13. Sulkamo V. (2018) “IoT from cyber security perspective” Master’s thesis, School of Technology, Communication and Transport. Accessed at https://www.theseus.fi/bitstream/handle/10024/151498/IoT%20from%20cyber%20security %20perspective.pdf?sequence=1&isAl

14. ENISA (2019) “Good Practices for Security of Internet of Things”. Retrieved from https://www.enisa.europa.eu/publications/good-practices-for-security-of-iot.

Статтю подано до редакції 01.10.2019 р.

УДК 681.5 DOI: 10.33111/mise.98.2

Бабинюк О.І. к. е. н.,
доцент кафедри комп’ютерної математики та інформаційної безпеки,
Нагірна А.М., к. фіз.-мат. наук,
доцент кафедри комп’ютерної математики та інформаційної безпеки,
Нагорна О.В.,
студентка 3-го курсу спеціальності «Кібербезпека»,
Київський національний економічний університет імені Вадима Гетьмана

Babynjuk O. I. PhD in Economic,
Associate Professor of the
Computer Mathematics and Information Security Department,
Nahirna A.M., PhD in Physics and Mathematics,
Associate Professor of the
Computer Mathematics and Information Security Department,
Nahorna O.V.,
3rd year Student at the "Cybersecurity" speciality,
Kyiv National Economic University named after Vadym Hetman

АЛГОРИТМ ШИФРУВАННЯ LUCIFER ТА ЙОГО КРИПТОАНАЛІЗ

LUCIFER ENCRYPT ALGORITHM AND ITS CRYPTO ANALYSIS

Анотація. *Необхідність обміну інформацією між відправником та отримувачем передбачає дотримання конфіденційності даних. Для забезпечення захисту даних від третіх осіб сприяло виникненню методу шифрування даних. Шифрування початкового (вхідного) тексту здійснюється за допомогою певного ключа, який зберігається лише у двох осіб: відправника та отримувача. Відсутність у третіх осіб ключа не дає їм можливості розшифрувати зашифровані дані, але сприяло пошуку методів розшифрування зашифрованих даних, що спричинило виникненню поняття криптографічного аналізу.*

З появою криптографічного аналізу шифрування стало викликом, почали з’являтися нові та більш безпечні, порівняно з попередниками, методи шифрування.

Обраний алгоритм «Lucifer» є потужним алгоритмом шифрування, перевагою цього алгоритму є те, що значна кількість новітніх алгоритмів базується саме на цьому методі шифрування, яскравим прикладом нащадку Lucifer є стандарт DES.
Можна, можливо, посваритись із назвою цього алгоритму. А як щодо Playfair чи шифру Hill? Але LUCIFER, частина експериментальної криптографічної системи, розробленої IBM, був прямим родоначальником DES, також розробленим IBM.
Як і DES, LUCIFER був ітераційним блоковим шифром, використовуючи круглі Feistel. Тобто, LUCIFER скремтував блок даних, виконавши крок шифрування на цьому блоці кілька разів, і використовуваний крок включав прийняття ключа для цього кроку та половини цього блоку, щоб обчислити вихід, який потім застосували ексклюзивно-АБО до інша половина блоку. Потім половинки блоку поміняли місцями, щоб обидві половини блоку були змінені рівною кількістю разів.
До речі, ця стаття посилається на LUCIFER як фактично реалізований і описаний у статті в журналі Cryptologia Артура Соркіна. У статті в Scientific American обговорювали плани LUCIFER на більш загальному рівні та описували, що по суті є різним видом блокових шифрів. LUCIFER шифрував блоки з 128 біт, і він використовував 128-бітний ключ.
F-функція в LUCIFER мала високу ступінь симетрії і могла бути реалізована в умовах операцій над одним байтом правої половини повідомлення одночасно. Однак я опишу тут LUCIFER тим самим загальним способом, як описано DES.
***Ключові слова**: шифр, алгоритм, криптоаналіз, ключ, шифрування, Люцифер.*

