. $For \quad j = 0, \quad j < r-1, \quad j{+}{+} \quad do$
3.1. $SubBytes_{Luna}(state)$;
3.2. $ShiftRows(state)$;
3.3. $MixColumns(state)$;
3.4. $AddKeyMod2(state, subkey[j+2])$;
4. $SubBytes_{Luna}(state)$;
5. $ShiftRows(state)$;
6. $AddKeyMod2(state, subkey[r+1])$;
7. $InvExtraMix(state, subkey[r+2])$;
8. *return state*;

а)

Neptun

Input: 128-бітний вхідний блок даних $state$,
128-бітні розширені ключі $subkey[i]$, $i = \overline{0, 2\cdot r}$.
Output: 128-бітний вихідний блок даних.

1. $AddKeyMod2(state, subkey[0])$;
2. $For \quad j = 0, \quad j < r-1, \quad j{+}{+} \quad do$
2.1. $SubBytes_{Neptun}(state, subkey[2\cdot j+1])$;
2.2. $ShiftRows(state)$;
2.3. $MixColumns(state)$;
2.4. $AddKeyMod2(state, subkey[2\cdot j+2])$;
3. $SubBytes_{Neptun}(state, subkey[2\cdot r-1])$;
4. $ShiftRows(state)$;
5. $AddKeyMod2(state, subkey[2\cdot r])$;
6. *return state*;

б)

Рис. 1. Псевдокод процедури зашифровування алгоритмів шифрування *Luna* (а) та *Neptun* (б)

Операція $MixColumns(state)$ являє собою лінійне перетворення послідовності $state$. У даній операції блок даних $state$ розбивається на дві частини по 8 байт (перші два 4-байтних стовпчика утворюють одну 8-байтну частину, інші два − другу частину), кожна з яких розглядається як поліном над полем $GF(2^8)$ з 8 термами, який перемножують за модулем x^8+1 з фіксованим поліномом $c(x)$ степені 7 (див. рис. 2), що дозволяє забезпечити число галузей активізації 9. Поліном $c(x)$: $c(x) = 3x^7 + 7x^6 + x^5 + 3x^4 + 7x^3 + 4x^2 + 1Dx + 1$ (коефіцієнти

представлені в 16-ричній формі). У якості поліному, що не приводиться, обрано поліном: $m(x)=x^8+x^7+x^5+x^4+x+1$.

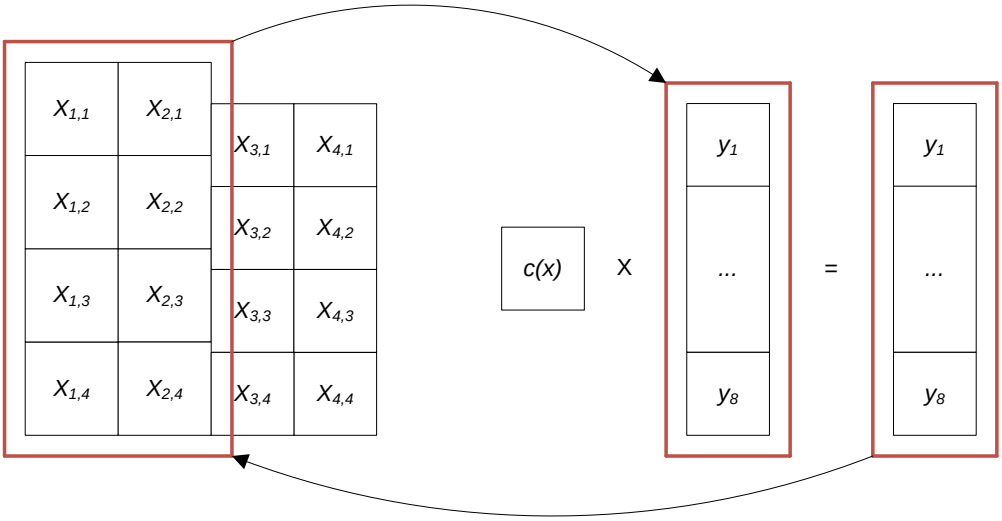
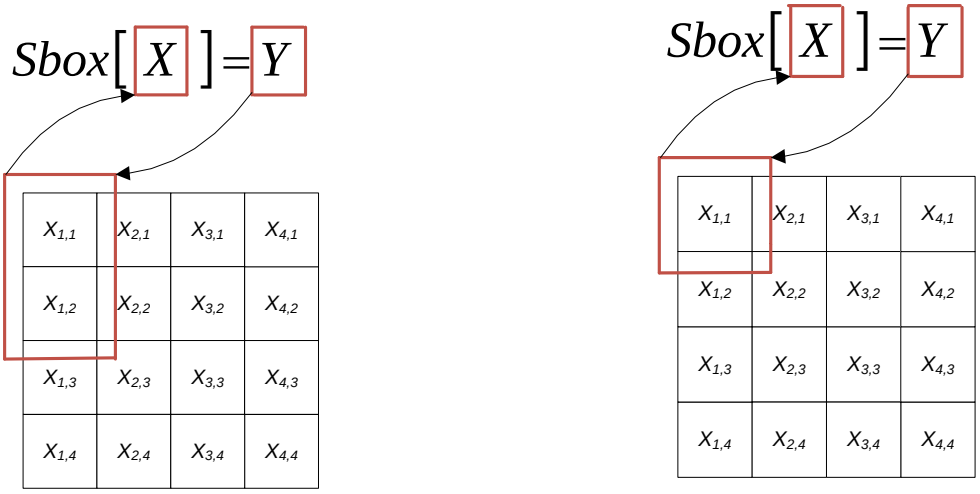


