МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки

КРУК Олег Васильович

Алгоритм захисту інформації на основі символьного розщеплення / Algorithm of Information Protection Based on Symbolic Splitting

спеціальність: 125 – Кібербезпека та захист інформації

освітньо-професійна програма –Кібербезпека

Кваліфікаційна робота

Виконав студент групи КБм -21
О.В. Крук

Науковий керівник
д.т.н., професор М.М. Касянчук

Кваліфікаційну роботу допущено
до захисту:

«____» _____ ___2024 р.

Завідувач кафедри
_____ В.В.Яцків

ТЕРНОПІЛЬ – 2024

Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки
Освітній ступінь «магістр»
спеціальність: 125 – Кібербезпека та захист інформації
освітньо-професійна програма – Кібербезпека

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ В.В.Яцків
«____» _____ 2023 року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
КРУК ОЛЕГ ВАСИЛЬОВИЧ

1. Тема кваліфікаційної роботи:
Алгоритм захисту інформації на основі символьного розщеплення / Algorithm of Information Protection Based on Symbolic Splitting

керівник роботи д.т.н., професор М.М. Касянчук
затверджені наказом по університету від 12 грудня 2023 року № 753
2. Строк подання студентом закінченої кваліфікаційної роботи 12 грудня 2024 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:
– аналіз сучасних підходів для забезпечення захисту інформації;
– розробка методу захисту інформації за допомогою символьного розщеплення;
– розробка математичної моделі захисту інформації на основі символьного розщеплення;
– визначення етапів роботи алгоритмів захисту та відновлення простого розщеплення символу;
– розробка системи захисту символів з розщепленням при відображенні інформації різними поєднаннями чисел;
– розробка статистичного експерименту для дослідження запропонованого алгоритму.

5. Перелік графічного матеріалу у роботі:
– схема захисту та відновлення повідомлень за методом Вернама;
– блок-схема ключа захисту;
– заміна тексту Encryption ASCII кодами;
– ключ захисту для різних рівнів символьного розщеплення;

– результати тесту Колмогорова-Смирнова;

– тест серій для квадратичного конгруентного генератора з використанням ДГА;

– ентропія як міра випадковості для лінійного конгруентного генератора без використання ДГА.

6. Консультанти розділів кваліфікаційної роботи

	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 12 грудня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строки виконання етапів кваліфікаційної роботи	Примітка
1	Аналіз сучасних підходів для забезпечення захисту інформації	12.2023 р. – 03.2024 р.	
2	Математичні моделі символьного розщеплення та їх застосування при захисті текстової інформації	03.2024 р. – 06.2024 р.	
3	Експериметнальні результати захисту інформації за допомогою символьного розщеплення	06.2024 р. – 11.2024 р.	

Студент _____ Крук О.В.
(підпис)

Керівник роботи _____ д.т.н., професор М.М.Касянчук

АНОТАЦІЯ

Випускна кваліфікаційна робота на тему „Алгоритм захисту інформації на основі символьного розщеплення” на здобуття освітнього ступеня «Магістр» зі спеціальності 125 „Кібербезпека та захист інформації” освітньо-професійної програми «Кібербезпека» написана обсягом 81 сторінок і містить 35 ілюстрацій, 5 таблиць, 1 додаток та 30 джерел за переліком посилань.

Метою випускної кваліфікаційної роботи є розробка алгоритму захисту інформації на основі символьного розщеплення

Методи дослідження. Математичні методи моделювання, чисельні методи, статистичні методи.

Результати дослідження. Здійснено аналіз сучасних підходів для захисту інформації, що дало змогу дослідити можливості існуючих методів та обгрунтувати актуальність використання символьного розщеплення для захисту інформаційних текстових потоків. Розроблено метод захисту інформації на основі символьного розщеплення, що дало змогу розробити відповідну математичну модель та блок-схему. Розроблено числові приклади застосування методу захисту на основі простого та узагальненого символьного розщеплення, які дали змогу продемонструвати переваги розробленого алгоритму. Розроблено систему захисту символів з розщепленням при відображенні інформації різними поєднаннями чисел та проведено статистичне тестування для випадкових чисел. Обгрунтовано використання ентропії для символьного розщеплення.

Результати роботи можуть успішно застосовуватися для захисту інформації, зокрема текстової, на основі символьного розщеплення.

КЛЮЧОВІ СЛОВА: РОЗЩЕПЛЕННЯ, СИМВОЛ, ЗАХИСТ ІНФОРМАЦІЇ, МОДЕЛЬ, СТАТИСТИЧНИЙ ТЕСТ.

ABSTRACT

The graduate work on the topic „Algorithm of Information Protection Based on Symbolic Splitting” for Master’s degree on speciality 125 "Cybersecurity " is written on 81 pages and contains 35 illustrations, 5 tables, 1 suplement and 30 references.

The aim of graduate work is to develop an information protection algorithm based on symbolic splitting.

Research methods. Mathematical modeling methods, numerical methods, statistical methods.

Results of the study. An analysis of modern approaches to information protection was carried out, which made it possible to explore the possibilities of existing methods and justify the relevance of using symbolic splitting to protect information text flows. A method of information protection based on symbolic splitting has been developed, which has made it possible to develop a corresponding mathematical model and a flowchart. Numerical examples of the application of the protection method based on simple and generalized symbolic splitting have been developed, which have made it possible to demonstrate the advantages of the developed algorithm. A system of symbol protection with splitting when displaying information by various combinations of numbers has been developed and statistical testing for random numbers has been conducted. The use of entropy for symbolic splitting has been justified.

The results of the work can be successfully applied to the protection of information, in particular text information, based on symbolic splitting.

Keywords: SPLITTING, SYMBOL, INFORMATION PROTECTION, MODEL, STATISTICAL TEST.

ЗМІСТ

ВСТУП

В сучасних умовах розвитку інформаційних технологій і комп'ютеризації роль інформації як одного із найбільш цінних і важливих активів в різних сферах неупинно зростає [1-4]. Захист текстової інформації при передачі по каналах зв'язку є важливим завданням для бізнес-застосунків і ряду інших областей життя сучасного суспільства [5-7]. Дослідження проблем розробки, удосконалення та застосування методів і засобів захисту інформації в процесі її передачі та зберігання набуло особливої важливості не тільки в державі, дипломатичних, військових, але також у банківських, комерційних та інших сферах, пов'язаних з широким кругом соціально-економічних проблем [8-10].

Захист текстової інформації має свою історію. В США К. Шеннон в 1944 р. створив основи теорії секретного зв'язку. В його роботах викладена теорія так званих секретних систем, які служать, фактично, математичною моделлю шифрів, які доповнюю алгебраїчні та інші властивості шифрів деяких елементів ймовірнісними властивостями, що дозволяє формалізувати багато установок задач синтезу і аналізу шифрів.

В роботах К. Шеннона і Вернама була дана можливість захисту інформації, що володіє властивістю абсолютної стійкості. Але на практиці були виявлені деякі труднощі в застосуванні цих результатів, у числі яких – необхідність використання одноразової гами для кожного вихідного символу і необхідність вказівки загальної довжини вихідного тексту. Ці обмеження приводять до складності реалізації методів захисту, тоді як порушення цих умов полегшує задачу зловмиснику [11]. Тому виникає необхідність створення методу захисту інформації, який сприяв би збільшення стійкості при менш строгих умовах.

Проблеми захисту інформації віддавна турбували людське суспільство. Потреба в захисті інформації виникла з необхідності таємної передачі військових, дипломатичних та інших повідомлень. Вона зародилась майже одночасно з самою технологією письма. Криптографія стала широко

застосовуватися не тільки в державних, дипломатичних, військових сферах, але також в банківських, комерційних та інших додатках [12-15].

Тому вивчення альтернативи у вигляді процедури розщеплення слід визнати дуже актуальною задачею. Дослідження властивостей запропонованого нового методу захисту текстової інформаці є актуальною задачею, яка виникає як у зв'язку із передачами по інформаційних мережах, так і з виникненням таких нових способів зберігання інформації, як хмарна технологія.

Мета роботи. Метою даної роботи є розробка алгоритму захисту інформації на основі символьного розщеплення.

Для вирішення поставленої мети вирішуються наступні **завдання**:

– аналіз сучасних підходів для забезпечення захисту інформації;

– розробка методу захисту інформації за допомогою символьного розщеплення;

– розробка математичної моделі захисту інформації на основі символьного розщеплення;

– визначення етапів роботи алгоритмів захисту та відновлення простого розщеплення символу;

– розробка системи захисту символів з розщепленням при відображенні інформації різними поєднаннями чисел;

– розробка статистичного експерименту для дослідження запропонованого алгоритму.

Об’єкт дослідження. Процес захисту інформації на основі символьного розщеплення.

Предмет дослідження. Методи і засоби захисту інформації на основі символьного розщеплення.

Методи дослідження. Математичні методи моделювання, чисельні методи, статистичні методи.

Наукова новизна одержаних результатів.

1. Здійснено аналіз сучасних підходів для захисту інформації, що дало змогу дослідити можливості існуючих методів та обгрунтувати актуальність

використання символьного розщеплення для захисту інформаційних текстових потоків.

2. Розроблено метод захисту інформації на основі символьного розщеплення, що дало змогу розробити відповідну математичну модель та блок-схему.

3. Розроблено числові приклади застосування методу захисту на основі простого та узагальненого символьного розщеплення, які дали змогу продемонструвати переваги розробленого алгоритму.

Практичне значення отриманих результатів. Розроблено систему захисту символів з розщепленням при відображенні інформації різними поєднаннями чисел та проведено статистичне тестування для випадкових чисел. Обгрунтовано використання ентропії для символьного розщеплення.

Публікації та апробація КР.

1. Крук О., Голембйовський М., Басістий В. Математична модель захисту інформації на основі символьного розщеплення. Збірник матеріалів науково - практичної конференції молодих вчених, аспірантів та студентів «Кібербезпека та комп'ютерно-інтегровані технології» (КБКІТ-2024), Тернопіль, 2024. С.118-121 [16].

2. Карпів В., Крук О., Казьмірчук Н. Імовірнісний аналіз стійкості методу розщеплення для захисту інформації. Матеріали науково-практичного симпозіуму «Захист інформації». Тернопіль, 2024. С.89-92 [17].

1 АНАЛІЗ СУЧАСНИХ ПІДХОДІВ ДЛЯ ЗАБЕЗПЕЧЕННЯ ЗАХИСТУ ІНФОРМАЦІЇ

1.1 Основні поняття і визначення

Шифрування − практичний спосіб представлення секретної інформація з метою її захисту від незаконного читання та модифікації [18-20].

В сучасних методах шифрування здійснюється математичними перетвореннями вихідного тексту за допомогою алгоритмів, в яких осмислені повідомлення передаються як беззмістовні [21-24]. Осмислене повідомлення, що є вихідним для вибраного алгоритму захисту, називається вихідним текстом. Неосмислене повідомлення, яке є результатом роботи алгоритму захисту, називається зашифрованим текстом (або захищеним текстом). Шифруючі перетворення повинні бути оберненими, щоб мати можливість відновити початкову інформацію. Процес зворотного перетворення називається відновленням.

Ключ захисту – це визначений секретний стан деяких параметрів захисту, пов'язані з перетворенням даних, який забезпечує вибір тільки однієї вибірки з усіх можливих варіантів для даного алгоритму перетворень [25].

Криптосистема або система захисту – це набір, що складається з наступних компонентів: алгоритм захисту E, алгоритм відновлення D, алгоритм генерації ключів G, множина ключів захисту K, множина ключів відновлення K', простір вихідних повідомлень M і простір захищених текстів C.

Взагалі кажучи, всі системи захисту з ключем діляться на симетричні та асиметричні системи щодо захисту і відновлення. В симетричних системах етапи захисту і відновлення здійснюються за допомогою одного і того ж ключа, як показано на рисунку 1.1. Такий ключ заздалегідь передається по окремому захищеному надійному каналу обміну ключової інформації. Симетричні системи захисту отримали в літературі також другу назву: криптосистеми з секретним ключем.

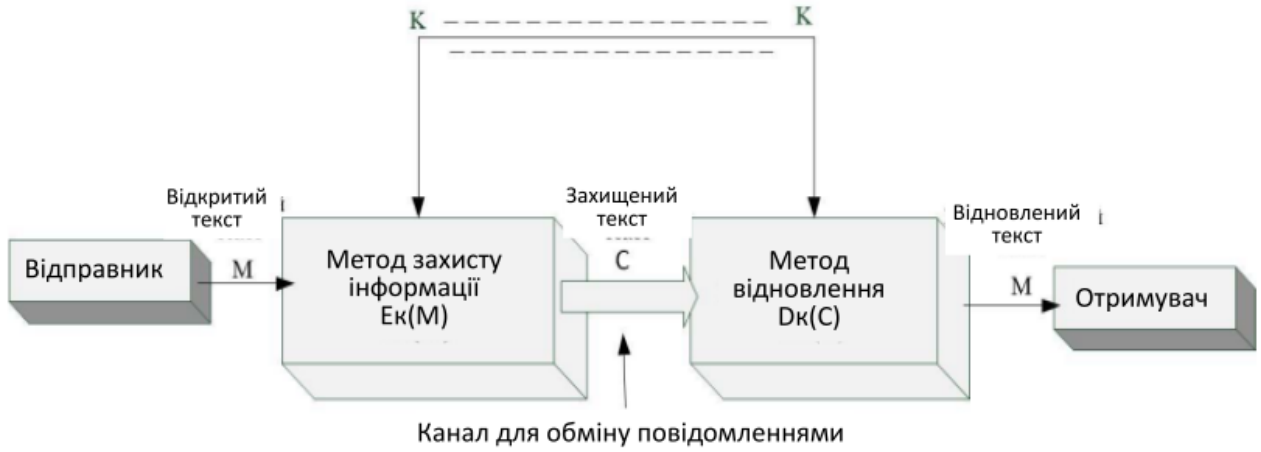


Рисунок 1.1 – Узагальнена схема симетричної системи захисту

В асиметричних системах захисту інформації етапи захисту і відновлення використовують різні ключі, як показано на рисунку 1.2. В залежності від застосування один ключ буде відкритим, тобто загальнодоступним, а другий необхідно зберігати в секреті. Завдяки умові, що один з ключів захисту не обов'язково тримати в секреті, асиметричні криптосистеми отримали в літературі також іншу назву: криптосистеми з відкритим ключем.

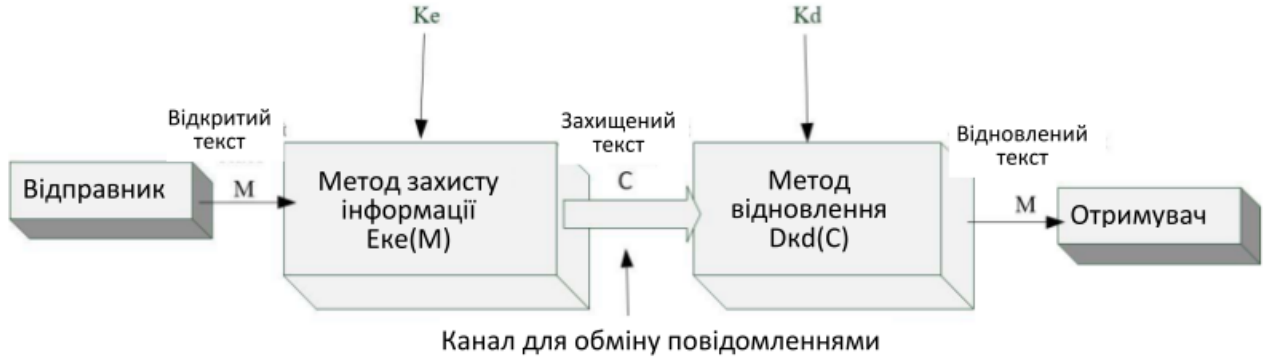


Рисунок 1.2 − Узагальнена схема асиметричної системи захисту з відкритим ключем

Згідно літературних джерел, всі відомі симетричні методи захисту, наприклад, DES, AES, Rijndael, гамування, TEA, IDEA, ГОСТ 28147-89, MARS, RC6, Serpent, twoFish тощо, і асиметричні алгоритми захисту, наприклад, RSA, Рабіна, Ель-Гамаль, Мак-Еліса тощо не використовують можливість заміни кожного символу тексту, що передається, на послідовність із *k* цілих чисел, що забезпечувало б більш високий ступінь безпеки.

1.2 Деякі методи заміни, що грунтуються на операціях модульної арифметики

Свою назву шифр Цезаря отримав по імені римського імператора Гая Юлія Цезаря, який застосував його для захисту при переписці з Ціцероном.

Метод Цезаря грунтується на підстановках: кожна буква замінюється іншою. З кінця алфавіту слідує циклічний перехід на його початок, як показано на рисунку 1.3.

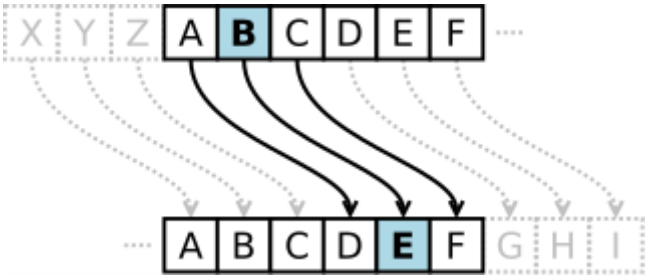


Рисунок 1.3 − Метод Цезаря

При розмірі алфавіту n правило захисту інформації Цезаря записується таким чином:

$$c=(m+k) \bmod n, \qquad (1.1)$$

де m і c − номер букви відповідно для відкритого повідомлення та захищеного тексту;

k − ключ захисту (в методі Цезаря k=3);

n − кількість букв в алфавіті.