***Abstract.** The need to exchange information between the sender and the recipient implies the confidentiality of the data. Protection of the data promoted the encryption method. The source (input) text is encrypted with a specific key, which is stored only by two people: the sender and the recipient. The lack of a key does not allow stranger to decrypt encrypted data, but facilitated the search for methods of decrypting encrypted data, which led to the concept of cryptographic analysis.*
With the advent of cryptographic analysis, encryption has become a challenge. That's why new and more secure encryption methods have emerged (than their predecessors).
The chosen Lucifer algorithm is a powerful encryption algorithm, the great advantage of this algorithm is that a large number of the newest algorithms are based on this particular encryption method, a clear example of the descendant of Lucifer is the DES standard.
One could perhaps quarrel with the title of this section. What about Playfair, or the Hill cipher? But LUCIFER, part of an experimental cryptographic system designed by IBM, was the direct ancestor of DES, also designed by IBM.
Like DES, LUCIFER was an iterative block cipher, using Feistel rounds. That is, LUCIFER scrambled a block of data by performing an encipherment step on that block several times, and the step used involved taking the key for that step and half of that block to calculate an output which was then applied by exclusive-OR to the other half of the block. Then, the halves of the block were swapped, so that both halves of the block would be modified an equal number of times.
Incidentally, this page refers to LUCIFER as actually implemented, and described in an article in the journal Cryptologia by Arthur Sorkin. An article in Scientific American discussed plans for LUCIFER on a more general level, and described what was essentially a different kind of block cipher. LUCIFER enciphered blocks of 128 bits, and it used a 128-bit key.
The F-function in LUCIFER had a high degree of symmetry, and could be implemented in terms of operations on one byte of the right half of the message at

a time. However. I will describe LUCIFER here in the same general fashion that DES is described.
***Keywords**: cipher, algorithm, cryptanalysis, key, encryption, Lucifer.*

Вступ: Люцифер (Lucifer) — це блоковий шифр, розроблений компанією IBM на початку 70-х років. Проект Lucifer містить у собі кілька алгоритмів з даною назвою.

Власне система є комбінацією методів підстановки та перестановки, яка складається з послідовності чергування відповідних блоків. При цьому використовувався ключ для керування станами блоків, завдовжки 128 біт.

Хоча Lucifer має більший розмір блоку та ключа, ніж DES, він значно вразливіший до атак диференціального криптоаналізу, а також слабкий за своїм ключовим розкладом. Дану проблему можна вирішити застосувавши сильний потоковий шифр як до алгоритму, так і після нього, після даних дій слабкі місця перестають бути такими вразливими [11].

Постановка проблеми: З розвитком інформаційного простору (в темпі геометричної прогресії), створенні нових засобів комунікації (наприклад, соціальних мереж) та взаємодії постає проблема у захищеності даних, що обмінюються, тому необхідно покращувати методи захисту та обміну інформації. Методи захисту даних як раз і вивчає галузь кібербезпеки.

Виклад основного матеріалу: Lucifer (надалі Люцифер) був дослідним проектом фірми в 1970-х роках по створенню криптостійкого блокового шифру, в результаті досліджень було створено (визначено) два методи побудови стійких, до злому, симетричних шифрів, а саме: мережі Фейстеля та підстановко-перестановочної мережі. Таким чином Люцифер побудував міцну основу сфери сучасної симетричної криптографії. Проектом «Люцифер» керували та розвивали два лідери, тепер відомі криптографи, Хорст Фейстель (англ. Horst Feistel) і Дон Копперсміт (англ. Don Coppersmith). Найвідомішим нащадком даного проекту (даного шифрування) алгоритм DES.

Цікава історія трапилася із назвою даної програми, спершу Хорст реалізуючи алгоритм шифрування найменував його простою назвою «демонстрація» (англійською demonstration), але більш ранні версії APL обмежували довжину символів імені файлу, тому було природно скоротити назву до п’яти символі, тобто «демон» (англійською demon). Колега Хорста Лінн Сміт запропонувала найменування «Люцифер», яке й було обране.

Lucifer часто називають «першим алгоритмом шифрування для цивільних застосувань». Насправді Lucifer не є єдиним алго-

ритмом, а являє собою сукупність (сімейство) пов'язаних між собою алгоритмів. Надалі розглянемо варіанти даного алгоритму.