Рис.2. Схема виконання операції $MixColumns(state)$

У Операціях $SubBytes_{Luna}(state)$ і $SubBytes_{Neptun}(state, subkey[i])$ виконується таблична заміна відповідно кожних 16 та 8 біт блоку даних $state$ за визначеною таблицею замін (див. відповідно рис. 3а та 3б). У алгоритмі *Luna* використовується одна таблиця замін 16×16, а у алгоритмі *Neptun* − 16 таблиць, при чому вибір конкретної таблиці у кожному раунді залежить від розширеного ключа (використання динамічно змінюваних таблиць замін ускладнить його криптоаналіз та дозволить динамічно керувати процесом розсіювання інформації). Таблиці замін були побудовані таким чином, щоб були відсутні фіксовані точки, а також, щоб виконувались рівності для параметрів: $\Delta=\Lambda=2^{-14}$ для таблиці замін алгоритму *Luna* та $\Delta=\Lambda=2^{-6}$ для кожної таблиці замін алгоритму *Neptun*.



а) *б)*

Рис. 3. Схема виконання операції табличної заміни алгоритмів *Luna* (а) та *Neptun* (б)

Запропоновані таблиці замін побудовані за допомогою обрахунку зворотнього елементу поля $(C/X)^{-1}\in GF(2^q)$ з подальшим виконанням афінного перетворення над полем $GF(2)$: $S(X)=M\cdot(C/X)^{-1}+V$ (була обрана методика генерації таблиць замін, яка схожа на методики, що використовувалися при побудові таблиць замін алгоритмів *AES*, *ADE*, *Лабіринт*), де $X,C,V\in GF(2^q)$, а M − квадратна не вироджена матриця над полем $GF(2)$ розміром $q\times q$. Для алгоритму *Luna* $q=16$, а для *Neptun* − $q=8$.

Параметри C, V та M для таблиці підстановок алгоритмів *Luna* та *Neptun* відповідно наведені в шістнадцятеричному вигляді в табл. 2 та табл.3 (кожний рядок матриці M відображений у вигляді одного шістнадцятеричного числа).