Для відновлення захищеного тексту потрібно використовувати зворотне співвідношення:

$$m=(c-k) \bmod n, \qquad (1.2)$$

При використанні української мови n=33, що дорівнює кількості букв в алфавіті.

В афінній системі підстановок Цезаря застосовується одночасно операції множення і додавання за модулем m, афінний метод узагальнює метод Цезаря.

Перетворення для захисту в афінній системі підстановок Цезаря визначається таким чином:

$$E_k: m \rightarrow E_k(m);\ E_k(m)=(am+b) \bmod n, \tag{1.3}$$

де $k=(a, b)$ − ключ захисту;

a, b - цілі числа, $0 \le a, b < n$;

a, n - взаємно прості числа;

m − відкритий текст;

$E_k(m)$ - зашифрований текст.

Відтворення відкритого тексту виконується наступним рівнянням:

$$D_k(c)=a^{-1}(c-b) \bmod n, \tag{1.4}$$

де $k=(a^{-1}, b)$ - ключ відновлення;

a^{-1} - обернене до a число за модулем n, $aa^{-1} \bmod n = 1$;

c - зашифрований текст.

Метод Хілла є узагальненням афінної системи підстановок Цезаря. Він був винайдений у 1929 році американським математиком Лестером Хіллом.

В цьому методі кожні l букв відкритого тексту замінюються l буквами зашифрованого тесту. При l=3 метод може бути описаний наступними рівняннями:

$$\begin{aligned} c_1&=(k_{11}m_1+ k_{12}m_2+ k_{13}m_3) \bmod n,\\ c_2&=(k_{21}m_1+ k_{22}m_2+ k_{23}m_3) \bmod n,\\ c_3&=(k_{31}m_1+ k_{32}m_2+ k_{33}m_3) \bmod n, \end{aligned} \tag{1.5}$$

де m_1, m_2, m_3 – відкритий текст;

c_1, c_2, c_3 - зашифрований текст;

k_{11}, k_{12}, k_{13}, k_{21}, k_{22}, k_{23}, k_{31}, k_{32}, k_{33} - ключі захисту.

Цей метод може бути представлений у матричній формі:

$$\begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} \bmod n. \quad (1.6)$$

Або його ще можна представити у вигляді добутку векторів:

$$C=(KM) \bmod n, \quad (1.7)$$

де M - вектор-стовбець висоти l, який являє собою відкритий текст;

C - вектор-стовбець висоти l, який являє собою зашифрований текст;

K - матриця $l \times l$, яка являє собою ключі захисту.

Метод Віженера складається з багатократного застосування методу Цезаря з різними значеннями зсуву. Для відновлення і захисту може використовуватися таблиця алфавітів, яка називається таблицею Віженера або квадратом Віженера. Таблиця для відновлення українського алфавіту показано на рисунку 1.4.

Метод захисту Віженера визначається за такою формулою:

$$c_i=(m_i+k_i) \bmod n, \quad (1.8)$$

де c_i - букви зашифрованого тексту;

m_i - букви відкритого тексту;

k_i – букви ключа.

Метод відновлення Віженера можна записати таким чином:

$$m_i=(c_i-k_i) \bmod n. \quad (1.9)$$

	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я
а	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я
б	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а
в	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б
г	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в
ґ	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г
д	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ
е	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д
є	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е
ж	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є
з	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж
и	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з
і	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и
ї	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і
й	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї
к	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й
л	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к
м	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л
н	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м
о	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н
п	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о
р	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п
с	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р
т	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с
у	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т
ф	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у
х	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф
ц	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х
ч	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц
ш	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч
щ	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш
ь	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ
ю	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь
я	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю

Рисунок 1.4 − Таблиця Віженера для українського алфавіту

Всі вищеперераховані рівняння для методів захисту мають один недолік, який полягає в тому, що використані розміри модулів збігаються з розміром відповідного алфавіта. Цей недолік відсутній в запропонованому методі, де

величина модуля змінюється на кожному кроці роботи системи і не пов'язана з загальним розміром алфавіта.

Крім того, метод Цезаря і його модифікації використовують операцію поділу з остачею (ділення за модулем або ділення націло), що забезпечує однозначну заміну кожної букви якоюсь іншою буквою. До цих пор в літературі по захисту інформації не висувалось ніяких узагальнених операцій ділення націло. В даній роботі наводиться узагальнення цієї операції, при якій кожен символ замінюється на послідовність k цілих чисел (k - рівень розщеплення).

1.3 Потоковий метод захисту

Потоковий метод захисту – це симетричний метод захисту інформації, в якому кожен елемент відкритого тексту шифрується в залежності від ключа потоку і місця цього елемента в текстовому потокі.

Існує два типу потокових методів: синхронний потоковий метод (СПШ) і самосинхронізований потоковий метод (асинхронний потоковий метод (АПШ)).

У синхронному потоковому методі потік ключів формується незалежно від захищеного і вихідного тексту. Процес захисту синхронного потокового методу може бути описаний такими рівняннями:

$$\begin{aligned} \sigma_{i+1} &= f(\sigma_i, k), \\ z_i &= g(\sigma_i, k), \\ c_i &= h(z_i, m_i), \end{aligned} \qquad (1.10)$$

де σ_0 – початковий стан і може бути визначений з ключа;

k – ключ;

f − функція переходу між станами;

g − функція, яка генерує потік ключів z_i;

h − функція виведення, яка об'єднує ключовий потік і вихідний текст m_i для створення зашифрованого тексту c_i.

Схема захисту з використанням методу синхронного потоку показано на рисунку 1.5.

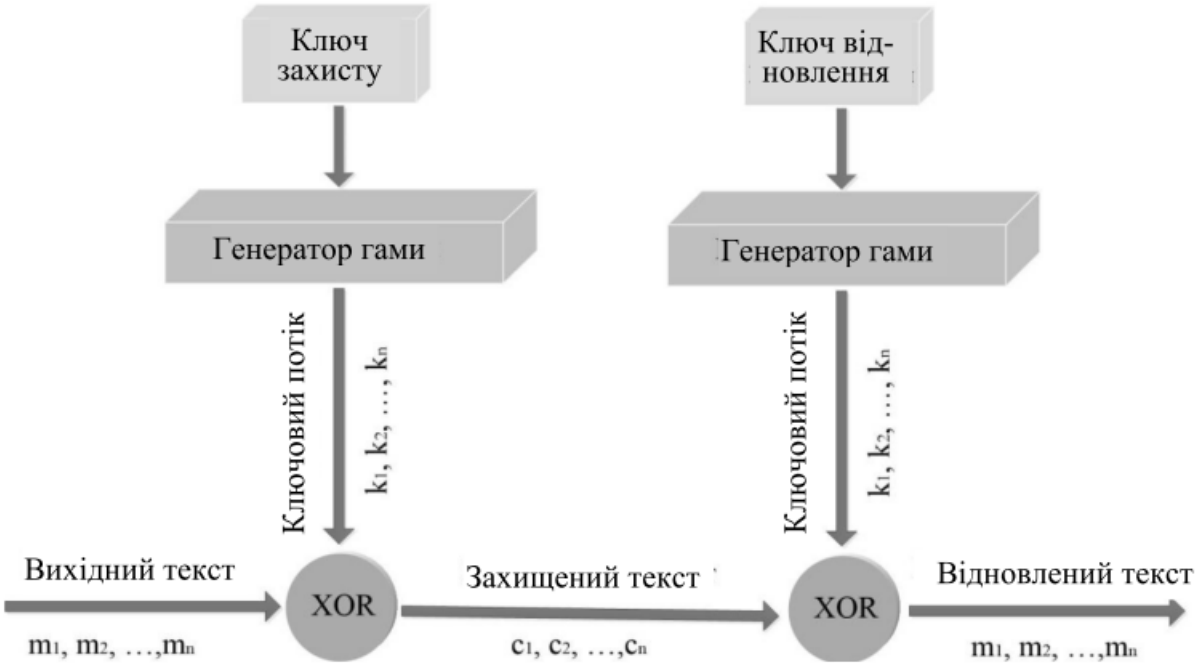


Рисунок 1.5 – Схема захисту з використанням синхронного потокового методу

У потоковому методі, що самосинхронізується, потік ключів створюється функцією ключа та фіксованим числом знаків шифротексту.

Процес захисту самосинхронізованого потокового методу може бути описаний такими рівняннями:

$$\begin{aligned} \sigma_i &= (c_{i-t}, c_{i-t+1}, \ldots, c_{i-1}), \\ z_i &= g(\sigma_i, k), \\ c_i &= h(z_i, m_i), \end{aligned} \qquad (1.11)$$

де $\sigma_0 = (c_{i-t}, c_{i-t+1}, \ldots, c_{i-1})$ - початковий стан;

h – функція виведення, яка об'єднує ключовий потік і вихідний текст m_i для створення зашифрованого тексту c_i, тобто функція захисту інформації.

Схема захисту з використанням потокового методу, що самосинхронізується, показана на рисунку 1.6.

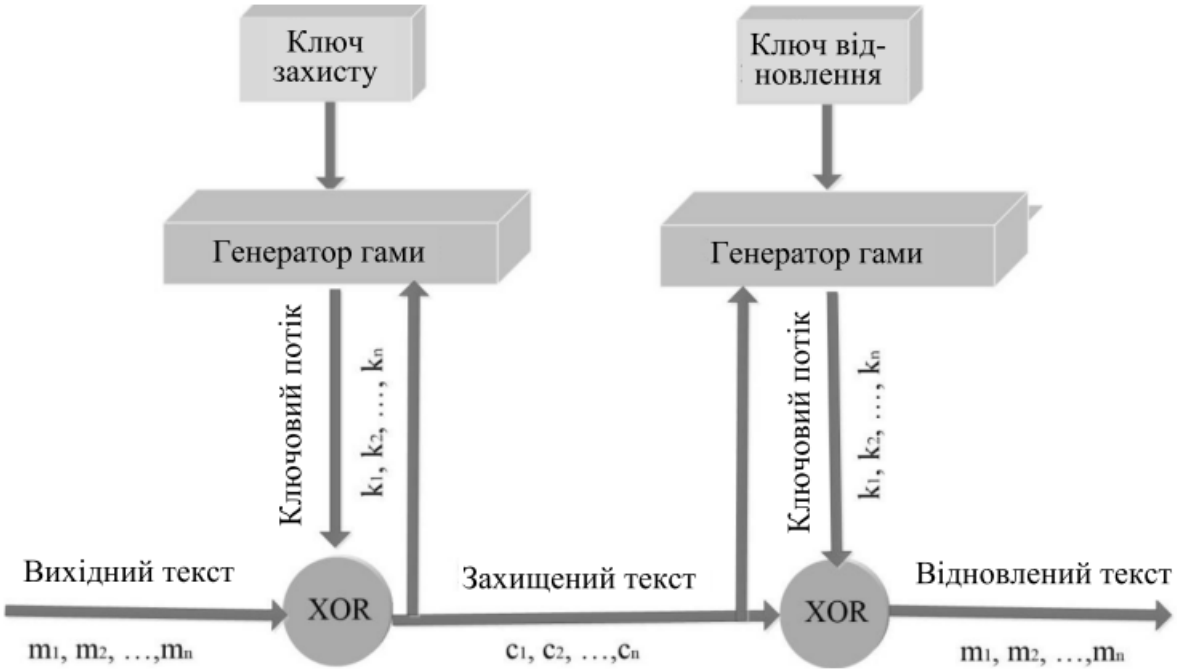


Рисунок 1.6 − Схема захисту з використанням самосинхронізованого потокового методу

Захист методом гамування та методом Вернама є двома прикладами потокового методу захисту, які мають деякі недоліки, які вдається усунути у запропонованому методі захисту інформації шляхом розщеплення.

1.4 Генератори псевдовипадкових чисел

Проблема генерації випадкових чисел на електронно-обчислювальній техніці стояла давно, але з розвитком криптографії їй стало приділятися особлива увага. Випадкові числа застосовуються при створенні сеансових та закритих ключів для асиметричних криптосистем, під час підписання документів у схемах спільного використання процедури захисту та при генерації ключів.

У багатьох системах захисту потрібні джерела (генератори) випадкових чисел, які використовуються, наприклад, для генерації ключів захисту, одноразових шифроблокнотів, початкового вектора у схемах об'єднання блоків, одноразових випадкових чисел, при хешуванні паролів у схемах цифрового підпису тощо.

Можна говорити про два різновиди генераторів, що створюють істинно випадкові числа та псевдовипадкові числа. Кожен із них має свої переваги та недоліки. ГПВЧ використовують детермінований алгоритм для їхнього створення [26].

Стандартні генератори випадкових чисел (ГВЧ) виробляють статистично випадкові послідовності чисел. Однак далеко не завжди вони відповідають потребам криптографів, оскільки система захисту, заснована на таких ГВЧ, може виявитися нестійкою.

В ідеалі ГВЧ мають бути засновані на деякій кількості істинно випадкових фізичних процесів або вимірах фізичних властивостей об'єктів. Наприклад, можна використовувати шуми, радіоактивний розпад у різних електронних приладах, вимірювання температури, квантових флуктуацій вакууму в якості джерела ентропії для ГВЧ, фотоелектричний ефект, вимірювання реакції користувача, наприклад, рух мишки тощо.

Практичні прилади, засновані на подібних фізичних процесах, можуть бути неприйнятними через низьку швидкість видачі результатів, значних габаритів, схильності до зовнішніх впливів або їх високої собівартості. Основні недоліки ГВЧ полягають у тому, що вони є повільними, неефективними, дорогими, і послідовність створюваних чисел не може бути зроблена відтворюваною. Тому в багатьох додатках використовуються програмні псевдовипадкові генератори, що виробляють детерміновані послідовності чисел.

Методи отримання псевдовипадкових чисел докладно описані в літературі, серед яких найбільш популярні такі:

− метод Фібоначчі із запізнюваннями. Послідовності Фібоначчі можуть бути описані відповідно до наступної ітеративної формули:

$$X_{n+1}=(X_n+X_{n+1}) \bmod m, \qquad (1.12)$$

де m – параметр модуля;

X_0 – початкове значення;

– лінійний конгруентний метод. Генератор визначається таким рекурентним співвідношенням:

$$X_{n+1}=(aX_n+c) \bmod m, \tag{1.13}$$

де X_n - послідовність псевдовипадкових чисел;

a – множник;

c – приріст;

X_0 – початкове значення;

– квадратичний конгруентний метод. Генератор визначається таким рекурентним співвідношенням:

$$X_{n+1}=(aX_n^2+ bX_n +c) \bmod m, \tag{1.14}$$

де X_n - послідовність псевдовипадкових чисел;

a, b, c – константи;

- алгоритм Блюм-Блюма-Шуба визначається такою формулою:

$$X_{n+1}=X_n^2 \bmod m. \tag{1.15}$$

Будь-який ГПВЧ рано чи пізно «зациклюється», тобто, зрештою, числа утворюють цикл, який повторюється нескінченне число разів. Цикл, який повторюється, називається періодом. Довжина періоду у послідовності має різні значення.

Прикладами відомих криптостійких ГПВЧ є ISAAC, RC4, Snow, SEAL. Перевагами даних ГПВЧ є ефективність, швидкодія, відтворюваність. Вони практично не вимагають ресурсів пам'яті та компактні.

Недоліком є передбачуваність ПВЧ, заснована на знанні попередніх послідовностей, наявність періодичності та невипадковості, оскільки між значеннями є залежність.

Все це призводить до необхідності створення методу чи генератора, який дозволить об'єднати переваги обох типів генераторів ГПВЧ та ГВЧ для підвищення рівня надійності. Це означає, що потрібно створити генератор, який має такі властивості: він ефективний, відтворюваний, швидкодіючий, забезпечує неперіодичність та випадковість, недорогий та гарантуючий непередбачуваність випадкових чисел. Цього можна досягнути за допомогою так званого детермінованого генетичного алгоритму (ДГА) [27-30].

ДГА є евристичним алгоритмом, що призначений для покращення ймовірнісних характеристик деяких ГПВЧ. Він застосовувався в експериментах при генерації випадкових чисел для гам, що використовуються в методі захисту шляхом використання символьного розщеплення.

Результати можна отримати на основі статистичних тестів, які включають частотний тест, критерій серій, коефіцієнт кореляції Пірсона та обчислення ентропії для трьох описаних вище типів ГПВЧ. Порівнянню піддаються послідовності, створені кожним з ГПВЧ без будь-якої модифікації, з тією ж послідовністю, до членів якої застосовуються оператори ДГА. Тести можна здійснювати з за допомогою трьох пакетів програм: МАТЛАБ, Minitab та IBM SPSS Statistics.

Статистичне дослідження показало, що ДГА покращує якість випадковості чисел, згенерованих деякими традиційними ГПВЧ відповідно до вибраних критеріїв.

1.5 Захист методом гамування

Принцип методу гамування полягає у генерації гами за допомогою датчика ПВЧ та накладення отриманої гами на відкриті дані оборотним чином (використовуючи додавання за модулем 2). Накладання гами можна здійснити кількома способами, наприклад, за допомогою такої формули:

$$t_ш = t_о \oplus t_г, \tag{1.16}$$

де $t_ш$, $t_о$,$t_г$ – ASCII коди відповідно зашифрованого символу, вихідного символу та гами;

⊕ - побітова операція "виключаюче АБО".

Відновлення тексту відбувається аналогічно:

$$t_о=t_ш⊕t_г. \quad (1.17)$$

Схема захисту шляхом гамування показана на рисунку 1.7.

