КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 1 (13), 2021
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2021.13.183201
УДК 004.056

**Смірнова Тетяна Віталіївна**
кандидат технічних наук, доцент кафедри кібербезпеки та програмного забезпечення
Центрально український національний технічний університет, Кропивницький, Україна
ORCID ID:0000-0001-6896-0612
*sm.tetyana@gmail.com*

**Бурмак Юлія Анатоліївна**
здобувач науково-дослідної лабораторії протидії кіберзагрозам в авіаційній галузі
Національний авіаційний університет, Київ, Україна
ORCID ID: 0000-0002-5410-6260
*julburmac@gmail.com*

**Улічев Олександр Сергійович**
кандидат технічних наук, старший викладач кафедри кібербезпеки та програмного забезпечення
Центральноукраїнський національний технічний університет, Кропивницький, Україна
ORCID ID: 0000-0003-3736-9613
*askin79@gmail.com*

**Усік Павло Сергійович**
доктор філософії (PhD), старший викладач кафедри кібербезпеки та програмного забезпечення
Центральноукраїнський національний технічний університет, Кропивницький, Україна
ORCID ID: 0000-0002-3268-342X
*mr.usik@ukr.net*

**Доренський Олександр Павлович**
кандидат технічних наук, доцент кафедри кібербезпеки та програмного забезпечення
Центрально український національний технічний університет, Кропивницький, Україна
ORCID ID: 0000-0002-7625-9022
*dorensky@ukr.net*

# СТІЙКА ФУНКЦІЯ ШИФРУВАННЯ УДОСКОНАЛЕНОГО МОДУЛЯ КРИПТОГРАФІЧНОГО ЗАХИСТУ ІНФОРМАЦІЇ В ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ СИСТЕМАХ

**Анотація.** У роботі проведено аналіз вимог до побудови систем забезпечення конфіденційності даних на базі криптоалгоритмів, визначено ключові аспекти і шляхи удосконалення існуючих методів і систем шифрування даних. Методи дослідження. Основні теоретичні положення роботи отримані з використанням методів теорії захисту інформації. Об'єктом дослідження є процес забезпечення конфіденційності даних в інформаційно-комунікаційних системах управління технологічними процесами на базі хмарних технологій. Предметом дослідження є стійка функція шифрування для забезпечення удосконаленого модуля криптографічного захисту інформації в інформаційно-комунікаційних системах. Метою даної роботи є розроблення стійкої функції шифрування удосконаленого модуля криптографічного захисту інформації для забезпечення конфіденційності даних в інформаційно-комунікаційних системах управління технологічними процесами на базі хмарних технологій. Розроблено метод генерації криптографічних ключів, щоб покращити швидкість генерації ключів, з його використанням удосконалено функцію шифрування (для забезпечення удосконаленого модуля) на основі відомого і ефективного алгоритму RC6, що дозволило підвищити швидкість криптографічної обробки даних та перевірити криптостійкість алгоритму проти спеціалізованих атак лінійного та диференціального криптоаналізу.

© Т.В. Смірнова, Ю.А. Бурмак, О.С. Улічев, П.С. Усік, О.П. Доренський, 2021.

ISSN 2663 - 4023

**Ключові слова:** стійка функція шифрування, криптографічний захист, інформаційно-комунікаційна система.

## 1. ВСТУП

На сучасному етапі розвитку хмарних технологій, існує завдання захисту даних, які зберігаються у відповідних інформаційно-комунікаційних системах. Останні події, пов'язані з атаками на різні хмарні сервіси потребують розроблення нових або удосконалення існуючих механізмів захисту інформації. Одними з таких механізмів є програмні модулі криптографічного захисту даних, у яких необхідно реалізовувати вибір стійких методів шифрування та гешування, а також синхронізацію секретного ключа. У якості зазначених процедур можуть використовуватись відомі криптографічні методи і засоби, стійкі до лінійного, диференціального, алгебраїчного, квантового та інших відомих видів криптоаналізу.

## 2. АНАЛІЗ ПУБЛІКАЦІЙ ТА ПОСТАНОВКА ЗАВДАННЯ ДОСЛІДЖЕННЯ

Сьогодні серед множини методів захисту інформації особливе місце займають криптографічні методи [1]. У теперішній час в месенджерах і інших застосунках використовуються наступні відомі програмні модулі криптографічного захисту даних: MTProto 1.0 [2] – модуль, який використовується для шифрування повідомлень при передаванні клієнтами Telegram; Signal Protocol [3] – використовується для шифрування миттєвих повідомлень Facebook Messenger; TLS Skype [4] – для миттєвих повідомлень використовується TLS (безпека на рівні транспорту) для шифрування повідомлень між клієнтом Skype та службою чату, коли вони надсилаються безпосередньо між двома клієнтами Skype. Проведений порівняльний аналіз розглянутих модулів захисту інформації у сучасних інформаційно-комунікаційних системах та мережах (ІКСМ). за такими критеріями, як використовувані криптоалгоритми, швидкість роботи (ШР), зручність для користувачів (ЗК) і кросплатформеність (КП), показав, що розглянуті програмні модулі мають низку недоліків і можуть бути удосконалені за рахунок використання сучасних процедур безпеки. Зважаючи на зазначене, важливим завданням є удосконалення модуля криптографічного захисту інформації для забезпечення конфіденційності та цілісності даних у сучасних ІКСМ. На теперішній час розробляється удосконалений модуль криптографічного захисту інформації, який за рахунок фіксування інформації про ідентифікатор користувача, ідентифікатор сесії, час відправлення, довжину повідомлення та його порядковий номер, а також використання нової процедури формування сеансового ключа для шифрування, дозволить забезпечити конфіденційність і цілісність даних в ІКСМ. Опис удосконаленого модуля криптографічного захисту інформації буде наведений у наступних роботах. Для ефективного використання цього модуля важливим є вибір криптостійких методів шифрування $F_{enc}$ та гешування $F_{hash}$, а також синхронізація секретного ключа $authKey$. У якості функцій $F_{enc}$ та $F_{hash}$ можуть бути використані зокрема й алгоритми, запропоновані у роботах [5-13], стійкі до лінійного, диференціального, алгебраїчного, квантового та інших відомих видів криптоаналізу. Областю застосування запропонованих підходів є хмарні системи які описані у [14-17].

**Методи дослідження**. Основні теоретичні положення роботи отримані з використанням методів теорії захисту інформації.

**Об’єктом дослідження** є процес забезпечення конфіденційності даних в інформаційно-комунікаційних системах управління технологічними процесами на базі хмарних технологій.

**Предметом дослідження** є стійка функція шифрування для забезпечення удосконаленого модуля криптографічного захисту інформації в ІКС.

**Метою** даної роботи є розроблення стійкої функції шифрування удосконаленого модуля криптографічного захисту інформації для забезпечення конфіденційності даних в інформаційно-комунікаційних системах управління технологічними процесами на базі хмарних технологій.

## 3. ОСНОВНА ЧАСТИНА ДОСЛІДЖЕННЯ

### 3.1 Управління ключовими даними

Алгоритми шифрування зазвичай не є секретними і публікуються відкрито або майже відкрито (у деяких випадках). Основне навантаження щодо захисту інформації методами шифрування несуть ключі. Адміністрування ключів покликане додати їм необхідні властивості і забезпечити нормальне функціонування на всіх стадіях життя (використання) ключів. Стадіями життя ключів є:

– генерація або формування;

– розподіл;

– верифікація і автентифікація;

– зберігання;

– використання;

– модифікація;

– ліквідація або утилізація.

Криптостійкість деяких алгоритмів шифрування (або майже всіх) сильно залежить від того наскільки непередбачувані числа, що видає ГПВЧ, який використовує той або інший алгоритм шифрування. У зв'язку цим виникає поняття криптостійкості ГПВЧ, чим більше непередбачуваний ГСВЧ, тим вище його криптостійкість. Тому для розробленої системи був запропонований новий метод генерації криптографічних ключів.

### 3.2. Процедура генерації криптографічних ключів

На вхід подається вектор ініціалізації (ключ). Далі виконується розширення ключа (формується $t$ 32-бітних ключів, які утворюють згенерований криптографічний ключ), що складається з трьох етапів:

ISSN 2663 - 4023

1. Відбувається вирівнювання ключа шифрування, в рамках якого він (якщо його розмір в байтах *b* не кратний *w*/8, тобто розміру слова в байтах) доповнюється нульовими байтами до найближчого більшого розміру *c*, кратного *w*/8, де *w* – довжина слова в бітах, *c* – кількість 32-бітних слів у масиві ключів, *b* – довжина ключа.