Таблиця 2

Параметри C, V та M для побудови таблиці замін алгоритму шифрування *Luna*

M	C	V
{652, CA4, 1948, 3290, 6520, CA40, 9481, 2903, 5206, A40C, 4819, 9032, 2065, 40CA, 8194, 329}	1787	2544

Таблиця 3

Параметри C, V та M для кожної таблиці підстановок алгоритму шифрування *Neptun*

Індекс таблиці замін	M	C	V
1	{ 91, 23, 46, 8C, 19, 32, 64, C8 }	95	E0
2	{ 83, 7, E, 1C, 38, 70, E0, C1 }	E1	E4
3	{ AB, 57, AE, 5D, BA, 75, EA, D5 }	14	EA
4	{ EA, D5, AB, 57, AE, 5D, BA, 75 }	54	5
5	{ 94, 29, 52, A4, 49, 92, 25, 4A }	B2	B0
6	{ AB, 57, AE, 5D, BA, 75, EA, D5 }	50	7A
7	{ 64, C8, 91, 23, 46, 8C, 19, 32 }	DD	F8
8	{ C4, 89, 13, 26, 4C, 98, 31, 62 }	F9	97
9	{ 2F, 5E, BC, 79, F2, E5, CB, 97 }	1D	B2
10	{ FE, FD, FB, F7, EF, DF, BF, 7F }	95	8E
11	{ D6, AD, 5B, B6, 6D, DA, B5, 6B }	13	43
12	{ A4, 49, 92, 25, 4A, 94, 29, 52 }	22	35
13	{ E9, D3, A7, 4F, 9E, 3D, 7A, F4 }	D9	B9
14	{ 7F, FE, FD, FB, F7, EF, DF, BF }	EE	54
15	{ 8A, 15, 2A, 54, A8, 51, A2, 45 }	3B	EA
16	{ D3, A7, 4F, 9E, 3D, 7A, F4, E9 }	91	E7

У операції $ShiftRows(state)$ виконується побайтний зсув елементів матриці $state$: елементи i-го ($i = 2, 3$) рядка послідовності $state$ циклічно зсуваються вправо на 2 елементи.

У алгоритмі *Luna* вхідне та вихідне відбілювання виконується операціями $ExtraMix(state, subkey)$ та $InvExtraMix(state, subkey)$, які являють собою 2-раундову мережу Файстеля, в якій за допомогою розширеного ключа, таблиці замін та операцій динамічного циклічного зсуву, забезпечується початкове та кінцеве відбілювання даних (див. рис. 4).

Під змінними a_i та k_i ($i = \overline{1,4}$) мається на увазі відповідно i-та колонка блоку даних $state$ та розширеного ключа $subkey$. $Sbox(x)$ – операція табличної заміни кожних 16 біт. + та $\oplus$ – додавання за модулем 2^{32} та 2 відповідно. Процедура розшифрування алгоритмів *Luna* та *Neptun* аналогічна процедурі зашифровування (див. рис. 1а, 1б). Тільки розширені ключі подаються в зворотному порядку, використовуються зворотні таблиці замін та зворотна до $MixColumns(state)$ операція (множення на поліном $d(x) = 7Ax^7 + A1x^6 + F8x^5 + EEx^4 + 20x^3 + 89x^2 + EBx + 51$). *Процедура розширення ключа.* Для формування розширених ключів $128n$ бітний секретний ключ K ($n = 1, 2, 4$) розбивається на n частин по 128 біт ($a_l, l = \overline{1,n}$). Кожну з яких розкладають на чотири частини $k^l{}_i$ ($i = \overline{1,4}$), які разом з допоміжними 32 бітними змінними A, B, C, D, E, F, y_i ($i = \overline{1,4}$) подаються на вхід процедури розширення ключа, псевдокод якої наведений на рис. 5.

1. $t = a_1,\ u = a_2$
2. $t = t \oplus k_3,\ u = u \oplus k_4$
3. $t = t + u,\ t = Sbox(t)$
4. $u = u <<< 7,\ u = u + t$
5. $t = t >>> 1,\ u = Sbox(u)$
6. $t = t + k_1,\ u = u + k_2$
7. $a_3 = a_3 \oplus t,\ a_4 = a_4 \oplus u$
8. $t = a_3,\ u = a_4$
9. $t = t \oplus k_1,\ u = u \oplus k_2$
10. $t = t + u,\ t = Sbox(t)$
11. $u = u <<< 7,\ u = u + t$
12. $t = t >>> 1,\ u = Sbox(u)$
13. $t = t + k_3,\ u = u + k_4$
14. $a_1 = a_1 \oplus u,\ a_2 = a_2 \oplus t$

а)

1. $t = a_3,\ u = a_4$
2. $t = t \oplus k_1,\ u = u \oplus k_2$
3. $t = t + u,\ t = Sbox(t)$
4. $u = u <<< 7,\ u = u + t$
5. $t = t >>> 1,\ u = Sbox(u)$
6. $t = t + k_3,\ u = u + k_4$
7. $a_1 = a_1 \oplus u,\ a_2 = a_2 \oplus t$
8. $t = a_1,\ u = a_2$
9. $t = t \oplus k_3,\ u = u \oplus k_4$
10. $t = t + u,\ t = Sbox(t)$
11. $u = u <<< 7,\ u = u + t$
12. $t = t >>> 1,\ u = Sbox(u)$
13. $t = t + k_1,\ u = u + k_2$
14. $a_3 = a_3 \oplus t,\ a_4 = a_4 \oplus u$