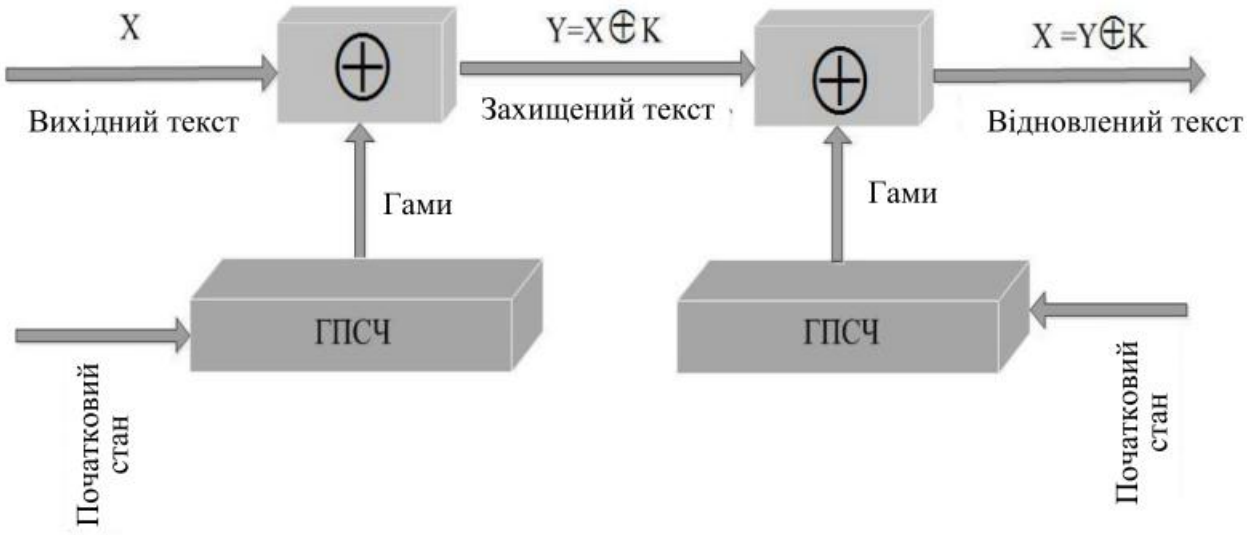


Рисунок 1.7 − Схема захисту методом гамування

У цій схемі обидва ГПВЧ повинні бути абсолютно однаковими, і довжина псевдовипадкової послідовності (гами) має дорівнювати довжині вихідних даних.

Прикладом системи гамування є A5 - потоковий алгоритм захисту, який використовується в європейській системі мобільного цифрового зв'язку GSM (Group Special Mobile) для захисту телефонних сеансів. Тобто алгоритм A5 використовується для захисту інформації між базовою станцією та абонентом для забезпечення конфіденційності переданих даних між ними. В алгоритмі захисту необхідно створювати однакові гами у відправника та одержувача, що

не може бути досягнуто за допомогою ГВЧ, оскільки він не забезпечує відтворюваності, тому при захисті методом гамування цей недолік долається під час використання ГПВЧ. Але ГПВЧ має період, і існує передбачуваність випадкових чисел, оскільки між ними є залежність. Тому при захисті методом гамування виникають ті ж самі проблеми.

1.6 Захист методом Вернама

Метод симетричного захисту був винайдений співробітником AT&T Гілбертом Вернамом у 1917 році. Метод захисту Вернама вважається прикладом системи з абсолютною стійкістю, тобто, якщо вона використовується правильно, то не може бути зламаною.

Метод захисту Вернама може бути описаний такими рівняннями:

$$Y=X\oplus K=(x_1\oplus k_1, x_2\oplus k_2, \ldots, x_l\oplus k_l,) - \text{метод захисту Вернама,} \quad (1.18)$$

$$X=Y\oplus K=(y_1\oplus k_1, y_2\oplus k_2, \ldots, y_l\oplus k_l,) - \text{метод відновлення Вернама,} \quad (1.19)$$

де $X=x_1, x_2, \ldots, x_l$ – двійкове вихідне повідомлення довжини l

$K=k_1, k_2, \ldots, k_l$ – двійковий ключовий рядок тієї ж довжини l;

$Y=y_1, y_2, \ldots, y_l$ – зашифрований текст.

Схема захисту методом Вернама показано рисунку 1.8.

Метод Вернама має серйозну проблему, пов'язану з необхідністю доставки одержувачу такого ж ряду гам, як у відправника, або з потребою безпечного зберігання ідентичного ряду гам у відправника та одержувача. Ці недоліки методу захисту Вернама не властиві методу розщеплення.

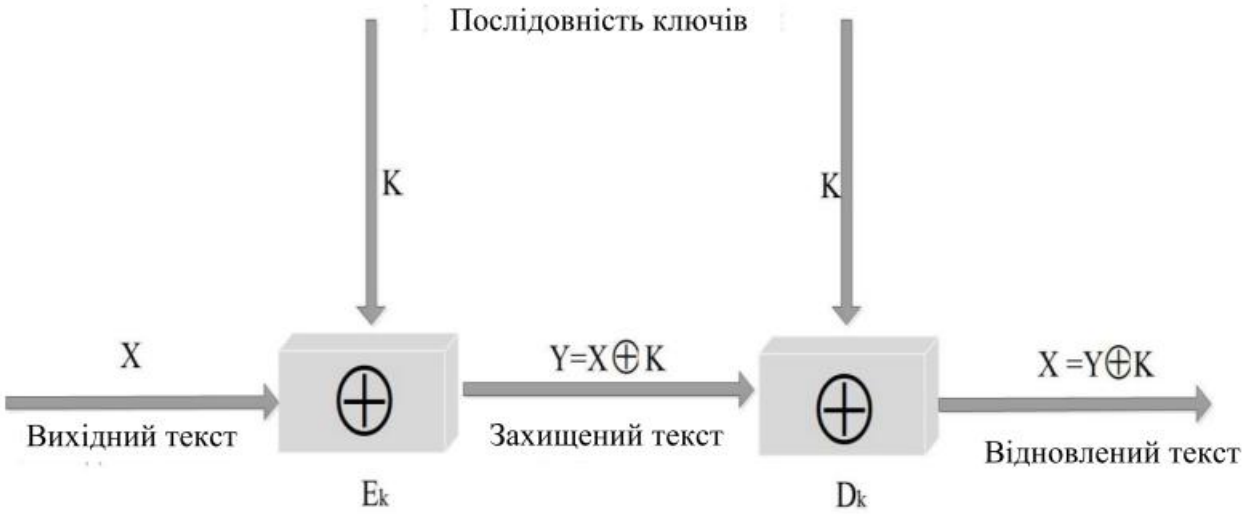


Рисунок 1.8 − Схема захисту та відновлення повідомлень за методом Вернама

1.7 Абсолютно стійкі методи захисту

За Шенноном досконала стійкість шифру означає, що захищений текст *C* та відкритий текст *M* статистично незалежні, тобто для всіх можливих значень *M*, *C* справедливі такі ймовірності:

$$\Pr(M/C)=\Pr(M). \tag{1.20}$$

Наведена вище формула дозволяє зробити висновок, що отримання захищеного тексту *C* не дає криптоаналітику жодної інформації про відкритий текст *M*.

У термінах ентропії визначення досконалої стійкості шифру можна уявити такою рівністю:

$$H(M/C)=H(M). \tag{1.21}$$

Необхідною та достатньою умовою для досконалої стійкості шифру є те, що всім *M* і кожного *C* виконується така рівність:

$$\Pr(C/M)=\Pr(C). \tag{1.22}$$

Це означає, що за умови зашифрування тексту *M* ймовірності отримання конкретного захищеного тексту *C* одинакові всім *M*.

Прикладом стійкого методу є метод захисту Вернама.

Досконалий секретний метод захисту має такі недоліки:

− абсолютно стійкі методи захисту надають криптоаналітику інформацію максимально можливої довжини вихідного повідомлення;

− кожна гама використовується лише один раз;

− практично дуже дорогий за необхідними ресурсами, оскільки виникає проблема необхідності зберігання гам.

Всі перераховані вище наведені недоліки стійкого методу захисту невластиві при захисті інформації методом розщеплення.

2 МАТЕМАТИЧНІ МОДЕЛІ СИМВОЛЬНОГО РОЗЩЕПЛЕННЯ ТА ЇХ ЗАСТОСУВАННЯ ПРИ ЗАХИСТІ ТЕКСТОВОЇ ІНФОРМАЦІЇ

2.1 Метод захисту інформації за допомогою символьного розщеплення

Розщеплення при потоковій передачі означає заміну кожного символу в послідовності на ланцюжок чисел. Розщеплення забезпечує глибокий захист інформації, що передається від дій різного роду зловмисників.

Розщеплення передбачає, що кожному вихідному символу відповідає k чисел, де k – рівень розщеплення. Більш того, одержуваний захищений текст досить важкий для розкриття, оскільки в гамі захисту база r є змінною і той самий символ буде представлений при розщепленні щоразу різними поєднаннями. Ця властивість у галузі захисту інформації є новою.

Основою більшості механізмів захисту інформації є шифрування даних. Шифрування – це перетворення даних на нечитабельну форму за допомогою ключів.

У процесі симетричного захисту інформації відправник генерує відкритий текст вихідного повідомлення M, яке має бути передане законному одержувачу незахищеним каналом. Припустимо, що за каналом стежить зловмисник з метою перехопити і розкрити повідомлення, що передається.

Для того, щоб несанкціонований користувач не міг дізнатися зміст повідомлення M, відправник шифрує це повідомлення за допомогою оборотного перетворення до E_k і отримує захищений текст $C=E_k(M)$, який відправляє одержувачу.

Отримувач, прийнявши захищений текст C, читає його за допомогою зворотного перетворення $D=E_k^{-1}$ і отримує вихідне повідомлення M:

$$D_k(C)=E_k^{-1}(E_k(M))=M. \quad (2.1)$$

У схемі симетричної системи захисту з одним ключем, коли однакові ключі використовуються в блоці захисту та в блоці відновлення тексту, це означає, що

будь-хто, хто має доступ до ключа захисту, може розкрити повідомлення. Іншими словами, ключ захисту повинен бути доступний лише тому, кому призначено повідомлення.

Якщо одержувачу відомий ключ захисту, то розщеплення довільного символу S є ін'єктивним і оборотним, що відкриває можливість однозначного відновлення цього символу на приймальному кінці.

Нехай база r перевищує максимальне значення кодів у вибраній кодовій таблиці символів. Тоді розщепленням рівня k для символу S з кодом a відповідно до зазначеної кодової таблиці і r називається представлення S у вигляді ряду відповідних цілих чисел $\delta^{(2)}, \delta^{(3)}, \delta^{(4)}, \ldots, \delta^{(k-1)}, \delta^{(k)}, q^{(k)}$. Тут числа $\delta^{(i)}$ обчислюються за такою формулою:

$$\delta^{(i)}=r \bmod q^{(i-1)}, \text{ де } q^{(i-1)} = \left\lfloor \frac{r}{q^{(i-2)}} \right\rfloor, i=2, 3, \ldots, k. \qquad (2.2)$$

Назвемо отриманий таким чином ряд цілих чисел $\delta^{(2)}, \delta^{(3)}, \delta^{(4)}, \ldots, \delta^{(k-1)}, \delta^{(k)}, q^{(k)}$ результатом розщеплення.

2.2 Математичні моделі захисту інформації на основі символьного розщеплення

Захист тексту та його відновлення відбуваються за допомогою ГПВЧ, який вважається відомим і на приймальному, і на передаючому кінці. Зазвичай мається на увазі, що ГПВЧ створює непередбачувану послідовність псевдовипадкових чисел для зовнішнього спостерігача.

Від ГПВЧ надходить величина r_i, необхідна як під час передачі з використанням розщеплення, так і при відновленні кожного символу. В результаті розщеплення в момент t створюються цілі числа $\delta^{(2)}, \delta^{(3)}, \delta^{(4)}, \ldots, \delta^{(k-1)}, \delta^{(k)}, q^{(k)}$. Передбачається, що величина $r_i>0$ перевершує максимальне значення символів за обраною кодовою таблицею.

У нашій моделі потім надходять випадкові величини r_{i+1}, r_{i+2}, r_{i+3}, …, r_{i+k}, які використовуються для додаткового захисту кожного компонента розщеплення цього символу шляхом гамування.

Тоді модель кроку захисту символу $S(t)$ з кодом $a(t)$ приводить до такого результату:

$$Y = \begin{cases} r_i \oplus a, & \text{при} \quad k = 1; \\ \delta^{(2)}, \delta^{(3)}, \delta^{(4)}, ..., \delta^{(k-1)}, \delta^{(k)}, q^{(k)}, & \text{при} \quad k > 1. \end{cases} \tag{2.3}$$

При гамуванні отримується такий результат захисту:

$$\begin{cases} \delta^{(j)} \oplus r_{i+j-1}, & \text{де} \quad j = 2,3,...,k; \quad \text{при}\, k > 1; \\ q^{(k)} \oplus r_{i+k}. \end{cases} \tag{2.4}$$

Модель кроку відновлення символу після гамування дає такий результат:

$$\begin{cases} \delta^{(j)} \oplus r_{i+j-1}, & \text{де} \quad j = 2,3,...,k; \quad \text{при}\, k > 1; \\ q^{(k)} \oplus r_{i+k}. \end{cases} \tag{2.5}$$

Тоді результат відновлення розщепленого символу:

$$\begin{cases} r_i \oplus Y, & \text{при} \quad k = 1; \\ \dfrac{\left(r_i - \delta^{(j)}\right)}{q^{(j)}}, & \text{де} \quad j = k, k-1, ..., 3, 2; \quad \text{при}\, k > 1. \end{cases} \tag{2.6}$$

Звідси процедура узагальненого розщеплення, при якій на кожному кроці формування величин використовується нова величина $r_i > q^{(i)}$, матиме такий вигляд:

$$\begin{cases} \delta^{(2)}, \delta^{(3)}, \delta^{(4)}, ..., \delta^{(k-1)}, \delta^{(k)}, q^{(k)}, \text{ де} \\ \delta^{(2)} = r_1 \bmod a, \; q^{(2)} = \left\lfloor \dfrac{r_1}{a} \right\rfloor; \\ \delta^{(3)} = r_2 \bmod q^{(2)}, \; q^{(3)} = \left\lfloor \dfrac{r_2}{q^{(2)}} \right\rfloor; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \delta^{(k)} = r_{k-1} \bmod q^{(k-1)}, \; q^{(k-1)} = \left\lfloor \dfrac{r_{k-2}}{q^{(k-2)}} \right\rfloor; \\ q^{(k)} = \left\lfloor \dfrac{r_{k-1}}{q^{(k-1)}} \right\rfloor. \end{cases} \tag{2.7}$$

Правило відновлення вихідного символу під час використання методу узагальненого розщеплення та відомому векторі значень $\vec{r}$ полягає в послідовному відновлення величин, що входять до (2.7), у зворотному порядку в порівнянні з їх надходженням на приймальний вузол. Результат відновлення узагальненого розщепленого символу матиме такий вигляд:

$$\begin{cases} \dfrac{\left(r_i - \delta^{(j)}\right)}{q^{(j)}}, \quad \text{де} \quad j = k, k-1, ..., 3, 2; \quad \text{при } k > 1; \\ \qquad\qquad\qquad i = k-1, k-2, ..., 2, 1 \quad \text{при } k > 1. \end{cases} \tag{2.8}$$

2.3 Блок-схема методу захисту на основі символьного розщеплення

Узагальнена схема симетричного методу захисту на основі символьного розщеплення показано рисунку 2.1.

На рисунках 2.2 і 2.3 показано блок-схему методу розщеплення при k=1 та $k \geq 2$.

