ВСЕУКРАЇНСЬКИЙ КОНКУРС СТУДЕНТСЬКИХ НАУКОВИХ РОБІТ З ПРИРОДНИЧИХ, ТЕХНІЧНИХ ТА ГУМАНІТАРНИХ НАУК

НАУКОВА РОБОТА

ЗА СПЕЦІАЛЬНІСТЮ **«Кібербезпека»**

на тему:

«СИМЕТРИЧНИЙ АЛГОРИТМ НЕЛІНІЙНОГО ШИФРУВАННЯ З МОЖЛИВІСТЮ СТЕГАНОГРАФІЧНОГО ЗАСТОСУВАННЯ»

під шифром “Криптосистема”

Хмельницький 2021

ЗМІСТ

ВСТУП 3

1 ДОСЛІДЖЕННЯ СУЧАСНИХ СИМЕТРИЧНИХ КРИПТОСИСТЕМ ТА ПРИНЦИПІВ ЇХ ФУНКЦІОНУВАННЯ 4

2 МОДИФІКАЦІЯ ІСНУЮЧИХ АЛГОРИТМІВ ШИФРУВАННЯ З ВИКОРИСТАННЯМ ПРИНЦИПУ НЕЛІНІЙНОСТІ 8

3 ВДОСКОНАЛЕННЯ МЕТОДУ НЕЛІНІЙНОГО ШИФРУВАННЯ НА ОСНОВІ РОЗГАЛУЖЕНОСТІ КРИПТОГРАФІЧНОГО АЛГОРИТМУ.. Error! Bookmark not defined.14

3.1 Розробка симетричної нелінійної криптосистеми 14

3.2 Генерація ключа 19

3.3 Криптографічна стійкість алгоритму 20

3.4 Створення програмного прототипу криптосистеми 21

4 МОДИФІКАЦІЯ АЛГОРИТМУ ДЛЯ ПРИХОВАНОГО ЗБЕРІГАННЯ ТА ПЕРЕДАЧІ ІНФОРМАЦІЇ 26

ВИСНОВКИ 29

ПЕРЕЛІК ПОСИЛАНЬ 30

ДОДАТОК А КОПІЇ АКТІВ ВПРОВАДЖЕННЯ 33

ДОДАТОК Б КОПІЇ НАУКОВИХ ПРАЦЬ ПО ТЕМІ ДОСЛІДЖЕННЯ 34

ВСТУП

Актуальність дослідження. Інформаційна безпека та захист інформації є вкрай актуальною проблемою на сьогодні. Необхідність захищати персональну, конфіденційну інформацію, корпоративну, або державну таємницю регулярно виникає як у рядових користувачів, так і у приватних підприємців та державних структур, адже своєчасна інформація, як ресурс, з кожним днем набуває все більшу цінність.

З дуже давніх часів людство використовувало безліч способів захисту важливих даних. Найвдаліші з них вдосконалювалися, модернізувалися та дійшли до нас у вигляді таких наукових напрямків як криптографія та стеганографія.

Сучасна криптографія є математично-комбінаторною наукою, що вивчає способи перетворення інформації з метою її захисту від несанкціонованого використання та з можливістю дешифровки перетвореної інформації назад у осмислений вигляд, зі збереженням цілісності та ідентичності вхідних та вихідних даних. Проблемою сучасної криптографії є гігантський прорив у області електронних обчислювальних машин, в силу чого незламні до цього криптосистеми стали вразливими. Тому саме стійкість до автоматизованого криптоаналізу, з використанням комп’ютеризованих обчислень, фактична потужність яких збільшується мало не щодня, є головною вимогою до сучасних криптосистем.

Стеганографія також є невід’ємною складовою сучасного захисту інформації, оскільки криптографічне повідомлення, яке представляє собою неосмислений набір байт, неодмінно приверне до себе увагу, в той час як стеганографічне повідомлення у вигляді зображення, чи будь-яких інших даних, що самі по собі жодної цінності не представляють, з великою ймовірністю залишиться не поміченим.

Комбінація криптографічної та стеганографічної складової гарантує високий рівень захисту даних, а розробка алгоритмів, що включатимуть в себе обидві зазначені складові, є вкрай актуальною задачею.

Існуючі сьогодні симетричні алгоритми шифрування, такі як Triple DES, IDEA, AES та інші, надійно захищають інформацію, відносно поточного рівня розвитку обчислювальної техніки, проте вони не можуть використовуватись у стеганографічних цілях, оскільки вихідний шифротекст у цих алгоритмах є важкопрогнозованим. Створення алгоритму шифрування, з можливістю впливати на вихідний шифротекст, зробить можливим його змішане крипто-стеганографічне використання.

Об'єктом дослідження є шифрування інформації в інформаційно-телекомунікаційних системах, з застосуванням криптографічних методів захисту.

Предметом дослідження є методи і алгоритми криптографічного та стеганографічного захисту інформації.

Метою дослідження є створення криптостійкого алгоритму шифрування з можливістю комбінованого криптографічно-стеганографічного використання.

Науковою новизною дослідження є вдосконалення криптографічного методу нелінійного шифрування на основі гілкового розгалуження алгоритму та потокового вибору випадкового варіанту шифрування, з метою реалізації надійного криптографічно-стеганографічного захисту даних.

Методи дослідження: апарат теорії алгоритмів, теорії захисту інформації, системного аналізу, комп'ютерного та математичного моделювання.

Практична цінність дослідження полягає у прикладній направленості вдосконаленого методу та реалізованих на його основі алгоритмів шифрування, а також їх використання для стеганографічного приховування даних.

Публікації. За результатами наукової роботи опубліковано 1 фахова стаття і 2 тези доповідей на міжнародній і всеукраїнській конференціях.

1 ДОСЛІДЖЕННЯ СУЧАСНИХ СИМЕТРИЧНИХ КРИПТОСИСТЕМ ТА ПРИНЦИПІВ ЇХ ФУНКЦІОНУВАННЯ

На сьогоднішній день існує велика кількість симетричних криптосистем, що забезпечують достатній захист інформації. Найвідомішими серед них є такі алгоритми шифрування як DES, DESX, Triple DES, IDEA, AES та інші. В деяких з них були знайдені вразливості, що дають можливість провести успішний криптоаналіз, інші ж до цих пір умовно вважаються незламними. Проте, не дивлячись на це, серйозною загрозою для подібних алгоритмів є стрімкий розвиток комп’ютерних технологій та, як наслідок, швидке зростання практичної швидкості обчислювальної потужності [1, 2, 3].

Яскравим прикладом цього є алгоритм DES: в часи його розробки довжина ключа в 56 біт здавалася цілком достатньою, а атака повним перебором ключів здавалась абсолютно не практичною. Проте лише за кілька десятиліть дана атака стала можливою практично в домашніх умовах, з використанням обчислювальних потужностей, доступних рядовому користувачу [3, 4].

Подібно до цього, з появою нових технологій, одною з яких, теоретично, може бути квантовий комп’ютер, який за прогнозами може обігнати найкращі з існуючих комп’ютерів в мільйон раз, за швидкістю обчислень, існуючі симетричні алгоритми та класичний підхід до криптографії в цілому може виявитись під загрозою [3]. Одним з варіантів уникнення даної загрози, на нашу думку, може бути створення нелінійних алгоритмів шифрування, що додатково ускладнять криптографічний аналіз даних шифрів.

Практично будь-яка криптосистема будується на основі простоти одностороннього перетворення: маючи ключ можна легко та швидко перетворити (зашифрувати) повідомлення та, у випадку симетричних криптосистем, на основі того ж самого ключа можна провести зворотне перетворення (дешифрування) шифрованого повідомлення. При цьому перед криптоаналітиком, у якого немає даного ключа, а є лише шифроване повідомлення, стоїть значно важче завдання, оскільки існує непрактична

кількість способів якими це повідомлення можна розшифрувати, при цьому існує лише один, в результаті якого можливо отримати початкове повідомлення [5-8].

В сучасній криптографії, в цілому, існує два напрямки симетричних криптосистем: потокові та блочні. Сучасні блокові криптосистеми з симетричними ключами шифровують n-бітові блоки вхідних даних, або розшифровують n-бітовий блок зашифрованих даних. Алгоритми шифрування або дешифрування використовують k-бітовий ключ. В той час потокові алгоритми шифрування, як правило, орієнтовані на роботу в реальному часі. Вони працюють із малими частинками інформації: з бітами, або байтами, що дозволяє виконувати шифрування, або дешифрування потокових даних [2, 4, 9].

Відповідно до принципу Керкхоффса, якого дотримуються більшість криптологів, стійкість криптосистеми не повинна залежати від принципу шифрування, а повинна базуватися виключно на її ключах [2, 10]. При тому що даний принцип не є деякою істиною в останній інстанції, оскільки існує багато прикладів того як відсутність інформації про принцип шифрування серйозно ускладнювала криптоаналіз, вказаний підхід все ж є одним з головних безпекових вимог до сучасних алгоритмів шифрування, адже криптосистема, що пройшла відповідну перевірку буде гарантовано стійкою в незалежності від контексту її використання [10].

Поєднання криптографічного алгоритму, стійкість якого була доведена за принципом Керкхоффса, з стеганографічними інструментами, різноманітними прийомами заплутування та обфускації, що додатково ускладнять роботу криптоаналітика, в комплексну систему захисту дозволять гарантувати надійний захист важливих даних.

Відповідно до ідеї Джона Неша, основою криптографічної стійкості шифру є складність обчислень, відповідно до якої і проводиться оцінка криптосистеми. Також Неш висунув гіпотезу, відповідно до якої складність обчислень перебуває у експоненційній залежності, від довжини ключа [2, 11].

Таким чином, збільшення параметрів (розміру) ключа та варіативності шифрування, що збільшує складність обчислень, підвищує стійкість та надійність алгоритму [2, 6, 7].

Отже, сучасні симетричні криптосистеми – це деякі алгоритми перетворення інформації, що відбувається на основі деякого криптографічного ключа, який сам по собі представляє базову складність алгоритму.

Постановка задачі. На основі проаналізованої інформації, стають зрозумілими основні вимоги до сучасної криптосистеми, що може претендувати на надійність та криптостійкість в умовах стрімкого розвитку технологій та постійному нарощуванні обчислювальної потужності комп'ютерів.

Головна криптостійкість алгоритму повинна базуватися на складності обчислень, необхідних для криптоатаки: дана складність повинна бути суттєво вищою як за поточний рівень обчислювальної потужності комп'ютерної техніки, так і за цей показник в майбутньому, з урахуванням згаданого зростання. Складність обчислень повинна підкріплюватись великим розміром ключа, а середній прогнозований час, затрачений на реалізацію атаки повинен бути абсолютно не практичним.

Блочна структура шифрування дозволить забезпечити додаткову криптостійкість, оскільки вона створює залежність одного блоку від іншого, що вимусить криптоаналітика підходити до аналізу шифротексту комплексно, досліджуючи все повідомлення в цілому.

Завданням роботи є створення алгоритму шифрування інформації, який відповідав би сучасним стандартам безпеки, забезпечував високу швидкодію роботи та був легким в модернізації, з можливістю широкого практичного застосування, зокрема стеганографічного.

Проектування алгоритму з заявленими характеристиками дозволить забезпечити високий рівень захисту важливих даних та відповідає актуальній тенденції забезпечення цифрової безпеки та захисту конфіденційних даних.

2 МОДИФІКАЦІЯ ІСНУЮЧИХ АЛГОРИТМІВ ШИФРУВАННЯ З ВИКОРИСТАННЯМ ПРИНЦИПУ НЕЛІНІЙНОСТІ

Перед початком проектування криптосистеми спробуємо інтегрувати запропонований принцип нелінійного шифрування в один із вже існуючих алгоритмів шифрування. На основі цього ми зможемо проаналізувати те як зміняться властивості алгоритму.

У якості базової криптосистеми ми оберемо симетричний алгоритм шифрування DES. Це загальновідомий та добре досліджений шифр, з відносно простою схемою шифрування на основі мережі Фейстеля [2, 12].

В класичній формі DES складається з таких етапів:

- початкова перестановка;
- шістнадцять раундів мережі Фейстеля, що складаються з:
  - розширення 32-бітного підблоку до 48-бітного;
  - побітове додавання раундового 48-бітного ключа до розширеного підблоку за модулем 2;
  - додавання за модулем два блоку та раундового ключа;
  - підстановка за двовимірними S-блоками, в результаті якої блок зменшується до 32-х біт;
  - бітова перестановка;
  - побітове додавання отриманого блоку з другим 32-бітовим підблоком за модулем 24
- кінцева перестановка.

Алгоритм працює з 64-бітними блоками, що в процесі шифрування діляться на два 32-бітних підблоки та 56-бітним ключем.

Для зручності, раундові операції від розширення підблоку до бітової перестановки ми розглядатимемо як загальну криптографічну функцію, як це прийнято для мережі Фейстеля та позначатимемо f.

Дана функція чітко прописана в алгоритмі, іншими словами в процесі шифрування коли на початку раунду в мережу Фейстеля подається чергових два підблоки з раундовим ключем існує єдиний варіант вихідних підблоків.

Ми можемо модифікувати цю частину алгоритму з використанням запропонованого підходу нелінійного шифрування. Для цього опишемо характеристики даної функції: на вхід вона приймає 32-бітний блок даних та 48-бітний раундовий ключ, на вихід подається 32-бітний блок даних, сформований на основі вхідних даних.