Структура алгоритму Люцифер зразка (тобто «Люцифер1») червня 1971 року являє собою SP-мережа (або підстановлювально-перестановчу мережу) «сендвіч» з шарів двох типів, які використовуються по черзі. Перший тип шару P-блок розрядності 128 біт, за ним йде другий шар, що являє собою 32 модулі, кожен з яких складається з двох 4-бітних S-блоків, чиї відповідні входи закорочені й на них подається одне і те ж значення з виходу попереднього шару.

Даний варіант алгоритму шифрує дані блоками по 48 бітів, використовуючи при цьому 48-бітний ключ шифрування. У процесі шифрування виконується 16 раундів перетворень (це рекомендоване автором алгоритму кількість раундів), в кожному з яких над оброблюваним блоком даних проводяться такі дії (рис. 1):

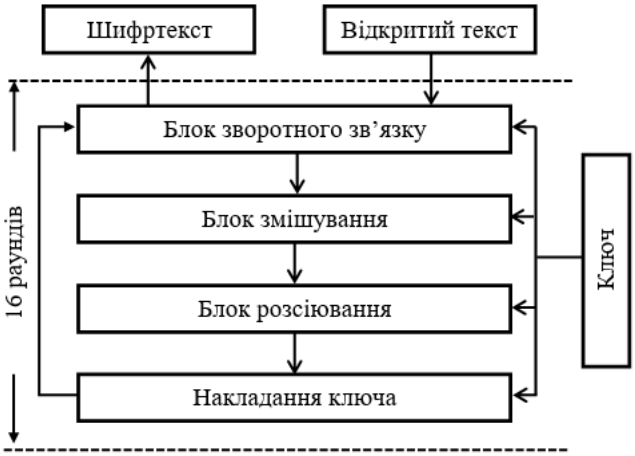

Рис. 1. Узагальнена схема

1. перемішування (або змішування) перетворення. Виконується нелінійне перетворення даних, отриманих в результаті попередньої операції, шляхом табличної заміни, що залежить від значення конкретного біта ключа раунду. Причому така залежність є досить простою: якщо значення керуючого біта дорівнює 1, виконується таблична заміна S_0, в іншому випадку використовується таблиця S_i;

2. розсіювання, що складається з перенаправлення вхідних бітів таким чином, що значення всіх вхідних бітів міняються міс-

цями за певним законом. Ця операція виконується абсолютно незалежно від значення ключа шифрування;

3. накладення ключа. Виконується шляхом побітового застосування операції XOR до результату попередньої операції і відповідним бітам ключа раунду. В кожному раунді шифрування потрібно 108 бітів ключової інформації:

– 48 бітів для блоку зворотного зв'язку;
– 12 бітів для блоку перемішування;
– 48 бітів для блоку накладення ключа [8, 9, 12].

Надалі буде описана друга версія цього шифрування, а саме — «Люцифер 2», що була опублікована у 1975 році.

Цей шифр має такі характеристики (операція або ітерація надалі буде замінена словом раунд):

– 128-бітний ключ, даний простір є досить великим для ключа, навіть за сьогоднішніми стандартами;
– 128-бітові блоки, дана кількість простору також вважається достатньою й сьогодні;
– для обробки 128-бітові блоки розділяють по 16 байт;
– схема Фейстеля складається з 16 раундів;
– у кожному раунді 8 байт правої половини R_i блоку (дорівнює 64 біти) обробляються квазі паралельно. Іншими словами, кожний раунд опрацьовує 8 блоків, тобто 1 байт, незалежно від інших;
– кожен раунд складається (включає в себе операції) з заміни і перестановки. Між деякими ключами біти додаються (операція XOR);
– нелінійність входить в алгоритм тільки шляхом заміни. Нещодавно додані ключі бітів обробляються лінійним способом на даному етапі (фактичному раунді), але згодом переходять до нелінійного заміщення наступного раунду;
– заміщення одного байту починається з розкладанням на дві половини по 4 біта, кожна з яких окремо трансформується підстановкою:

$$S_0, S_1: \ F_2^4 \to F_2^4 \ , \qquad (1)$$

де S_0 і S_1 є фіксованою заміною, що використовується в процесі шифрування.

Лише присвоєння S_0 і S_1 4-бітовим половинам змінюється залежно від певного ключового біта. Дану нелінійну карту булівського типу прийнято називати «*S*-коробки» («*S*-boxes» або *S*-бокси), в той час, коли вони постають ще меншими елементарними блоками карти ядра.