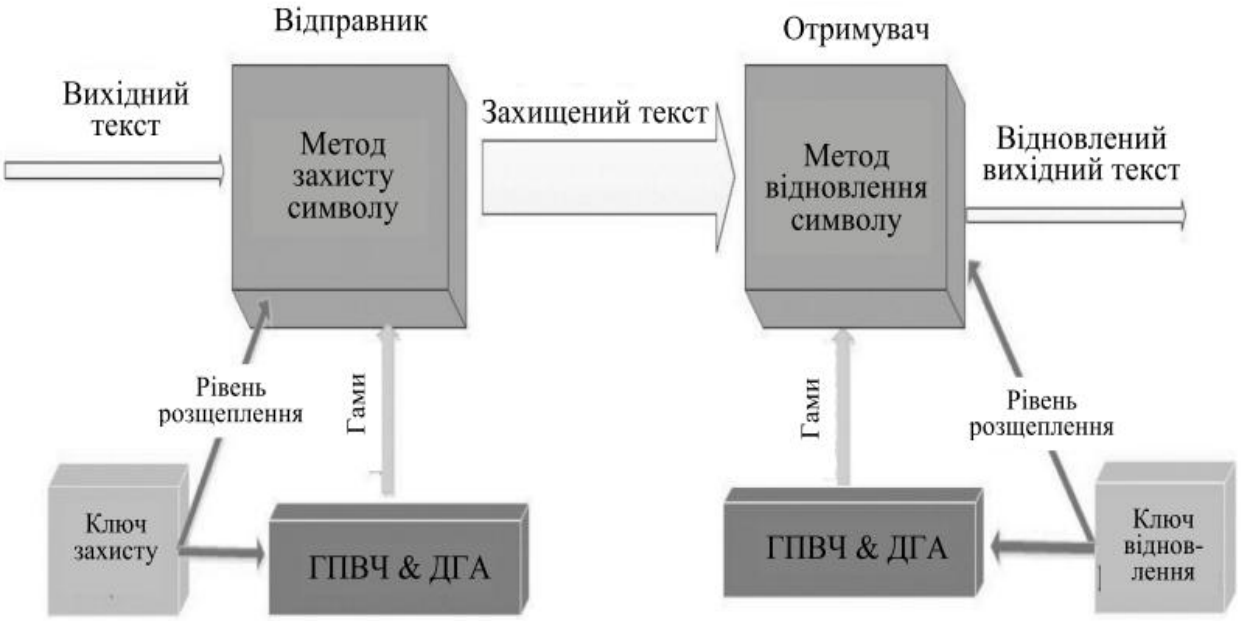


Рисунок 2.1 − Узагальнена схема методу захисту на основі символьного розщеплення

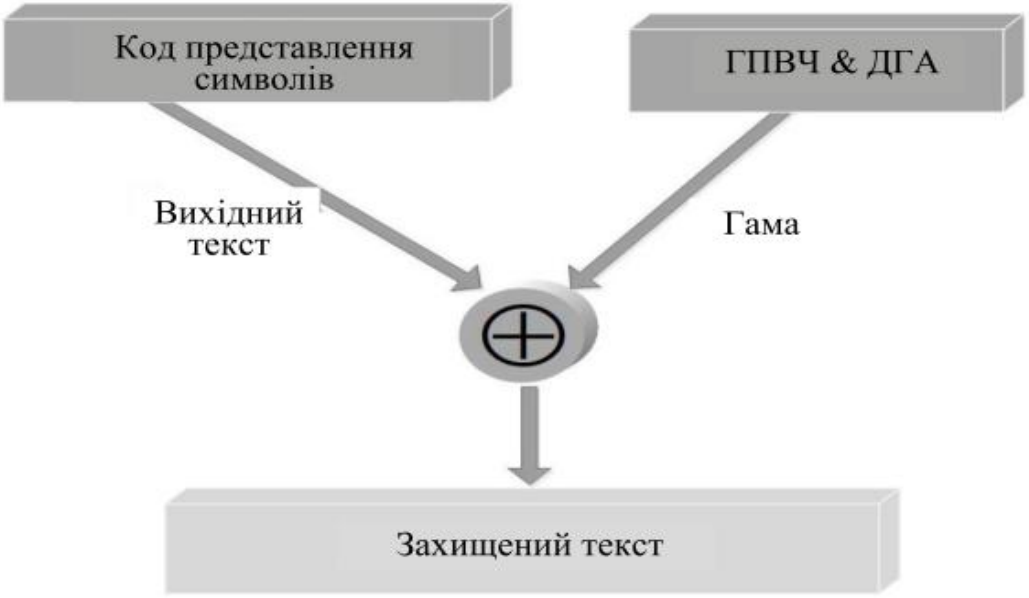


Рисунок 2.2 − Блок-схема методу розщеплення при k=1

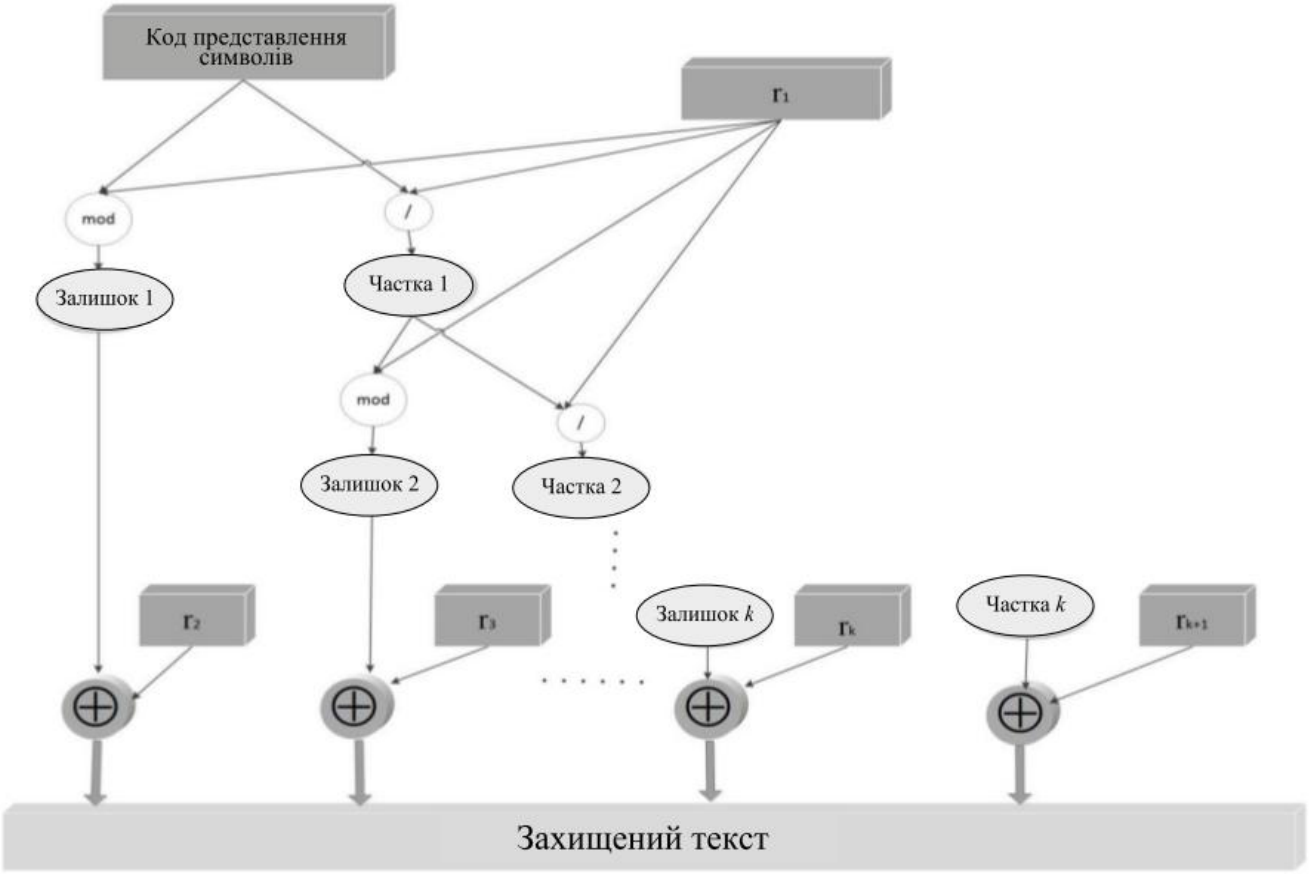


Рисунок 2.3 – Блок-схема методу розщеплення при $k \geq 2$

2.4 Генерація ключа захисту

В симетричних алгоритмах захисту із символьним розщепленням для захисту та відновлення повідомлення використовується один і той самий блок інформації (тобто ключ).

Ключ захисту містить інформацію про детермінований генетичний алгоритм, параметри ГПВЧ та рівні розщеплення. Параметри детермінованого генетичного алгоритму та ГПВЧ використовуються на етапі створення бази r у запропонованій процедурі розщеплення. Блок-схема ключа захисту показана на рисунку 2.4.

Ключ захисту містить багато параметрів, які забезпечують стійкість алгоритму та ускладнюють аналіз тексту сторонній особі. Зокрема, можуть бути такі параметри:

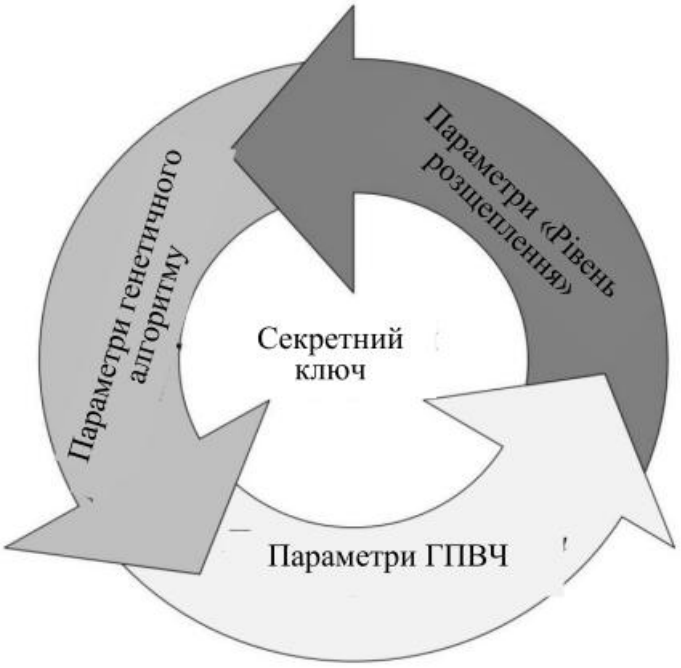


Рисунок 2.4 – Блок-схема ключа захисту

1) параметр “Рівень розщеплення символу” вказує кількість символів захищеного тексту, які отримуються під час розщеплення символу;

2) параметри детермінованого генетичного алгоритму, які включають “розмір генерації, кількість поколінь, довжину хромосоми, початкове та кінцеве значення набору;

3) параметри вибраного ГПВЧ:

а) у випадку генераторів Блюм-Блюма-Шуба та Фібоначчі параметри включають початкове значення та модуль;

б) у разі лінійного конгруентного генератора параметри включають початкове значення, модуль, множник та збільшення;

в) у разі квадратичного конгруентного генератора параметри включають “початкове значення, модуль, a, b та c.

Пункти 2) та 3) використовуються для створення базового r шляхом використання детермінованого генетичного алгоритму для підвищення рівня захисту.

Використання еволюційного генетичного алгоритму для захисту інформації є порівняно новим підходом у цій галузі, тому стандартна класифікація відсутня.

Застосування еволюційного генетичного алгоритму у системах захисту інформації доцільно класифікувати за трьома категоріями відповідно з цілями та завданнями:

1) еволюція з урахуванням генетичного алгоритму використовується для генерації ключів (відкритий та закритий ключі, ключ гами);

2) еволюція на основі генетичного алгоритму використовується для створення нових процесів захисту інформації;

3) еволюція на основі генетичного алгоритму використовується для покращення стандартного алгоритму захисту.

Процедура розщеплення буде створювати покоління чисел на основі арифметичної операції ділення з залишком. Це призводить до ще однієї концепції еволюції, заснованої на процедурі розщеплення, що дозволяє скоротити час виконання процесу еволюції в порівнянні з традиційними методами, заснованими на генетичному алгоритмі.

2.5 Етапи роботи алгоритмів захисту та відновлення простого розщеплення символу

Вхід: вихідний текст, тип генератора та ключ захисту. Якщо рівень розщеплення дорівнює одиниці, то потрібно:

1) згенерувати послідовність гами, позначити цю послідовність символом R_1 на основі обраного ГПВЧ, ключа захисту та генетичних операторів алгоритму;

2) конвертувати кожен символ вихідного тексту на його значення ASCII-коду, щоб сформувати послідовність цілих чисел. Позначимо цю послідовність символом S_{ASCII};

3) застосувати операцію XOR між частиною послідовності від генератора та послідовністю, представленою ASCII-кодом S_{ASCII}, яка отримана на кроці 2, щоб отримати нову послідовність, яка позначається символом $C=R_1 \oplus S_{ASCII}$. Послідовність C є захищеним текстом.

У випадку рівня розщеплення $k>1$ потрібно:

1) згенерувати послідовність гами, позначивши її символом R, на основі секретного ключа, обраного ГПВЧ та генетичних операторів алгоритму;

2) вибрати з R числа, які більші, ніж 256 (тобто максимальне значення символів за обраною кодовою таблицею ASCII), позначивши цю послідовність гам символом $r=\{r_1, r_2, \ldots, r_L\}$;

3) для кожного символу вихідного тексту виконати розщеплення k-1 разів та зберегти залишок ділення на кожному кроці, після цього зберегти останній результат, тобто частку. Позначити цю послідовність символом $P=\{p_1, p_2, \ldots, p_k, \ldots\}$;

4) захищений текст буде являти собою послідовність такого виду: $C=\{r_2 \oplus p_1, r_3 \oplus p_2, \ldots, r_{k+1} \oplus p_k, \ldots\}$.

Для відновлення числа на основі простого розщеплення використовується такий алгоритм.

Вхід: захищений текст $C=\{c_1, c_2, \ldots, c_k, \ldots\}$, тип генератора та ключ захисту. Якщо розщеплення рівнів дорівнює 1, то потрібно:

1) згенерувати послідовність гами, позначити цю послідовність символом R_1 на основі обраного ГПВЧ, ключа захисту та генетичних операторів алгоритму;

2) застосувати операцію XOR між частиною послідовності R_1, яка отримана на кроці 1, і кожним символом захищеного тексту C, щоб створити послідовність, яку позначимо символом, де $S_{ASCII}=R_1 \oplus C$;

3) конвертувати послідовність S_{ASCII} в значення відповідного символу. Позначимо цю послідовність символом S_{char}. Це буде відновлений вихідний текст.

Якщо є k>1 рівнів розщеплення, то необхідно:

1) згенерувати послідовність гами R на основі вибраного ГПВЧ, ключа захисту та генетичних операторів алгоритму;

2) вибрати гами із згенерованої послідовності R, які більші, ніж 256. Позначимо цю послідовність гам символом $r=\{r_1, r_2, \ldots, r_L\}$;

3) якщо позначити операцію XOR символом ⊕, а захищений текст літерою (C), то послідовність $S_{результат}$ буде мати вигляд: $S_{результат}=\{r_2 \oplus c_1,\ r_3 \oplus c_2, \ldots, r_{k+1} \oplus c_k, \ldots\}$;

4) для кожного k числа послідовності $S_{результат}$ слід відняти гаму з залишку, розділити результат на частку і далі повторити ці операції k-1 раз, після цього зберегти останній результат у послідовності S_{ASCII}. Далі конвертувати послідовність S_{ASCII} в значення відповідного символу. Позначити цю послідовність через S_{char}, яка буде вихідним текстом і представлятиме собою послідовність, піддану раніше захисту.

2.6 Числові приклади застосування методу захисту на основі простого символьного розщеплення

Приклад 1. Нехай рівень розщеплення символу дорівнює одиниці, тобто k=1.

Вхід: вихідний текст та ключ захисту.

Вихідний текст: Genetic Encryption

Нехай ключ захисту має такі параметри (для генератора Фібоначчі): модуль = 1000, початкове значення = 4, розмір генерації = 100, кількість поколінь = 5, довжина хромосоми = 12, початкове значення набору = 50, кінцеве значення = 150, рівень розщеплення = 1.

Етапи захисту:

1) згенерувати гаму R_0 = {... ,371, 618, 989, 607, 596,203,799,2,801, 803, 604, 407, 11, 418, 429, 847, 276, 123, ... };

2) замінити кожен символ вихідного тексту на його значення ASCII коду S_{ASCII} (таблиця 2.1).

3) застосувати операцію XOR (⊕) до R_0 та S_{ASCII}, щоб побудувати C={308, 527, 947, 570, 544,162,892,34,868, 845, 575, 485, 114, 466, 473, 806, 379, 21}.

Таблиця 2.1 – Заміна тексту Genetic Encryption ASCII кодами

Символ	ASCII код	Символ	ASCII код	Символ	ASCII код
G	71	пробіл	32	t	116
e	101	E	69	i	105
n	110	n	110	o	111
e	101	c	99	n	110
t	116	r	114		
i	105	y	121		
c	99	p	112		

Приклад 2: кількість рівнів розщеплення символу k >1.

Початковий текст: Encryption.

Нехай ключ захисту має такі параметри (для генератора Фібоначчі): модуль = 1000, початкове значення = 4, розмір генерації = 100, кількість поколінь = 5, довжина хромосоми = 9, початкове значення набору = 50, кінцеве значення = 150, рівень розщеплення = 3.

