**Міністерство освіти і науки України**
**Тернопільський національний технічний університет імені Івана Пулюя**

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету )
Кафедра комп'ютерних наук
(повна назва кафедри)

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр
(назва освітнього ступеня)

на тему: Методи захисту інформації в незахищених каналах зв'язку

Виконав: студент VI курсу, групи СНм-61
спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)

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

Керівник _____ Литвиненко Я.В.
(підпис) (прізвище та ініціали)

Нормоконтроль _____ Дуда О.М.
(підпис) (прізвище та ініціали)

Завідувач кафедри _____ Боднарчук І.О.
(підпис) (прізвище та ініціали)

Рецензент _____ Жаровський Р.О.
(підпис) (прізвище та ініціали)

Тернопіль
2023

Міністерство освіти і науки України
**Тернопільський національний технічний університет імені Івана Пулюя**

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Боднарчук І.О.
(підпис) (прізвище та ініціали)
«_26_» _грудня 2023 р.

# ЗАВДАННЯ
## НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Семенюк Володимир Олександрович
(прізвище, ім'я, по батькові)

1. Тема роботи Методи захисту інформації в незахищених каналах зв'язку

Керівник роботи Литвиненко Ярослав Володимирович, д.т.н., професор кафедри КН
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «_24_» _листопада_ 2023 року № _4/7-1099_____

2. Термін подання студентом завершеної роботи 24 грудня 2023р.

3. Вихідні дані до роботи Наукові публікації про методи захисту інформації. Збір та аналіз відомостей про методи захисту інформації.

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ. 1 Види загроз безпеці інформації. 2 Методи й засоби захисту інформаці. Blowfish.
3. Програмна реалізація алгоритму захисту інформації.
4 Охорона праці та безпека в надзвичайних ситуаціях. Висновки. Додатки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1 Тема. 2 Мета, задачі, дослідження. 3 Актуальність дослідження.
4 Переваги алгоритму Blowfish. 5 Програмна реалізація інтерфейсу. 6. Результати застосування. 7. Висновки.

6. Консультанти розділів роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---|---|---|---|
| | | завдання видав | завдання прийняв |
| Охорона праці | Сенчишин В.С., доцент | | |
| Безпека в надзвичайних ситуаціях | Клепчик В.М., ст. викладач | | |

7. Дата видачі завдання 24 листопада 2022 р.

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів роботи | Термін виконання етапів роботи | Примітка |
|---|---|---|---|
| 1. | Ознайомлення з завданням до кваліфікаційної роботи | 25.11.2023 | *Виконано* |
| 2. | Підбір наукових джерел по обраній тематиці | 26.11.2023-28.11.2023 | *Виконано* |
| | Аналіз джерел, які стосуються математичного забезпечення | | |
| 3. | Аналіз джерел, які стосуються програмного забезпечення, огляд систем. Підбір даних для опрацювання по обраній темі роботи. | 29.11.2023-1.12.2023 | *Виконано* |
| 4. | Виконання дослідження згідно мети кваліфікаційної | 2.12.2023-4.12.2023 | *Виконано* |
| | роботи | | |
| 5. | Оформлення розділу «Види загроз безпеці інформації» | 5.12.2023-7.12.2023 | *Виконано* |
| 6. | Оформлення розділу «Методи й засоби захисту. | 8.12.2023-10.12.2023 | *Виконано* |
| | інформаці. Blowfish» | | |
| 7. | Оформлення розділу «Програмна реалізація | 11.12.2023-13.12.2023 | *Виконано* |
| | алгоритму захисту інформації» | | |
| 8. | Виконання завдання до підрозділу «Охорона праці» | 14.12.2023-15.12.2023 | *Виконано* |
| 9. | Виконання завдання до підрозділу «Безпека в | 16.12.2023-17.12.2023 | *Виконано* |
| | надзвичайних ситуаціях» | | |
| 10. | Оформлення кваліфікаційної роботи | 18.12.2023-19.12.2023 | *Виконано* |
| 11. | Нормоконтроль | 19.12.2023-20.12.2023 | *Виконано* |
| 12. | Перевірка на плагіат | 21.12.2023 | *Виконано* |
| 13. | Попередній захист кваліфікаційної роботи | 22.12.2023 | *Виконано* |
| 14. | Захист кваліфікаційної роботи | 27.12.2023 | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Студент _____ Семенюк В.О.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Литвиненко Я.В.
(підпис) (прізвище та ініціали)