Алгоритм дуже добре підходить для реалізації в апаратних засобах, зокрема, для 8-бітних архітектур. Не є актуальним для ПЗ, так як має численні бітові перестановки.

Способом складання карти ядра з безлічі маленьких (ідентичних або схожих) *S*-боксів часто користуються і на сьогоднішній день. Емпіричне правило є таким: чим менше *S*-боксів, тим більше раундів необхідно виконати задля досягнення захищеності [1].

При розгляді другої версії алгоритму важливо розглянути такі поняття: ключовий розклад, циклічна мапа, *S*-бокси, перестановка.

Спершу розглянемо ключовий розклад. Позначимо 16 байтів ключа $k \in F_2^{128}$ як $k = (k_0, \dots, k_{15}) \in (F|\ |2^8)^{16}$.

Раунд (або ж цикл) включає вибір $\alpha_i(k) = \left(\alpha_{ij}(k)\right)_{0 \le j \le 7}$ з восьми байтів $\alpha_{ij}(k) = k_{7i+j-8mod16}$.

Ця формула виглядає доволі складною, однак описує досить просту схему вибору, відображену у табл. 1.

Таблиця 1

СХЕМА КЛЮЧОВОГО РОЗКЛАДУ

Раунд (цикл)	Позиція							
	0	1	2	3	4	5	6	7
1	0	1	2	3	4	5	6	7
2	7	8	9	10	11	12	13	14
3	14	15	0	1	2	3	4	5
4	5	6	7	8	9	10	11	12
5	12	13	14	15	0	1	2	3
6	3	4	5	6	7	8	9	10
7	10	11	12	13	14	15	0	1
8	1	2	3	4	5	6	7	8
9	8	9	10	11	12	13	14	15
10	15	0	1	2	3	4	5	6
11	6	7	8	9	10	11	12	13
12	13	14	15	0	1	2	3	4
13	4	5	6	7	8	9	10	11
14	11	12	13	14	15	0	1	2
15	2	3	4	5	6	7	8	9
16	9	10	11	12	13	14	15	0

З кожним новим циклом вибір циклічно зсувається на 7 позицій. Необхідно зазначити, що кожен байт опиняється в кожному положенні рівно один раз. Положення повідомляє, до якого байту з фактичних 64-бітових блоків відносяться фактичний ключовий байт. Крім того, $\alpha_{i0}(k)$ байт у положенні 0 служить як «контрольним байтом».

Таблиця 2

ЛІВА ТА ПРАВА ЧАСТИНА АЛГОРИТМУ ЦИКЛІЧНОЇ МАПИ

64	64							
L	*R*							
	r_0	r_1	r_2	r_3	r_4	r_5	r_6	r_7
	8	8	8	8	8	8	8	8

Розглянемо циклічну мапу. В *i*-ому раунді (циклі) вхід, тобто права 64-бітна частина фактичного блоку, розділена на вісім байтів $r = (r_0, \ldots, r_7)$, відображено у табл. 2.

j-ий номер байту перетворюється в наступний спосіб цього раунду (циклу), зображено на рис. 2:

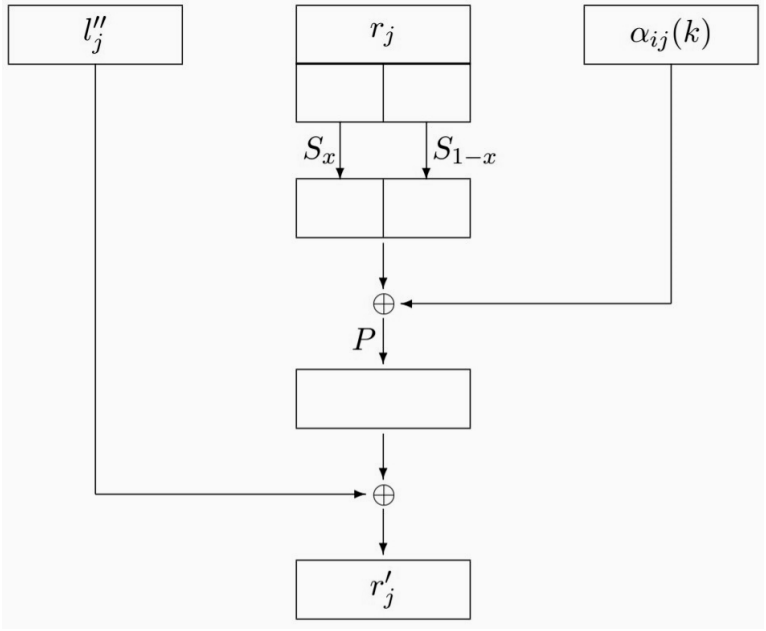

Рис. 2. Метод перетворення *j*-ого байту

На цій схемі $l_j^{''}$ є фіксованою вибіркою, що складається з восьми бітів лівої сторони актуального блоку.

Контрольний біт трансформації $\alpha_{i1}(k) = (b_0,\ldots,b_7)$ обирається справа наліво, $x=b_{7-j}$.

Виразність мапи ядра f прослідковується на зображенні на рис. 2, явна формула не є компактною, тому вона опускається.

S-бокси $S_0, S_1: F_2^4 \to F_2^4$ наведені таблицею значень. При цьому 4-бітові блоки записуються шістнадцятковою системою. Описані S-бокси відображені табл. 3 і 4.

Таблиця 3

S-БОКС $S_0(X)$, ОПИСАНИЙ 16-вою СИСТЕМОЮ

$x =$	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S_0(x) =$	C	F	7	A	E	D	B	0	2	6	3	1	9	4	5	8

Таблиця 4

S-БОКС $S_1(X)$, ОПИСАНИЙ 16-вою СИСТЕМОЮ

$x =$	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S_1(x) =$	7	2	E	9	3	B	0	4	C	D	1	A	6	F	8	5

Перестановки P переставляє біти байту таким чином:

$$P: F_2^8 \to F_2^8, (z_0, z_1, z_2, z_3, z_4, z_5, z_6, z_7)$$
$$\to (z_2, z_5, z_4, z_0, z_3, z_1, z_7, z_6).$$

Побітове додавання лівої половини до перетвореної (або трансформованої) правої частини наслідує престановку, опи сану такими діями:

Розділити ліву половину фактичного блоку на вісім байт:

$$L = (l_0, l_1, l_2, l_3, l_4, l_5, l_6, l_7).$$

Циклічно обернених після кожного кроку. Тоді при r_j вони знаходяться в позиції $(l_0', \ldots, l_7') = (l_j, \ldots, l_{7+j \bmod 8})$.

Нарешті побудувати байт $l_j^{''}$ в якості $l_j^{''} = (Bit0 з l_7', Bit1 з l_6', Bit0 з l_2', \ldots)$ і т.д. в порядку (7, 6, 2, 1, 5, 0, 3, 4), та додати його до r_j.

Третя версія, як і друга, використовує мережу Фейстеля, вона оперує над 32-бітними блоками з 64-бітовим ключем і 128-бітними блоками з 128-бітними ключами. Необхідно відзначити, що у третій версії раундова функція шифрування доволі спрощена — спочатку зашифрований підблок пропускався через шар 4-бітних *S*-блоків (аналогічно верствам в SP-мережах, тільки *S*-блок є сталим і не залежить від ключа), потім до нього додавався раундовий ключ, після чого результат пропускався через P-блок [14].

Стверджують, що четвертий варіант схожий одночасно на попередні. Як і в другому варіанті, тут виконується розбиття 128-бітного блоку, який шифрується, на два підблока, один з яких обробляється таким чином:

1. Виконується накладення ключа раунду шляхом застосування операції XOR бітам оброблюваного підблока і 64-бітного фрагмента ключа раунду KX_t (t — номер поточного раунду).

2. Керована ключем таблична заміна виконується досить схоже на перший та третій варіанти алгоритму Люцифер. Таблиці визначені в такий спосіб (табл. 5).

3. Виконується фіксована перестановка (P) бітів підблоку згідно табл. 6.