2. Виконується початкова ініціалізація масиву розширених ключів $K_0 \dots K_{t-1}$, використовуючи арифметичну прогресію, визначену константами $a_{32}$, $b_{32}$ , $c_{32}$ і $d_{32}$, це відбувається таким чином:

$$k_0 = a_{32}$$

$$k_{i+1} = \big((k_i + d_{32}) \times c_{32} - b_{32}\big) mod 2^{32}$$

Константи визначені таким чином:

$$a_{32} = Odd((e-2)2^{32} + 2^{32} \div 4)$$

$$b_{32} = Odd((\theta - 1)2^{32})$$

$$c_{32} = Odd((e-2)2^{32})$$

$$d_{32} = Odd((\theta - 1)2^{32} + 2^{32} \div 3)$$

де $e = 2.718281828459$ ... (основа натурального алгоритму); $\theta = 1.618033988749$ ...(золотий перетин); *Odd(x)* – непарне ціле число, найближче до *x*. Ці константи в двійковому і шістнадцятковому вигляді мають таке значення:

$$a_{32} = 11110111111000010101000101100011 = F7E15163$$

$$b_{32} = 10011110001101110111100110111001 = 9E3779B9$$

$$c_{32} = 10110111111000010101000101100011 = B7E15163$$

$$d_{32} = 11110011100011001100111100001111 = F38CCF0F$$

3. Забезпечується змішування ключа користувача за допомогою циклічного виконання наступних дій:

$$A = K_i = S\big[\big((K_i + A - B) <<< 3\big) mod C\big]$$

$$B = Kl_j = \big(Kl_j + S[(A+B) mod 2^{32}]\big) <<< (C - A)$$

$$C = (K_i - A + S[B]) <<< C$$

$$i = (i+1) mod t$$

$$j = (j+1) mod 4$$

де:

- *i, j, A, B, C* і *D* – тимчасові змінні, їх початкові значення дорівнюють нулю;
- *S*[*x*] – таблиця підстановок;
- *Kl* – (32**l*) – бітний вектор ініціалізації.

Розмір ключа дорівнює 32*l*, при 128 бітах $l = 4$.

ISSN 2663 - 4023

Виконується $m = 15t$ ітерацій циклу.

Схема циклічних ітерацій представлена на рис. 1.

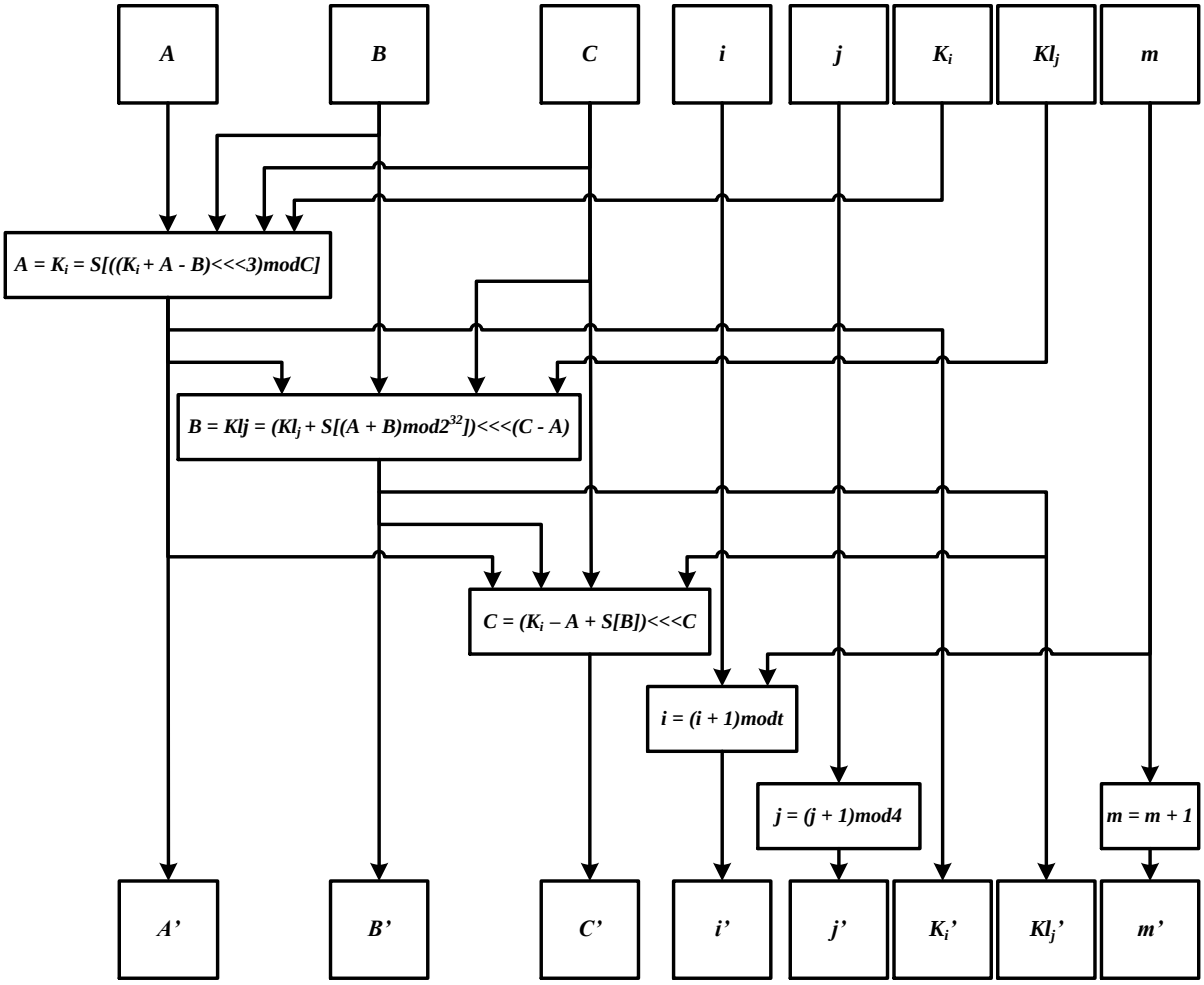


*Рис. 1. Схема циклічних ітерацій третьої алгоритмічної частини*

### 3.3. Обґрунтування та опис удосконалення функції шифрування

Для удосконалення прототипу, яким є алгоритм шифрування RC6, пропонується така методика:

1. *Ускладнити процедуру обрахунку U і T.*

У даних процедурах пропонується ввести блок підстановок (вибраний із множини, так званих , гранично-нелінійних бієктивних перетворень; побудований на основі конструкції Ніберг-Дінга), операцію додавання за модулем $2^{32}$ і множення за *модулем* $2^{32}$, що забезпечить захист від алгебраїчних атак, лінійного та диференціального криптоаналізу, інтерполяційної атаки. Також пропонується ввести операції динамічного-циклічного зсуву (залежить від розширених ключів)– це дозволить динамічно керувати процесом розсіювання інформаційних даних.

2. *Збільшити кількість підключів.*

Для забезпечення динамічного керування процесом розсіювання інформаційних даних, також це ускладнить диференціальний та лінійний криптоаналіз.

*3. Замінити часткове вхідне та вихідне відбілювання – повним.*
Це ускладнить проведення лінійного та диференціального криптоаналізу.

4. *Змінення раундової функції.*
Після кожних двох раундів добавлена операція *RoundColumns*, що призведе до ще більшого лавинного ефекту запропонованого алгоритму.

*5. Зменшена кількість раундів*
Це дозволяє підвищити швидкість шифрування даних без втрати криптостійкості.

В основу запропонованого шифру був покладений алгоритм шифрування RC6, який побудований на основі мережі Фейстеля. У RC6 відкритий текст розбивається на чотири 32-бітні підблоки *A, B, C, D* над якими виконуються наступні перетворення [10]:

1) Часткове вхідне відбілювання. Відбілюються підблоки *B* і *D* за допомогою підключів: $B = B + K_0 \bmod 2^{32}$, $D = D + K_1 \bmod 2^{32}$;

2) 20 раундових перетворень. Для кожного раунду $i$ $(i = \overline{1,..,20})$ спочатку з підблоків *B* і *D* обраховуються допоміжні 32-бітні підблоки *U, T*. За допомогою яких змінюють підблоки *A* і *C*, після чого до *A* і *C* додається за модулем $2^{32}$ значення відповідних підключів:

$$A = \left( \left( A \oplus T \right) <<< U \right) + K_{2i} \bmod 2^{32},$$

$$C = \left( \left( C \oplus U \right) <<< T \right) + K_{2i+1} \bmod 2^{32}.$$

У кінці раунду підблоки зсуваються: $(A, B, C, D) = (B, C, D, A)$;

3) Часткове вихідне відбілювання. Відбілюються підблоки *A* і *C* за допомогою підключів: $A = A + K_{42} \bmod 2^{32}$, $C = C + K_{43} \bmod 2^{32}$.