Криптографічна функція алгоритму DES відповідає переліченим характеристикам, проте класичний її вигляд, як можна помітити, далеко не єдиний який, який забезпечує виконання даних характеристик. До прикладу зміна принципу початкового розширення блоку ніяк не вплине на загальні характеристики алгоритму, проте якщо ми зашифруємо деяке повідомлення алгоритмом з класичною функцією, а потім зашифруємо це ж саме повідомлення алгоритмом з функцією, яка по-іншому виконує початкове розширення блоків то, при ідентичності всіх інших параметрів, вихідний шифротекст буде різним (Рис. 2.1) [7].

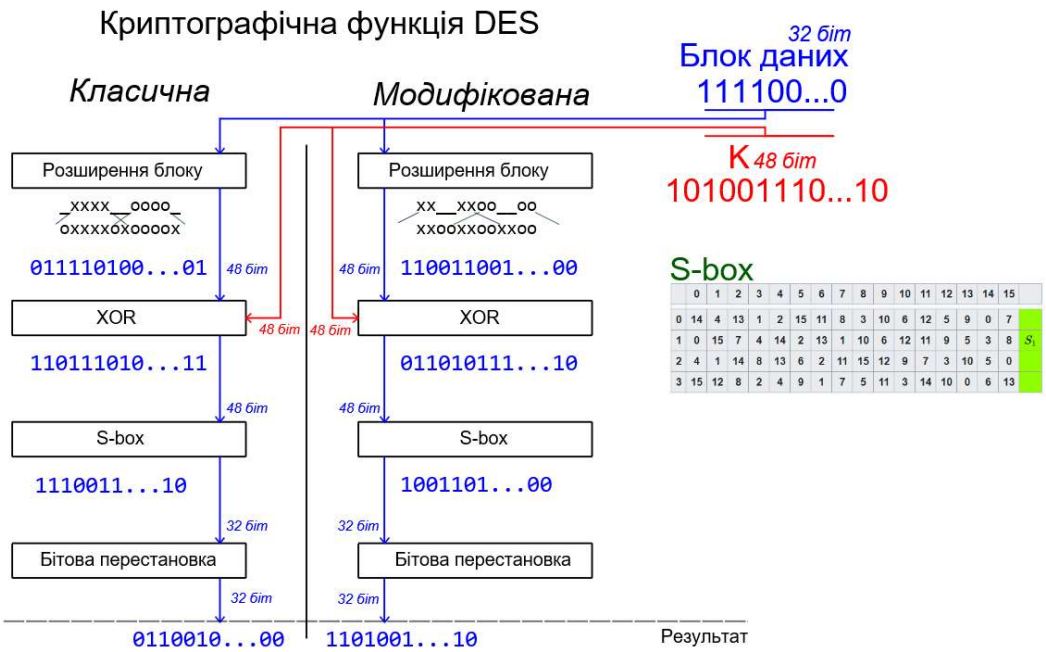


Рисунок 2.1 – Класична та варіант альтернативної криптографічної функції алгоритму DES

Також слід зазначити, що маючи на руках лише шифротексти без ключів, по результату шифрування неможливо сказати яка функція шифрування була використана.

Звісно, наведений варіант альтернативної функції не єдиний можливий: існує безліч способів незначним чином модифікувати класичну функцію, щоб вихідні дані при цьому щоразу різнилися, без зміни базових основ алгоритму. До прикладу ми можемо змінити порядок бітової перестановки, додати циклічний зсув на n біт до, або після розширення блоку, проводити гамування блоку, переставляти місцями біти на парних позиціях після заміни за S-блоком, проводити матричну перестановку тощо.

Таким чином, припустимо, ми створили вісім модифікованих функцій, переконавшись що кожна з них дає різний вихідний блок, при однакових вхідних, а наші модифікації не створили яких-небудь вразливостей. Використання алгоритму з будь-якою із цих функцій за класичною схемою даватиме однозначний вихідний результат, хоча й різний. Проте що якщо використати всі вісім отриманих функцій, а не лише одну з них?

Випадково присвоїмо функціям ідентифікатори від 0 до 7. Тепер при виконанні шифрування, перед раундами, кидатимемо монетку: обиратимемо випадкове число в діапазоні $[0; n)$, де n – кількість функцій, у нашому випадку $n = 8$. Припустимо нам випало число п’ять. Це означатиме, що в наступних шістнадцяти раундах у якості криптографічної функції ми використовуватимемо функцію з ідентифікатором 5. Після завершення усіх операцій для даного блоку, в кінцевий шифротекст допишемо поточний ідентифікатор. Для наступного блоку ми знову кинемо монетку і так до кінця. В результаті ми отримаємо деякий шифротекст, розмір якого буде приблизно на 4,5% більше, ніж шифрований текст при класичному використанні алгоритму (Рис.2.2)[6].

У вказаному випадку, при перехопленні шифрованого повідомлення, криптоаналітик отримає самі шифроблоки, а також ідентифікатори використаних функцій. Ці ідентифікатори не дають жодного уявлення про

функцію, яку вони ідентифікують, проте в даній ситуації аналітик зможе однозначно сказати про те чи однакові функції були використані, в тому, чи іншому випадку, елементарно порівнявши індекси в кінці блоків. На підставі цього він зможе згрупувати блоки, в яких були використані однакові функції та досліджувати їх окремо.

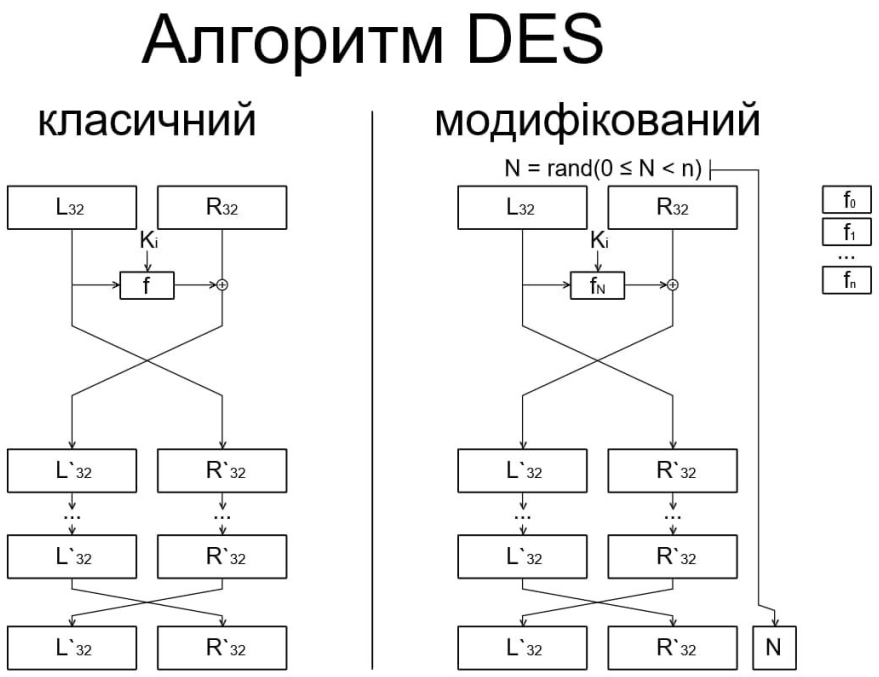


Рисунок 2.2 – Схема роботи звичайного та модифікованого алгоритму DES

Для того щоб позбавити криптолога такої можливості, зробимо ще одну модифікацію: індексів буде більше ніж функцій. У наведеному прикладі, це можна зробити двома способами: або зменшити кількість функцій, тоді в нас буде, до прикладу, 4 функції та 8 індексів, або збільшити кількість індексів, тоді матимемо 8 функцій та 16 індексів [6-8].

При генерації ключа, ми як і раніше випадково генеруємо відповідності між функціями та ідентифікаторами, проте вже не у відношенні $1:1$, а у відношенні $1:m$, де одній функції відповідатиме m індексів. При шифруванні тепер ми обиратимемо відразу два випадкових числа: перше – щоб вибрати функцію, друге – щоб вибрати один індекс, серед всіх, що відповідають цій функції.

Також варто зазначити що кількість індексів оптимально обирати рівною 2^n, а також ці індекси повинні рівномірно розподілятися між функціями. Це дозволить усунути надлишковість та максимально ефективно використовувати діапазон значень ідентифікаторів, при шифруванні, а також забезпечить однакову ймовірність випадання функцій.

Тепер, перехопивши повідомлення, криптоаналітик не зможе однозначно сказати чи були використані однакові функції в блоках з різними індексами, адже це може бути як інший ідентифікатор тієї ж самої функції, так й ідентифікатор іншої функції.

Цікавою властивістю, як в першому прикладі, так і в другому є те що ми отримуємо різні шифровані повідомлення при однаковому ключі та інших вхідних даних. Це досягається завдяки описаній вище випадковій генерації, на основі якої, в даному випадку, обирається те яка функція буде застосована.

Отримана модифікація алгоритму DES не змінила жодних ключових моментів та принципів функціонування криптосистеми, проте вона додала варіативності і заявленої нелінійності в процес шифрування. Дані нововведення не сильно відображатимуться на швидкодії алгоритму, оскільки єдина нова затрата по часу – генерація випадкових чисел. Також можна помітити що у даній модифікації, порівняно з класичним алгоритмом, з’явились нові характерні риси, а саме:

- повторне шифрування з ідентичними вхідними параметрами дасть різні вихідні шифротексти;
- перехопивши різні шифровані повідомлення, важко з’ясувати чи будуть вони однаковими при дешифруванні, чи ні;
- одна і та ж сама криптографічна функція в різних ключах позначатиметься різними ідентифікаторами, відповідно при перехоплені повідомлень, шифрованих різними ключами, аналітику буде важко встановити відповідність між однією і тією ж функцією в різних шифрованих повідомленнях;

- компрометація деякого відкритого повідомлення не гарантує компрометації цього ж самого повідомлення, зашифрованого повторно тим самим ключем;
- складність обчислень при атаці грубої сили суттєво виростає, пропорційно кількості альтернативних функцій, оскільки у найгіршому випадку, навіть за умови що всі ці функції є у відкритому доступі, аналітику доветься перебирати усі можливі шифротексти усіх ймовірних функцій, після чого намагатися комбінувати їх різними можливими способами.

Таким чином, запропонована модифікація, на прикладі алгоритму DES, є прикладом того що введення розгалуженості у вже існуючий алгоритм, ціною незначних затрат дозволить покращити характеристики алгоритму шифрування. В описаній модифікації надійність шифрування не погіршується на жодному з етапів: запропонований алгоритм зберігає усі ключові аспекти оригіналу, лише розширюючи та покращуючи його характеристики, збільшуючи затрати, необхідні для проведення успішної криптоатаки на нього.

Також нелінійність функції у даному алгоритмі зовсім не єдиний аспект, до якого даний принцип можна застосувати: аналогічним чином можна ввести варіативність на інших етапах даного алгоритму. Наприклад, можна різними способами зчіплювати вихідні підблоки, на основі деяких випадкових факторів модифікувати раундові ключі тощо[6, 8].

Ще кращих результатів можна досягнути, ввівши деякий алгоритм кінцевого перетворення, який би «затирав» використані ідентифікатори.

Наведений приклад наглядно демонструє принцип запропонованого нелінійного алгоритму, в якому існувало б кілька паралельних гілок, усі з яких є відомими, а в процесі шифрування випадковим чином обиралася б одна, при чому аналітику невідомо яка саме гілка була обрана.

3 ВДОСКОНАЛЕННЯ МЕТОДУ НЕЛІНІЙНОГО ШИФРУВАННЯ НА ОСНОВІ РОЗГАЛУЖЕНОСТІ КРИПТОГРАФІЧНОГО АЛГОРИТМУ

3.1 Розробка симетричної нелінійної криптосистеми

На основі запропонованого підходу нелінійного шифрування, шляхом реалізації алгоритму з великою кількістю можливих варіантів роботи, серед яких обирається один випадковий, можна модифікувати існуючі криптосистеми, покращуючи їх характеристики, що було продемонстровано на прикладі алгоритму DES.

Метод нелінійного шифрування даних визначатиметься такими ключовими характеристиками:

- деякий алгоритм шифрування, спроектований, або модифікований за даним методом повинен включати в себе деякий модуль, на базі якого може бути створено алгоритмічне розгалуження;
- розгалуження гілок алгоритму повинно починатися з однієї точки та зводитись в одну точку, інакше кажучи, всі гілки повинні бути взаємозамінні;
- кінцевий результат роботи алгоритму (шифротекст) повинен мати деяке маркування, на основі якого, маючи ключ, можливо однозначно відновити яка гілка алгоритму була виконана.

На основі перелічених характеристик, ми спроектуємо новий алгоритм шифрування та на його основі наглядно дослідимо переваги та недоліки вказаного методу.

Основою даного алгоритму буде доволі проста за своєю сутністю блокова (поліалфавітна) підстановка. Не дивлячись на простоту, даний метод шифрування перевірений часом і за умови його грамотного використання, як наприклад у шифрі Віженера, він гарантує досить високу криптостійкість алгоритму [6, 13, 14].

Базовий принцип роботи алгоритму полягатиме у підстановці блоків довжиною m біт через випадкову підстановочну систему. Для можливості дешифрування кожна підстановочна система матиме деяку кількість ідентифікаторів, які зчіплюватимуться з вихідними блоками.