Таблиця 5

ВХІДНІ ЗНАЧЕННЯ ТА БЛОКИ

	Вхідні значення															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S_o	12	15	7	10	14	13	11	0	2	6	3	1	9	4	5	8
S_j	7	2	14	9	3	11	0	4	12	13	1	10	6	15	8	5

Таблиця 6

ТАБЛИЦЯ ЗНАЧЕНЬ БІТІВ

10	21	52	56	27	1	47	38	18	29	60	0	35	9	55	46
26	37	4	8	43	17	63	54	34	45	12	16	51	25	7	62
42	53	20	24	59	33	15	6	50	61	28	32	3	41	23	14
58	5	36	40	11	49	31	22	2	13	44	48	19	57	39	30

Значення вхідного біта 10 (біти нумеруються зліва направо, починаючи з 0-го) поміщається в вихідний біт 0, значення 21-го біта — в біт 1 і т. д.

4. Результат попередньої операції накладається на необроблений підблок операцією XOR.

5. Субблоки міняються місцями (у всіх раундах, крім останнього).

Всього виконується 16 раундів описаних вище перетворень.

Процедура розширення ключа вирішує завдання отримання 16 ключів раундів по 72 біта кожен (64 біта KX_t і 8 бітів KS_t) з вихідного 128-бітного ключа шифрування. Розширення виконується досить простим способом:

– як фрагмента KS_t використовується перший байт ключа шифрування (вважаючи байти ключа зліва направо, починаючи з 1);

– в якості фрагмента KX_t використовуються перші 8 байтів ключа шифрування (т. е. перший байт ключа використовується в обох фрагментах);

– ключ шифрування циклічно зсувається вліво на 7 байтів, після чого аналогічно «набираються» фрагменти ключа для наступного циклу.

Четвертий варіант може бути реалізований як програмно, так і апаратно.

Криптографічні аналізи перших двох варіантів алгоритму не здобули поширення, деякої «популярності», тому детально розглянемо криптографічний аналіз інших варіантів алгоритму.

У 1991 році Елі Біхам і Аді Шамір досліджували варіант № 3. Для визначеності вони використовували перестановку P з алгоритму DES, а в якості таблиць S_0 і S_i були взяті рядки 3 і 4 таблиці замін S {алгоритму DES}. Згідно 8-раундовому варіанту № 3 з 32-бітовим блоком, розкривається при наявності всього 24 обраних відкритих текстів і відповідних їм шифротекст шляхом виконання порядку 221 тестових операцій шифрування. У тій же роботі Біхам і Шамір описали можливу атаку на аналогічний алгоритм з 128-бітовим блоком, для успішного здійснення якої потрібно 60–70 обраних відкритих текстів і порядку 253 операцій шифрування.

Крім того, Біхам і Шамір стверджували, що варіант № 4 є ще слабшим. Останнє твердження було доведено в роботі, яку опублікували Ішаі Бен-Аройо і Елі Біхам в 1993 році. У ній змальована атака на варіант № 4 алгоритму Lucifer, в якому виявилося підмножина ключів, яка мала ризиковий результат: близько 55 % усіх можливих значень ключа, слабких до диференціального криптоаналізу. При використанні ключа шифрування з даної підмножини алгоритм розкривається при наявності 236 обраних відкритих текстів [2, 4].

Література

1. Артур Соркін: Люцифер, криптографічний алгоритм. (1984), 22–41.
2. Суканя С. Систематизація 256-бітового легкого блочного шифру Марвіна «підрозділ Інституту інженерії та менеджменту» / Суканя Саха — Колката. — 12 с.
3. Криптологія Частина II: Шифри Бітблока — Saarstraße 21 D-55099 Майнц, 2016. — 153 с. — (Informatik der Johannes-Gutenberg-Universit¨at). — (К. Поммеренінг, Бітблокові шифри).
4. Junod, Pascal & Canteaut, Anne. Розширений лінійний криптоаналіз шифрів блоку та потоку. — IOS Press, 2011. — ст. 2.
5. Наберіть «Ш» для шифру // Вибрані області в криптографії: 13-й міжнародний семінар, SAC 2006, Монреаль, Канада, 17–18 серпня 2006 р .: переглянуті вибрані документи. — Спрингер, 2007. — ст. 77. Криптографічні булеві функції та програми. — Академічна преса, 2009. — ст. 164.
6. Кац, Джонатан. Вступ до сучасної криптографії / Джонатан Кац, Єгуда Лінделл. — CRC Press, 2008. — ст. 166–167.
7. Габидулин Э. М., Кшевецкий А. С., Колыбельников А. И. Защита информации: учебное пособие — М.: МФТИ, 2011. — 225 ст.
8. Баричев С. Г., Гончаров В. В., Серов Р. Е. Основы современной криптографии — 3-е изд. — М.: Диалог-МИФИ, 2011. — 176 ст.
9. LUCIFER: the first block cipher [Електронний ресурс] — Режим доступу до ресурсу: http://www.quadibloc.com/crypto/co0401.htm.
10. Хоффман Л. Современные методы защиты информации / пер. с англ. — М. Сов. радио, 1980. — 264 с.