б)

Рис.4. Послідовність операцій $ExtraMix(\ )$ (а) та $InvExtraMix(\ )$ (б)

Під операцією $Sbox(X)$, для алгоритмів *Luna* та *Neptun*, мається на увазі таблична заміна відповідно кожних 16 та 8 біт (у алгоритмі *Luna* використовується таблиця замін побудована на основі параметрів з табл.2., а у алгоритмі *Neptun* − на основі параметрів з першого рядку табл.3). $Mix(y_1, y_2, y_3, y_4)$ - операція лінійного розсіювання. У даній операції змінні y_i розбивається на дві частини по 8 байт (старші біти y_i утворюють першу 8-байтну частину, а молодші − другу), кожна з яких розглядається як поліном над полем $GF(2^8)$ з 8 термами, який перемножують за модулем $x^8 + 1$ з фіксованим поліномом $c(x)$ степені 7. Де $c(x) = 3x^7 + 7x^6 + x^5 + 3x^4 + 7x^3 + 4x^2 + 1Dx + 1$. У якості поліному, що не приводиться, обрано поліном: $m(x) = x^8 + x^7 + x^5 + x^4 + x + 1$. Початкові значення змінних A, B, C, D, E, F, y_i наведені в шістнадцятирічному вигляді у табл. 4.

Таблиця 4

Початкові значення змінних A, B, C, D, E, F, y_i

Змінна	Початкові значення
A	13C5E572
B	BC6FF4AD
C	4C1371E1
D	89F8D170
E	01069DA9
F	C5F52BD7
y_1	3B106B7A
y_2	7E15CEC1
y_3	23B0C13E
y_4	37A763D2

Програмна оптимізація алгоритмів шифрування Luna та Neptun. У алгоритмах *Luna* та *Neptun* широко використовуються алгебраїчні операції в кінцевих полях. Безпосереднє

виконання цих операцій призвело б до в край неефективної їхньої програмної реалізації. Однак байтова структура алгоритмів відкриває широкі можливості щодо їх оптимізації [17].

Input: a_l, *KolSubKey*
$A, B, C, D, E, F, y_1, y_2, y_3, y_4$.
Output: 128-бітні розширені ключі.
1. *For* $k = 0,\ k < KolSubKey,\ k{+}{+}$ *do*
1.1. $subkey[k] = 0;$
2. *For* $l = 0,\ l < n,\ l{+}{+}$ *do*
2.1. *For* $k = 0,\ k < KolSubKey,\ k{+}{+}$ *do*
2.1.1. *For* $j = 0,\ j < 7,\ j{+}{+}$ *do*
2.1.1.1 $A = \left(\left(A \oplus k^l{}_2\right) <<< \left(D \oplus k^l{}_1\right)\right) \oplus y_3;$
2.1.1.2. $k^l{}_1 = Sbox\left(\left(A \oplus k^l{}_1\right) + B\right);$
2.1.1.3. $B = Sbox\left(B \oplus k^l{}_1\right) >>> \left(C \oplus k^l{}_3\right);$
2.1.1.4. $k^l{}_2 = Sbox\left(\left(B \oplus k^l{}_2\right) <<< k^l{}_4\right) \oplus A;$
2.1.1.5. $y_1 = Sbox\left(\left(\left(Sbox\left(y_1 \oplus k^l{}_2\right) <<< k^l{}_1\right) \oplus E\right) \oplus y_4\right);$
2.1.1.6. $C = Sbox\left(\left(C \oplus F\right) + y_1\right) \oplus D;$
2.1.1.7. $y_2 = Sbox\left(\left(\left(y_2 \oplus C\right) >>> k^l{}_2\right) \oplus k^l{}_1\right);$
2.1.1.8. $Mix\left(y_1, y_2, y_3, y_4\right);$
2.1.1.9. $D = \left(\left(D \oplus k^l{}_4\right) <<< \left(A \oplus k^l{}_3\right)\right) \oplus y_1;$
2.1.1.10. $k^l{}_3 = Sbox\left(\left(D \oplus k^l{}_3\right) + E\right);$
2.1.1.11. $E = Sbox\left(E \oplus k^l{}_3\right) >>> \left(F \oplus k^l{}_1\right);$
2.1.1.12. $k^l{}_4 = Sbox\left(\left(E \oplus k^l{}_4\right) <<< k^l{}_2\right) \oplus D;$
2.1.1.13. $y_3 = Sbox\left(\left(\left(Sbox\left(y_3 \oplus k^l{}_4\right) <<< k^l{}_3\right) \oplus B\right) \oplus y_2\right);$
2.1.1.14. $F = Sbox\left(\left(F \oplus C\right) + y_3\right) \oplus A;$
2.1.1.15. $y_4 = Sbox\left(\left(\left(y_4 \oplus F\right) >>> k^l{}_4\right) \oplus k^l{}_3\right);$
2.1.1.16. $Mix\left(y_1, y_2, y_3, y_4\right);$
2.1.2 $temp[l][k] = y_1 \mid y_2 \mid y_3 \mid y_4;$
2.1.3 $subkey[k] = subkey[k] \oplus temp[l][k].$