# АНОТАЦІЯ

Методи захисту інформації в незахищених каналах зв’язку // Кваліфікаційна робота освітнього рівня «Магістр» // Семенюк Володимир Олександрович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп’ютерно-інформаційних систем і програмної інженерії, кафедра комп’ютерних наук, група СНн-61 // Тернопіль, 2023 // С. , рис. – , табл. – , кресл. – , додат. – , бібліогр. – .

***Ключові слова:*** захист інформації, шифрування, безпека, методи захисту інформації, Blowfish.

Кваліфікаційна робота присв’ячена розробці програмного забезпечення яке дозволяє здійснювати захист інформації.

У першому розділ в процесі проектування було розглянуто різні алгоритми шифрування, зважено їхні переваги і недоліки.

У другому розділі розглянуті методи криптографічного захисту інформації та описаний Blowfish.

У третьому розділі та побудовано програму – текстовий редактор з вбидованим в нього алгоритмом шифрування повідомлень за допомогою Blowfish. Вона є поєднанням механізму захисту з функціями текстового редактора. Написана для даної роботи програма може ефективно використовуватись для шифрування повідомлень для подальшої передачі незахищеними каналами або зберігання на загальнодоступних носіях інформації.

У четвертому розділі роботи описані основні питання охорони праці та безпеки в надзвичайних ситуаціях.

Об’єктом дослідження даної магістерської роботи є основні види загроз безпеці інформації та методи й засоби її захисту, криптографічні методи захисту інформації.

# ANNOTATION

Methods of protecting information in unprotected communication channels // Qualification work of the educational level "Master" // Semenyuk Volodymyr Oleksandrovich // Ternopil National Technical University named after Ivan Pulyu, Faculty of Computer Information Systems and Software Engineering, Department of Computer Sciences, group CHn-61 // Ternopil, 2023 // C. , fig. - , tab. - , chair. - , add. – , bibliography - .

***Keywords*:** information protection, encryption, security, information protection methods, Blowfish.

The qualification work is devoted to the development of software that allows for information protection.

In the first section, various encryption algorithms were considered in the design process, their advantages and disadvantages were weighed.

In the second chapter, methods of cryptographic protection of information are considered and Blowfish is described.

In the third section, a program is built - a text editor with an algorithm for encrypting messages using Blowfish built into it. It is a combination of a protection mechanism with the functions of a text editor. The program written for this work can be effectively used to encrypt messages for further transmission through unsecured channels or storage on publicly available media.

The fourth chapter of the work describes the main issues of labor protection and safety in emergency situations.

The object of research of this master's thesis is the main types of threats to information security and methods and means of its protection, cryptographic methods of information protection.

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ІС – Інформаційних систем.

ПЗ – Програмне забезпечення.

ПК – Персональний комп’ютер.