References

1. Arthur Sorkin: Lucifer, a cryptographic algorithm. Cryptologia 8 (1984), 22–41.
2. Sukanya S. Systematization of a 256-bit lightweight block cipher by Marvin "unit of the Institute of Engineering and Management" / Sukanya Saha — Kolkata. — 12 sec.
3. Cryptology Part II: Bitblock Ciphers — Saarstraße 21 D-55099 Mainz, 2016. — 153 p. — (Informatik der Johannes-Gutenberg-Universit¨at). — (K. Pommerening, Bitblock Ciphers).
4. Junod, Pascal & Canteaut, Anne. Advanced Linear Cryptanalysis of Block and Stream Ciphers. — IOS Press, 2011 — Art. 2.
5. Dial ‘C’ for Cipher // Selected Areas in Cryptography: 13th International Workshop, SAC 2006, Montreal, Canada, August 17–18, 2006: revised selected papers. — Springer, 2007 — Art. 77.
6. Cryptographic Boolean functions and applications. — Academic Press, 2009. — Art. 164.

7. Katz, Jonathan. Introduction to Modern Cryptography / Jonathan Katz, Yehuda Lindell. — CRC Press, 2008. — p. 166-167.
8. Gabidulin E. M., Kshevetsky AS, Kolybelnikov AI Information protection: a textbook — M .: MFTI, 2011. — 225 Art.
9. Barichev SG, Goncharov VV, Serov RE Fundamentals of modern cryptography — 3rd ed. — M. : Dialogue-MIFI, 2011. — 176 Art. 10. LUCIFER: the first block cipher [Electronic resource] — Resource access mode: http://www.quadibloc.com/crypto/co0401.htm.
10. Hoffman L. Modern methods of protection of information / trans. with English. — M. Sov. Radio, 1980. — 264 p.

Статтю подано до редакції 08.09.2019 р.

УДК 330.3 DOI: 10.33111/mise.98.3

Бєгун А.В., к. е. н.,
професор кафедри інформаційного менеджменту,
Гриценко К.А.,
студентка 3-го курсу спеціальності "Системний аналіз",
Київський національний економічний університет імені Вадима Гетьмана

Biehun A.V., PhD in Economics,
Professor of the Information Management Department,
Hrytsenko K.A.,
3rd year Student at the "System analysis" speciality,
Kyiv National Economic University named after Vadym Hetman

СИСТЕМНИЙ АНАЛІЗ ІНВЕСТИЦІЙНОЇ ПОЛІТИКИ УКРАЇНИ

SYSTEM INVESTMENT OF UKRAINE INVESTMENT POLICY

Анотація. *Кожна держава світу зобов'язана проводити та реалізовувати раціональну економічну політику. Це явище вважається складним, так як складається з чималої кількості взаємопов'язаних і невід'ємних елементів. Однією зі складових економічної політики є інвестиційна політика держави.*
Проблема інвестицій в країні пов'язана не тільки з обсягами, але і структурою, а також ефективістю їх використання, що вимагає значних нормотворчих і організаційних зусиль як від влади, так і від окремих підприємств і фінансових інститутів.
Інвестиції, як дуже важливий параметр, яким визначається макроекономіка, роблять значний вплив на неї. Вони являють собою капітал, який викуповує компанія, для того щоб збільшити свою продуктивність і максимізувати прибуток.
За своїм впливом інвестиції запускають так звану ланцюгову реакцію, що сприятливо позначається на економіці як окремого підприємства,