Етапи захисту:

1) вибрати гаму, члени якої більші, ніж 256, із згенерованої послідовності, наприклад R={... ,470, 301, 293, 429,315,371, 341, 418, 421, 408, 380, 332, 404, …};

2) замінити кожен символ вихідного тексту на значення його ASCII-коду (таблиця 2.2);

Таблиця 2.2 – Заміна тексту Encryption ASCII кодами

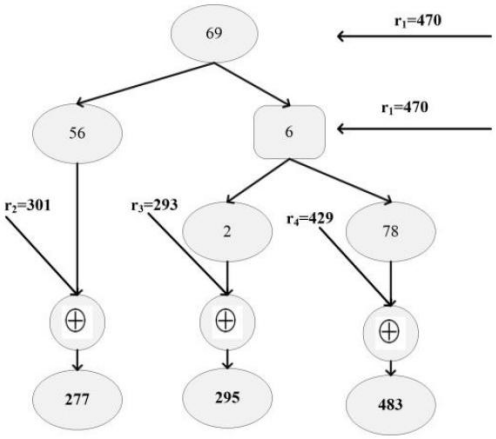
Символ	ASCII код	Символ	ASCII код
E	69	p	112
n	110	t	116
c	99	i	105
r	114	o	111
y	121	n	110

3) для кожного символу вихідного тексту виконати розщеплення *k*-1=2 рази;

ASCII код (E) = 69

Гама = 470301293429

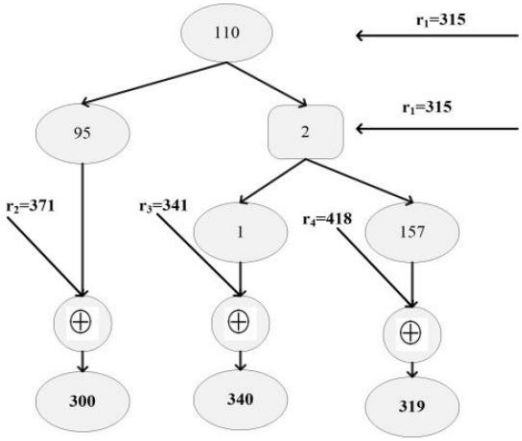
Результат розщеплення символу = =277295483



ASCII код (n) = 110

Гама = 315371341418

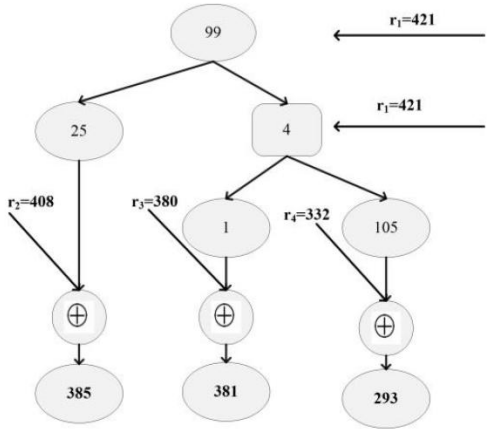
Результат розщеплення символу = 300340319



ASCII код (c) = 99

Гама = 421408380332

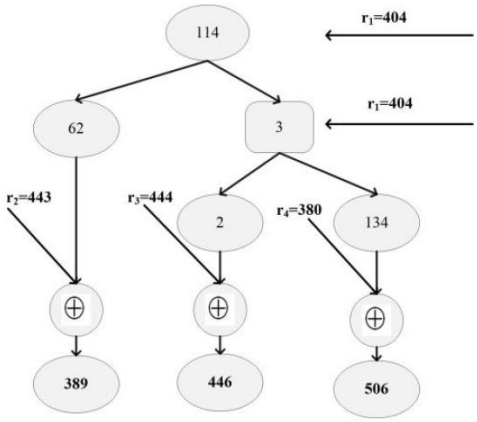
Результат розщеплення символу = =385381293



ASCII код (r) = 114

Гама = 404443444380

Результат розщеплення символу = 389446506



ASCII код (y) = 121

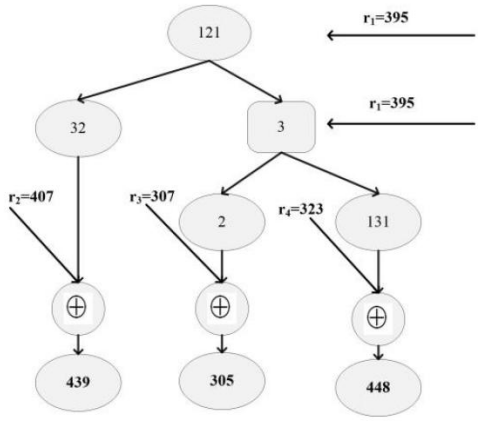
Гама =395407307323

Результат розщеплення символу = 439305448

ASCII код (p) = 112

Гама =374425396428

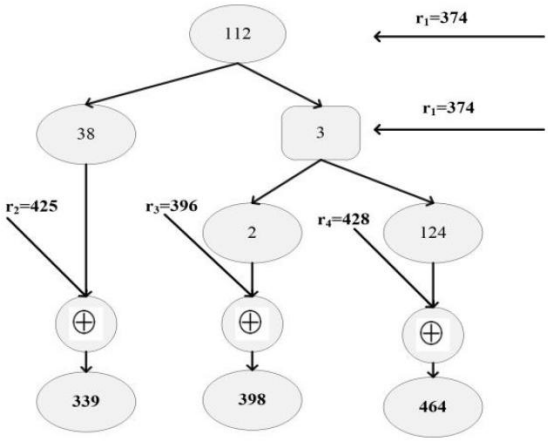
Результат розщеплення символу = 339398464



ASCII код (t) = 116

Гама = 379427498351

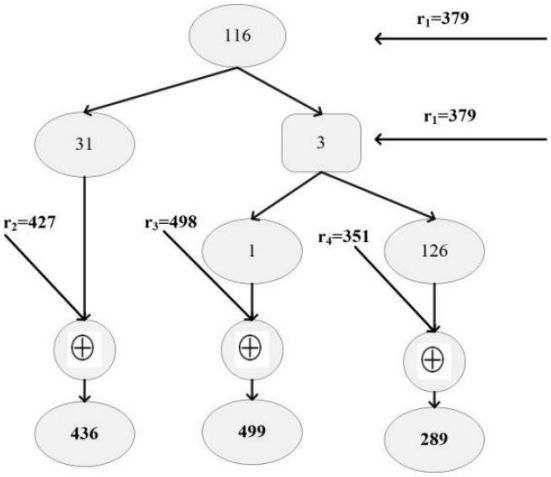
Результат розщеплення символу = 436499289



ASCII код (i) = 105

Гама = 473349447349

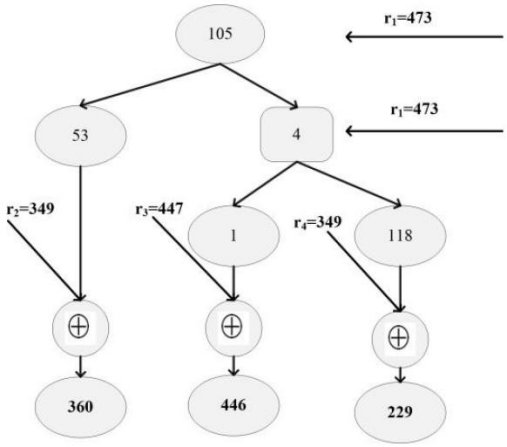
Результат розщеплення символу = 360446229



ASCII код (o) = 111

Гама = 415509423421

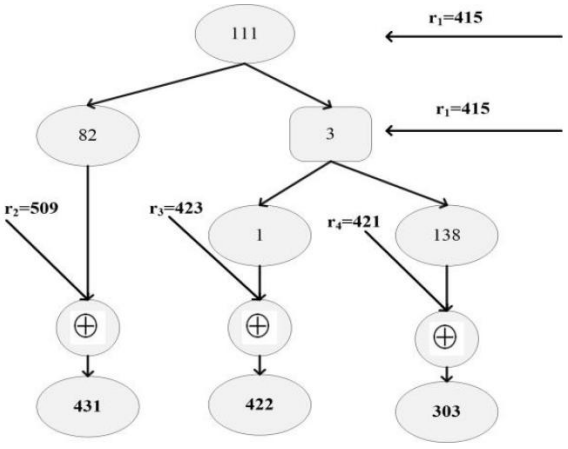
Результат розщеплення символу = 431422303

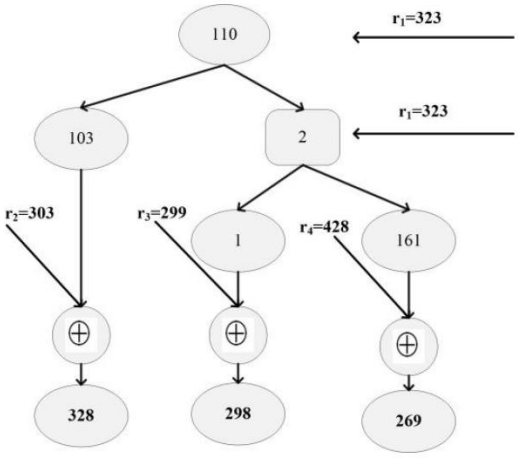


ASCII код (n) = 110

Гама = 323303299428

Результат розщеплення символу = 328298269





4. Захищений текст виглядатиме так: 277 295 483 300 340 319 385 381 293 389 446 506 439 305 448 339 398 464 436 499 289 360 446 229 431 422 303 328 298 269.

2.7 Чисельні приклади застосування методу захисту на основі узагальненого символьного розщеплення

Етапи захисту:

1) вибрати гаму, члени якої більші, ніж 256, із згенерованої послідовності R ={470 301 293 429 315 371 341 418 421 408 380 332 404 443 444 380 395 407 307 323 ... };

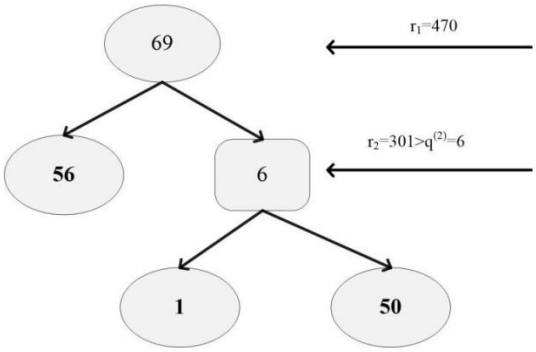
2) замінити кожен символ вихідного тексту на відповідне значення ASCII коду;

3) для кожного символу вихідного тексту виконати правило узагальненого розщеплення k-1=2 рази. На відміну від простого розщеплення, на цьому етапі необхідно виконати додаткові кроки, щоб перевірити можливості використання значення гами r_i, якщо воно задовольняє умові $r_i > q^{(i)}$, де i=2, 3, …, k-1, або пропустити це значення до наступного значення, якщо гама не задовольняє цю умову;

ASCII код (E) = 69

Гама = 470 301

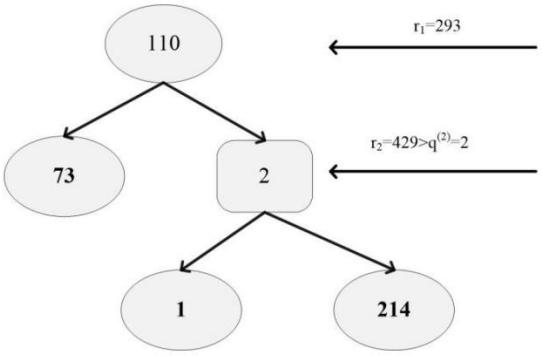
Результат узагальненого розщеплення символу = 56 1 50



ASCII код (n) = 110

Гама = 293429

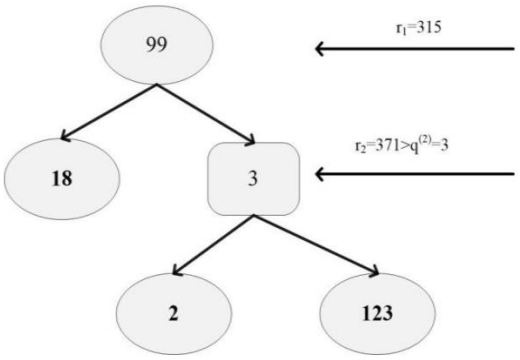
Результат узагальненого розщеплення символу = 73 1 214



ASCII код (c) = 99

Гама = 315 371

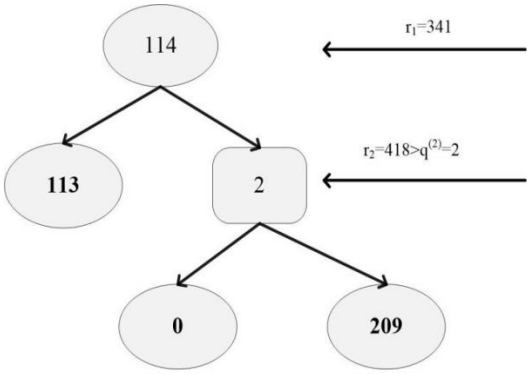
Результат узагальненого розщеплення символу = 18 2 123



ASCII код (r) = 114

Гама = 341 418

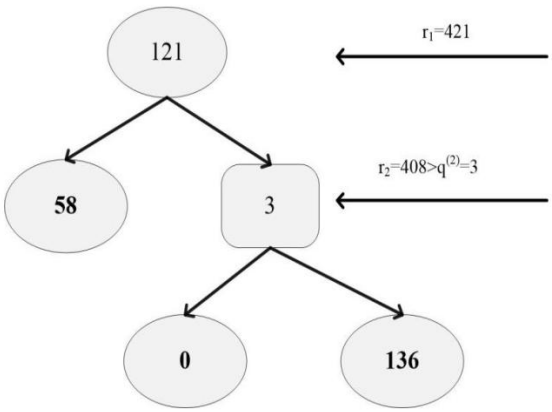
Результат узагальненого розщеплення символу = 113 0 209



ASCII код (y) = 121

Гама = 421 408

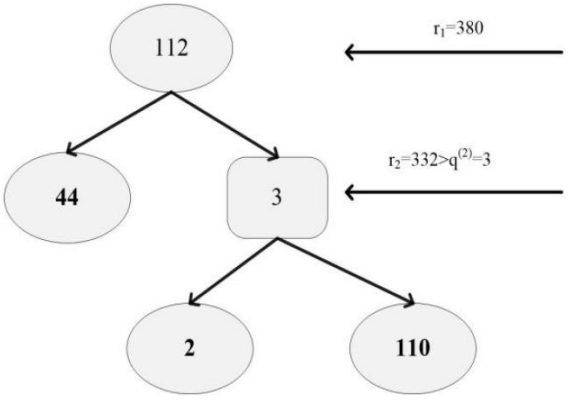
Результат узагальненого розщеплення символу = 58 0 136



ASCII код (p) = 112

Гама = 380 332

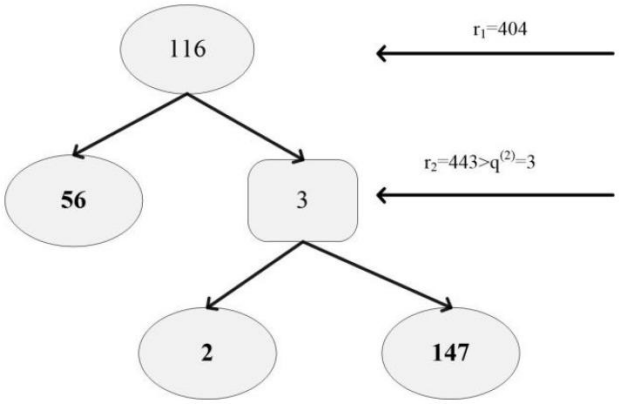
Результат узагальненого розщеплення символу = 44 2 110



ASCII код (t) = 116

Гама = 404 443

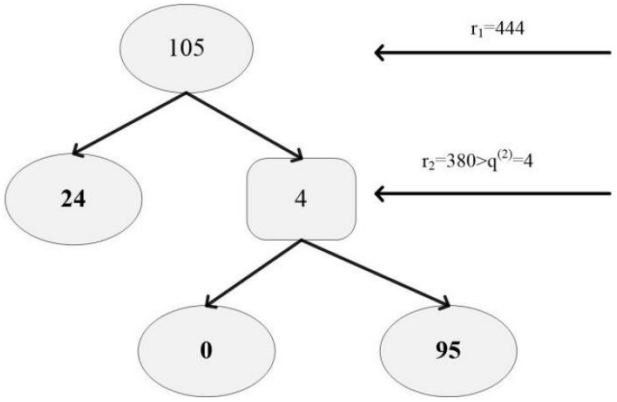
Результат узагальненого розщеплення символу = 56 2 147



ASCII код (i) = 105

Гама = 444 380

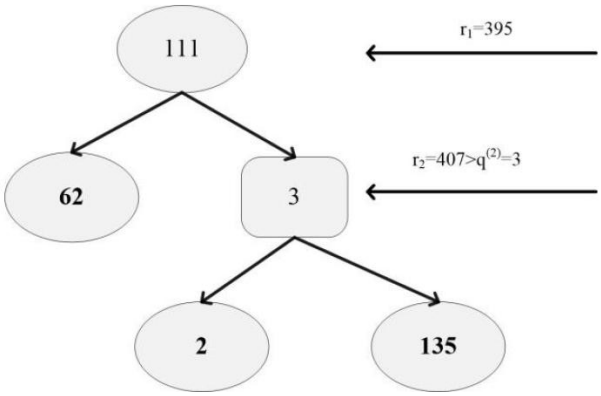
Результат узагальненого розщеплення символу = 24 0 95



ASCII код (o) = 111

Гама = 395 407

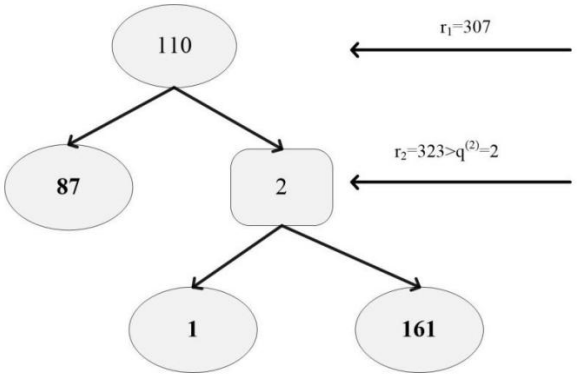
Результат узагальненого розщеплення символу = 62 2 135



ASCII код (n) = 110

Гама = 307 323

Результат узагальненого розщеплення символу = 87 1 161



4) захищений текст виглядатиме так: 56 1 50 73 1 214 18 2 123 113 0 209 58 0 136 44 2 110 56 2 147 24 0 95 62 2 135 87 1 161.

Відновлення вихідного тексту із зашифрованого тексту методом узагальненого розщеплення відбувається з використанням наступних кроків:

1) розділити отриманий зашифрований текст на поєднання з *k*=3 цілих чисел: 56 1 50 73 1 214 18 2 123 113 0 209 58 0 136 44 2 110 56 2 147 24 0 95 62 2 135 87 1 161;

2) розділити згенеровану послідовність гам на поєднанні з *k*-1=2 цілих чисел або більше k-1 цілих чисел: 470 301 293 429 315 371 341 418 421 408 380 332 404 443 444 380 395 407 307 323;

3) процес вилучення символів виглядатиме так:

- гама = 470 301; зашифрований текст = 56 1 50

$$q^{(2)} = \frac{r_2 - \delta^{(3)}}{q^{(3)}} = \frac{301-1}{50} = 6;$$

$$a = \frac{r_1 - \delta^{(2)}}{q^{(2)}} = \frac{470-56}{6} = 69;$$

Відображення значення 69 в таблицю кодів ASCII приводить до розшифрування першого символу E;

- гама = 293 429; зашифрований текст = 73 1 214;

$$q^{(2)} = \frac{r_2 - \delta^{(3)}}{q^{(3)}} = \frac{429-1}{214} = 2;$$

$$a = \frac{r_1 - \delta^{(2)}}{q^{(2)}} = \frac{293-73}{2} = 110;$$

Відображення значення 110 в таблицю кодів ASCII відповідає символу n;

- гама = 315 371; зашифрований текст = 18 2 123;

$$q^{(2)} = \frac{r_2 - \delta^{(3)}}{q^{(3)}} = \frac{371-2}{123} = 3;$$

$$a = \frac{r_1 - \delta^{(2)}}{q^{(2)}} = \frac{315-18}{3} = 99;$$

Відображення значення 99 в таблицю кодів ASCII приводить до символу c;

- гама = 341 418; зашифрований текст = 113 0 209;

$$q^{(2)} = \frac{r_2 - \delta^{(3)}}{q^{(3)}} = \frac{418-0}{209} = 2;$$

$$a = \frac{r_1 - \delta^{(2)}}{q^{(2)}} = \frac{341-113}{2} = 114;$$

Відображення значення 114 у таблицю кодів ASCII веде до символу r.