Ключ алгоритму складатиметься з n підстановочних систем (S-блоків), де n обирається довільно, під час генерації ключа. Кількість блоків підстановки обирається користувачем, виходячи з прийнятного балансу між безпекою та розміром ключа: чим більша кількість блоків – тим більша ентропія шифрування та як наслідок – складність обчислень, проте в той же час чим більша кількість блоків – тим більший розмір ключа, який і без того в даному алгоритмі набагато більший, ніж в інших.

Розмір блоку даних також вибирається на етапі генерації ключа. Умовна особа, що генерує ключ (далі – Аліса) може самостійно обрати як розмір блоку даних, так і розмір ідентифікаторів, або ж дані значення можуть обратись випадково. Закономірність при виборі розміру блоку аналогічна до вибору довжини ключа: чим більший розмір блоку, тим більша складність обчислень для успішного криптоаналізу, проте при цьому суттєво збільшується розмір ключа.

Складність обчислень даного алгоритму та, відповідно, криптостійкість базується на ідеї, протилежній шифру Віженера (Вернама) та подібних йому, оскільки в них по-замовчуванню вважається відомим перелік підстановочних алфавітів, а в таємниці, фактично, тримається те які з цих алфавітів та в якій послідовності були використані. В запропонованому алгоритмі логіка будується від зворотного: те які підстановочні блоки (алфавіти) були використані генерується випадково для кожного блоку та передається практично відкрито разом із шифрованим повідомленням, хоча й алгоритм кінцевого перетворення і нелінійність зчеплення блоку та ідентифікатора мають за мету максимально ускладнити виведення криптоаналітиком цих ідентифікаторів, проте це більше додаткова міра захисту.

Головна складність криптосистеми полягає у відсутності у криптолога систем криптографічних підстановок, які випадковим чином генеруються щоразу при створенні ключа. Відновлення цих систем взаємовідносин між вхідними та вихідними блоками, які створюються випадкоао, без будь-яких об’єктивних залежностей, є набагато важчою задачею, ніж відновлення порядку використання відомих залежностей. Так наприклад для 8-бітних блоків даних одна система підстановок складатиметься з 256 восьмибітних елементів. Кількість можливих варіантів розміщення 256 елементів випадковим чином, відповідно до законів комбінаторики, складає 256!, серед яких генерується один випадковий. Навіть з урахуванням того що використовуються не всі елементи підстановочної системи, грубий перебір навіть десятої частини можливих варіантів є абсолютно непрактичною задачею, навіть з урахуванням розвитку технологій. При цьому в даному алгоритмі використовується не одна така система [6-8].

Випадково перемішати велику кількість елементів відносно легко, але відтворити дану випадкову послідовність знову, не знаючи її, вкрай важко. Саме на цій умовно односторонній функції будується базова складність запропонованої криптосистеми.

Недоліком такого підходу є розмір. По-перше, мова йде про розмір повідомлення, адже воно в будь-якому випадку буде збільшуватись в процесі шифрування за рахунок ідентифікаторів..

По-друге, страждає також і розмір ключа. В той час як ключ алгоритму DES складав 56 біт, алгоритм AES має ключ 128, або 256 біт, то криптографічний ключ даного алгоритму займатиме щонайменше близько кілобайту, з урахуванням максимально компактного запису та стискання.

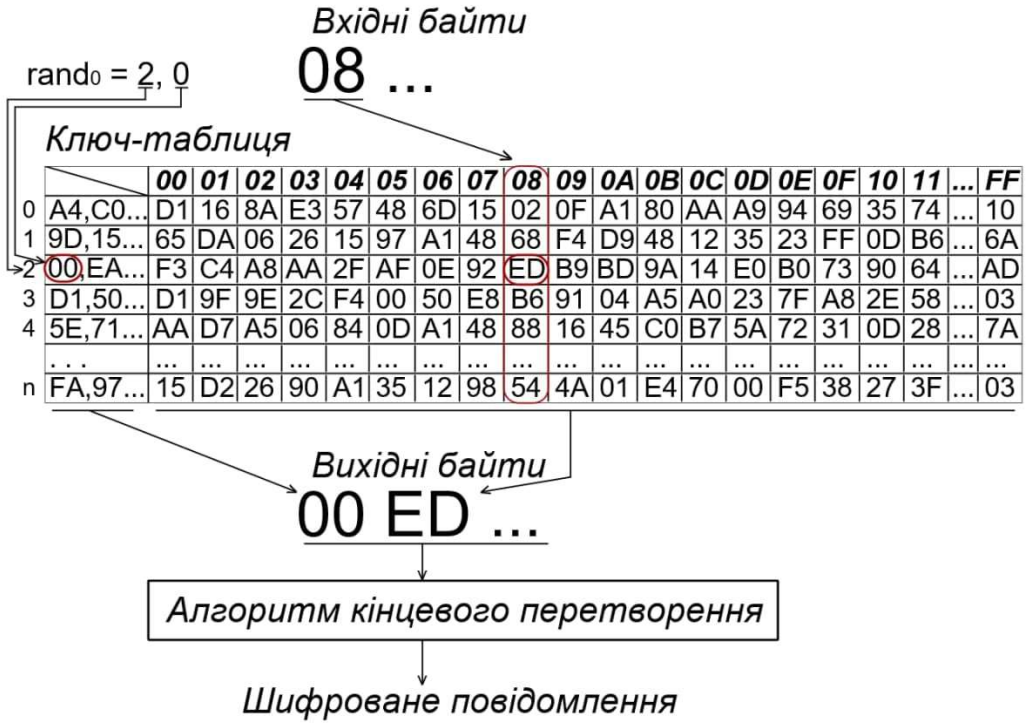


Рисунок 3.1 – Схема роботи алгоритму шифрування

Це не є однозначним недоліком, оскільки розмір навіть у декілька кілобайт давно перестав бути чимось суттєвим в часи, коли одне фото на смартфоні займає до десятка мегабайт. Разом з цим, відповідно до згаданої раніше гіпотези Джона Неша, складність обчислень експоненційно залежить від розміру ключа, великий за розміром ключ повинен забезпечувати високий рівень захисту.

І все ж, в умовах критичної економії пам’яті існують варіанти суттєвого зменшення розміру ключа, звичайно зі зниженням рівня складності обчислень аналізу. До них ми перейдемо в наступному розділі.

Загальна схема роботи алгоритму шифрування (Рис. 3.1) складається з таких частин:

- генерація ключа шифрування, з ключовими параметрами;
- представлення вхідного повідомлення у двійковому вигляді;
- генерація випадкових значень (номеру алфавіту та ідентифікатора) для кожного n-бітного вхідного блоку;
- підстановка вхідних блоків за випадково обраним алфавітом;

- зчеплення підставленого блоку та ідентифікатора використаної підстановочної системи (алфавіту);
- алгоритм кінцевого перетворення.

Дешифрування відбувається у зворотній послідовності, що характерно для абсолютної більшості симетричних шифрів.

Зчеплення вихідного блоку даних з ідентифікатором перестановочної системи також матиме варіанти виконання (Рис. 3.2). Вибір варіанту зчеплення може проводитись для кожного блоку, проте тоді виникне необхідність у додаткових ідентифікаторах, для всього повідомлення, або для ключа.

Ідентифікатор	*Блок заміни*
11111111	00000000

Варіанти зчеплення ідентифікатора та блоку:

- просте зчеплення	1111111100000000
- змішування	1010101010101010
- кільцеве	1111000000001111
- перехрестне	1111000011110000
- комбіноване	1010101011110000

Рисунок 3.2 – Варіанти зчеплення блоку даних та ідентифікатора

Варіативність зчеплення блоку та ідентифікатора створить додаткову перешкоду для криптоаналітика, оскільки якщо він не зможе достовірно отримати пару «ідентифікатор – блок», можливість будь-яких досліджень та аналізу майже повністю нівелюється.

Алгоритм кінцевого перетворення (АКП) деяким чином змінює проміжний шифротекст, маючи завдання остаточно ускладнити, а в ідеальному випадку унеможливити, розбір шифротексту на пари «ідентифікатор – блок», змішавши їх зі значеннями блоків даних. В цілому, на роль даного алгоритму може підійти спрощена версія будь-якого існуючого алгоритму шифрування, починаючи від звичайного додавання за модулем два, на основі деякої гами, закінчуючи мережею Фейстеля, або матричними перестановками, на кшталт перестановок алгоритмів SQUARE та AES.

Перед АКП стоять такі вимоги:

- алгоритм повинен «змішувати» ідентифікатори та блоки даних, а не видозмінювати кожен з них окремо;
- ключ алгоритму повинен генеруватися на основі ключ-таблиці загального алгоритму. Рекомендовано щоб у якості ключа АКП виступала хеш-сума, або деяка інша проекція ключ-таблиці, а відповідно усіх систем підстановки, навіть тих, що не використовувались при шифруванні;
- АКП не повинен бути складним для обчислень та не повинен використовувати більше ресурсів, ніж базова криптографічна підстановка.

У якості алгоритму кінцевого перетворення даної криптосистеми ми використаємо блокове додавання за модулем два, з розміром блоку в 64 біти, в режимі зчіплення блоків. Ключем АКП слугуватиме хеш-зліпок підстановочних систем ключа.

3.2 Генерація ключа

Криптографічний ключ даного алгоритму сильно відрізняється по своїй суті від ключів інших систем шифрування, таких як DES, AES, IDEA тощо. У наведених в приклад симетричних криптосистемах ключ являє собою деякий блок біт фіксованого розміру, який за певними закономірностями трансформується у набір раундових ключів, що використовуються в криптографічних ітераціях.

Ідентифікатори *Вхідні блоки* *Ключ-таблиця*

		00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	...	FF
0	A4,C0...	D1	16	8A	E3	57	48	6D	15	02	0F	A1	80	AA	A9	94	69	35	74	...	10
1	9D,15...	65	DA	06	26	15	97	A1	48	68	F4	D9	48	12	35	23	FF	0D	B6	...	6A
2	00,EA...	F3	C4	A8	AA	2F	AF	0E	92	ED	B9	BD	9A	14	E0	B0	73	90	64	...	AD
3	D1,50...	D1	9F	9E	2C	F4	00	50	E8	B6	91	04	A5	A0	23	7F	A8	2E	58	...	03
4	5E,71...	AA	D7	A5	06	84	0D	A1	48	88	16	45	C0	B7	5A	72	31	0D	28	...	7A
	. . .	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
n	FA,97...	15	D2	26	90	A1	35	12	98	54	4A	01	E4	70	00	F5	38	27	3F	...	03

Системи (алфавіти) підстановки

Рисунок 3.3 – Криптографічний ключ (ключ-таблиця) алгоритму

У запропонованому алгоритмі ключем є структура даних (Рис. 3.3), що складається з набору алфавітів підстановок, з відповідними ідентифікаторами, а також з набору параметрів, що випадково генеруються та вирішують особливості функціонування алгоритму. В наслідок цього розмір ключа даного алгоритму незрівнянно більший, ніж у перелічених вище.

Дана особливість навряд стане серйозним недоліком при практичному використанні, оскільки з побутової точки зору розмір файлу в кілька кілобайт давно перестав вважатися великим.

Отже якщо такий розмір ключа нас влаштовує то нам підійде звичайний режим генерації ключа, що є основним в даному алгоритмі.

Звичайна генерація ключа складається з таких етапів:

- вибір кількості систем підстановки (алфавітів);
- вибір розмірності блоку даних та ідентифікаторів;
- обрахування кількості елементів в одному алфавіті, за формулою $n = 2^p$, де p – це розмір блоку даних в двійковому представленні;
- генерація для кожного алфавіту випадкової послідовності (розміщення) блоків даних в діапазоні $[0; 2^p - 1]$;
- генерація для кожного алфавіту деякої встановленої, або випадкової кількості ідентифікаторів обраної розмірності, з урахуванням того що кожен ідентифікатор повинен зустрічатись лише один раз;

- генерація додаткових параметрів ключа: типу зчеплення блоків з ідентифікаторами, випадкових параметрів АКП, якщо такі є та інших необхідних параметрів.

Тип зчеплення блоків, в залежності від реалізації алгоритму, може записуватись в ключ у вигляді бітової маски, або у вигляді ідентифікатора маски, якщо вони прописані в коді програми.

Криптостійкість генератора випадкових чисел, що використовується для генерації ключа є одною з ключових вимог безпеки даного алгоритму. Значення повинні бути максимально випадкові, без можливості встановлення будь-яких закономірностей між ними. Рекомендується використання апаратних засобів рандомної генерації, а при відсутності можливості їх використання, можна приймати деякі випадкові значення від користувача (рух мишки по екрану, натискання випадкових клавіш, або однієї з різним часовим інтервалом тощо).

У випадку критичних обмежень на пам’ять, або на канал зв’язку, коли передача ключа розміром у кілька кілобайт є неможливою, або вкрай неприйнятною, можлива реалізація векторного генератора ключа.

Векторна генерація ключа – це алгоритм генерації криптографічного ключа, який створює ключ-таблицю, аналогічну за характеристиками до таблиці, створеної звичайною генерацією, на основі деякого бітового вектору.

Одним з варіантів реалізації такого алгоритму є генератори псевдовипадкових чисел, на основі деякого ініціалізуючого вектору. В такому випадку вхідний набір біт можна розділити на декілька частин, з яких одні стануть ініціалізаторами псевдовипадкової генерації для підстановочних алфавітів, а на основі інших можна створити деякий шум, наприклад шум Перліна, та провести гамування.