У запропонованому методі шифрування пропонується замінити часткове вхідне та вихідне відбілювання – повним. Змінюються процедура обрахунку *U, T*: вводиться блок підстановок (представлений нижче).

Також, збільшується кількість підключів раундів, тепер в кожному раунді використовується по 6 підключів. Внесені зміни до раундової функції: використовується змінений порядок операцій та введена операція множення на поліном кожні два раунди. Крім того, для розширення ключів − використовується розроблений генератор псевдовипадкових чисел (генерується $t = 6r + 8$ розширених ключів).

***Представлення вхідних та вихідних даних алгоритму шифрування***

До вхідних даних алгоритму належать:

– відкритий текст;

ISSN 2663 - 4023

– ключ шифрування (секретний ключ у відповідності з яким виконується розширення підключів).

На виході отримуємо шифротекст. Вхідні, вихідні блоки даних алгоритму представляються у вигляді чотирьох 32-бітних підбоків *A, B, C* і *D*.

***Параметри алгоритму***

Розмір блоку та довжина ключа шифрування. Запропонований метод шифрування підтримує довжину блоку даних у 128 бітів з підтримкою ключа шифрування довжиною 128.

Алгоритм шифрування – це складна процедура, що складається з попередньої та фінальної рандомізації, між якими відбуваються ітеративні (циклові) перетворення шифрування (зашифрування та розшифрування). Мінімальне допустиме число раундів шифрування $(r)$, а отже і кількість циклів $\left(n = \frac{r}{2}\right)$ залежить від довжини ключа шифрування. При довжині ключа 128 бітів $r = 12$, відповідно кількість циклів $n = 6$.

***Процедура зашифрування***

На вхід процедури подаються підключі $K_i$ і відкритий текст, який розбивається на підблоки *A, B, C, D*. Спочатку виконується повне початкове відбілювання (рандомізація) підблоків *A, B, C* і *D*. Потім виконуються *r* раундових перетворень. Далі виконується повне кінцеве відбілювання підблоків *A, B, C* і *D*. Отримані у результаті зашифрування підблоки об'єднують у шифротекст. Загальна схема алгоритму зашифрування зображена на рис. 2.

***Початкове та кінцеве відбілювання***

Перед початком шифрування даних всі підблоки відбілюють за допомогою відповідних підключів: $A = (A \oplus K_0)$, $B = (B + K_1)$, $C = (C \oplus K_2)$, $D = (D + K_1)$. Для підблоків *A* і *C* ця операція виконується додаванням за модулем 2 кожного байту вказаного підблоку з кожним байтом вказаного підключа, а для *B* і *D* – додаванням за модулем $2^{32}$. Така ж операція виконуються наприкінці шифрування, але для *A* і *C* вже використовується додаванням за модулем $2^{32}$, а для *B* і *D* – додаванням за модулем 2: $A = (A + K_{6r+4})$, $B = (B \oplus K_{6r+5})$, $C = (C + K_{6r+6})$, $D = (C \oplus K_{6r+7})$.

Схема операцій відбілювання (початкової та фінальної) наведена на рис. 3.

***Раундові перетворення***

Для кожного раунду *i* ($i = \overline{1,..,r}$) виконується наступне:

1) Послідовності *B* і *D* подаються на вхід функцій *Ft*() і *Fu*() відповідно. В результаті отримують допоміжні 32-ох бітні послідовності *T* і *U*.

2) Додають кожний байт блоків *A* і *C* з кожними байтом послідовностей *T* і *U* використовуючи додавання за модулем $2^{32}$ та за модулем 2 відповідно: $A = (A + T) \bmod 2^{32}$, $C = (C + U) \bmod 2$.

3) Виконують циклічний по бітний зсув елементів блоків $A$ і $C$ в залежності від елементів блоків $U$ і $T$: $A=\left(A<<<U\right)$, $C=\left(C<<<T\right)$.

Приклад при зсуві $x$ на $y$ (для оптимізації беремо $y$ mod32):

$$Z=\left(X<<<Y\right)\bmod 32.$$

4) Додають кожний байт блоків $A$ і $C$ з кожними байтом підключів $K_{6i+4}$ і $K_{6i+6}$ $(0\le i\le (r-1))$ використовуючи додавання за модулем 2 та за модулем $2^{32}$ відповідно: $A=\left(A+K_{6i+4}\right)\bmod 2$, $C=\left(C+K_{6i+6}\right)\bmod 2^{32}$.

5) У кінці раунду підблоки зсуваються вліво: $(A,B,C,D)=(B,C,D,A)$. Схема заміщення послідовностей показана на рис. 4.

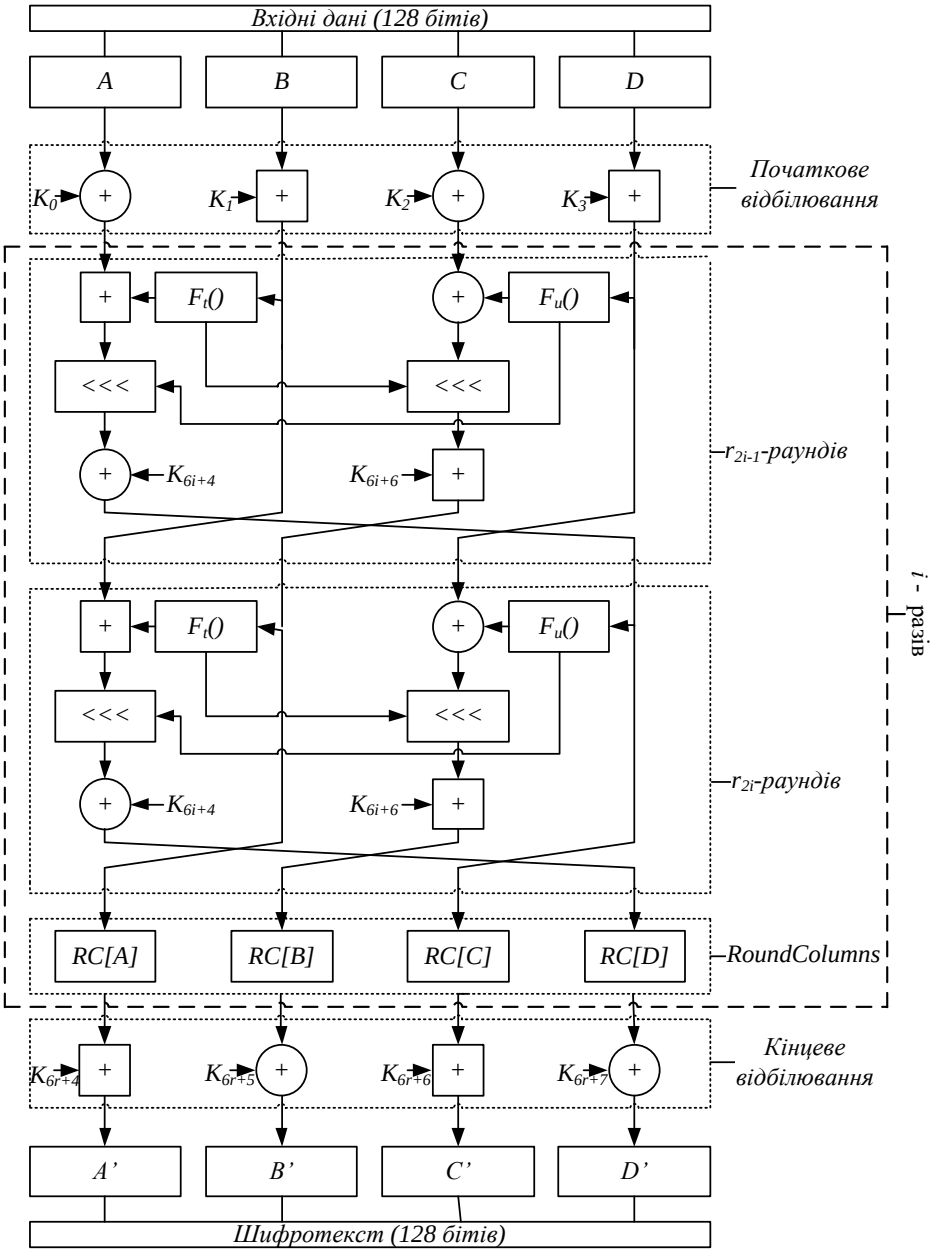


*Рис. 2. Загальна схема роботи процедури зашифрування*

ISSN 2663 - 4023

Таким чином в якості базової схеми раундового перетворення взято мережі Фейстеля. Схема раундового перетворення наведена на рис. 5.