- гама = 421 408; зашифрований текст = 58 0 136;

$$q^{(2)} = \frac{r_2 - \delta^{(3)}}{q^{(3)}} = \frac{408 - 0}{136} = 3;$$

$$a = \frac{r_1 - \delta^{(2)}}{q^{(2)}} = \frac{421 - 58}{3} = 121;$$

У таблиці кодів ASCII значення 121 відповідає символу y;

- гама = 380 332; зашифрований текст = 44 2 110;

$$q^{(2)} = \frac{r_2 - \delta^{(3)}}{q^{(3)}} = \frac{332 - 2}{110} = 3;$$

$$a = \frac{r_1 - \delta^{(2)}}{q^{(2)}} = \frac{380 - 44}{3} = 112;$$

Відображення значення 112 у таблицю кодів ASCII приводить до символу p;

- гама = 404 443; зашифрований текст = 56 2 147;

$$q^{(2)} = \frac{r_2 - \delta^{(3)}}{q^{(3)}} = \frac{443 - 2}{147} = 3;$$

$$a = \frac{r_1 - \delta^{(2)}}{q^{(2)}} = \frac{404 - 56}{3} = 116;$$

Відображення значення 116 таблицею кодів ASCII призводить до символу t;

- гама = 444 380; зашифрований текст = 24 0 95;

$$q^{(2)} = \frac{r_2 - \delta^{(3)}}{q^{(3)}} = \frac{380 - 0}{95} = 4;$$

$$a = \frac{r_1 - \delta^{(2)}}{q^{(2)}} = \frac{444 - 24}{4} = 105;$$

Відображення значення 105 таблицями кодів ASCII призводить до розкриття символу i;

- гама = 395 407; зашифрований текст = 62 2 135;

$$q^{(2)} = \frac{r_2 - \delta^{(3)}}{q^{(3)}} = \frac{407 - 2}{135} = 3;$$

$$a = \frac{r_1 - \delta^{(2)}}{q^{(2)}} = \frac{395 - 62}{3} = 111;$$

Відображення значення 111 в таблиці кодів ASCII вказує на символ o;

- гама = 307 323; зашифрований текст = 87 1 161;

$$q^{(2)} = \frac{r_2 - \delta^{(3)}}{q^{(3)}} = \frac{323 - 1}{161} = 2;$$

$$a = \frac{r_1 - \delta^{(2)}}{q^{(2)}} = \frac{307 - 87}{2} = 110;$$

Відображення значення 110 таблицею кодів ASCII призводить до символу n;

4) відновлений вихідний текст виглядатиме так: Encryption.

3 ЕКСПЕРИМЕТНАЛЬНІ РЕЗУЛЬТАТИ ЗАХИСТУ ІНФОРМАЦІЇ ЗА ДОПОМОГОЮ СИМВОЛЬНОГО РОЗЩЕПЛЕННЯ

3.1 Система захисту символів з розщепленням при відображенні інформації різними поєднаннями чисел

Один і той же символ відображається у системі з розщепленням різними поєднаннями двох чисел при k=2, як показано на рисунках 3.1 та 3.2.

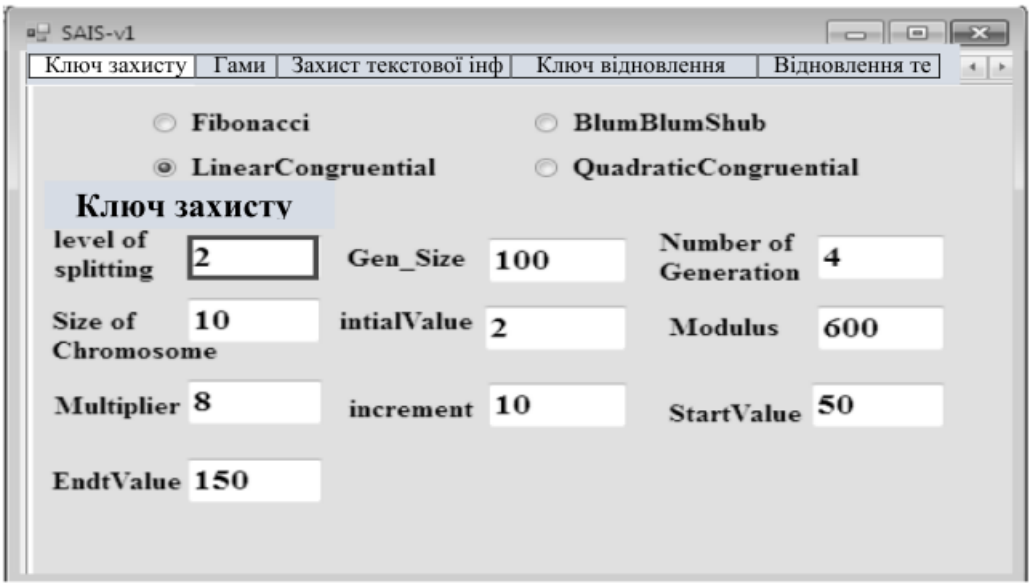


Рисунок 3.1 − Ключ захисту (рівень символьного розщеплення k=2)

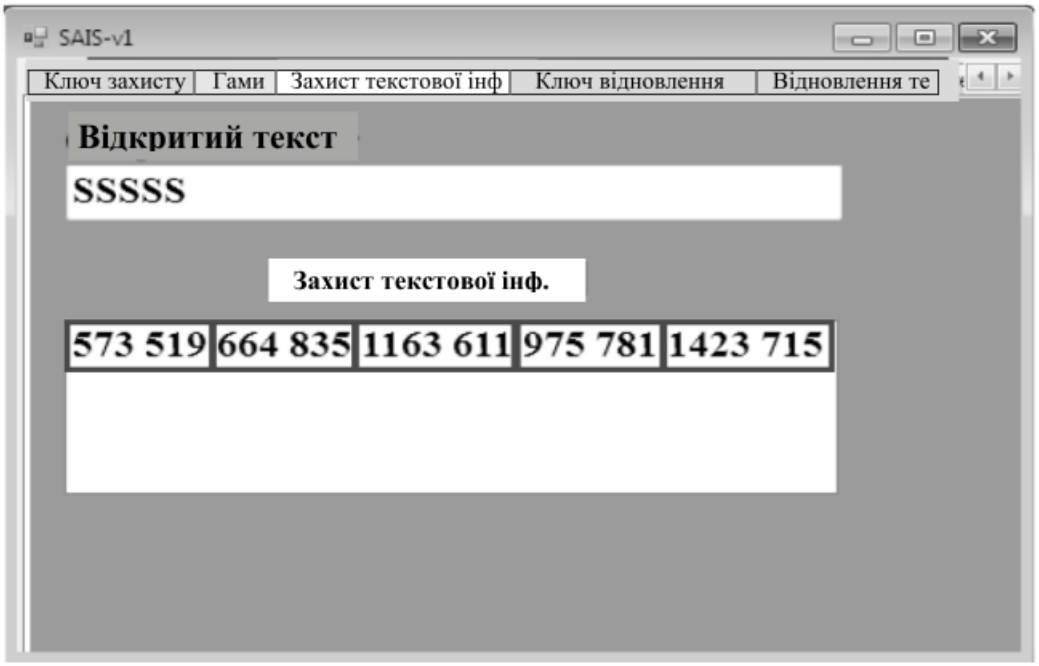


Рисунок 3.2− Захищений текст відображається різними поєднаннями чисел (при рівні розщеплення k=2)

Кожний символ відображається різними поєднаннями трьох чисел при k=3, як показано на рисунках 3.3 та 3.3.

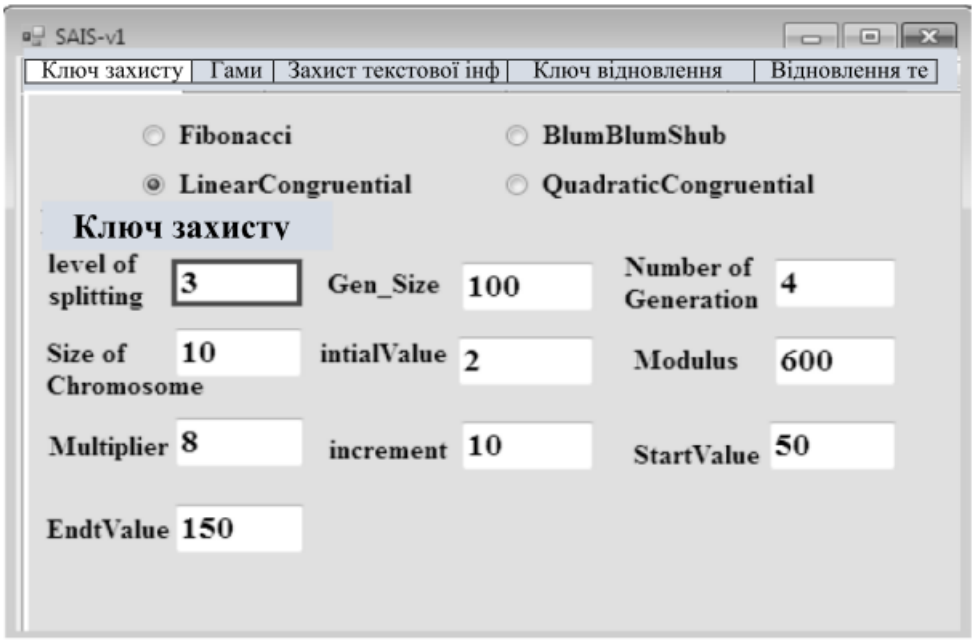


Рисунок 3.3 – Ключ захисту (рівень символьного розщеплення k=3)

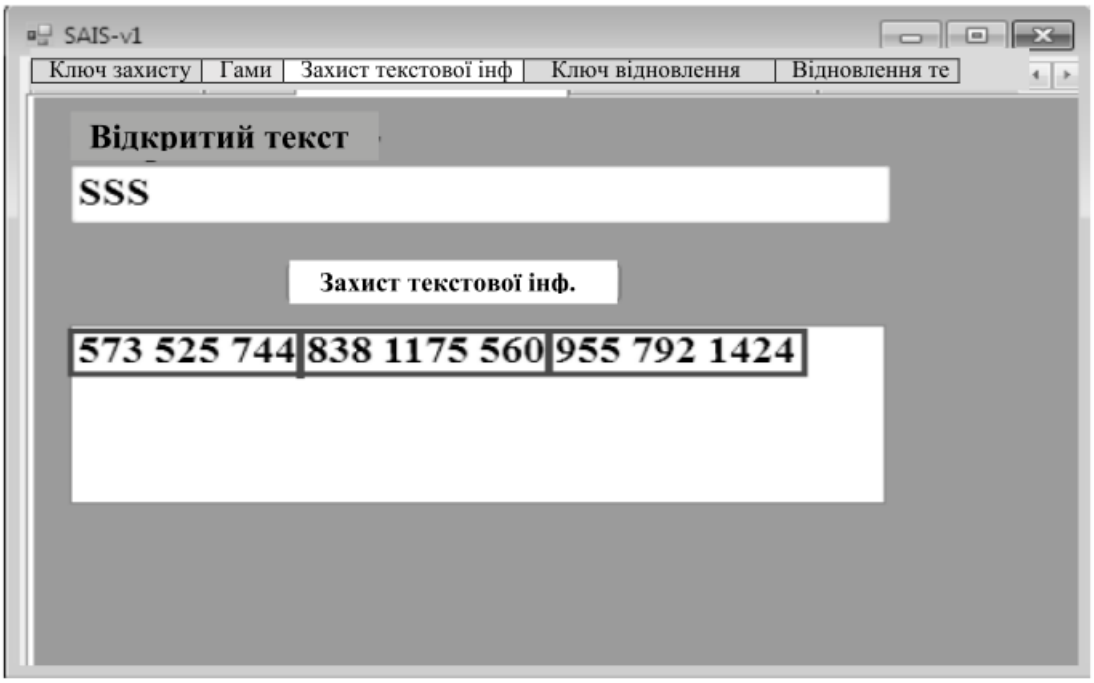


Рисунок 3.4 − Захищений текст відображається різними поєднаннями чисел (при рівні розщеплення k=3)

Отже, запропонований метод розщеплення суттєво ускладнює відновлення вихідного тексту несанкціонованим користувачем, що спирається на семантичний зміст цього тексту.

3.2 Статистичні тести для випадкових чисел, які створюються ГПВЧ з використанням ДГА

Статистичні тести виконуються для двох послідовностей випадкових чисел. У першому випадку послідовність створюється традиційним ГПВЧ без будь-якої корекції, тоді як у другому – до цієї послідовності застосовуються оператори ДГА. Тести виконуються за допомогою чотирьох програмних комплексів: МАТЛАБ, C#, Minitab та IBM SPSS Statistics, для трьох типів ГПВЧ:

а) вихідні псевдовипадкові числа генеруються за допомогою лінійного конгруентного генератора з наступними параметрами: множник=7, приріст = 10, модуль = 988, початкове значення = 7, розмір генерації = 100;

б) аналогічний тест застосовується також для генератора Блюм-Блюма-Шуба з наступними параметрами: модуль = 988, початкове значення = 7, розмір генерації=100 (значення параметрів такі ж, як пункті а);

в) квадратичний конгруентний ГПВЧ.

Тест застосовується також до квадратичного конгруентного генератора з такими параметрами: $a = 4$, $b = 5$, $c = 7$, модуль = 988, початкове значення = 7, розмір генерації = 100 (схожі значення параметрів, як у пунктах а і б).

Частотний тест перевіряє на рівномірність розподілу та реалізується з допомогою тесту Колмогорова – Смирнова, який порівнює емпіричну функцію розподілу змінної із заданим теоретично законом розподілу. Він застосовується для перевірки гіпотези про належності вибірки деякому фіксованому закону розподілу. Ми будемо використовувати цей тест для перевірки того, що два емпіричні розподіли відповідають одному й тому закону. Розглядаються нульова та альтернативна гіпотези:

- H0 – дані близькі до нормального розподілу;

- H1 – дані не можна вважати близькими до нормального розподілу.

Рівень статистичної значущості вибирається α=0.01 (у криптографії прийнято α брати в інтервалі від 0.001 до 0.01).

Результат застосування тесту Колмогорова – Смирнова (у пакеті Minitab) до послідовності, що створюється лінійним конгруентним генератором без використання ДГА, показаний рисунку 3.5.

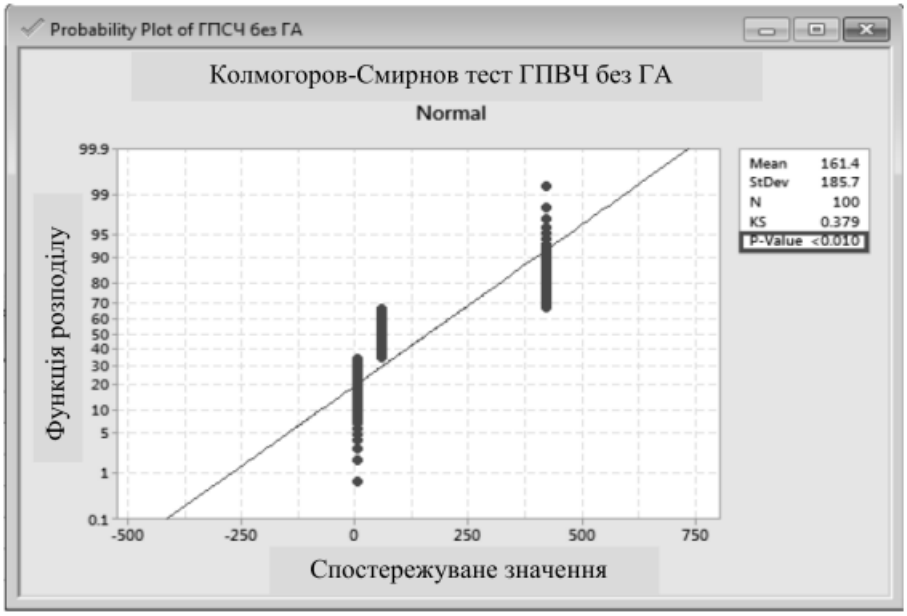


Рисунок 3.5 – Тест Колмогорова – Смирнова для лінійного конгруентного генератора без використання ДГА

З тесту випливає такий висновок: $P<a=0.010$, і ми відкидаємо нульову гіпотезу, що дані близькі до нормального розподілу, та приймаємо альтернативну гіпотезу H1, що дані слід вважати близькими до нормального розподілу. З рисунка 3.5 видно, що емпіричні (аактичні) функції дуже далекі від нормального розподілу.

Результати застосування тесту Колмогорова – Смирнова в (пакеті Minitab) до послідовності, що породжується лінійним конгруентним генератором з застосуванням ДГА, показані рисунку 3.6.