Таким чином на основі деякого бітового псевдоключа ми генеруємо ключ-таблицю, проте використання даного методу несе в собі ряд загроз, оскільки навіть за умови ідеальної реалізації самого алгоритму та відсутності в ньому вразливостей, складність обчислень для успішного проведення криптоатаки

суттєво зменшиться, оскільки аналітику необхідно буде вже не відновити цілу таблицю, а лише відтворити цей псевдоключ. Простіше кажучи, якщо розмір псевдоключа, наприклад, 256 біт, то складність обчислення при атаці грубою силою зведеться до перебору 2^{256} ключів.

Використання того чи іншого методу генерації ключа залишається на розсуд шифрувальника, адже обидва способи мають переваги та недоліки. В програмній реалізації алгоритму, в рамках даної роботи, ми використаємо звичайну генерацію ключа.

3.3 Криптографічна стійкість алгоритму

Криптографічна стійкість будь-якого алгоритму шифрування є головним критерієм його оцінки та в подальшому – можливості його практичного використання [2, 14-16].

В рамках даної роботи ми проведемо первинне дослідження запропонованого алгоритму шифрування на криптостійкість, а також розглянемо базові техніки криптоаналізу.

Найбільш універсальною атакою на криптосистеми є атака грубої сили, а отже саме на неї слід орієнтуватись в першу чергу.

Запропонований алгоритм надійно захищений від атаки грубою силою, в першу чергу завдяки величезному по криптографічних розмірах ключу та нелінійності, яка додає серйозні складності для аналізу. Оскільки зчіплення блоків відбувається різними способами, в залежності від параметрів ключа, аналітик не знає яким чином вони поєднуються і які біти належать ідентифікатору, а які – блоку даних, навіть не враховуючи алгоритм кінцевого перетворення. В зв’язку з цим навіть сама організація подібної атаки буде достатньо проблематичною [14].

Кількість можливих варіантів підстановочних систем ключа N_a можна обрахувати за формулою $N_a = (2^p)! * n$, де p – розмір блоку даних в двійковому вигляді, а n – кількість алфавітів.

Загальна кількість можливих ключів N обраховується за формулою $N = N_a \times N_{id} \times N_p$, де N_a – кількість варіантів підстановочних алфавітів, N_{id} – кількість можливих ідентифікаторів, а N_p – кількість додаткових параметрів.

Таким чином навіть при п'яти перестановочних системах, з розміром блоку в 7 байт, кількість можливих згенерованих алфавітів сягне понад $1{,}9 \times 10^{216}$, враховуючи ще кількість можливих ідентифікаторів та додаткових параметрів ключа, навіть зі швидкістю перебору у мільярд ключів за секунду, що практично не можливо на сьогоднішній день, на перебір навіть тисячної частини цих ключів піде абсолютно непрактичний час.

Окрім атаки грубої сили даний алгоритм стійкий перед усіма видами класичних криптоатак, таких як частотний аналіз, атака по масці та інших лінійних видів аналізу.

У алгоритмі не помічено вразливостей перед математичним аналізом, адже в ньому майже відсутні алгебраїчні ітерації, а головну криптографічну складність представляє собою заміна блоку байт на випадковий.

Що стосується атак на основі пари «відкритий текст – шифрований текст», то відновити ключ навіть у такому випадку не вдасться, проте це залежить від АКП та його надійності, від того чи знає аналітик який тип зчеплення блоків було використано та від кількості пар відкритих та шифрованих повідомлень, що має на руках криптолог.

Також надійність даної системи залежить від згенерованого ключа: при критично малому розмірі вхідного блоку (менше п'яти біт) та при малій кількості підстановочних систем (менше трьох-чотирьох), складність обчислень сильно знижається. Оптимальними для використання є розмір блоку даних у 7-10 біт, при наявності 6-12 систем підстановки, не менше ніж по три ідентифікатори на кожну.

В цілому, опираючись на проведені дослідження, даний криптографічний алгоритм показав себе як надійний метод шифрування з достатньо високою криптографічною стійкістю. Практичне використання даного алгоритму потребує багатостороннього криптографічного тестування та профільного пошуку вразливостей в майбутньому.

3.4 Створення програмного прототипу криптосистеми

Програмна реалізація описаного алгоритму буде виконана методами об’єктно-орієнтованого програмування, з використанням мови програмування C#, на основі фреймворку .NET та інтерфейсу Windows Forms. Середовищем розробки виступить Microsoft Visual Studio 2019.

Розробку програми розпочнемо з побудови ієрархії класів, необхідних для роботи. Всього у нас буде три класи: «NonlinearCrypt», «NonlinearCryptKey» та «NonlinearCryptBitsBlock».

Клас «NonlinearCryptBitsBlock» описуватиме n-бітний блок даних, з інкапсульованими змінними Size, типу «int» та BitArray, типу «bool[]», для зберігання бітів. Також цей клас матиме конструктор, що прийматиме значення розміру блоку, а також набір методів для задання, повернення, порівняння та заповнення блоку бітів. Екземпляр даного класу зберігатиме блок бітів фіксованого розміру.

Клас «NonlinearCryptKey» описуватиме ключ алгоритму, він міститиме змінні-параметри Alphabets, що представляє собою двовимірний масив об’єктів класу «NonlinearCryptBitsBlock», Identifiers – складений двовимірний масив блоків бітів, AlphabetsCount, DataBlockSize та IdentifiersSize, типу «int», а також конструктор, що задає новий екземпляр ключа на основі масиву алфавітів та масиву ідентифікаторів, методи для порівняння та генерації ключів.

Клас «NonlinearCrypt» містить масив відкритих та шифрованих байт, ключ – об’єкт класу «NonlinearCryptKey», конструктори, та методи Crypt та Decrypt.

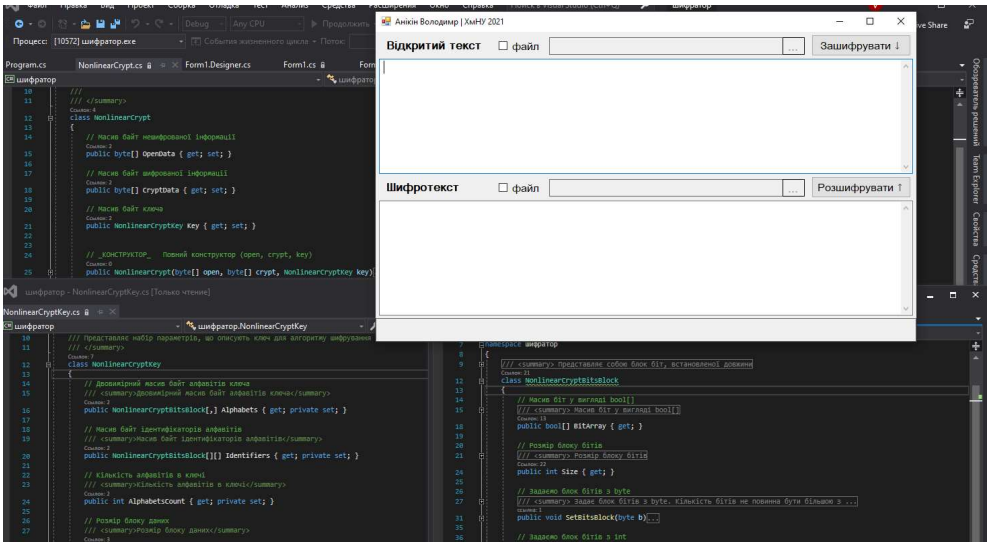


Рисунок 3.4 – Вигляд програмного коду та користувацького інтерфейсу.

Разом з цим, інструментами інтерфейсу Windows Forms налаштуємо клас «Form1» з графічним користувацьким інтерфейсом (Рис.3.4).

Користувацький інтерфейс буде представлений двома областями: для відкритого та шифрованого текстів, кнопками «Зашифрувати» та «Розшифрувати». Дані можна буде ввести у символьному вигляді, або вказати шлях до файлів, з яких будуть зчитані масиви байт. Після натискання відповідних кнопок, користувачу буде запропоновано обрати файл з ключем.

Алгоритми генерації ключів, шифрування та дешифрування будуть реалізовані за допомогою операторів бітового зсуву та операторів бітової логіки.

Це дозволить реалізувати програмне забезпечення, працююче за схемою представленого алгоритму, з великою швидкістю роботи та можливістю легкого запуску на будь якому комп’ютері з операційною мережею Windows.

Для зручності символьної передачі повідомлень, буде також використана реалізація кодування, що заміняє бітові тетради на літери латинського алфавіту, що дозволить комфортно працювати та передавати шифровані повідомлення через месенджери та інші канали символьного зв’язку.

4 МОДИФІКАЦІЯ АЛГОРИТМУ ДЛЯ ПРИХОВАНОГО ЗБЕРІГАННЯ ТА ПЕРЕДАЧІ ІНФОРМАЦІЇ

Як можна було помітити у попередніх розділах, характерною рисою запропонованого алгоритму є те, що шифровані повідомлення одного й того ж відкритого повідомлення, при однаковому криптографічному ключі виходять різні після кожного шифрування. Це пов’язано з генерацією випадкових параметрів в процесі шифрування.

Шифруючи одне і те ж повідомлення багато разів можна помітити що шифротекст практично ніколи не повторюється, при умові що повідомлення достатньо велике. Фактично, оскільки системи підстановки в ключі генеруються випадково, будь-який блок вхідного повідомлення може бути зашифрований будь-яким іншим.

Виходячи з цього, виникає питання: «А чи можна створити такий ключ, при якому деяке конкретне повідомлення, зашифроване одним конкретним методом, серед можливих, на виході дало б шифротекст у вигляді іншого осмисленого повідомлення?».

Дослідивши особливості алгоритму, можна однозначно відповісти, що це дійсно можливо.

Рисунок 4.1 – Умовна модель стеганографічного режиму роботи алгоритму (на рівні символів)

Найпростіше дана задача вирішується при такій послідовності дій:

- представляємо вхідне повідомлення у двійковому вигляді – це те що нам необхідно зашифрувати;
- представляємо друге «публічне» повідомлення у двійковому вигляді – це наш «шифротекст» який ми маємо отримати;
- «розшифровуємо» наш «шифротекст» та отримуємо деякий набір байт, який ми повинні отримати внаслідок підстановки;
- заповнюємо ключ-таблицю, слідуючи блок за блоком по вхідному повідомленню;
- заповнюємо невикористані поля ключ-таблиці підходящими значеннями таким чином, щоб отримати повноцінний криптографічний ключ.

На практиці на деяких з перелічених етапів можуть виникнути труднощі.

По-перше, розмір «публічного» повідомлення повинен бути більшим за вхідне в декілька разів.

По-друге, якщо слідувати за логікою звичайного шифрування, то ключ для АКП формується на основі всієї ключ-таблиці, а відповідно до наведеної вище логіки необхідно спочатку розшифрувати байти вихідного повідомлення. Таким чином ми отримуємо замкнене коло, де ми не можемо створити ключ, не маючи готової ключ-таблиці, і не можемо сформувати ключ-таблицю, не маючи розшифрованих АКП байт. Вирішується даний парадокс або відмовою від АКП для стеганографічних повідомлень взагалі, проте це створює загрозу компрометації ключа, за умови що аналітик дізнається про приховане повідомлення, або зміною принципу генерації ключа для АКП для стеганографічних повідомлень. Наприклад даний ключ можна генерувати на основі лише деяких підстановочних систем, а не відразу усієї ключ-таблиці.

По-третє, може виникнути ситуація, коли деякому вхідному блоку неможливо скласти відповідність до необхідного вихідного, оскільки така комбінація вже зайнята. В такому випадку слід або змінити розмірність блоків

чи ідентифікаторів, або пропустити даний блок, залишивши його зі «сміттям», та позначивши його деяким чином.

Також є сенс в ключах для стеганографічного використання робити велику довжину ідентифікаторів, це дозволить мінімізувати ситуації, коли виникатиме колізія значень.

Даний спосіб використання запропонованого алгоритму має великий потенціал практичного застосування. Згенерований таким методом криптографічний ключ по своїх властивостях та можливостях цілком відповідатиме характеристикам ключа, згенерованого звичайним способом, а значить він буде придатним і до криптографічного використання.

Так, наприклад, можна передавати деяке шифроване повідомлення, з нешифрованими інструкціями. Увагу спеціалістів, що перехоплять його, в першу чергу приверне шифротекст, а інструкції до нього вони приймуть за технічну документацію, в якій нічого цікавого немає. Проте на ділі сам шифротекст буде лише обманним маневром, а справжню цінність матимуть саме дані інструкції, адже якщо їй розшифрувати деяким ключем, то ми і отримаємо справжнє конфіденційне повідомлення.

Існує безліч інших практичних застосувань наведеній технології, зокрема можна зашифрувати повідомлення таким чином, щоб шифротекст повторював шифротекст іншого повідомлення, зашифрованого іншим алгоритмом, як наслідок дешифрування одного шифротекста різними алгоритмами даватиме різні повідомлення.

Отже, стеганографічний метод використання даного нелінійного алгоритму можливий та має багато сфер практичного застосування. Алгоритми з подібними можливостями мало розповсюджені на сьогоднішній день та мають значний потенціал розвитку в майбутньому.

ВИСНОВКИ

У даній роботі висвітлена актуальна тематика криптографічного захисту інформації та запропоновано вдосконалений метод нелінійного шифрування представлений, у вигляді алгоритму, а також у вигляді модифікацій до існуючих криптосистем. Запропонований алгоритм також має стеганографічний режим роботи, що значно розширює область його практичного застосування.