***Функції*** $Ft()$ ***і*** $Fu()$

На рис. 6 зображена схема роботи функцій $Ft()$ і $Fu()$ для $i$-го раунду $(i=\overline{1,..,r})$. Як видно вони практично однакові, за винятком підключів, які в них використовуються, та констант $l$ , $m$.

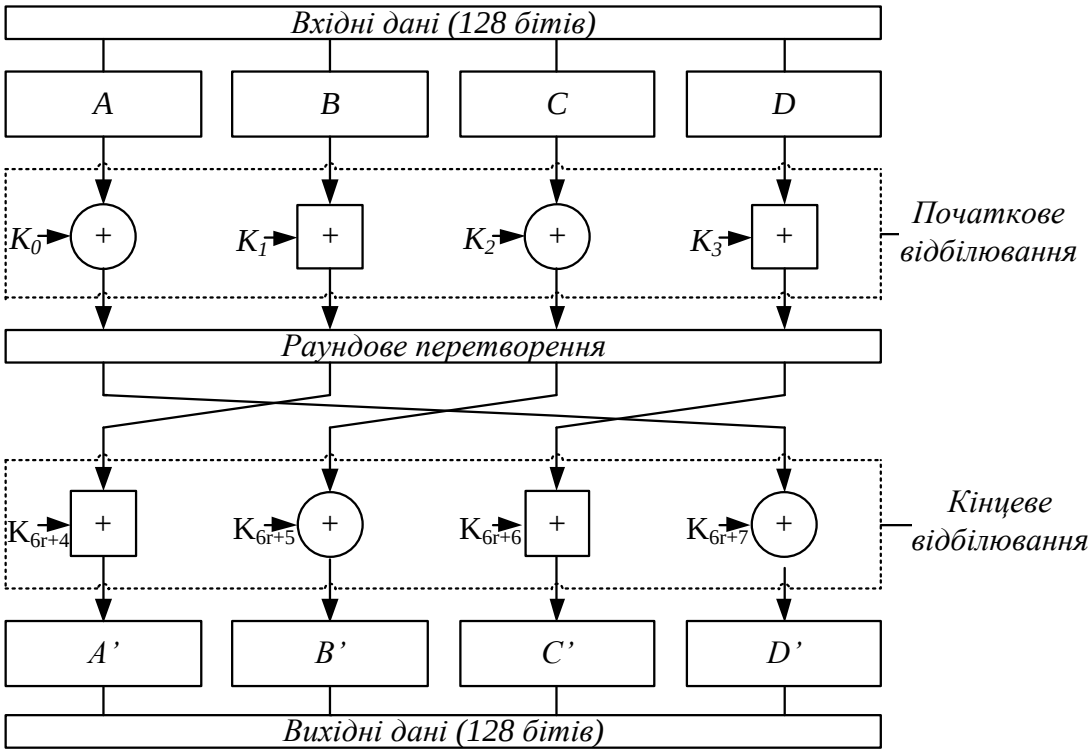


*Рис. 3. Початкова та фінальна рандомізації в алгоритмі*

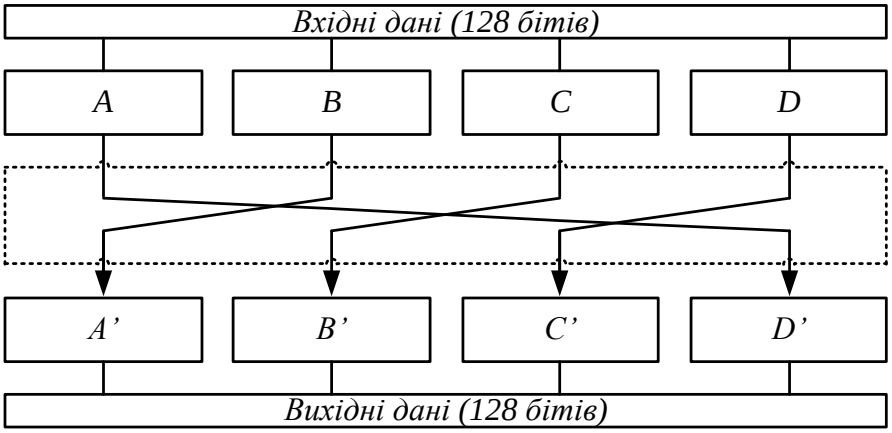


*Рис. 4. Заміщення матриць*

ISSN 2663 - 4023

Опишемо послідовність виконання функцій:

1) Кожний байт початкового значення підблоків $U$ і $T$ заміняють використовуючи таблицю підстановок ($S$-блок) (див. табл.2): $U = S(U)$, $T = S(T)$. Для заміни кожний байт $U$ і $T$ розбивають на дві частини: молодші 4 біти будуть означати необхідний стовпець, старші – необхідний рядок, їх перетин у таблиці і буде результатом. Наприклад, якщо байт який потрібно замінити *={53}*, то результат заміни необхідно шукати на перетині рядка з індексом "5" та стовпця з індексом "3", в результаті отримаємо *{ed}*.

2) Кожен байт $U$ і $T$ перемножують на розраховані для кожного раунду константи $l$ і $m$ за модулем $2^{32}$: $U = (U * m)$ і $T = (T * l)$. Приклад такої операції:

$$U = (l * U) \bmod 2^{32},\ T = (m * M) \bmod 2^{32},.$$

3) Обраховують допоміжні послідовності $UU$ і $TT$: $UU = (U + K_{6i+7})$, $TT = (T + K_{6i+5})$.

4) Кожен байт $UU$ і $TT$ циклічно зсувають вліво на розраховані для кожного раунду константи $l$ і $m$ відповідно: $UU = (UU <<< l)$, $TT = (TT <<< m)$.

5) Перемножують матриці $U$ і $UU$, $T$ і $TT$: $U = (U * UU)$, $T = (T * TT)$ за модулем $2^{32}$.