Рис. 5. Псевдокод процедури розширення ключа

Так, при реалізації алгоритму *Luna* можливо реалізувати 2-х байтову заміну, зсув та множення результату на відповідні стовпчики матриці M – як одну заміну 16 біт на 64 біта, а для агоритму *Neptun* реалізувати байтову заміну, зсув та множення елементу матриці $state$ на стовпчик матриці M – як одну заміну 8 біт на 64 біта. У такому випадку повний раунд алгоритму *Luna* складатиметься лише з 8-ми підстановок 16 на 64 біта та 8-ми додавань за модулем 2. Аналогічно для виконання повного раунду алгоритму *Neptun* необхідно виконати лише 16-ть підстановок 8 на 64 біта і 16-ть додавань за модулем 2. У такий спосіб, за рахунок виділення більшої кількості оперативної пам'яті та виконання попередніх обрахунків, можна зменшити кількість операцій в раунді та досягти приріст швидкості криптообробки.

3. Дослідження алгоритмів *Luna* та *Neptun* за методикою Nist STS. Властивості псевдовипадкових послідовностей утворених за допомогою алгоритмів *Luna* та *Neptun* (у режимі лічильника) було досліджено у середовищі статистичних тестів NIST STS (методика тестування описана в роботі [18]). Статистичні портрети програмних реалізацій алгоритмів *Luna* і *Neptun* наведені на рис. 6 та 7 відповідно.

У табл. 5 для порівняння наведені результати тестування послідовностей сформованих на основі алгоритмів Luna, Neptun, ГОСТ 28147-2009, Калина. Як видно з результатів (табл. 5), генератори на основі алгоритмів шифрування Luna, Neptun пройшли комплексний контроль за методикою NIST STS і мають кращі результати за генератори на основі інших алгоритмів.

Рис.6. Статистичний портрет алгоритму шифрування *Luna* у режимі лічильника

Рис.7. Статистичний портрет алгоритму шифрування *Neptun* у режимі лічильника

Таблиця 5

Результати тестування послідовностей

Генератор	Кількість тестів, в яких тестування пройшло	
	99% послід.	96% послід.
BBS	133 (70,3%)	189 (100%)
ГОСТ 28147-2009	132 (69,8%)	188 (99,4%)
Калина	136 (72,0%)	189 (100%)
Luna	141 (74,6%)	189 (100%)
Neptun	140 (74,0%)	189 (100%)

4. Дослідження швидкісних характеристик алгоритмів *Luna* та *Neptun*. На основі описаного п.2 способу програмної оптимізації були реалізовані алгоритми шифрування *Luna, Neptun, AES, Калина, ДСТУ ГОСТ 28147-2009* на мові програмування С++. Так, при

реалізації алгоритму *Luna* було представлено 2-х байтову заміну, зсув та множення результату на відповідні стовпчики матриці M − як одну заміну 16 біт на 64 біта. Аналогічно, при реалізації алгоритмів *Neptun* та *Калина* було представлено операцію байтової заміни, зсуву та множення елементу матриці *state* на стовпчик матриці M − як одну заміну 8 біт на 64 біта. При реалізації алгоритму *ГОСТ 28147-2009* кожні дві таблиці замін 4 на 4 біти об'єднувалися у таблицю 8 на 8 біт, що дозволило зменшити кількість операцій підстановок з 8 до 4. При реалізації алгоритму *AES* було представлено операцію байтової заміни, зсуву та множення елементу матриці *state* на стовпчик матриці M − як одну заміну 8 біт на 32 біта.