У роботі було досягнуто таких результатів:

1. Розроблено та запропоновано нелінійну схему шифрування, на основі вдосконаленого методу, використання якої можливе як у самостійному вигляді, так для модифікацій існуючих криптосистем.
2. Продемонстрована можливість модифікації існуючих алгоритмів шифрування з використанням методу нелінійного шифрування. На прикладі алгоритму DES показані нові властивості, що дає нелінійна модифікація алгоритму: варіативність шифротексту, де кожен блок може бути зашифрований 8-ми способами та як наслідок – багатократне збільшення складності обчислень при криптоаналізі.
3. Розроблено та представлено алгоритм шифрування, орієнтований на нелінійне шифрування, шляхом виконання потокових випадкових підстановок, з широким спектром практичного використання.
4. Проведено криптологічне дослідження розробленого алгоритму.
5. Запропонована варіація алгоритму, орієнтована на стеганографічне використання та приховування факту шифрування.
6. Методами об’єктно-орієнтованого програмування створено програмне забезпечення, що працює за описаними алгоритмами.

ПЕРЕЛІК ПОСИЛАНЬ

1. Інформаційна безпека : навчальний посібник / Ю.Я. Бобало, І.В. Горбатий, М.Д. Кіселичник, А.П. Бондарєв, С.С. Войтусік, А.Я. Горпенюк, О.А. Нємкова, І.М. Журавель, Б.М. Березюк, Є. І. Яковенко, В.І. Отенко, І.Я. Тишик ; за заг. ред. д-ра техн. наук, проф. Ю.Я. Бобала та д-ра техн. наук, доц. І.В. Горбатого. – Львів : Видавництво Львівської політехніки, 2019. – 580 с.
2. Прикладна криптологія: системи шифрування : підручник / О.Г. Корченко, В.П. Сіденко, Ю.О. Дрейс. – К. : ДУТ, 2014. – 448 с.
3. Інформаційний та кіберпростори: проблеми безпеки, методи та засоби боротьби. / В.Л. Бурячок, С.В. Толюпа, В.В. Семко, Л.В. Бурячок, П.М. Складанний, Н.В. Лукова-Чуйко – К.: ДУТ - КНУ, 2016. – 178 с.
4. Елементарна теорія чисел та елементи криптографії: підручник / О. І. Клесов. - Київ : ТВіМС, 2016. – 412 с.
5. Основы криптографии/ Г.В. Басалова. – М.: ИНТУИТ, 2016. - 283 с.
6. Анікін В.А. Симетрична криптосистема з нелінійним шифруванням та можливістю контролю шифротексту з метою маскування / В. А. Анікін, В. М. Джулій, І.В. Муляр, В.С. Орленко, В.Ю. Тітова // Вісник Хмельницького національного університету. Технічні науки. – 2020. – № 6. – С. 12-19.
7. Анікін В.А. Симетричний алгоритм нелінійного шифрування з можливістю стеганографічного застосування / В.А. Анікін, І.В. Муляр // «Інтелектуальний потенціал – 2020» – збірник наукових праць молодих науковців і студентів / Колектив авторів – Хмельницький: ПВНЗ УЕП, 2020. – Ч.2. – С. 93-97.
8. Анікін В. А. Симетрична поліалфавітна криптосистема на основі випадкової генерації коефіцієнту перестановки / В. А. Анікін , В. О. Бойчук // Тези доповідей XVI Міжнародної науково - практичної конференції "

Військова освіта і наука : сьогодення та майбутнє ", 27 листоп. 2020 р. – Київ : ВІКНУ , 2020. – Т. 1. – С. 20

9. Кодирование и защита информации / П.Б. Акмаров .— Ижевск : ФГБОУ ВО Ижевская ГСХА, 2016 .— 136 с.
10. Криптологія: навч.-метод. посіб. / Л.Я. Глинчук – Луцьк: Вежа-Друк, 2014. – 164 с.
11. Криптоанализ шифрсистемы ACBF/ И. В. Боровкова, И. А. Панкратова. – Прикладная дискретная математика. Приложение, Томск. – 2019. – С. 90-93.
12. Криптологія у прикладах, тестах і задачах: навч. посібник / Т.В. Бабенко, Г.М. Гулак, С.О. Сушко, Л.Я. Фомичова. - Д.: Національний гірничий університет, 2013. - 318 с.
13. Методичні вказівки і завдання до лабораторних робіт з курсу «Комп'ютерна криптографія» [Електронний ресурс]/ укладач: О.М. Гапак. – Ужгород: УжНУ, 2015. – режим доступу: https://www.uzhnu.edu.ua/uk/infocentre/get/10972
14. Криптографические методы защиты информации: учебное пособие [Електронний ресурс]./ Э.А. Болелов. – М.: МГТУ ГА, 2013. – режим доступу: http://89.218.140.251/ek/Болелов%20Э.А.%20Криптографические%20методы%20защиты%20информации.%20Часть%202.%20Ассиметричные%20криптосистемы.pdf
15. Технології захисту інформації / Ю. А. Тарнавський – Київ : КПІ ім. Ігоря Сікорського, 2018. – 162 с.
16. Муляр І.В. Ітераційно-геометричний метод для стійкого перцептуального хешування зображення / В. М. Джулій, Ю. П. Кльоц, І. В. Муляр, В. М. Чешун // Вісник Хмельницького національного університету. Технічні науки. – 2020. – № 1. – С. 76–79

ДОДАТОК А

КОПІЇ АКТІВ ВПРОВАДЖЕННЯ

АКТ

Про впровадження результатів наукової роботи Анікіна В.А. на тему «Симетричний алгоритм нелінійного шифрування з можливістю стеганографічного застосування»

Комісія Хмельницького національного університету в складі:

зав. кафедри кібербезпеки та комп'ютерних систем і мереж к.т.н., доц. Кльоца Ю.П.;

к.т.н., доцента кафедри кібербезпеки та комп'ютерних систем і мереж Муляра І.В.;

к.т.н., доцента кафедри кібербезпеки та комп'ютерних систем і мереж Чорненького В.І.

склали цього акта в тому, що результати наукової роботи Анікіна В.А. (криптографічний алгоритм, на основі принципу нелінійного шифрування та програмний прототип криптосистеми) з 8.02.2021 впроваджені в навчальний процес кафедри кібербезпеки та комп'ютерних систем і мереж в курсах «Алгоритми та структури даних в системах захисту», «Стеганографія та комп'ютерна графіка»

Підпис дійсний
Засвідчую
Начальник відділу кадрів
І.С.Мартинюк
ВІДДІЛ КАДРІВ
Ідентифікаційний код 02071234

Ю.П.Кльоц.
(підпис) (прізвище та ініціали)

І.В. Муляр
(підпис) (прізвище та ініціали)

В.І. Чорненький
(підпис) (прізвище та ініціали)

ДЕРЖАВНИЙ КОНЦЕРН«УКРОБОРОНПРОМ»
ДЕРЖАВНЕ ПІДПРИЄМСТВО «НОВАТОР»
вул.Тернопільська,17, м.Хмельницький, 29018, Україна, тел. +380 (382) 67-15-74, факс: +380 (382) 78-80-13
E-mail: centre@novator-tm.com, www.novator-tm.com, Код ЄДРПОУ 22987900

_____ №_____ На №_____ від _____

ЗАТВЕРДЖУЮ
Державне підприємство «Новатор»,
м. Хмельницький

АКТ

Про впровадження результатів наукової роботи Анікіна В.А. на тему «Симетричний алгоритм нелінійного шифрування з можливістю стеганографічного застосування»

Представниками державного підприємства «Новатор»

- заступник директора ДП «Новатор» Володченко В.О.
- інженер Кадинський Д.С.;

складено даний акт в зв'язку, запровадженням результатів наукової роботи Анікіна В.А., а саме:

- запропонованих методів криптографічного та стеганографічного захисту інформації;
- програмних реалізацій алгоритмів захисту інформації.

Результати наукової роботи впроваджені в можливе робоче використання з метою захисту важливої робочої інформації.

ДЕРЖАВНЕ ПІДПРИЄМСТВО «НОВАТОР» Україна м. Хмельницький Ідентифікаційний код 22987900

_____ Володченко В.О.
(підпис) (прізвище та ініціали)

_____ Кадинський Д.С.
(підпис) (прізвище та ініціали)

ДОДАТОК Б

КОПІЇ НАУКОВИХ ПРАЦЬ ПО ТЕМІ ДОСЛІДЖЕННЯ

к.т.н. Бойчук В.О. (ХмНУ)
Анікін В.А. (ХмНУ)

Симетрична поліалфавітна криптосистема на основі випадкової генерації коефіцієнту перестановки

Криптографія - це прикладна наука, яка розробляє і впроваджує системи захисту інформації шляхом перетворення вихідних осмислених повідомлень в зашифровані повідомлення, які неможливо розшифрувати без ключа дешифрування, хоча вони уразливі для криптоаналітичних атак.

В будь-які часи люди намагалися захистити різноманітну важливу інформацію, ховаючи, маскуючи, або видозмінюючи її. Класична криптографія ґрунтується на припущенні, що ніхто не може вирішити певну складну задачку за реалістичний проміжок часу, або покладатися на аргументи теорії інформації.

Пропонується метод шифрування оснований на принципі поліалфавітної байтової перестановки, де коефіцієнт перестановки для кожного байта обирається випадково. Дешифрування можливе завдяки присвоєнню унікальних ідентифікаторів для кожного байту. Кожен алфавіт може мати безліч ідентифікаторів, проте будь-який ідентифікатор може належати лише одному алфавіту заміни.

Криптографічним ключем у даній криптосистемі є таблиця з довільною кількістю випадково перемішаних відкритих алфавітів (де відкритий алфавіт – це послідовність байт від 0 до 256), кожному з яких належить набір ідентифікаторів.

Алгоритм шифрування складається з наступних етапів: представлення відкритої інформації у байтовому вигляді, генерація випадкового коефіцієнту перестановки (номеру алфавіту та ідентифікатора) для кожного байту, перестановка відкритих байт з байтами випадково обраних алфавітів, формування вихідного шифротексту у форматі «ідентифікатор-байт», кінцеве «забілення».

Алгоритм дешифрування є протилежним до алгоритму шифрування.

Таким чином досягається висока ентропія зашифрованої інформації, а випадковість та відсутність закономірностей у перестановках унеможливлює та серйозно ускладнює різні види криптоаналізу. Надійність даного алгоритма напряму залежить від стійкості генератора випадкових чисел.

Список використаних джерел:

1. Jara Vera V, Sánchez Ávila C (2019) Graphemic-phonetic diachronic linguistic invariance of the frequency and of the Index of Coincidence as cryptanalytic tools. PLoS ONE 14(3): e0213710 10.1371/journal.pone.0213710.

Симетричний алгоритм нелінійного шифрування з можливістю стеганографічного застосування

Анікін В.А.
Науковий керівник – к.т.н.,доц. Муляр І.В.
Хмельницький національний університет

На сьогоднішній день захист інформації є одним з найбільш актуальних напрямків кібербезпеки, враховуючи стрімкий ріст інформаційних технологій та збільшення цінності інформації як ресурсу. Вже давно як рядові користувачі, так і великі корпорації діджиталізують особисті дані, серед яких велика кількість персональної, конфіденційної інформації, даних що становлять корпоративну, лікарську, чи навіть державну таємницю.

Більше того, дана інформація пересилається через публічні мережі, зберігається на онлайн ресурсах, що окрім комфорту та зручності створює серйозні безпекові загрози.

Для усунення, чи мінімізації цих загроз, для збереження цілісності, доступності та конфіденційності – трьох базових аспектів захисту інформації, впродовж багатьох століть люди використовують шифрування, яке на сьогоднішній день вивчає наука під назвою криптографія.

Криптографія – це прикладна наука, яка розробляє і впроваджує системи захисту інформації шляхом перетворення вихідних осмислених повідомлень в зашифровані повідомлення, які неможливо, або вкрай важко розшифрувати без криптографічного ключа, хоча вони й вразливі для криптоаналітичних атак [1].

В будь-які часи люди намагалися захистити різноманітну важливу інформацію, ховаючи, маскуючи, або видозмінюючи її. Класична криптографія ґрунтується на припущенні, що ніхто не може вирішити певну складну задачку за реалістичний проміжок часу, або покладатися на аргументи теорії інформації [2].

Сучасна криптографія є математично-комбінаторною наукою, що вивчає способи перетворення інформації з метою її захисту від несанкціонованого використання.

Викликом перед сучасною криптографією є значний прорив у області комп'ютерних електронно-обчислювальних машин, в силу чого незламні до цього криптосистеми стали вразливими. Через це в сучасній криптографії, починаючи з другої половини минулого століття стійкість криптосистеми оцінюється на основі складності обчислень, необхідних для успішного проведення криптоатаки, з використанням усіх можливих ресурсів. Даний підхід запропонував криптолог Джон Неш і він залишається актуальним до сьогодні [3].

Стеганографія також є невід'ємною складовою сучасного захисту інформації, оскільки криптографічне повідомлення, яке представляє собою неосмислений набір байт, неодмінно приверне до себе увагу, в той час як стеганографічне повідомлення у вигляді зображення, чи будь-яких інших даних, що самі по собі жодної цінності не представляють, з великою ймовірністю залишиться не поміченим.

Комплексні алгоритми захисту, що включають в себе як криптографічний так і стеганографічний аспект здатні забезпечити надійний захист важливої інформації.

На підставі цього пропонується використання нелінійності у криптографічних алгоритмах яка з одного боку забезпечить додаткову складність обчислень, необхідних для криптоаналізу шифрованих повідомлень, а з іншого – дасть можливість отримати множину можливих

шифротекстів, при одному і тому ж вхідному повідомленні та ключі, що дасть підґрунтя для стеганографічного використання даного алгоритму.