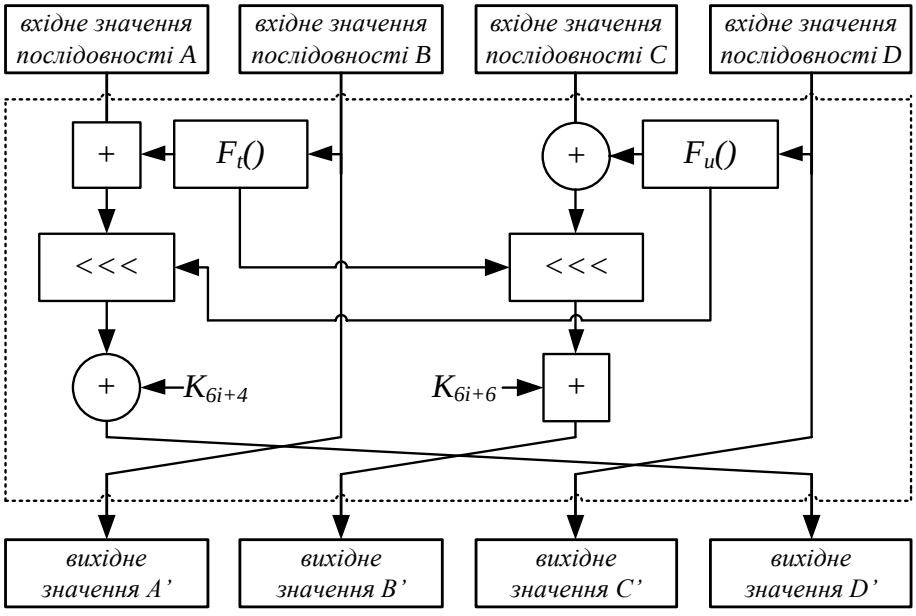


*Рис. 5. Раундове перетворення алгоритму*

ISSN 2663 - 4023

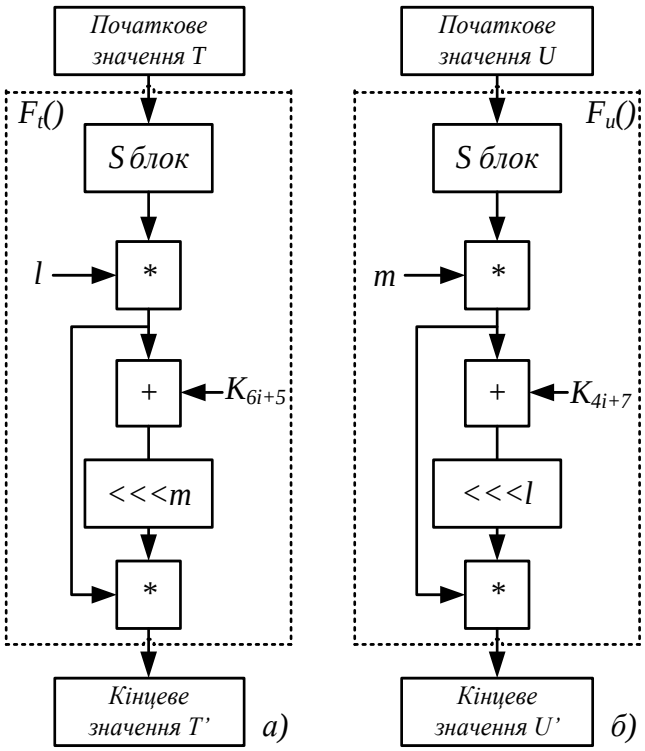


*Рис. 6. Для і-ого раунду функції: а) Ft() ; б) Fu()*

*Таблиця 1*

**Таблиця підстановок**

| | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **a** | **b** | **c** | **d** | **e** | **f** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
| **0** | 1F | 1B | 9B | AC | FC | D8 | 28 | B2 | 40 | B8 | EF | 6D | CB | F4 | 22 | 24 |
| **1** | 3A | D1 | 85 | F7 | 3B | AE | 60 | 46 | 94 | 8A | DB | D2 | 6C | B7 | D4 | 39 |
| **2** | DE | 17 | 1A | A | F6 | BE | 89 | 4A | B0 | C3 | 14 | D0 | 11 | 4E | 82 | BF |
| **3** | B1 | 2F | 11 | E8 | BD | E0 | 48 | 47 | E | C0 | C8 | 91 | F0 | 37 | 8C | 3F |
| **4** | 57 | 43 | 84 | 16 | F5 | 1 | DC | CE | E7 | 2B | 3D | 10 | E9 | 42 | 9D | F1 |
| **5** | 1E | C4 | F | FA | D6 | C1 | 74 | F2 | 3C | 81 | 19 | 93 | 20 | 56 | 53 | 4C |
| **6** | 70 | BA | 1D | 92 | D7 | 54 | ED | 64 | 6F | FE | 76 | 4F | A1 | 13 | EC | 86 |
| **7** | 58 | C7 | 5D | 3E | C6 | E1 | 5B | 4D | 5F | 4 | AF | B6 | 3 | A9 | 34 | 55 |
| **8** | 75 | DF | 68 | 75 | 98 | AD | EA | 2A | B5 | E3 | 15 | 8D | 6B | B | 7E | A8 |
| **9** | 5 | 8E | 99 | 5 | 8 | 2 | 52 | 26 | 83 | EB | 6 | 45 | 44 | E2 | 31 | D |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **a** | 35 | B3 | D9 | 35 | 36 | E4 | F8 | CA | CC | 7B | 33 | CF | BC | E6 | 63 | 9E |
| **b** | 6A | EE | 72 | 6A | A7 | 1C | 67 | 21 | 50 | 27 | AB | 2C | 41 | C | 25 | 9A |
| **c** | 32 | F3 | B9 | 32 | 23 | D3 | 9 | DD | A2 | C9 | 97 | 96 | 7 | 9C | 95 | 7F |
| **d** | CD | B4 | 7C | CD | 80 | 2D | 77 | 73 | 59 | D5 | 0 | 49 | 69 | 87 | A4 | 90 |
| **e** | 7D | A | 8B | 7D | 62 | 2E | 5A | 79 | E5 | 61 | 18 | FB | DA | A5 | 4B | 6E |
| **f** | A3 | 8F | FF | A3 | 7A | 9F | A6 | 78 | 29 | 12 | C2 | 65 | FD | 30 | F9 | 51 |

***Перемішування в колонках (RoundColumns)***

У ході перетворення *RoundColumns* виконується послідовна обробка всіх стовпчиків поточного стану. Блоки *A+ B, C+D* формують дві 8-байтні колонки, кожна з яких розглядається як поліном над полем $GF(2^8)$ з 8 термами, а в ході перетворення виконується множення цього полінома за модулем $x^8 + 1$ на фіксований поліном *c(x)*, де

$$c(x) = \{01\}x^7 + \{05\}x^6 + \{01\}x^5 + \{08\}x^4 + \{06\}x^3 + \{07\}x^2 + \{04\}x + \{01\}$$

Ця операція перетворює кожен стовпець матриці станів в новий стовпець. Це фактично матричне множення стовпця матриці станів і квадратної матриці констант (взята з «Калина»). Матриця станів формується з блоків даних *A+B, C +D*, кожен з яких утворює відповідний стовпець матриці, що складається з 8-ох байт. Байти в стовпці матриці станів і в матриці констант інтерпретуються як слова по 8 бітів (або поліноми) з коефіцієнтами в *GF*(2). Додавання – це використання операції ВИКЛЮЧНЕ АБО (XOR) до слів по 8 біт.

Тобто це перетворення еквівалентне матричному множенню над $GF(2^8)$ вихідного 8-байтного вектора *a* на фіксовану матрицю, результат заноситься в 8-байтний вектор *b*.

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} = \begin{bmatrix} 01 & 01 & 05 & 01 & 08 & 06 & 07 & 04 \\ 04 & 01 & 01 & 05 & 01 & 08 & 06 & 07 \\ 07 & 04 & 01 & 01 & 05 & 01 & 08 & 06 \\ 06 & 07 & 04 & 01 & 01 & 05 & 01 & 08 \\ 08 & 06 & 07 & 04 & 01 & 01 & 05 & 01 \\ 01 & 08 & 06 & 07 & 04 & 01 & 01 & 05 \\ 05 & 01 & 08 & 06 & 07 & 04 & 01 & 01 \\ 01 & 05 & 01 & 08 & 06 & 07 & 04 & 01 \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix}$$

Порядок обчислення елементів підсумкового вектора *b* можна пояснити так:

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} = \begin{bmatrix} 01 \times a_0 \oplus 01 \times a_1 \oplus 05 \times a_2 \oplus 01 \times a_3 \oplus 08 \times a_4 \oplus 06 \times a_5 \oplus 07 \times a_6 \oplus 04 \times a_7 \\ 04 \times a_0 \oplus 01 \times a_1 \oplus 01 \times a_2 \oplus 05 \times a_3 \oplus 01 \times a_4 \oplus 08 \times a_5 \oplus 06 \times a_6 \oplus 07 \times a_7 \\ 07 \times a_0 \oplus 04 \times a_1 \oplus 01 \times a_2 \oplus 01 \times a_3 \oplus 05 \times a_4 \oplus 01 \times a_5 \oplus 08 \times a_6 \oplus 06 \times a_7 \\ 06 \times a_0 \oplus 07 \times a_1 \oplus 04 \times a_2 \oplus 01 \times a_3 \oplus 01 \times a_4 \oplus 05 \times a_5 \oplus 01 \times a_6 \oplus 08 \times a_7 \\ 08 \times a_0 \oplus 06 \times a_1 \oplus 07 \times a_2 \oplus 04 \times a_3 \oplus 01 \times a_4 \oplus 01 \times a_5 \oplus 05 \times a_6 \oplus 01 \times a_7 \\ 01 \times a_0 \oplus 08 \times a_1 \oplus 06 \times a_2 \oplus 07 \times a_3 \oplus 04 \times a_4 \oplus 01 \times a_5 \oplus 01 \times a_6 \oplus 05 \times a_7 \\ 05 \times a_0 \oplus 01 \times a_1 \oplus 08 \times a_2 \oplus 06 \times a_3 \oplus 07 \times a_4 \oplus 04 \times a_5 \oplus 01 \times a_6 \oplus 01 \times a_7 \\ 01 \times a_0 \oplus 05 \times a_1 \oplus 01 \times a_2 \oplus 08 \times a_3 \oplus 06 \times a_4 \oplus 07 \times a_5 \oplus 04 \times a_6 \oplus 01 \times a_7 \end{bmatrix}$$

Всі операції множення виконуються над полем $GF(2^8)$. Операція лінійного розсіювання (перемішування в колонці) представленого алгоритму використовує поліноміальне подання байтів у полі $GF(2^8)$, яке утворене поліномом, що не приводиться. Для представленого алгоритму шифрування в якості поліному, що не приводиться, використовується:

$$m(x) = x^8 + x^4 + x^3 + x^2 + 1,$$

або {01}{1d} у шістнадцятковому поданні. Слід зазначити, що цей поліном, що не приводиться, не збігається з утворюючим поліномом AES.