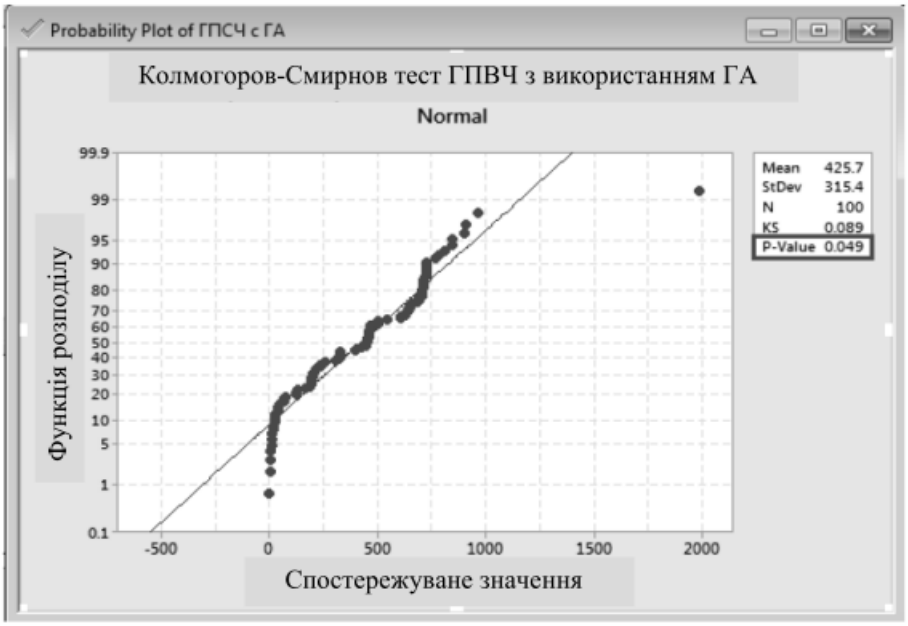


Рисунок 3.6 – Тест Колмогорова – Смирнова для лінійного конгруентного генератора з використанням ДГА

З результатів випливає, що оскільки P=0.049>a=0.010, то приймається нульова гіпотеза, тобто розподіл даних близький до нормального.

На рисунку 3.6 також видно, що емпіричні (фактичні) функції розподілу дуже близькі до теоретичного.

Результат застосування тесту Колмогорова – Смирнова у пакеті IBM SPSS Statistics до послідовності генератора Блюм-Блюма-Шуба без використання ДГА показаний рисунку 3.7.

One-Sample Kolmogorov-Smirnov Test

		BlumWithout GA
N		100
Normal Parameters[a,b]	Mean	605.2200
	Std. Deviation	208.25196
Most Extreme Differences	Absolute	.326
	Positive	.317
	Negative	-.326
Kolmogorov-Smirnov Z		3.261
Asymp. Sig. (2-tailed)		.000

a. Test distribution is Normal.
b. Calculated from data.

Рисунок 3.7 - Тест Колмогорова - Смирнова для генератора Блюм-Блюма-Шуба без використання ДГА

З тесту випливає такий висновок: оскільки P=0.00<a=0.010), то нульова гіпотеза, що дані мають розподіл, близький до нормального, відхиляється та приймається альтернативна гіпотеза H1, що розподіл не близький до нормальному.

Результат застосування тесту Колмогорова – Смирнова у пакеті IBM SPSS Statistics до послідовності генератора Блюм-Блюма-Шуба з використанням генетичного алгоритму показаний рисунку 3.8.

One-Sample Kolmogorov-Smirnov Test

		BlumWithGA
N		100
Normal Parameters[a,b]	Mean	475.7600
	Std. Deviation	223.42961
Most Extreme Differences	Absolute	.151
	Positive	.090
	Negative	-.151
Kolmogorov-Smirnov Z		1.507
Asymp. Sig. (2-tailed)		.021

a. Test distribution is Normal.
b. Calculated from data.

Рисунок 3.8 - Тест Колмогорова-Смирнова для генератора Блюм-Блюма- Шуба з використанням ДГА

З тесту випливає, що P=0.021>a=0.010, тобто нульова гіпотеза, що розподіл даних близький до нормального, не відкидається.

У таблиці 3.1 підсумовуються результати застосування тесту Колмогорова- Смирнова, і на рисунку 3.9 показані результати цього підсумовування.

3.3 Критерій серій

Цей тест використовується для перевірки того, чи є послідовність дійсно випадковою, в ній перевіряються викиди вище або нижче за деяку константу

(зазвичай середнього рівня). У тесті рахується фактичне число поява серій різної довжини, порівнюються показники відповідних лічильників з очікуваними значеннями за критерієм хі-квадрат та перевіряється розташування чисел у послідовності, щоб підтвердити гіпотезу про незалежність.

Таблиця 3.1 – Результати тесту Колмогорова-Смирнова

Тип генератора	Без використання ДГА	З використанням ДГА
Лінійний конгруентний генератор	0	0,049
Блюм-Блюм-Шуф генератор	0	0,021

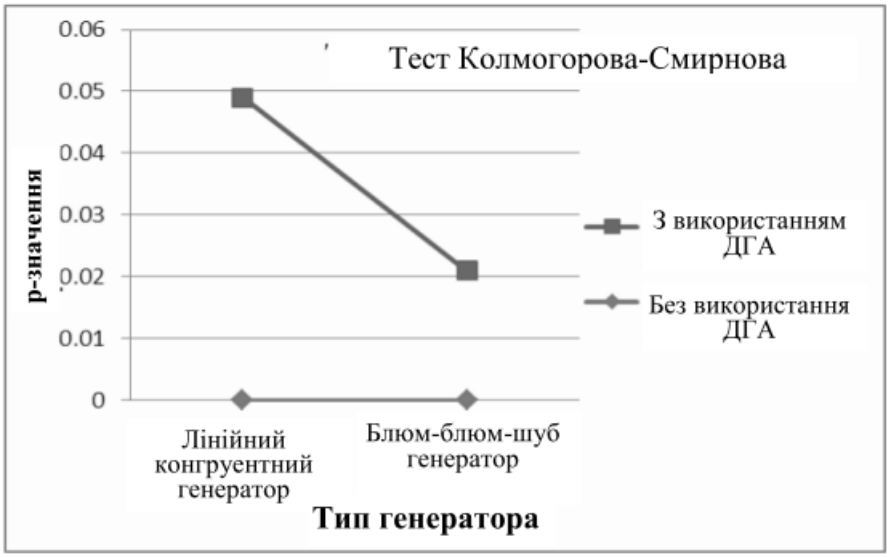


Рисунок 3.9 - Результати тесту Колмогорова-Смирнова

Нульова та альтернативна гіпотеза:

H0 – розподіл у послідовності є випадковим;

H1 – розподіл у послідовності не носить суто випадковий характер, рівень статистичної значимості: α=0.01.

Результати застосування тесту для лінійного конгруентного ГПВЧ з пакету Minitab до послідовності, що породжується лінійним конгруентним генератором без використання ДГА, показані на рисунку 3.10.

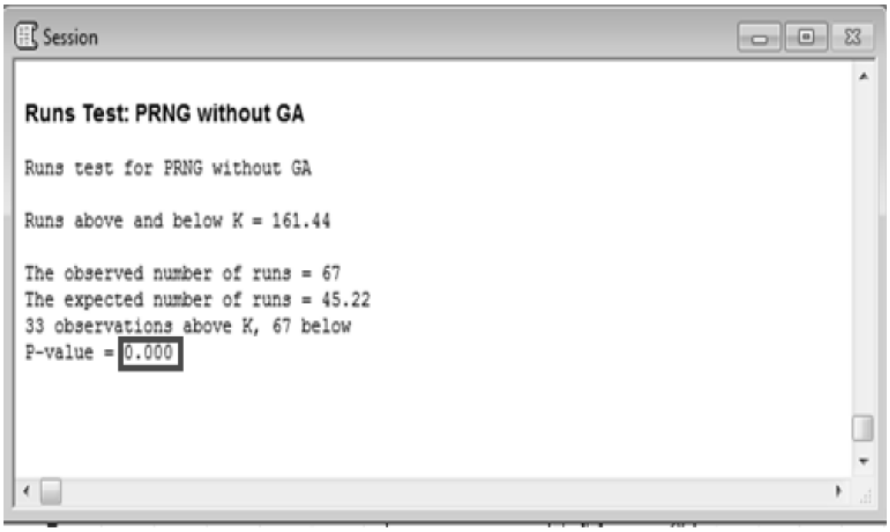


Рисунок. 3.10 – Тест серій для лінійного конгруентного генератора без використання ДГА

З тесту логічно робиться такий висновок: оскільки p=0,00<a=0,010, то нульова гіпотеза H0 про випадковість розподілу для послідовності відхиляється, а береться альтернативна гіпотеза H1.

Результати застосування тесту з пакету Minitab до послідовності, що породжується лінійним конгруентним генератором з використанням ДГА, показані рисунку 3.11.

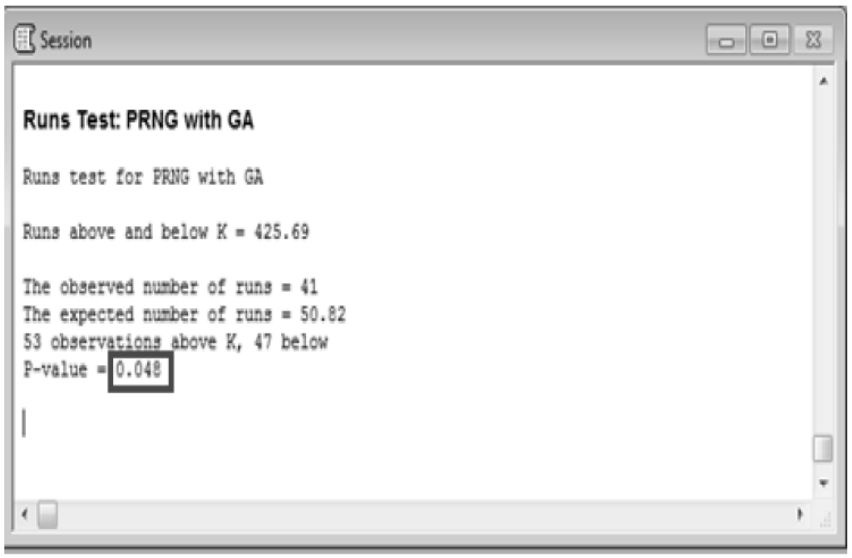


Рисунок 3.11 – Тест наявності серій для лінійного конгруентного генератора з використанням ДГА

З тесту випливає такий висновок: оскільки p=0,048>a=0,010, то нульова гіпотеза H0 про випадковість розподілу для послідовності не відхиляється.

Результат застосування тесту на серії Вальда-Вольфовіца в пакеті Minitab до послідовності генератора Блюм-Блюма-Шуба без використання ДГА показана на рисунку 3.12.

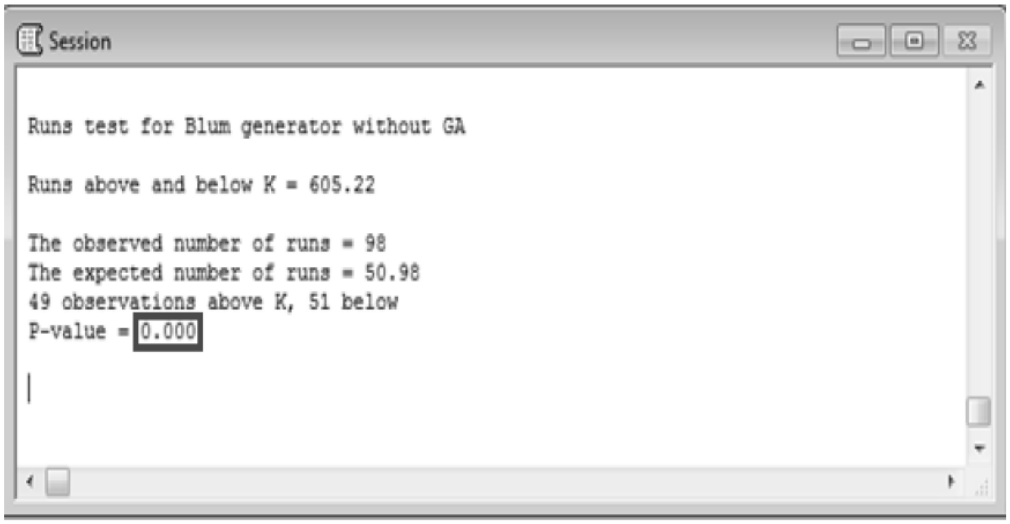


Рисунок 3.12 – Тест серій для генератора Блюм-Блюма-Шуба без використання ДГА

З тесту випливає висновок: p=0,00<a=0,010, тоді нульова гіпотеза H0 про випадковість розподілу для послідовності відхиляється, а альтернативна гіпотеза H1 не відхиляється.

Результат застосування тесту на серії Вальда-Вольфовіца з пакету Minitab до послідовності генератора Блюм-Блюма-Шуба з використанням ДГА показана на рисунку 3.13.

З тесту можна зробити такий висновок: оскільки p=0,361>a=0,010, то нульова гіпотеза H0 про випадковість розподілу для послідовності не відхиляється.

Результат застосування тесту послідовностей у пакеті Minitab до послідовності, що створюється квадратичним конгруентним генератором без використання ДГА, показаний рисунку 3.14.

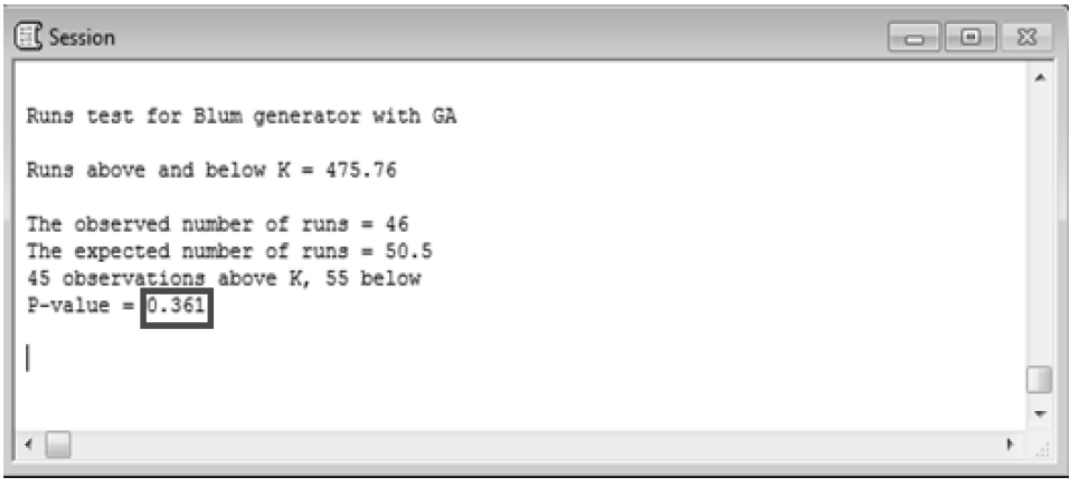


Рисунок 3.13 - Тест серій для генератора Блюм-Блюма-Шуба з використанням ДГА

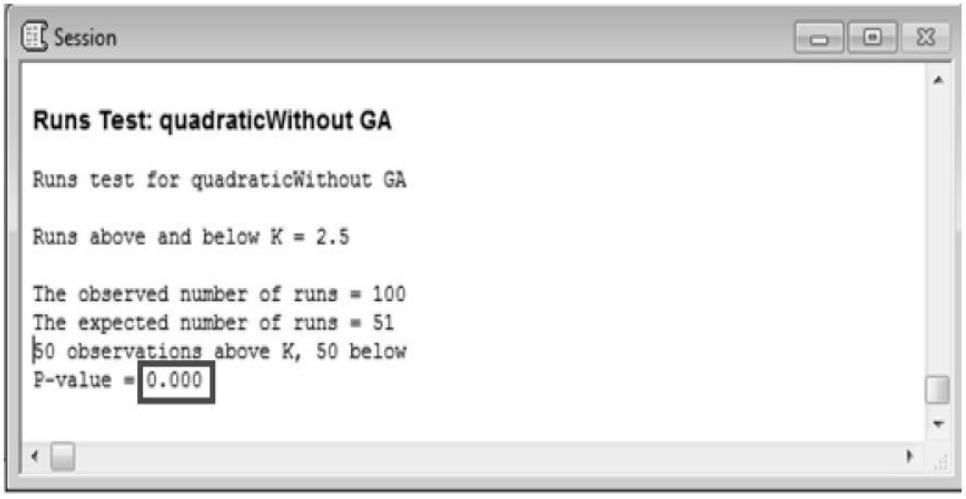


Рисунок 3.14 – Тест серій для Квадратичного конгруентного генератора без використання ДГА

З тесту випливає наступне: оскільки p=0,00<a=0,010, то нульова гіпотеза H0 про випадковість розподілу у цій послідовності відхиляється, а альтернативна гіпотеза H1 не відхиляється.

Результат застосування тесту послідовностей у пакеті Minitab до послідовності для квадратичного конгруентного генератора з використанням ДГА показаний рисунку 3.15.

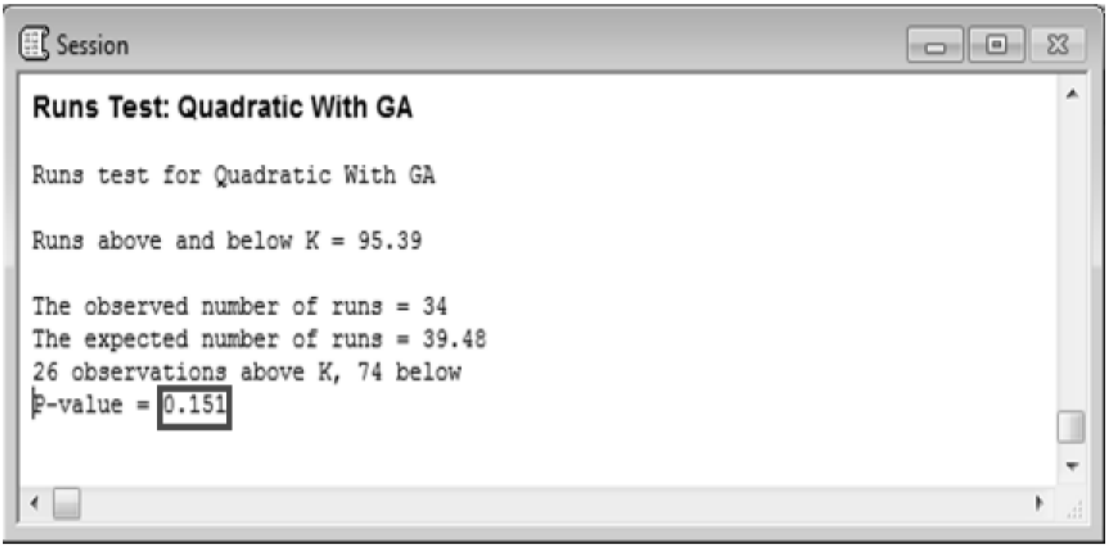


Рисунок 3.15 – Тест серій для квадратичного конгруентного генератора з використанням ДГА

Оскільки значення параметра p=0,151>a=0,010, то в цьому випадку нульова гіпотеза H0 про випадковість розподілу для послідовності не відкидається.