# ЗМІСТ

ВСТУП ..... 6
1 ВИДИ ЗАГРОЗ БЕЗПЕЦІ ІНФОРМАЦІЇ ..... 9
1.1 Канали зв’язку ..... 9
1.1.1 Загрози безпеці інформації ..... 10
1.2 Класифікація алгоритмів шифрування ..... 12
1.3 Переваги блочного шифрування ..... 15
1.4 Поняття алгоритм Blowfish ..... 16
1.5 Висновок до першого розділу ..... 17
2 МЕТОДИ Й ЗАСОБИ ЗАХИСТУ ІНФОРМАЦІ. BLOWFISH ..... 18
2.1 Захист інформації в каналах зв'язку ..... 19
2.2 Захист інформації в месенджерах ..... 21
2.3 Застосування Blowfish для захисту інформації ..... 22
2.4 Висновок до другого розділу ..... 23
3 ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМУ ЗАХИСТУ ІНФОРМАЦІЇ ..... 24
3.1 Мережі Фейстела ..... 24
3.2 Опис Blowfish ..... 25
3.3 Безпека Blowfish ..... 28
3.4 Реалізація програмного забезпечення ..... 29
3.5 Висновок до третього розділу ..... 31
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ ..... 32
4.1 Джерела іонізуючого, електромагнітного та віброакустичного випромінювання ..... 32
4.2 Соціальні небезпеки ..... 37
4.3 Висновок до четвертого розділу ..... 39
ВИСНОВКИ ..... 40
ПЕРЕЛІК ДЖЕРЕЛ ..... 41
ДОДАТКИ

# ВСТУП

**Актуальність теми.** В наш час інформаційні технології розвиваються надзвичайно швидкими темпами. Дедалі більше сфер людської діяльності модулюються і обробляються за допомогою комп’ютерних систем. Такий науковий прогрес призвів до того, що інформаційна безпека не тільки стає обов'язковою, вона також являється однією з характеристик інформаційних систем (ІС). Існує досить великий клас систем обробки інформації, при розробці яких фактор безпеки відіграє ключову роль (наприклад, банківські інформаційні системи).

Під безпекою ІС розуміють захищеність системи від випадкового або навмисного втручання в нормальний процес її функціонування, від спроб викрадення (несанкціонованого одержання) інформації, модифікації або фізичного руйнування її компонентів. Інакше кажучи, це здатність протидіяти різним негативним впливам на ІС.

В наш час світ стурбований станом захисту національних інформаційних ресурсів у зв'язку з розширенням доступу до них через відкриті інформаційні мережі типу Internet. Крім того, що повсюдно збільшується число комп'ютерних злочинів, реальною стала загроза інформаційних атак на більш високому рівні для досягнення політичних і економічних цілей.

До найбільш уразливих місць, через які зазвичай намагаються проникнути зловмисники, належать відкриті системи, системи, що підтримують технологію підключення периферійних пристроїв у режимі plug-and-play, засоби централізованої віддаленої підтримки, канали комутації віддаленого доступу та недостатньо надійні технології шифрування. Разом з тим компанії цілком в змозі повністю забезпечити досить надійний захист інформації при відносно невеликих витратах.

Зловживання інформацією, яка циркулює в ІС або передається каналами зв'язку, удосконалювалися не менш інтенсивно, ніж методи захисту від них. На сьогодні для забезпечення захисту інформації потрібна не просто розробка приватних механізмів захисту, а реалізація системного підходу, який включає

комплекс взаємозалежних заходів (використання спеціальних технічних і програмних засобів, організаційних заходів, нормативно-правових актів, морально-етичних заходів протидії і т.д.). Комплексний характер захисту випливає з комплексних дій зловмисників, які прагнуть будь-якими способами добути важливу для них інформацію.

Канал зв'язку — в інформаційних технологіях засіб для передачі сигналів між пристроями, які розташовані на відстані один від одного.

**Мета і задачі дослідження.** Метою даної кваліфікаційної роботи освітнього рівня «Магістр» є аналіз методів шифрування та розробка програми для шифрування інформації яка може бути використана в незахищених каналах зв’язку. Для досягнення поставленої мети потрібно виконати ряд завдань, зокрема:

- Проаналізувати стан досліджень в області захисту даних.
- Провести огляд існуючих на даний час методів шифрування інформації.
- Обгрунтувати та застосувати алгоритм шифрування інформації Blowfish.
- Розробити програмне забезпечення для шифрування ініормації, яке може бути використано в незахищених каналах зв’язку.