Рис. 7 пояснює порядок виконання перетворення *RoundColumns* для поточного стану шифру.

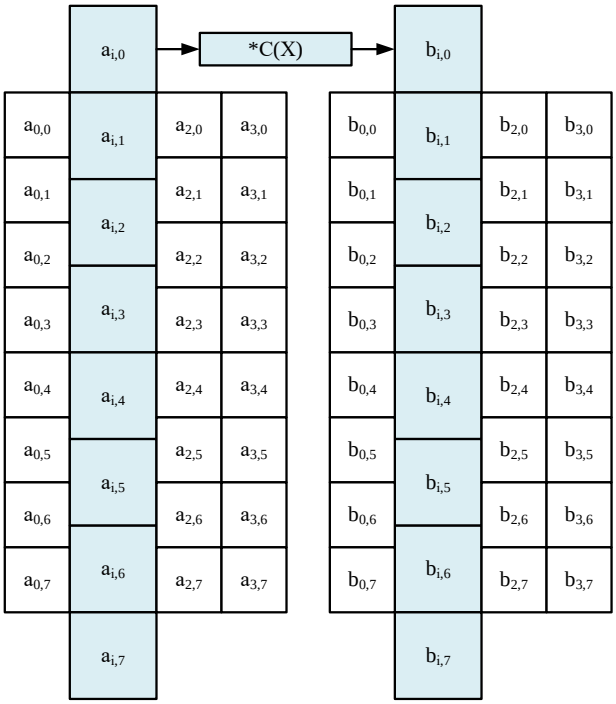


*Рис. 7. Порядок виконання перетворення RoundColumns*

***Процедура розширення підключів***

Процедура розширення ключів виконується за допомогою розробленого генератора псевдовипадкових чисел, у якості вектора ініціалізації виступає секретний ключ. Генерується $t = 6r + 8$ розширених ключів.

***Розрахунок констант l і m для кожного раунду***

Для *i*-ого раунду *l* і *m* розраховуються так:

ISSN 2663 - 4023

$$l = S\left[\left(S\left[K_{6i+8}\right] + K_{6i+9}\right) \bmod 2^{32} <<< 7\right],$$

$$m = S\left[\left(K_{6i+8} + S\left[K_{6i+9}\right]\right) \bmod 2^{32}\right] <<< 11.$$

У алгоритмі передбачено те, що змінні $l$ і $m$ не можуть дорівнювати нулю. Якщо $l$ і $m$ дорівнює нулю, то $l = a_{32} + 1$, $m = b_{32} + 3$. Для цього відбувається додавання константи у кожні з формул.

***Процедура розшифрування***

При розшифруванні підключі використовуються в зворотному порядку, накладання підключів замість додавання по модулю $2^n$ виконується відніманням, зрушення субблоків виконується на початку раунду і у зворотний бік, а також замість процедури *RoundColumns* використовується зворотня до неї – *InvRoundColumns*. Перетворення *f()* не зазнало змін (Рис. 9). Загальна схема роботи процедури зашифрування представлена на рис. 8.

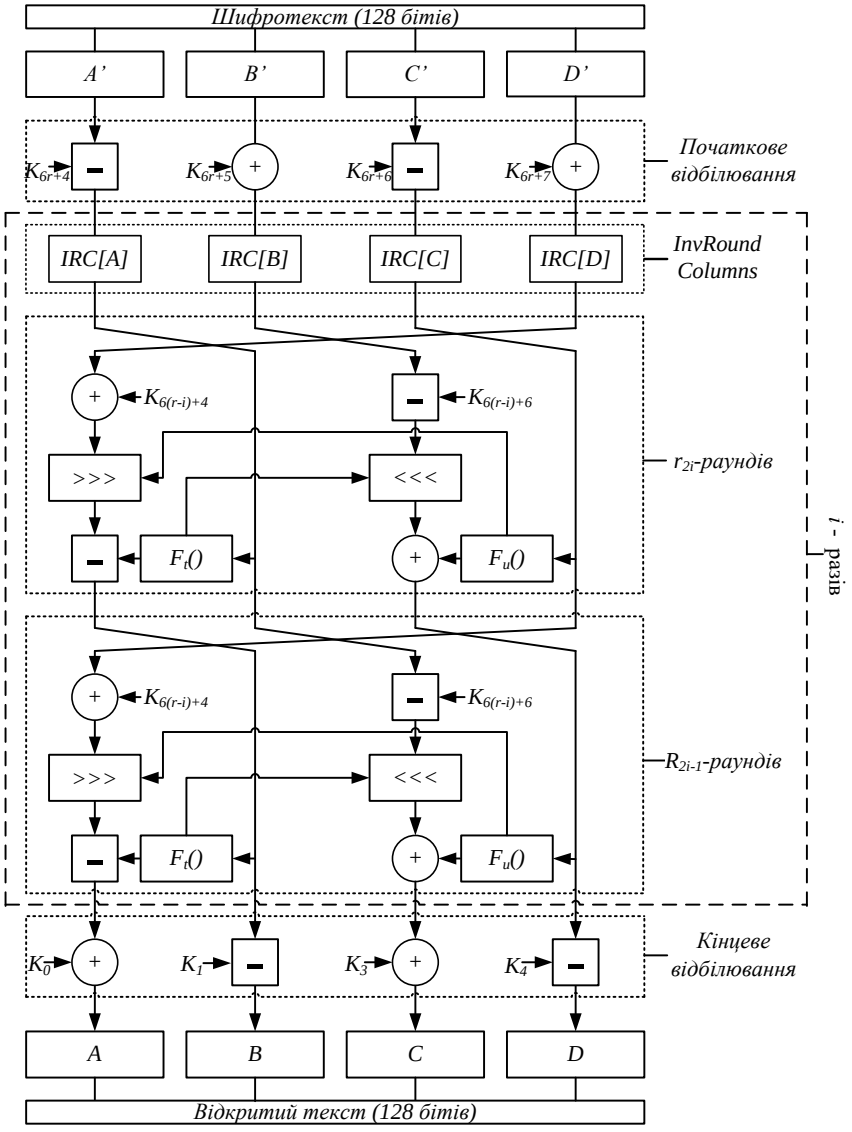


*Рис. 8. Загальна схема роботи процедури розшифрування*

ISSN 2663 - 4023

Зворотне для *RoundColumns* перемішування в колонках (*InvRoundColumns*) полягає в множенні кожного стовпчика на зворотній для $c(x)$ поліном $d(x)$:

$$d(x) = \{95\}x^7 + \{76\}x^6 + \{A8\}x^5 + \{2F\}x^4 + \{49\}x^3 + \{D7\}x^2 + \{CA\}x + \{AD\}$$

Рис. 10 показує матриці констант, використовувані для перетворень *RoundColumns* та *InvRoundColumns*. Ці дві матриці інверсні одна до одної, коли елементи інтерпретуються як слова з 8-ми бітів (або поліноми) з коефіцієнтами в $GF(2^8)$.

Якщо дві матриці констант інверсні одна до одної, то легко довести, що ці два перетворення також інверсні один одному.