Після розробки програмних засобів було проведено експериментальне дослідження, яке показало, що в однакових умовах запропоновані алгоритми шифрування *Luna* та *Neptun* швидші від 1,09 до 2,93 раз за шифри ДСТУ ГОСТ 28147-2009, *Калина* та *AES* (див. табл. 6). Дослідження проводились на Intel(R) Core(TM)2 Duo T7300 2.0 GHz.

Таблиця 6

Порівняння швидкісних характеристик алгоритмів шифрування

Алгоритм шифрування	Швидкість (МБайт/с)
AES -128	37,2
Калина -128	34,9
ГОСТ 28147-2009	18,1
Luna -128	53,1
Neptun -128	40,9

5. Дослідження стійкості алгоритмів *Luna* та *Neptun* до лінійного та диференційного криптоаналізу. При розрахунку аналітичних верхніх оцінок параметрів, що характеризують практичну стійкість до лінійного та диференційного криптоаналізу за формулами (1)-(2) потрібно розрахувати параметри Δ та Λ, що залежать від таблиць підстановок. Для цього було розроблено спеціальне програмне забезпечення та побудовані відповідні таблиці за формулами (3)-(4). Після чого визначено максимальне значення в цих таблицях (за виключенням елементів які знаходились в 0-му рядку чи стовпцю). В результаті було визначено, що для алгоритму *Luna* $\Delta = \Lambda = 2^{-14}$, а для кожної таблиці *Neptun* $\Delta = \Lambda = 2^{-6}$.

У таблиці 7. наведено аналітичні верхні оцінки параметрів (за формулами (1)-(2)), що характеризують практичну стійкість алгоритмів шифрування *Luna* та *Neptun* до диференційного та лінійного криптоаналізу (при розрахунку показників вважали, що в алгоритмів *Luna* відсутні операції *ExtraMix* та *InvExtraMix*).

Таблиця 7

Аналітичні верхні оцінки стійкості до диференційного та лінійного криптоаналізу

Довжина ключа K (біт)	*Luna*		*Neptun*	
	Диференц. криптоаналіз	Лінійний криптоаналіз	Диференц. криптоаналіз	Лінійний криптоаналіз
128	$EDP(\Omega) \le 2^{-392}$	$ELP(\Omega) \le 2^{-392}$	$EDP(\Omega) \le 2^{-222}$	$ELP(\Omega) \le 2^{-222}$
256	$EDP(\Omega) \le 2^{-512}$	$ELP(\Omega) \le 2^{-512}$	$EDP(\Omega) \le 2^{-330}$	$ELP(\Omega) \le 2^{-330}$
512	$EDP(\Omega) \le 2^{-770}$	$ELP(\Omega) \le 2^{-770}$	$EDP(\Omega) \le 2^{-546}$	$ELP(\Omega) \le 2^{-546}$

Висновки. У роботі запропоновано нові алгоритми шифрування інформації для підвищення ефективності захисту електронних інформаційних ресурсів. Як видно з результатів експериментального дослідження алгоритми *Luna* та *Neptun* мінімум в 2 рази

швидші за наш національний стандарт шифрування ДСТУ ГОСТ 28147-2009. Крім того, вони пройшли комплексний контроль за методикою NIST STS і показали кращі результати за генератори на основі інших алгоритмів шифрування. Також показано, що запропоновані алгоритми практично стійкі до лінійного та диференційного криптоаналізу. Проте, у майбутньому потрібно перевірити їх на стійкість до інших методів криптоаналізу.