У таблиці 3.2 підсумовуються отримані результати для застосування описаного тесту серій і відповідно на рисунку 3.16 показані результати цього підсумовування.

Таблиця 3.2 - Підсумовування результатів тесту серій

Тип генератора	Лінійний конгруентний генератор	Блюм-Блюм-Шуб генератор	Квадратичний конгруентний генератор
Без використання ДГА	0	0	0
З використанням ДГА	0,048	0,361	0,151

3.4 Використання ентропії для символьного розщеплення

Порівняння результатів застосування ентропії як міри випадковості в пакеті MATLAB до послідовності лінійного конгруентного генератора без використання ДГА показана на рисунку 3.17, а з використанням ДГА - на рисунку 3.18.

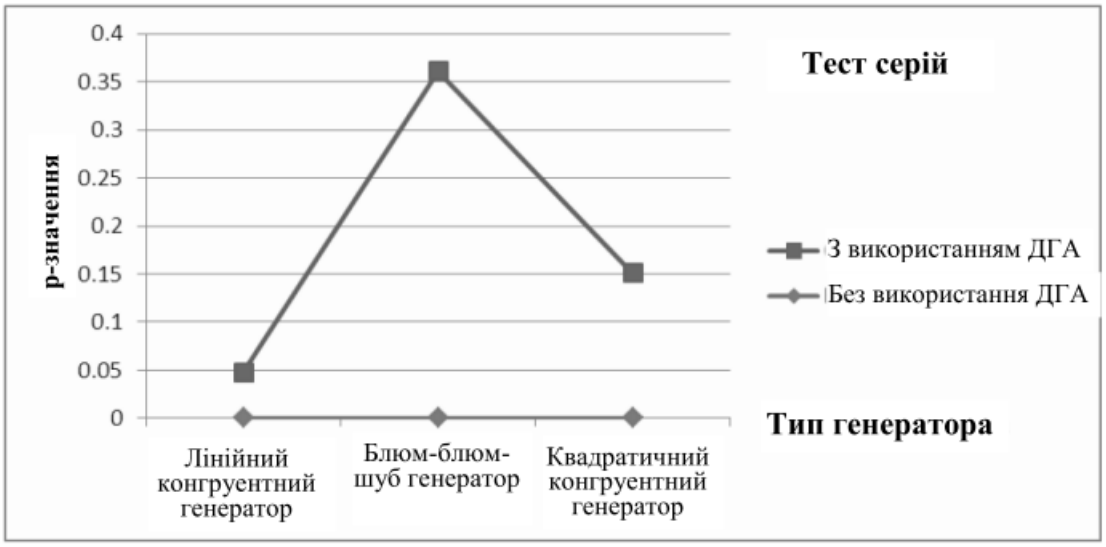


Рисунок 3.16 – Підсумовування результатів тестування серій

```
>> Entropy_Linear_Generator_Without_GA=Entropy(linearGeneratorWithoutGA)

Entropy_Linear_Generator_Without_GA =

    1.5848
```

Рисунок 3.17 – Ентропія як міра випадковості для лінійного конгруентного генератора без використання ДГА

```
>> Entropy_Linear_Generator_With_GA=Entropy(linearGeneratorWithGA)

Entropy_Linear_Generator_With_GA =

    6.3688
```

Рисунок 3.18 – Ентропія як міра випадковості для лінійного конгруентного генератора з використанням ДГА

Ентропія є мірою невпорядкованості чи непередбачуваності інформаційних елементів Чим вона вища, тим хаотичніше – непередбачувано заздалегідь – розподілено інформацію.

З тесту рисунків 3.17 і 3.18 видно, що ентропія для лінійного конгруентного генератора без використання ДГА =1.58 менша ентропії лінійного конгруентного генератора з використанням ДГА = 6.37. Тобто ентропія послідовності від лінійного конгруентного генератора з використанням ДГА вища, вона більш хаотична (непередбачувана) порівняно з послідовністю без використання ДГА.

Порівняння результатів застосування ентропії як міри випадковості в пакеті MATLAB до послідовності від генератора Блюм-Блюма-Шуба без використання ДГА показана на рисунку 3.19, з використанням ДГА - на рисунку 3.20.

```
>> Entrony_Blum_Generator_without_GA=Entropy(Blum_Blum_Shub_generator_without_GA)

Entrony_Blum_Generator_without_GA =

    1.1414
```

Рисунок 3.19 – Ентропія як міра випадковості для генератора Блюм-Блюма-Шуба без використання ДГА

```
>> Entrony_Blum_Generator_with_GA=Entropy(Blum_Blum_Shub_generator_with_GA)

Entrony_Blum_Generator_with_GA =

    5.3002
```

Рисунок 3.20 – Ентропія як міра випадковості для генератора Блюм-Блюма-Шуба з використанням ДГА

З тесту на рисунках 3.19 і 3.20 видно, що ентропія генератора Блюм-Блюма-Шуба без використання ДГА = 1.14<ентропія генератора Блюм- Блюма-

Шуба з використанням ДГА = 5.3. Тобто ентропія послідовності від генератора Блюм-Блюма-Шуба з допомогою ДГА вища, вона більш хаотично-непередбачувана, ніж послідовність без використання ДГА.

Порівняння результатів застосування ентропії як міри випадковості в пакеті MATLAB для послідовності від квадратичного конгруентного генератора без використання ДГА показано рисунку 3.21, а з використанням ДГА – на рисунку 3.22.

```
>> Entrony_quadratic_congruential_generator_without_GA=Entropy(quadratic_congruential_generator_without_GA)

Entrony_quadratic_congruential_generator_without_GA =

    1
```

Рисунок 3.21 – Ентропія як міра випадковості для конгруентного генератора без використання ДГА

```
>> Entrony_quadratic_congruential_generator_with_GA=Entropy(quadratic_congruential_generator_with_GA)

Entrony_quadratic_congruential_generator_with_GA =

    5.1103
```

Рисунок 3.22 – Ентропія як міра випадковості для квадратичного конгруентного генератора з використанням ДГА

З тесту на рисунках 3.21 і 3.22 випливає висновок, що ентропія квадратичного конгруентного генератора без використання ДДА=1<ентропії квадратичного конгруентного генератора з використанням ДГА=5,1103). Тобто ентропія послідовності квадратичного конгруентного генератора з використанням ДГА вища, вона більш хаотично-непередбачувана порівняно з послідовністю без використання ДГА.

У таблиці 3.3 підсумовуються результати застосування ентропії і на рисунку 3.23 показані результати цього підсумовування.

Таблиця 3.3 - Підсумовування результатів ентропії

Тип генератора	Лінійний конгруентний генератор	Блюм-блюм-шуб генератор	Квадратичний конгруентний генератор
Без використання ДГА	1,58	1,14	1
З використанням ДГА	6,37	5,30	5,11

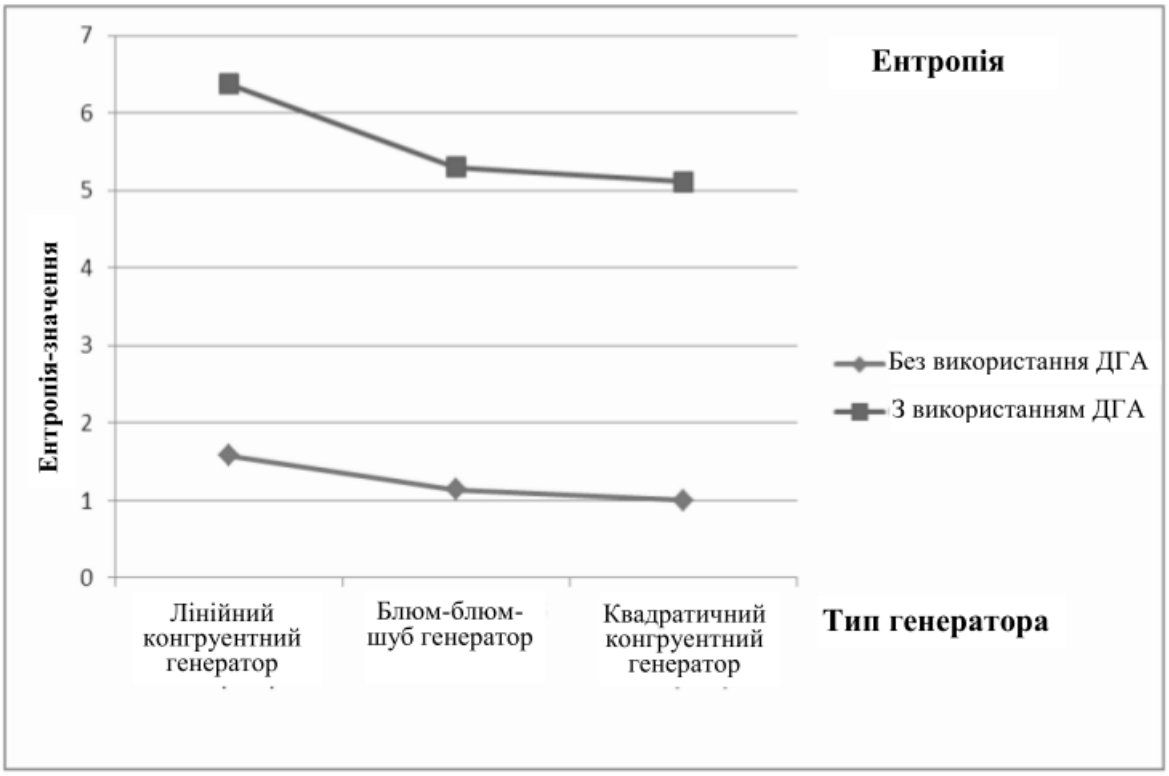


Рисунок 3.23 – Підсумовування результатів ентропії

Отже, запропонований метод символьного розщеплення істотно ускладнює відновлення вихідного тексту для несанкціонованих користувачів, які спираються на семантичний зміст цього тексту.

ВИСНОВКИ

1. Здійснено аналіз сучасних підходів для захисту інформації, що дало змогу дослідити можливості існуючих методів та обгрунтувати актуальність використання символьного розщеплення для захисту інформаційних текстових потоків.

2. Розроблено метод захисту інформації на основі символьного розщеплення, що дало змогу розробити відповідну математичну модель та блок-схему.

3. Розроблено числові приклади застосування методу захисту на основі простого та узагальненого символьного розщеплення, які дали змогу продемонструвати переваги розробленого алгоритму.

4. Розроблено систему захисту символів з розщепленням при відображенні інформації різними поєднаннями чисел та проведено статистичне тестування для випадкових чисел. Обгрунтовано використання ентропії для символьного розщеплення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бурячок В.Л., Гулак Г.М., Толубко В.Б. Інформаційний та кіберпростори: проблеми безпеки, методи та засоби боротьби: Підручник. К.: ДУТ, 2015. 449 с.

2. Бурячок В.Л., Толубко В.Б., Хорошко В. О., Толюпа С.В. Інформаційна і кібербезпека: соціотехнічний аспект: Підручник. К.: ДУТ, 2015. 288 с.

3. Власюк О. С. Національна безпека України: еволюція проблем внутрішньої політики: Вибр. наук. праці. К. : НІСД, 2016. 528 с.

4. Грищук Р.В., Даник Ю.Г. Основи кібернетичної безпеки: Монографія. Житомир: ЖНАЕУ, 2016. 636 с.

5. Пількевич І.А., Лобанчикова Н.М.,. Молодецька К.В. Захист інформації в автоматизованих системах управління: посібник. Житомир: Вид-во ЖДУ ім. І. Франка, 2015. 226 с.

6. Тарнавський Ю.А. Технології захисту інформації [Електронний ресурс]: підручник. К.: КПІ ім. Ігоря Сікорського, 2018. 162 с. Режим доступу до ресурсу: https://ela.kpi.ua/bitstream/123456789/23896/1/ TZI_book.pdf

7. Волокітін А.В., Маношкин А.П., Солдатенков А.В., Савченко С.А., Петров Ю.А. Інформаційна безпека державних організацій і комерційних фірм. К.: Юніор, 2012. 303 с.

8. Остапов С. Технології захисту інформації. Посібник. Родовід, 2014. 428 с.

9. Козловський А.В., Паночишин Ю.М., Погрішук Б.В. Комп'ютерна техніка та інформаційні технології: навч. посіб. К.: Знання, 2014. 463с.

10. Інформаційна безпека: навчальний посібник/ Ю. Я. Бобало, І. В. Горбатий, М. Д. Кіселичник, А. П. Бондарєв та інші; 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.

11. Антонюк А.О., Жора В.В. Теоретичні основи моделювання та аналізу систем захисту інформації: [монографія] / Ірпінь Національний університет ДПС України, 2010. 310 с.

12. Salami, Y., Khajevand, V., & Zeinali, E. Cryptographic algorithms: a review of the literature, weaknesses and open challenges. J. Comput. Robot. 2023. Vol.16(2). P.46-56.

13. Al-Husainy M.A.F., Al-Shargabi B., Aljawarneh, S. (2021). Lightweight cryptography system for IoT devices using DNA. Computers and Electrical Engineering. 2021, 95. P.107418.

14. Калушка М., Кулина С. Концепція та принципи гібридного шифрування. Кібербезпека державних інституцій та подолання кризових станів: матеріали III Міжнар. наук.-практ. конф. в 2 т. (Київ – Прага – Таллінн – Тернопіль), 14 листоп. 2024 р. / ІСЗЗІ КПІ ім. Ігоря Сікорського. Київ, 2024. Т. 1. С. 450.

15. Adki V., Hatkar S. A Survey on Cryptography Techniques. International Journal of Advanced Research in Computer Science and Software Engineering. 2016. Vol. 6 (6). P. 469-475.

16. Крук О., Голембйовський М., Басістий В. Математична модель захисту інформації на основі символьного розщеплення. Збірник матеріалів науково - практичної конференції молодих вчених, аспірантів та студентів «Кібербезпека та комп'ютерно-інтегровані технології» (КБКІТ-2024), Тернопіль, 2024. С.118-121.

17. Карпів В., Крук О., Казьмірчук Н. Імовірнісний аналіз стійкості методу розщеплення для захисту інформації. Матеріали науково-практичного симпозіуму «Захист інформації». Тернопіль, 2024. С.89-92.

18. Задірака В.К., Олексюк О.С. Комп'ютерна криптологія. Тернопіль, Київ, 2002. 504 с.

19. Jeffrey H., Jill P., Joseph H. An Introduction to Mathematical Cryptography. Berlin: Springer, 2008. 540 p.

20. Silverman J.H. The arithmetic of elliptic curves. New York: Springer. 2007. Vol. 106. P. 17-40.

21. Hoffstein J., Pipher J., Silverman J. An Introduction to Mathematical Cryptography. Springer Science+Business Media, LLC, 2008. 524 p.

22. Shoup V. A Computational Introduction to Number Theory and Algebra. Cambridge University Press, 2005. 517 p.

23. Івасьєв С.В. Методи та обчислювальні засоби рішення задач теорії чисел у базисах Радемахера – Крестенсона: дис. канд. техн. наук: 05.13.05. Тернопіль: ТНЕУ, 2016. 222 с.

24. Немеш І.В., Капустинський Р.І., Козбур Г.Є., Твердун Б.С. Метод ітераційного та часового аналізу k-арного алгоритму Евкліда для задач криптографії. Збірник матеріалів науково - практичної конференції молодих вчених, аспірантів та студентів «Кібербезпека та комп'ютерно-інтегровані технології» (КБКІТ-2023), Тернопіль, 2023. С.154-156.

25. Николайчук Я.М. Теорія джерел інформації. Тернопіль: ТзОВ: Терног раф, 2010. 536 с.

26. Rukhin A., Soto J., Nechvatal J., Smid M. A Statistical Test Suite For Random And Pseudorandom Number Generators For Cryptographic Applications. NIST Special Publication 800-22, 2001. 153 с.

27. Dutta S., Das T., Jash Sh., Patra D. A Cryptography Algorithm Using the Operations of Genetic Algorithm & Pseudo Random Sequence Generating Functions. International Journal of Advances in Computer Science and Technology. 2014. Vol.3, No.5. Pp. 325-330.

28. Mondal S., Mollah T.K., Samanta A., Paul S. A Survey on Network Security Using Genetic Algorithm. International Journal of Innovative Research in Science, Engineering and Technology. 2016. Vol. 5, No.1. Pp. 87-93.

29. Aarti S., Suyash A. Using Genetic Algorithm for Symmetric key Generation in Image Encryption. International Journal of Advanced Research in Computer Engineering & Technology. 2012. Vol. 1, No. 10. Pp.137-140.

30. Agarwal A. Secret Key Encryption Algorithm Using Genetic Algorithm. International Journal of Advanced Research in Computer Science and Software Engineering. 2012. Vol. 2, No. 4. Pp. 216-218.