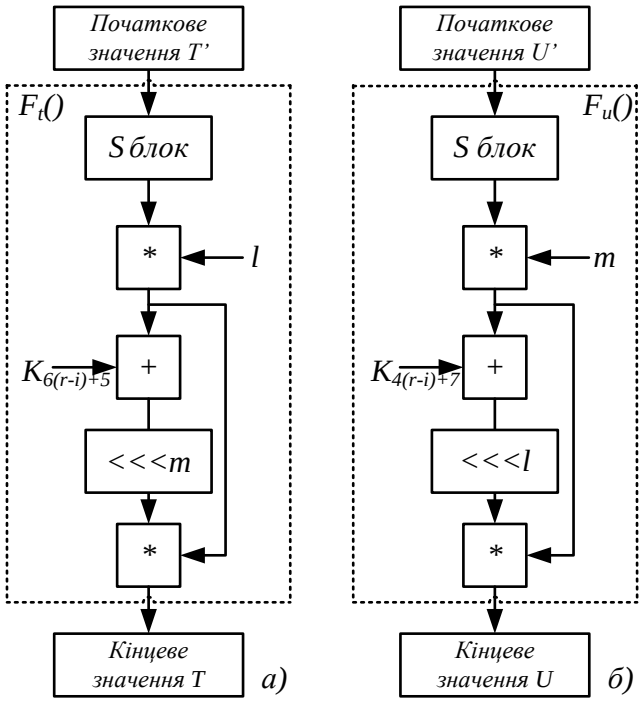


*Рис. 9. Для і-ого раунду при розшифруванні функції: а) Ft() ; б) Fu()*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01 | 01 | 05 | 01 | 08 | 06 | 07 | 04 |
| 04 | 01 | 01 | 05 | 01 | 08 | 06 | 07 |
| 07 | 04 | 01 | 01 | 05 | 01 | 08 | 06 |
| 06 | 07 | 04 | 01 | 01 | 05 | 01 | 08 |
| 08 | 06 | 07 | 04 | 01 | 01 | 05 | 01 |
| 01 | 08 | 06 | 07 | 04 | 01 | 01 | 05 |
| 05 | 01 | 08 | 06 | 07 | 04 | 01 | 01 |
| 01 | 05 | 01 | 08 | 06 | 07 | 04 | 01 |

C

Інверсія

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| AD | 95 | 76 | A8 | 2F | 49 | D7 | CA |
| CA | AD | 95 | 76 | A8 | 2F | 49 | D7 |
| D7 | CA | AD | 95 | 76 | A8 | 2F | 49 |
| 49 | D7 | CA | AD | 95 | 76 | A8 | 2F |
| 2F | 49 | D7 | CA | AD | 95 | 76 | A8 |
| A8 | 2F | 49 | D7 | CA | AD | 95 | 76 |
| 76 | A8 | 2F | 49 | D7 | CA | AD | 95 |
| 95 | 76 | A8 | 2F | 49 | D7 | CA | AD |

$C^{-1}$

*Рис. 10. Матриці перетворень RoundColumns та InvRoundColumns*

Розглянемо приклад описаних перетворень: Рис. 11 показує, як матриця станів зміниться при використанні перетворення *RoundColumns*. На рис. 11 також показано, що перетворення *InvRoundColumns* створює первинний текст.

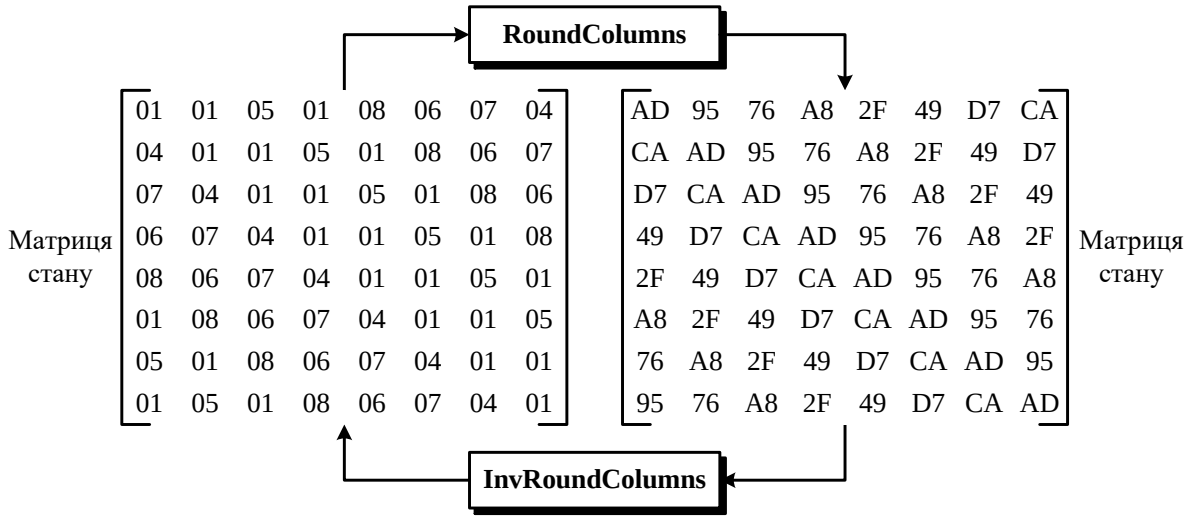


*Рис. 11. Перетворення RoundColumns в прикладі*

Варто зауважити, що байти, які рівні між собою в старій матриці станів, більше не рівні в новій матриці станів. Наприклад, два байти *01* в другому рядку змінено на *AD* і *95*.

## 4. ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У роботі проведено аналіз вимог до побудови систем забезпечення конфіденційності даних на базі криптоалгоритмів, визначено ключові аспекти і шляхи удосконалення існуючих методів і систем шифрування даних. Розроблено метод генерації криптографічних ключів, щоб покращити швидкість генерації ключів, з його використанням удосконалено функцію шифрування (для забезпечення удосконаленого модуля) на основі відомого і ефективного алгоритму RC6, що дозволило підвищити швидкість криптографічної обробки даних та перевірити криптостійкість алгоритму проти спеціалізованих атак лінійного та диференціального криптоаналізу.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1 Oppliger, R. (2021). *Cryptography 101: From Theory to Practice.* Artech.
2 Job, J, Naresh, V, Chandrasekaran, K. (2015). A modified secure version of the Telegram protocol (MTProto). У *2015 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*. IEEE. https://doi.org/10.1109/conecct.2015.7383884
3 Dion van Dam. (2019). *Analysing the Signal Protocol. A manual and automated analysis of the Signal Protocol*. Radboud University.
4 (2011). *TLS and SRTP for Skype Connect Technical Datasheet.* Skype.
5 Wu, Q. (2015). A Chaos-Based Hash Function. У *2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*. IEEE. https://doi.org/10.1109/cyberc.2015.13
6 Gnatyuk, S., Kinzeryavyy, V., Kyrychenko, K., Yubuzova, K., Aleksander, M., & Odarchenko, R. (2019). Secure Hash Function Constructing for Future Communication Systems and Networks. У *Advances in*

*Artificial Systems for Medicine and Education II* (p. 561–569). Springer International Publishing. https://doi.org/10.1007/978-3-030-12082-5_51.

7 Rajeshwaran, K., Anil Kumar, K. (2019). Cellular Automata Based Hashing Algorithm (CABHA) for Strong Cryptographic Hash Function. У *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*. IEEE. https://doi.org/10.1109/icecct.2019.8869146

8 Iavich, M., Iashvili, G., Gnatyuk, S., Tolbatov, A., Mirtskhulava, L. (2021). Efficient and Secure Digital Signature Scheme for Post Quantum Epoch. *Communications in Computer and Information Science*, 1486, 185-193.

9 Gnatyuk, S., Iavich, M., Kinzeryavyy, V., Okhrimenko, T., Burmak, Y., Goncharenko, I. (2020). Improved secure stream cipher for cloud computing. *CEUR Workshop Proceedings*, 2732, 183-197.

10 Gnatyuk, S., Akhmetov, B., Kozlovskyi, V., Kinzeryavyy, V., Aleksander, M., Prysiazhnyi, D. (2020). New Secure Block Cipher for Critical Applications: Design, Implementation, Speed and Security Analysis. У *Advances in Intelligent Systems and Computing* (p. 93–104). Springer International Publishing. https://doi.org/10.1007/978-3-030-39162-1_9.

11 Kuznetsov, A., Horkovenko, I., Maliy, O., Goncharov, N., Kuznetsova, T., Kovalenko, N. (2020). Non-Binary Cryptographic Functions for Symmetric Ciphers. У *2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T)*. IEEE. https://doi.org/10.1109/picst51311.2020.9467982.

12 Jintcharadze, E., Iavich, M. (2020). Hybrid Implementation of Twofish, AES, ElGamal and RSA Cryptosystems. У *2020 IEEE East-West Design & Test Symposium (EWDTS)*. IEEE. https://doi.org/10.1109/ewdts50664.2020.9224901.

13 Lee, T. R., Teh, J. S., Jamil, N., Yan, J. L. S., Chen, J. (2021). Lightweight Block Cipher Security Evaluation Based on Machine Learning Classifiers and Active S-Boxes. У *IEEE Access* (p. 134052-134064). https://doi.org/10.1109/ACCESS.2021.3116468.