ЛІТЕРАТУРА

1. Додаток 17 до «Конвенції про міжнародну цивільну авіацію». — 9 вид. — 2011 (Restricted).
2. Doc 30 «Політика ЄКЦА у сфері авіаційної безпеки». — 13 вид. — 2010 (Restricted).
3. Doc 8973/8 «Керівництво ІКАО з безпеки для захисту цивільної авіації від актів незаконного втручання». — 2011 (Restricted).
4. Advanced Encryption Standard (AES) [Electronic resource] : FIPS 197. — Electronic data (1 file: 279 457 byte). — Gaithersburg, Maryland, USA : NIST, 2001. — Mode of access: World Wide Web. — URL: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf. — Description based on screen.
5. Конкурс NESSIE (Новые европейские алгоритмы подписи, обеспечения целостности и шифрования) [Электр. ресурс] С.П. Панасенко. — Режим доступа: http://old.cio-world.ru/bsolutions/e-safety/340556.
6. Положення про проведення відкритого конкурсу криптографічних алгоритмів [Електронний ресурс] // Інститут кібернетики ім. В.М. Глушкова НАНУ; ДСТСЗІ. — Режим доступу: http://www.dstszi.gov.ua/dstszi/control/ru/publish/article;jsessionid=EE63A37FEF8F5B34030F1E38D7247DBC?art_id=48387&cat_id=92733.
7. Горбенко И.Д. Стандартизация алгоритмов шифрования. Требования к проекту національного стандарта блочного симметричного шифрования на современном этапе развития криптографии / И.Д. Горбенко, И.В. Лисицкая. // Радиотехника. — 2011. — С. 5-10.
8. Горбенко І.Д. Принципи побудування та властивості блокових симетричних IDEA подібних шифрів / І.Д. Горбенко, В.І. Долгов, Р.В. Олійников та ін. // Прикладная радиоэлектроника. — 2007. — Т. 6, № 2. — С. 158-173.
9. Головашич С.А. Спецификация алгоритма блочного симметричного шифрования «Лабиринт» / С.А. Головашич // Прикладная радиоэлектроника. — 2007. — Т. 6, № 2. — С. 230-240.
10. Панасенко С.П. Алгоритмы шифрования. Специальный справочник. — СПб. : БХВ-Петербург, 2009 — 576 с.
11. Алексейчук А.Н. Оценки практической стойкости блочного шифра «Калина» относительно методов разностного, линейного криптоанализа и алгебраических атак, основанных на гомоморфизмах / А.Н. Алексейчук, Л.В. Ковальчук, Е.В. Скрынник, А.С. Шевцов // Прикладная радио-электроника. — 2008. — Т.7, № 3. — С. 203-209.
12. Алексейчук А.Н. Оценки практической стойкости блочного шифра “Калина” относительного разностного, линейного и билинейного методов криптоанализа / А.Н. Алексейчук, Л.В. Ковальчук, Л.В. Скрынник, А.С. Шевцов // Материалы Четвертой международной конференции по проблемам безопасности и противодействию терроризму. МГУ им. М.В. Ломоносова. 30–31 октября 2008 г., Том. 2. – М.: МЦНМО, 2009. — С. 15-20.
13. Математичні основи криптоаналізу: навч. посібник / С.О.Сушко, Г.В. Кузнецов, Л.Я. Фомичова, А.В. Корабльов. – Д.: Національний гірничий університет, 2010. — 465 с.
14. Lai X. Markov ciphers and differential cryptanalysis / X. Lai, J. Massey, S. Murphy // Advances in Cryptology — EUROCRYPT’91, Proceedings. — Springer Verlag, 1991. — P. 17-38.
15. Matsui M. Linear cryptanalysis methods for DES cipher [Electronic resource] EUROCRYPT, Springer Verlag, 1998. — Mode of access: World Wide Web. — URL: http://www.cs.bgu.ac.il/~crp042/Handouts/Matsui.pdf.
16. Daemen J. Cipher and hash function design strategies based on linear and differential cryptanalysis: Ph. D. Thesis. — Katholieke Univ. Leuven, 1995.
17. Винокуров А. Сравнение российского стандарта шифрования, алгоритма ГОСТ 28147-89, и алгоритма Rijndael, выбранного в качестве нового стандарта шифрования США. / А.Винокуров, Э.Применко. // "Системы безопасности", М., изд. "Гротэк", 2001, №1,2.
18. Горбенко І.Д. Порівняльний аналіз алгоритмів генерації псевдовипадкових послідовностей / І.Д. Горбенко, Р.І. Мордвінов // Прикладная радиоэлектроника. — 2012. — Т. 11, № 2. — С. 188-190.

Надійшла: 12.11.2012 р.

Рецензент: д.т.н., професор Хорошко В.О.