**Об’єкт дослідження** Процеси шифрування інформації в незахищених каналах зв’язку.

**Предмет дослідження.** Методи шифрування інформації в тому числі симетричні та асиметричні.

**Наукова новизна одержаних результатів** кваліфікаційної роботи полягає у тому, що отримав подальший розвиток метод шифрування інформації Blowfish.

**Практичне значення одержаних результатів.** Розроблено проагарме забезпечення яке дозволяє шифрувати інформацію в незахищених каналах зв’язку.

**Апробація результатів магістерської роботи.** Основні результати проведених досліджень обговорювались на XI науково-технічній конференції «Інформаційні моделі, системи та технології» ІМСТ-2023 Тернопільського національного технічного університету імені Івана Пулюя (м. Тернопіль, 2023 р.).

**Публікації.** Основні результати кваліфікаційної роботи опубліковано у двох працях конференції (Див. додатки А).

**Структура й обсяг кваліфікаційної роботи.** Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, списку літератури з 28 найменувань та 2 додатків. Загальний обсяг кваліфікаційної роботи складає 62 сторінки, з них 43 сторінки основного тексту, який містить 7 рисунків.

# 1 ВИДИ ЗАГРОЗ БЕЗПЕЦІ ІНФОРМАЦІЇ

## 1.1 Канали зв’язку

Канали зв'язку - це шляхи або середовища, через які відбувається передача інформації між відправником та отримувачем. Канали зв'язку можуть бути фізичними (матеріальними) або безпровідними (безпровідними). Тут розглянемо деякі основні типи каналів зв'язку:

Паралельні Канали:

Опис: Використовуються для передачі бітів одночасно по кількох провідниках або каналах.

Застосування: Класичний приклад - паралельний інтерфейс для підключення пристроїв, таких як принтери або жорсткі диски, до комп'ютерів.

Серійні Канали:

Опис: Передача бітів в один ряд підряд через один провідник або канал.

Застосування: Серійні порти, такі як USB, RS-232, використовуються для з'єднання різних пристроїв.

Оптоволоконні Канали:

Опис: Використовують оптичні волокна для передачі світлових сигналів.

Застосування: Високошвидкісні мережі, телефонія, передача даних на великі відстані.

Безпровідні Канали:

Опис: Використовують електромагнітні хвилі для передачі сигналів без провідникового з'єднання.

Застосування: Мобільні телефони, Wi-Fi, Bluetooth, супутникові зв'язок.

Канали Низької Частоти (LF):

Опис: Використовуються для передачі сигналів на низьких частотах, зазвичай в області 30 кГц - 300 кГц.

Застосування: Системи ідентифікації (RFID), низькочастотний зв'язок.

Канали Середньої Частоти (MF):

Опис: Використовуються для передачі сигналів на середніх частотах, приблизно в області 300 кГц - 3 МГц.

Застосування: Деякі системи радіотрансляції.

Канали Високої Частоти (HF):

Опис: Використовуються для передачі сигналів на високих частотах, зазвичай в області 3 МГц - 30 МГц.

Застосування: Короткохвильова радіотрансляція, аматорське радіо.

Канали Дуже Високої Частоти (VHF) та Ультрависокої Частоти (UHF):

Опис: Використовуються для передачі сигналів на високих частотах, від 30 МГц до 300 МГц (VHF) та від 300 МГц до 3 ГГц (UHF).

Застосування: Телебачення, мобільна зв'язок, радіо.

Ці різновиди каналів використовуються в різних сферах комунікації і мають свої особливості та застосування.

### 1.1.1 Загрози безпеці інформації

Загрози безпеці інформації включають різноманітні можливості або події, які можуть призвести до порушення конфіденційності, цілісності та доступності інформації. Захист інформації від таких загроз є критичним завданням для забезпечення ефективності та безпеки різних організацій. Деякі загрози безпеки інформації включають:

Кібератаки:

Опис: Зловмисники можуть намагатися проникнути в інформаційні системи, використовуючи різні методи, такі як віруси, черв'яки, троянці, фішинг, інженерія соціальних мереж тощо.