14 Smirnova, T., Polishchuk, L., Smirnov, O., Buravchenko, K., Makevnin, A. (2020). RESEARCH OF CLOUDY TECHNOLOGIES AS A SERVICES. *Cybersecurity: Education, Science, Technique*, *3*(7), 43–62. https://doi.org/10.28925/2663-4023.2020.7.4362.

15 Smirnov, T., Solovykh, Y., Smirnov, O., Drieiev, O. (2019). Construction of Cloud information Technologies for Optimization of Technological Process of Restoration and Strengthening of Surfaces of Parts. *Central Ukrainian Scientific Bulletin. Technical Sciences*, (1(32)), 184–194. https://doi.org/10.32515/2664-262x.2019.1(32).184-194.

16 Smirnova, T.V., Smirnov S.A., Minaylenko, R.M., Dorensky, O.P., Sysoenko, S.V. (2020). Cloud automated system of intelligent decision support for technological processes. *Bulletin of Cherkasy State Technological University. Technical sciences*, 4, 84-92.

17 Smirnova, T.V., Buravchenko, K.O., Kravchenko, S.S., Gorbov, V.O., Smirnov, O.A. (2021). Cloud system to support decision-making of the technological process of restoration of surfaces of structures and parts of machines. *Modern information systems, 5*(4), 79-95.

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 1 (13), 2021
ISSN 2663 - 4023

**Tetiana Smirnova**
Candidate of Science (Engineering), Associate Professor of Cybersecurity & Software Academic Department
Central Ukrainian National Technical University, Kropyvnytskyi, Ukraine
ORCID ID:0000-0001-6896-0612
*sm.tetyana.gmail.com*

**Yuliia Burmak**
applicant for a research laboratory to combat cyber threats in the aviation industry
National Aviation University, Kyiv, Ukraine
ORCID ID: 0000-0002-5410-6260
*julburmac@gmail.com*

**Oleksandr Ulichev**
Candidate of Science (Engineering), Senior Lecturer of Cybersecurity & Software Academic Department
Central Ukrainian National Technical University, Kropyvnytskyi, Ukraine
ORCID ID: 0000-0003-3736-9613
*askin79@gmail.com*

**Pavlo Usik**
Doctor of Philosophy (PhD), Senior Lecturer of Cybersecurity & Software Academic Department
Central Ukrainian National Technical University, Kropyvnytskyi, Ukraine
ORCID ID: 0000-0002-3268-342X
*mr.usik@ukr.net*

**Oleksandr Dorenskyi**
Candidate of Science (Engineering), Associate Professor of Cybersecurity & Software Academic Department
Central Ukrainian National Technical University, Kropyvnytskyi, Ukraine
ORCID ID: 0000-0002-7625-9022
*dorensky@ukr.net*

# STABLE ENCRYPTION FUNCTION OF THE ADVANCED MODULE OF CRYPTOGRAPHIC PROTECTION OF INFORMATION IN INFORMATION AND COMMUNICATION SYSTEMS

**Abstract.** The paper analyzes the requirements for the construction of data privacy systems based on cryptographic algorithms, identifies key aspects and ways to improve existing methods and systems of data encryption. Research methods. The main theoretical provisions of the work are obtained using the methods of information security theory. The object of research is the process of ensuring the confidentiality of data in information and communication systems management systems based on cloud technologies. The subject of the study is a stable encryption function to provide an advanced module of cryptographic protection of information in information and communication systems. The aim of this work is to develop a stable encryption function of the advanced module of cryptographic protection of information to ensure data confidentiality in information and communication systems management processes based on cloud technologies. A method of generating cryptographic keys was developed to improve the speed of key generation, using an encryption function (to provide an advanced module) based on the well-known and efficient RC6 algorithm, which increased the speed of cryptographic data processing and tested cryptographic stability of the algorithm.

**Keywords:** stable encryption function, cryptographic protection, information and communication system.

## REFERENCES

1 Oppliger, R. (2021). *Cryptography 101: From Theory to Practice.* Artech.
2 Job, J, Naresh, V, Chandrasekaran, K. (2015). A modified secure version of the Telegram protocol (MTProto). In *2015 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*. IEEE. https://doi.org/10.1109/conecct.2015.7383884

3 Dion van Dam. (2019). *Analysing the Signal Protocol. A manual and automated analysis of the Signal Protocol*. Radboud University.
4 (2011). *TLS and SRTP for Skype Connect Technical Datasheet.* Skype.
5 Wu, Q. (2015). A Chaos-Based Hash Function. In *2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*. IEEE. https://doi.org/10.1109/cyberc.2015.13
6 Gnatyuk, S., Kinzeryavyy, V., Kyrychenko, K., Yubuzova, K., Aleksander, M., & Odarchenko, R. (2019). Secure Hash Function Constructing for Future Communication Systems and Networks. In *Advances in Artificial Systems for Medicine and Education II* (p. 561–569). Springer International Publishing. https://doi.org/10.1007/978-3-030-12082-5_51.
7 Rajeshwaran, K., Anil Kumar, K. (2019). Cellular Automata Based Hashing Algorithm (CABHA) for Strong Cryptographic Hash Function. In *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*. IEEE. https://doi.org/10.1109/icecct.2019.8869146
8 Iavich, M., Iashvili, G., Gnatyuk, S., Tolbatov, A., Mirtskhulava, L. (2021). Efficient and Secure Digital Signature Scheme for Post Quantum Epoch. *Communications in Computer and Information Science*, 1486, 185-193.
9 Gnatyuk, S., Iavich, M., Kinzeryavyy, V., Okhrimenko, T., Burmak, Y., Goncharenko, I. (2020). Improved secure stream cipher for cloud computing. *CEUR Workshop Proceedings,* 2732, 183-197.
10 Gnatyuk, S., Akhmetov, B., Kozlovskyi, V., Kinzeryavyy, V., Aleksander, M., Prysiazhnyi, D. (2020). New Secure Block Cipher for Critical Applications: Design, Implementation, Speed and Security Analysis. In *Advances in Intelligent Systems and Computing* (p. 93–104). Springer International Publishing. https://doi.org/10.1007/978-3-030-39162-1_9.
11 Kuznetsov, A., Horkovenko, I., Maliy, O., Goncharov, N., Kuznetsova, T., Kovalenko, N. (2020). Non-Binary Cryptographic Functions for Symmetric Ciphers. In *2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T)*. IEEE. https://doi.org/10.1109/picst51311.2020.9467982.
12 Jintcharadze, E., Iavich, M. (2020). Hybrid Implementation of Twofish, AES, ElGamal and RSA Cryptosystems. In *2020 IEEE East-West Design & Test Symposium (EWDTS)*. IEEE. https://doi.org/10.1109/ewdts50664.2020.9224901.
13 Lee, T. R., Teh, J. S., Jamil, N., Yan, J. L. S., Chen, J. (2021). Lightweight Block Cipher Security Evaluation Based on Machine Learning Classifiers and Active S-Boxes. In *IEEE Access* (p. 134052-134064). https://doi.org/10.1109/ACCESS.2021.3116468.
14 Smirnova, T., Polishchuk, L., Smirnov, O., Buravchenko, K., Makevnin, A. (2020). RESEARCH OF CLOUDY TECHNOLOGIES AS A SERVICES. *Cybersecurity: Education, Science, Technique*, *3*(7), 43–62. https://doi.org/10.28925/2663-4023.2020.7.4362.
15 Smirnov, T., Solovykh, Y., Smirnov, O., Drieiev, O. (2019). Construction of Cloud information Technologies for Optimization of Technological Process of Restoration and Strengthening of Surfaces of Parts. *Central Ukrainian Scientific Bulletin. Technical Sciences*, (1(32)), 184–194. https://doi.org/10.32515/2664-262x.2019.1(32).184-194.
16 Smirnova, T.V., Smirnov S.A., Minaylenko, R.M., Dorensky, O.P., Sysoenko, S.V. (2020). Cloud automated system of intelligent decision support for technological processes. *Bulletin of Cherkasy State Technological University. Technical sciences*, 4, 84-92.
17 Smirnova, T.V., Buravchenko, K.O., Kravchenko, S.S., Gorbov, V.O., Smirnov, O.A. (2021). Cloud system to support decision-making of the technological process of restoration of surfaces of structures and parts of machines. *Modern information systems*, *5*(4), 79-95.

(cc) BY-NC-SA

This work is licensed under Creative Commons Attribution-noncommercial-sharealike 4.0 International License.