Пропонується метод шифрування оснований на принципі поліалфавітної блочної підстановки, де система (алфавіт) підстановки для кожного блоку обирається на основі випадкових значень, що генеруються в процесі шифрування. Дешифрування можливе завдяки присвоєнню унікальних ідентифікаторів для кожної підстановочної системи. Кожен алфавіт може мати безліч ідентифікаторів, проте будь-який ідентифікатор може належати лише одному алфавіту заміни.

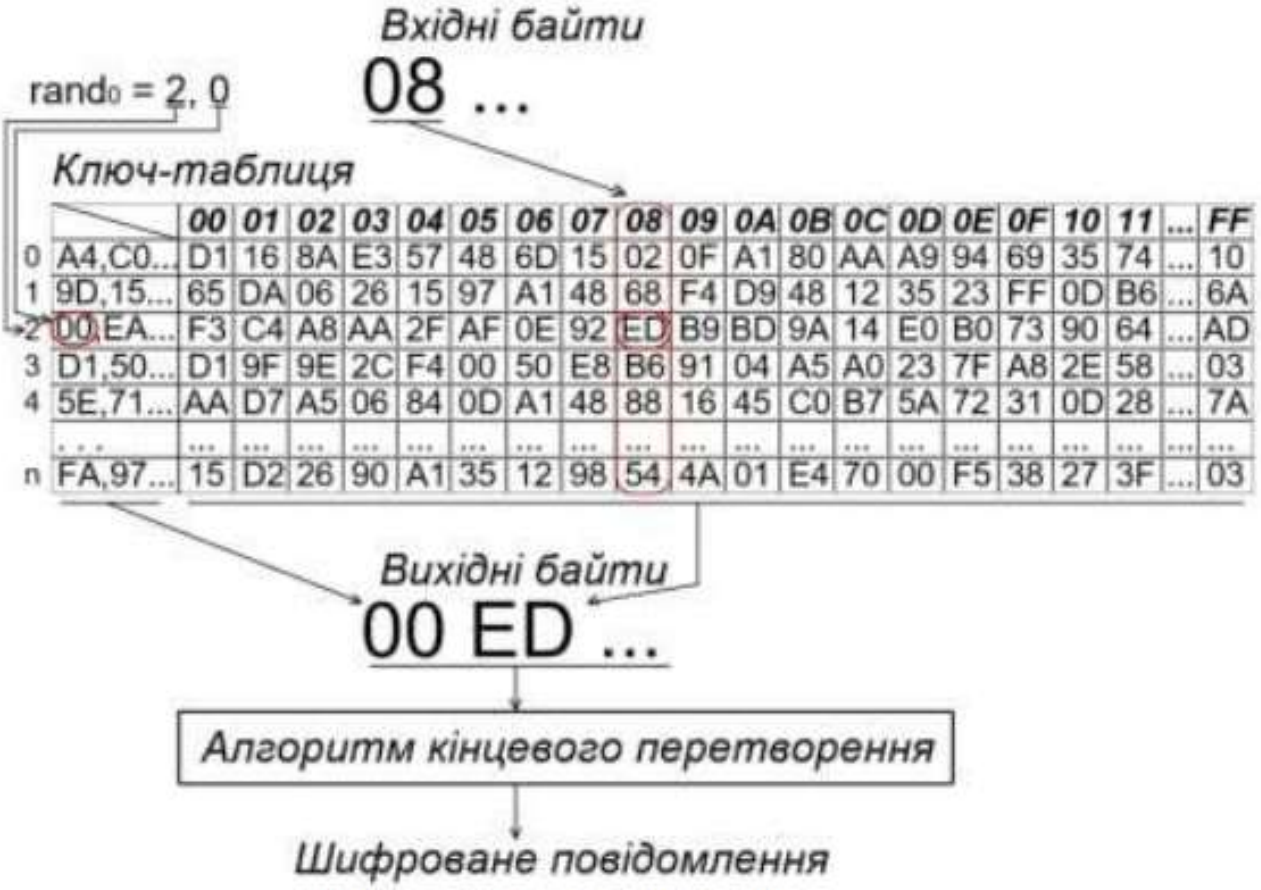
Логікою шифрування в даному алгоритмі буде заміна одного байту на інший, відповідно до деякої поліалфавітної таблиці перестановок. Дана таблиця міститиме в собі n рядків та 256 стовпців, якщо за умовний блок даних ми візьмемо один байт. В кожному рядку буде послідовність від 0 до 255, перемішана випадковим чином. Кількість рядків, а відповідно «алфавітів перестановки», може бути якою завгодно.

Кожному рядку присвоюється m випадкових двійкових ідентифікаторів. Розрядність цих ідентифікаторів також може бути обрана різна, проте вона повинна бути однаковою для всіх ідентифікаторів. Від обраної розрядності залежатиме, по-перше, кількість алфавітів, яку ми можемо визначити, наприклад якщо розрядність дорівнюватиме двом, то максимум у нас може бути 4 алфавіти, а з урахуванням рекомендації використовувати не менше двох ідентифікаторів на один алфавіт – всього 2. По-друге, від розрядності залежатиме те, наскільки збільшиться шифрований текст, порівняно з вхідним повідомленням. Якщо розрядність ідентифікатора буде рівною розрядності нашого блоку даних, а у нашому випадку – це 8 біт, то шифроване повідомлення буде вдвічі більше за вхідне. Чим менша розрядність ідентифікатора, по відношенню до розміру блоку даних, тим менше буде збільшення вихідного повідомлення при шифруванні і навпаки.

Дана таблиця замін, разом із відповідними ідентифікаторами, складатиме собою секретний ключ. Для її компактного запису можна використовувати різноманітні технології стиснення.

Схема шифрування даним методом показана на рис. 1: спочатку ми обираємо два випадкових числа, з яких перше – в діапазоні $[0; n)$, де n – кількість алфавітів, а друге – в діапазоні $[0; m)$, де m – кількість ідентифікаторів, відповідних даному алфавіту. Після цього ми заміняємо вхідний блок даних на вираз, що складається з обраного ідентифікатора та числа, що знаходиться в обраному алфавіті на позиції, номер якої відповідає числу, утвореного з блока вхідних даних. Тобто, якщо ми шифруємо 8-бітні блоки, і на вході, як приклад, отримали блок «10001010», що у десятковій формі відповідає числу 138, а випадкові числа випали 2 і 0, за умови, що вони відповідатимуть межам зазначених діапазонів, то шифрований вираз для даного блоку складатиметься з 0-го ідентифікатора 2-го алфавіту

перестановок та числа, що знаходиться на 138-й позиції у 2-му алфавіті. Такі перетворення почергово проводяться для кожного блоку даних, до кінця відкритого повідомлення.



		00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	...	FF
0	A4,C0...	D1	16	8A	E3	57	48	6D	15	02	0F	A1	80	AA	A9	94	69	35	74	...	10
1	9D,15...	65	DA	06	26	15	97	A1	48	68	F4	D9	48	12	35	23	FF	0D	B6	...	6A
2	00,EA...	F3	C4	A8	AA	2F	AF	0E	92	ED	B9	BD	9A	14	E0	B0	73	90	64	...	AD
3	D1,50...	D1	9F	9E	2C	F4	00	50	E8	B6	91	04	A5	A0	23	7F	A8	2E	58	...	03
4	5E,71...	AA	D7	A5	06	84	0D	A1	48	88	16	45	C0	B7	5A	72	31	0D	28	...	7A
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
n	FA,97...	15	D2	26	90	A1	35	12	98	54	4A	01	E4	70	00	F5	38	27	3F	...	03

Рисунок 1 – Схема шифрування

Криптографічним ключем у даній криптосистемі є таблиця з довільною кількістю випадково перемішаних підстановочних алфавітів, кожному з яких належить набір ідентифікаторів.

Алгоритм шифрування складається з наступних етапів: представлення відкритої інформації у бітовому (двійковому) вигляді, генерація випадкових значень (номеру алфавіту та ідентифікатора) для кожного n-бітного вхідного блоку, підстановка вхідних блоків за випадково обраним алфавітом, зчеплення підставленого блоку та ідентифікатора використаної підстановочної системи, кінцеве перетворення з метою «затирання» ідентифікаторів.

Алгоритм дешифрування є протилежним до алгоритму шифрування. Однозначне дешифрування у даному алгоритмі можливе завдяки ідентифікаторам, які однозначно вказують на те який підстановочний алфавіт було використано для конкретного блоку даних.

Відсутність складної ітеративності забезпечує високу швидкість роботи даного алгоритму.

Таким чином досягається висока ентропія зашифрованої інформації, а випадковість та відсутність закономірностей у перестановках унеможливлює та серйозно ускладнює різні види криптоаналізу. Надійність даного алгоритму напряму залежить від стійкості генератора випадкових чисел.

Стеганографічне функціонування алгоритму базується на припущенні: «Якщо з використанням нелінійного шифрування кожен n-бітний блок може бути замінено багатьма способами, в залежності від того яку підстановочну систему було обрано, а самі підстановочні системи генеруються без будь-яких закономірностей, то, фактично, будь-який вхідний блок після зашифрування може бути перетворений на будь-який вихідний блок, при чому ця відповідність є контрольованою».

Інакше кажучи, даний алгоритм дає можливість створити такий ключ, при якому шифротекст матиме деякий конкретний вигляд, а контрольована генерація ключа дасть можливість контролювати вихідний шифротекст.

На основі цього можна припустити, що можливо створити такий ключ, при якому деяке повідомлення, зашифроване одним із можливих способів на виході дасть шифротест, що представляє собою деяке інше осмислене повідомлення.

Перелік посилань

1. Jara Vera V, Sánchez Ávila C (2019) Graphemic-phonetic diachronic linguistic invariance of the frequency and of the Index of Coincidence as cryptanalytic tools. PLoS ONE 14(3): e0213710 10.1371/journal.pone.0213710
2. Cavaliere, Fabio, John Mattsson, and Ben Smeets. «The security implications of quantum cryptography and quantum computing.» Network Security 2020.9 (2020): 9-15.
3. Прикладна криптологія: системи шифрування : підручник / О.Г. Корченко, В.П. Сіденко, Ю.О. Дрейс. – К. : ДУТ, 2014. – 448 с.

УДК 004.891

В.А. АНІКІН
Хмельницький національний університет
В.М. ДЖУЛІЙ
Хмельницький національний університет
І.В. МУЛЯР
Хмельницький національний університет
В.С. ОРЛЕНКО
Хмельницький національний університет
В.Ю. ТІТОВА

СИМЕТРИЧНА КРИПТОСИСТЕМА З НЕЛІНІЙНИМ ШИФРУВАННЯМ ТА МОЖЛИВІСТЮ КОНТРОЛЮ ШИФРОТЕКСТУ З МЕТОЮ МАСКУВАННЯ

В статті запропонована проста симетрична перестановочна криптосистема, перестановки в якій відбуваються на основі потокової генерації випадкових чисел. Наведений криптографічний алгоритм, в зв'язку з нелінійністю процесу шифрування, здатний шифрувати одні і ті ж блоки інформації по-різному. Дана властивість може бути використана для формування шифротексту конкретного вигляду та, як наслідок, для прихованої передачі шифрованих повідомлень і повного, або часткового маскування факту криптографічного впливу.

Підстановочно-перестановочні алгоритми достатньо розповсюджені на сьогоднішній день. Вони зарекомендували себе як надійні криптостійкі шифри, що склали серйозну конкуренцію ітеративним математичним шифрам, на базі мережі Фейстеля, та поступово все більше та більше витісняють їх, займаючи їх місце.

В статті запропонований алгоритм шифрування, який має просту структуру та легкий в реалізації, але при цьому, прогнозовано, він є достатньо стійким для криптоатак, має високу швидкість роботи та використовує всі переваги сучасних алгоритмів шифрування та має потенціал для маскування шифрованих даних.

Розроблено схему шифрування, в якій перестановки відбуваються у випадковому порядку, та, як наслідок, стає можливою ситуація, коли одному відкритому тексту при одному і тому ж ключі відповідає безліч шифротекстів, що серйозно ускладнює роботу криптоаналітиків, ускладнюючи аналіз та злом шифру, а також суттєво збільшує область прикладного використання даної криптосистеми. Також дана криптосистема має стеганографічний потенціал, оскільки нелінійність, яка породжує варіативність шифротекстів, дозволяє, фактично, один і той же самий байт зашифрувати десятками, або навіть сотнями комбінацій.

Запропоновано новий підхід до створення шифрувальної варіативності та випадкового вибору, що дає можливість модифікації вже існуючих алгоритмів шифрування.

Ключові слова: криптографія, стеганографія, симетрична криптосистема, перестановочний шифр, випадкова генерація, захист інформації.

V. A. Anikin
Khmelnytsky National University, Khmelnytsky, Ukraine
V.M. Dzhuliy
Khmelnytsky National University, Khmelnytsky, Ukraine
I. V. Mulyar
Khmelnytsky National University, Khmelnytsky, Ukraine
V.S. Orlenko
Khmelnytsky National University, Khmelnytsky, Ukraine
V. Yu. Titova
Khmelnytsky National University, Khmelnytsky, Ukraine

SYMMETRIC CRYPTOSYSTEM WITH NONLINEAR ENCRYPTION AND THE POSSIBILITY OF CONTROL OF CIPHERTEXT FOR CONCEALMENT

The article proposes a simple symmetric permutation cryptosystem, in which permutations occur based on streaming generation of random numbers.

Substitution-permutation algorithms are quite common today. They have proven to be reliable crypto-strong ciphers that have seriously competed with iterative mathematical ciphers, based on the Feistel network, and are gradually displacing them more and more, taking their place.

The aim of the article is to propose an encryption algorithm that has a simple structure and is easy to implement. At the same time, it is estimated to be quite resistant to crypto attacks. This encryption algorithm has a high speed and uses all the advantages of permutation encryption algorithms.

Developed an encryption scheme in which permutations occur randomly, and as a result, a situation becomes possible when one plaintext with the same key corresponds to many ciphertexts, which seriously complicates the work of cryptanalysts, complicating the analysis and cracking of the cipher, and significantly increases the scope of application of this cryptosystem. Also, this cryptosystem has steganographic potential, because the nonlinearity that creates the variability of ciphertexts, allows the same byte to encrypt dozens or even hundreds of combinations.

This paper has clearly shown a new approach to the creation of encryption variability and random selection is proposed. It makes possible to modify existing encryption algorithms.

Key words: cryptography, steganography, symmetric cryptosystem, permutation cipher, random generation, information protection.

Вступ

У всі часи люди намагалися захищати інформацію, яку вважають важливою, постійно розвиваючи та удосконалюючи засоби для її захисту. Стислий Оксфордський словник англійської мови визначає криптографію як «мистецтво написання або розв'язування кодів". Це історично точно, але воно не враховує сучасної розповсюдженості галузі та її сьогоденних наукових основ [1]. На сьогоднішній день криптографія має вкрай широкий спектр застосувань, починаючи від класичного захисту інформації при зберіганні, чи передачі, закінчуючи контролем цілісності повідомлень та цифровими підписами, для верифікації даних.

Також здавна люди намагались приховувати ту чи іншу інформацію, різноманітними способами маскуючи її. Такі починання згодом переросли в науку під назвою стеганографія, що на сьогоднішній день широко використовується в сфері захисту інформації.

У розрізі цього, не видається дивним що здавна і до сьогодні люди працюють над розробкою нових і нових криптосистем: шифр Цезаря змінили криптосистеми простої заміни, їх, в свою чергу, витіснили поліалфавітні системи, на кшталт шифрів Гронсфельда та Віженера [2], а їх, пізніше, – ітеративні блочні шифри, на основі мережі Фейстеля, яскравим прикладом яких є DES [3], різноманітні модифікації якого широко використовуються до сьогоднішнього дня.

Разом із розвитком криптосистем, невпинно розвивався і криптоаналіз, накопичений інструментарій якого [4], у поєднанні з технічними можливостями сучасної обчислювальної техніки, задає вкрай високу планку вимог для сучасної криптографії вцілому та безпосередньо для використовуваних криптосистем.

На сьогоднішній день одними з найбільш захищених криптосистем у світі є шифри, на основі підстановочно–перестановочних систем, наприклад SQUARE та Rijndael, останній відомий на широкий загал як Advanced Encryption Standard (AES) [3, 5].

З огляду на це, можна стверджувати що підстановочно–перестановочні системи, спроектовані таким чином, щоб максимально використовувати сучасні обчислювальні потужності та досягати високої ентропії при шифруванні, здатні надійно захищати інформацію та є стійкими до сучасних криптоатак.

Також, безумовно, важливою складовою захисту інформації є непомітність. Логічно, що інформацію, яка виглядає цілісно, осмислено, без слідів того що в ній щось приховано, або зашифровано, навряд чи хто буде досліджувати без суттєвих на те підстав. Саме через це в сучасному світі, у сфері кібербезпеки, все частіше використовується стеганографія – наука, що до надає інструментарій для приховування однієї, умовно «таємної»,

інформації всередині іншої – публічної, тієї що не представляє для сторонніх осіб жодного інтересу. В зв'язку з цим, виникає великий попит на алгоритми, які підходять до захисту комплексно: вони здатні як захистити інформацію криптографічно, так і, повністю, або частково, замаскувати сам факт того що деяка інформація була прихована.

Огляд літературних джерел

Симетричні криптосистеми давно відомі науці та є в достатній мірі вивчені та проаналізовані. В зв'язку з цим, дослідження симетричних систем шифрування знайшло відбиток у великій кількості наукових статей та літературі. Дані напрацювання обов'язково слід враховувати при розробці криптосистеми.

Симетричні криптосистеми на протязі багатьох століть використовувались для захисту інформації та «закриття» важливої інформації від несанкціонованого використання [6, 7, 8]

Симетричні алгоритми шифрування перетворюють деяку конфіденційну інформацію, на основі криптографічного ключа, у шифровану форму. Зворотне перетворення, яке відновлює вхідну інформацію, відбувається при зворотному виконанні алгоритму, за умови, що у вас є такий цей ж самий ключ [8]. Сучасні симетричні криптосистеми, хоч і є більш витонченими, математично та комбінаторно обґрунтованими, все одно, як правило дотримуються таких принципів.

Також важливою складовою при розробці криптосистеми є оцінка інструментарію криптоаналізу, та сучасні можливості у сфері «взлому» систем шифрування. Для створення криптостійкого алгоритму слід на етапі проектування враховувати те, яким чином він буде поводитись під час криптоатаки та забезпечити достатній запас міцності та стійкості [9 - 11].

Постановка задачі

Необхідно створити простий у реалізації криптостійкий перестановочний симетричний алгоритм шифрування, який би відповідав сучасним вимогам безпеки та значно ускладнював існуючі способи криптоаналізу, при цьому з можливістю корегування вихідного шифротексту.

Основна ідея даного алгоритму полягає в тому, що Р-блок, за яким відбуватиметься перестановка, для кожного байту обирається випадково, на основі генератора випадкових чисел. Такий підхід значно ускладнить будь-який статистичний криптоаналіз, оскільки будь-які закономірності між відкритими байтами та зашифрованими будуть відсутні. Потокова випадкова генерація коефіцієнта перестановки дозволить досягти неоднорідності шифрування та відсутності закономірностей у ньому.

Також, через відсутність складної ітеративності, даний алгоритм повинен мати високу швидкість роботи, порівняно з аналогами.

Основна частина

Відповідно до поставленої задачі, криптосистема повинна відповідати сучасним вимогам безпеки, бути надійною та придатною для практичного використання, забезпечуючи високу надійність, а отже, даний шифр повинен бути стійким перед інструментарем сучасного криптоаналізу.

Одною з головних потужностей криптоаналізу на сьогодні є обчислювальна потужність комп'ютерної техніки, яка серйозно збільшується з кожним новим поколінням. Фактично будь-який симетричний шифр можливо взламати атакою «грубої сили», тобто повним перебором всіх можливих криптографічних ключів, та почерговими спробами розшифрування кожним з них. До сьогоднішнього дня даний метод – найбільш універсальний серед всіх існуючих, за умови що нам відомо тип алгоритму та принцип його роботи, при цьому не важко здогадатись, що ККД даного методу є вкрай низьким.

Для оцінки реальності загрози атаки грубої сили вводиться термін «практичний час», як деяка відповідність теоретичному часу, за який машина прогнозовано зможе підібрати ключ, до часу збереження актуальності та цінності прихованої інформації. Простіше кажучи, якщо умовна Єва перехопила зашифроване

повідомлення компанії конкурента, і на атаку грубої сили, прогнозовано буде затрачено 50 років, то такий час можна вважати непрактичним, адже за 50 років розшифроване повідомлення дасть Єві мінімум користі. При цьому обов'язково слід врахувати той факт, що якщо на сьогоднішній день прогнозований час атаки й складатиме 50 років, то абсолютно не факт, що через кілька років, з урахуванням нових технологій та потужностей, цей час не складатиме 50 годин, чи навіть хвилин. Окрім цього на сьогоднішній день все більшої популярності набувають розподілені обчислення, що переганяють за швидкістю роботи навіть передові суперкомп'ютери [12].

Таким чином запас міцності, в контексті стійкості криптосистеми до атаки грубої сили, як одна з базових характеристик надійності шифру повинна враховувати не лише обчислювальні потужності, існуючі сьогодні, а і їх прогнозоване зростання, яке є неминучим.

Слід враховувати й безліч інших існуючих криптоатак, такий як, наприклад, атака на основі перехопленого відкритого тексту, атака на основі підібраного відкритого тексту, диференціальний криптоаналіз та інші існуючі підходи.

Головна сила запропонованої криптосистеми – нелінійність та рандомізація алгоритму, саме це і повинно забезпечити його високу криптостійкість. В цілому, ідея полягає у створенні кількох можливих варіантів роботи шифру на тому, чи іншому етапі його виконання, серед яких випадковим чином обирається один. Для можливості дешифрування, позначається лише індекс обраного варіанту, який не дає криптоаналітику жодної додаткової інформації про те, що за ним стоїть.

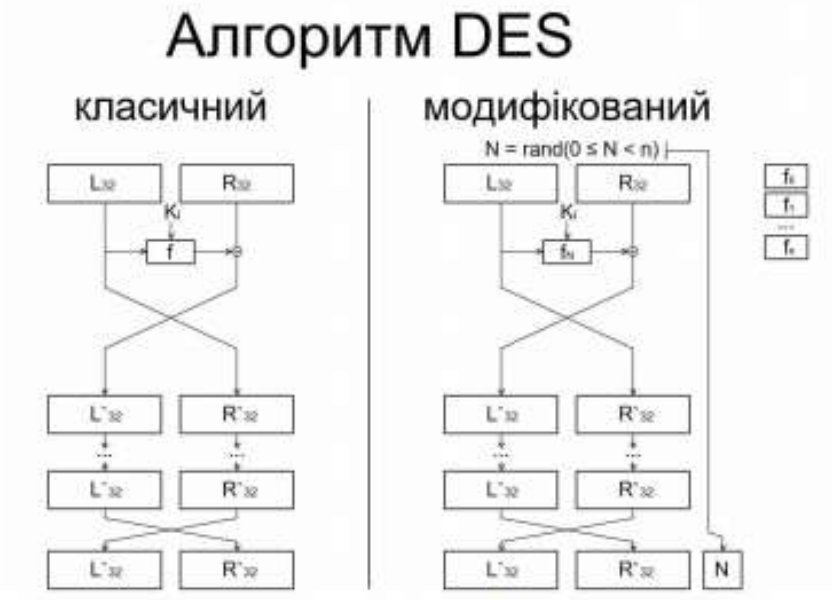


Рисунок 1 – Схема звичайного та модифікованого алгоритму DES

Даний підхід можна продемонструвати на прикладі алгоритму DES (Рис. 1).

В класичному представлені даного алгоритму у нас є одна функція, яка на основі раундового ключа та першого підблоку мережі Фейстеля, розміром 48 та 32 біта, відповідно, та повертає новий 32-бітний блок даних. Проте давайте змоделюємо ситуацію, при якій в даному алгоритмі у нас не одна функція, а, до прикладу, їх вісім. Кожна з цих функцій також прийматиме на вхід два блока у 48 та 32 біт та повертатиме, на їх основі, новий 32-бітний блок, при цьому кожна з них буде робити це по-різному. Якщо точніше, вони будуть і далі працювати за схемою класичної функції DES, проте, кожна з цих 8 функцій матиме деякі особливості: одна, наприклад, по-іншому розширюватиме вхідний блок, друга – перед розширенням проводитиме циклічний зсув вхідного блоку на

деяке число, третя – проводитиме аналогічний зсув, але вже після розширення, четверта – залишиться у класичному вигляді, без змін, а п'ята – додаватиме 27, за модулем 256, до кожного парного байту після розширення вхідного блоку і так далі. Як результат – кожна функція, в цілому, подібна до інших та кожну з них можна без проблем замінити іншою, проте 32-бітні блоки які буде повертати та, чи інша функція, як правило, будуть різні.

Тепер нам залишається лише незначно модифікувати загальний алгоритм шифрування: при генерації ключа, додатково, випадковим чином цим функціям буде присвоєно індекси від 0 до 9. Інформація про дані індекси та їх відповідності буде частиною секретного ключа. А при шифруванні, перед першим раундом кожного блоку, ми обиратимемо випадкове ціле число в діапазоні $[0; 8)$, яке і вказуватиме яку функцію з переліку ми використовуватимемо у наступних 16 раундах. Інформацію про те яка функція була використана можемо записати у вигляді 3-бітної комбінації в кінці, після зашифрованого 64-бітного блоку.

При такій модифікації алгоритму шифрований текст розшириться на n біт, де $n = \frac{N_{\text{повідомлення}} \times 67}{64}$, тобто, приблизно, на 4,5%.

В описаному випадку, при перехопленні шифрованого повідомлення, у криптоаналітика будуть на руках самі шифроблоки, а також індекси використаних функцій, без будь-якої важливої інформації, звичайно, якщо криптоаналітик знає принцип роботи даної модифікації. Проте в даній ситуації спеціаліст зможе однозначно сказати чи однакові функції були використані, в тому, чи іншому випадку, чи ні, елементарно зрівнявши ці самі індекси вкінці блоків. На основі даної інформації він зможе згрупувати блоки, в яких були використані однакові функції та досліджувати їх окремо.

Для того щоб позбавити криптоаналітика такої можливості, введемо ще одну зміну: індексів буде більше ніж самих функцій. На описаному вище прикладі, це можна зробити двома шляхами: або зменшити кількість функцій, тоді в нас буде, наприклад, 4 функції та 8 індексів, або збільшити кількість індексів, тоді при 8 функціях ми матимемо 16 індексів.

При генерації нового ключа, як і раніше, ми випадково генеруємо та записуємо відповідності між функціями та індексами, проте вже не у відношенні 1:1, а у відношенні 1:m, де одній функції відповідатиме кілька індексів. При шифруванні тепер ми кидатимемо монетку двічі: перший раз – щоб вибрати функцію, другий – щоб обрати один індекс, серед всіх, пов'язаних з даною функцією.

Слід зазначити що дані числа краще обирати рівним 2^n, а кількість функцій, рекомендовано, має бути кратною кількості індексів, при чому вони повинні бути рівномірно розподілені між функціями. Це дозволить в повній мірі, без надлишковості, використовувати діапазон значень, при шифруванні, а також забезпечить однакову ймовірність випадання функцій.

Тепер, перехопивши повідомлення, криптоаналітик не зможе однозначно сказати чи були використані однакові функції в блоках з різними індексами, адже це може бути як інший ідентифікатор тієї ж самої функції, так й ідентифікатор іншої функції.

Також звідси витікає цікава властивість даного алгоритму: зашифрувавши одне і те ж саме повідомлення, використовуючи один і той ж самий ключ, ми отримаємо різні шифротексти, оскільки тепер те як буде зашифровано той чи інший блок, залежатиме від випадкового показника, при шифруванні. Отже, криптоаналітик, перехопивший n зашифрованих повідомлень, не зможе однозначно виявити які серед них будуть ідентичними після дешифровки.

Приклад, наведений на основі криптографічного алгоритму DES, наглядно та в повній мірі демонструє принципи роботи, на базі яких пропонується створити окрему криптосистему.

Навіть у вигляді модифікації до вже існуючого алгоритму, така нелінійність, наявність кількох «сценаріїв» роботи, серед яких обирається випадковий, дозволяє значно ускладнити роботу криптоаналітиків та понизити ефективність машинного криптоаналізу. Причому слід зазначити, що в описаній модифікації зазначена

варіативність була додана лише для раундових функцій, хоча це зовсім не єдиний аспект, де даний підхід можна було б застосувати, навіть у випадку шифру DES.

Таким чином, аналогічно до наведеного прикладу ми можемо спроектувати окремий алгоритм, в якому описаний принцип нелінійності буде не модифікацією, а основою його роботи.

Логікою шифрування в даному алгоритмі буде заміна одного байту на інший, відповідно до деякої поліалфавітної таблиці перестановок. Дана таблиця міститиме в собі n рядків та 256 стовпців, якщо за умовний блок даних ми візьмемо один байт. В кожному рядку буде послідовність від 0 до 255, перемішана випадковим чином. Кількість рядків, а відповідно «алфавітів перестановки», може бути якою завгодно.

Кожному рядку, точно як і функції, у прикладі з DES, присвоюється m випадкових двійкових ідентифікаторів. Розрядність цих ідентифікаторів також може бути обрана різна, проте вона повинна бути однаковою для всіх ідентифікаторів. Від обраної розрядності залежатиме, по-перше, кількість алфавітів, яку ми можемо визначити, наприклад якщо розрядність дорівнюватиме двом, то максимум у нас може бути 4 алфавіти, а з урахуванням рекомендації використовувати не менше двох ідентифікаторів на один алфавіт – всього 2. По-друге, від розрядності залежатиме те, наскільки збільшиться шифрований текст, порівняно з вхідним повідомленням. Якщо розрядність ідентифікатора буде рівною розрядності нашого блоку даних, а у нашому випадку – це 8 біт, то шифроване повідомлення буде вдвічі більше за вхідне. Чим менша розрядність ідентифікатора, по відношенню до розміру блоку даних, тим менше буде збільшення вихідного повідомлення при шифруванні і навпаки.

Дана таблиця замін, разом із відповідними ідентифікаторами, складатиме собою секретний ключ. Для її компактного запису можна використовувати різноманітні технології стиснення.

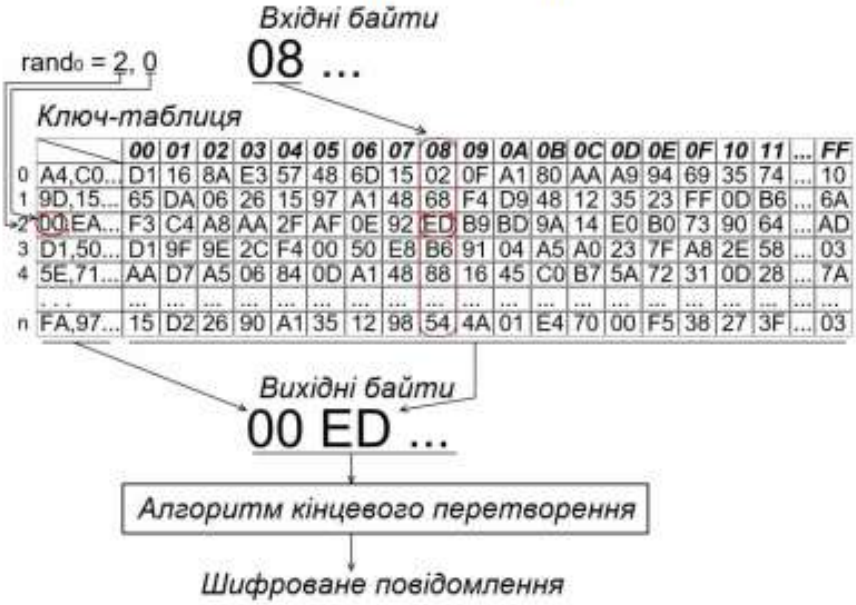


Рисунок 2 – Схема шифрування

Схема шифрування даним методом показана на рис. 2 та в цілому подібна до описаної на прикладі DES: спочатку ми обираємо два випадкових числа, з яких перше – в діапазоні $[0; n)$, де n – кількість алфавітів, а друге – в діапазоні $[0; m)$, де m – кількість ідентифікаторів, відповідних даному алфавіту. Після цього ми заміняємо вхідний блок даних на вираз, що складається з обраного ідентифікатора та числа, що знаходиться в обраному алфавіті на позиції, номер якої відповідає числу, утвореного з блока вхідних даних. Тобто, якщо ми шифруємо 8-бітні блоки, і на вході, як приклад, отримали блок «10001010», що у десятковій формі відповідає числу 138, а випадкові числа

випали 2 і 0, за умови, що вони відповідатимуть межам зазначених діапазонів, то шифрований вираз для даного блоку складатиметься з 0-го ідентифікатора 2-го алфавіту перестановок та числа, що знаходиться на 138-й позиції у 2-му алфавіті. Такі перетворення почергово проводяться для кожного блоку даних, до кінця відкритого повідомлення.

В прикладі з DES на цьому ми закінчували, проте в даному випадку не слід залишати шифроване повідомлення в такому вигляді, оскільки у обізнаного криптоаналітика, при перехопленні буде можливість проаналізувати однакові ідентифікатори та відповідно, на основі великої кількості повідомлень скласти деяку корсляцію. Щоб позбавити його такої можливості, ми використаємо два прийоми.

По-перше, ми додамо варіативності до принципу зчеплення ідентифікатора та блоку заміни. Це дозволить ускладнити розпізнавання ідентифікаторів та ускладнить криптоаналіз. Приклади зчеплень елементів продемонстровано на рис. 3.

По-друге, після формування вихідного масиву байт, після завершення замін, ми «змішаємо» його за допомогою алгоритму кінцевого перетворення. Найпростішим варіантом даного алгоритму може бути поблокове додавання за модулем 2, в режимі зчеплення блоків, де ключ та перший блок деяким чином формуються на основі таблиці. Даний алгоритм може бути реалізовано на основі мережі Фейстеля з невеликою кількістю раундів, на основі матричних перестановок, чи на основі будь-якого іншого існуючого алгоритму, що забезпечує надійне «розмиття».

Таким чином даний алгоритм буде стійким до атаки грубою силою, оскільки простим перебором відновити таблицю замін не можливо за практичний час, а за необхідності, розмір вхідного блока даних можна додатково збільшити. Кількість можливих варіантів перестановок для одного 8-бітного алфавіту складає 256!, а за умови що алфавітів багато, кількість можливих комбінацій забезпечує надійність ключа навіть з урахуванням розвитку технологій.

Слід зауважити, що при абсолютно випадковій генерації ключа, шифротекс, звісно, також вийде випадковий. При цьому, ми можемо піти від зворотнього: припустимо що в нас вже є шифротекст, який ми маємо отримати. Ми можемо згенерувати такий ключ, щоб при шифруванні конкретним способом, без викидання випадкових чисел, ми отримали необхідний нам шифротекст. При даному підході також з'являються додаткові вимоги до алгоритму кінцевого перетворення: він повинен мати змогу розшифрувати будь-який випадковий масив байт.

Ідентифікатор	*Блок заміни*
11111111	00000000

Варіанти зчеплення ідентифікатора та блоку:

- просте зчеплення	1111111100000000
- змішування	1010101010101010
- кільцеве	1111000000001111
- перехрестне	1111000011110000
- комбіноване	1010101011110000

Рисунок 3 Приклади зчеплення ідентифікатора та блоку заміни

Нелінійність алгоритму ускладнить машинний та людський криптоаналіз та затруднить статистичні дослідження.

Даний алгоритм також буде стійким на до атаки на основі відкритого тексту, оскільки, маючи на руках відповідне шифроване повідомлення, криптоаналітик матиме лише один з можливих варіантів шифротексту, при цьому криптоаналітик не матиме доступу до всіх інших можливих варіантів шифрування.

В цілому запропонований алгоритм представляє собою цікавий та перспективний метод криптографічного захисту, безумовно потребуючи розвитку та подальших досліджень. Окрім криптографічного, даний алгоритм представляє також стеганографічний інтерес, оскільки нелінійність, яка породжує варіативність шифротекстів, дозволяє, фактично, один і той ж самий байт зашифрувати десятками, або навіть сотнями комбінацій. Виходячи з цього, теоретично, можливо створити такий ключ, при якому деякий конфіденційний текст в зашифрованому вигляді представлятиме інший осмислений текст, що не викликатиме підозр. Даний аспект також потребує додаткових досліджень.

Висновки

Таким чином, алгоритми шифрування, що мають в собі елементи нелінійності, включають рівноможливі варіанти виконання, серед яких в процесі виконання обирається один, значно підвищують його стійкість та захищенність.

Запропонований алгоритм, який в повній мірі відповідає описаним принципам, проявляє цікаві властивості, а проведенні дослідження говорять про його достатню захищеність та стійкість.

Безумовно, наведені алгоритми потребують додаткового вивчення, проте вже на даному етапі можна говорити про те, що навіть модифікація вже існуючих алгоритмів, введення в них варіативних елементів, є легким способом підвищення їх надійності.

Окрім зазначеного, даний алгоритм може знайти розвиток у сфері стеганографії.

Літертура

1. Інформаційна безпека: навчальний посібник / [Ю. Я. Бобало, І. В. Горбатий, М. Д. Кіселичник, А. П. Бондарєв та ін.]; за заг. ред. д-ра техн. наук, проф. Ю. Я. Бобала та д-ра техн. наук, доц. І. В. Горбатого. – Львів: Видавництво Львівської політехніки, 2019. – 580 с.
2. Прикладна криптологія: системи шифрування : підручник / О. Г. Корченко, В. П. Сіденко, Ю. О. Дрейс. – К. : ДУТ, 2014. – 448 с.
3. Основы криптографии/ Г.В. Басалова. – М.: ИНТУИТ, 2016. - 283 с.
4. Cryptology and information security - past,present, and future role in society/ S. Bhattacharya. - International Journal on Cryptography and Information Security (IJCIS). – Vol. 9, No.1/2, 2019. – P. 13-36.
5. Муляр І.В. Ітераційно-геометричний метод для стійкого перцептуального хешування зображення / В. М. Джулій, Ю. П. Кльоц, І. В. Муляр, В. М. Чешун // Вісник Хмельницького національного університету. Технічні науки. – 2020. – № 1. – С. 76–79
6. Hybrid cryptographic technique using rsa algorithm and scheduling concepts/ M. Shankar, P. Akshaya// International Journal of Network Security & Its Applications (IJNSA)/ Vol.6, No.6. – 2014. - p. 39-48
7. Комп'ютерна стеганографічна обробка й аналіз мультимедійних даних: підручник. / Г. Ф. Конахович, Д. О. Прогонов, О. Ю. Пузиренко. – Київ: «Центр учбової літератури», 2018. – 558 с.
8. Digital Watermarking and Steganography: Fundamentals and Techniques / Edited by Frank Y. Shih. – Taylor & Francis Group, 2017. – 293 p
9. The Evolution of Cryptology [Електронний ресурс]/ G. R. Souza. – California State University, San Bernardino, 2016. – режим доступу: https://core.ac.uk/reader/55336770
10. Cryptology and communication security [Електронний ресурс]/ Shri Kant. – Defense Science Journal. – Vol. 62. - №1. – режим доступу: https://core.ac.uk/reader/333719963
11. Криптоанализ шифрсистемы ACBF/ И. В. Боровкова, И. А. Панкратова. – Прикладная дискретная математика. Приложение, Томск. – 2019. – С. 90-93.
12. Захист інформаційних ресурсів: криптографічні та стеганографічні методи захисту даних. Посібник для викладачів, вчителів та студентів інформатичних спеціальностей./ В.М. Франчук – К.: НПУ імені М.П. Драгоманова, 2014. – 120 с.