Витік Інформації:

Опис: Несанкціоноване розголошення конфіденційної інформації, часто з метою отримання прибутку або завдання шкоди організації.

Втрата Даних:

Опис: Випадки випадкового чи наміреного знищення або втрати даних, що може призвести до непередбачених наслідків.

Неавторизований Доступ:

Опис: Зловмисники можуть намагатися отримати несанкціонований доступ до систем, даних чи інших ресурсів.

Дефекти в Програмному Забезпеченні:

Опис: Наявність програмних або апаратних дефектів, що можуть бути використані для атак або витоку інформації.

Фізичні Загрози:

Опис: Загрози, пов'язані з фізичним доступом до інформаційних ресурсів, такі як крадіжка обладнання, пожежа, повідомлення про стихійні лиха тощо.

Соціальний Інженеринг:

Опис: Зловмисники можуть намагатися отримати інформацію, використовуючи маніпуляцію людей, наприклад, шляхом використання психологічних трюків або імітації авторитетних осіб.

Недостатня Автентифікація та Авторизація:

Опис: Слабкі системи автентифікації та авторизації можуть дозволяти несанкціонований доступ до інформаційних ресурсів.

Зміна Конфіденційності та Цілісності Даних:

Опис: Загрози, спрямовані на зміну чи вплив на конфіденційність та цілісність даних, такі як маніпулювання або фальсифікація інформації.

Відмова в Обслуговуванні (DoS) та Розподілена Відмова в Обслуговуванні (DDoS):

Опис: Атаки, що спрямовані на перевантаження системи або мережі, завдаючи шкоди їхній доступності.

Для ефективного захисту інформації важливо вживати широкий спектр заходів безпеки, таких як шифрування, аутентифікація, авторизація, резервне копіювання даних, антивірусні програми, фаєрволи та навчання персоналу щодо безпекових практик.

## 1.2 Класифікація алгоритмів шифрування

Алгоритми шифрування класифікуються за декількома параметрами, такими як використовуваний ключ, метод дії та можливість використання одного ключа для шифрування та розшифрування (симетричні або асиметричні). Ось деякі основні класифікації алгоритмів шифрування:

За Ключем:

**Симетричне (одноключове) Шифрування:**

Використовується один і той же ключ для шифрування та розшифрування даних.

Приклади: DES (Data Encryption Standard), AES (Advanced Encryption Standard), IDEA (International Data Encryption Algorithm).

**Асиметричне (двоключове) Шифрування:**

Використовує пару ключів: публічний та приватний. Публічний ключ використовується для шифрування, а приватний — для розшифрування.

Приклади: RSA (Rivest–Shamir–Adleman), ECC (Elliptic Curve Cryptography).

За Методом Дії:

**Блочне Шифрування:**

Дані розбиваються на блоки, і кожен блок оброблюється окремо.

Приклади: DES, AES, Blowfish.

**Потокове Шифрування:**

Дані шифруються біт за бітом або байтом.

Приклади: RC4 (Rivest Cipher 4), A5/1.

За Використанням:

**Шифри Заміни:**

Кожен символ чи блок замінюється іншим відповідно до заданого правила.

Приклади: Шифр Цезаря, Шифр Віженера.

**Шифри Перестановки:**

Символи переставляються місцями в певному порядку.

Приклади: Шифр Скайтала, Шифр Рейтца.

**Шифри Заміни та Перестановки:**

Комбінація методів заміни та перестановки.

Приклад: DES.

За Кількістю Ключів:

**Одноключові:**

Використовується тільки один ключ для обох операцій (шифрування та розшифрування).

Приклади: DES, AES.

**Двоключові:**

Використовується пара ключів, один для шифрування, інший для розшифрування.

Приклад: RSA.

Ці класифікації допомагають у розумінні особливостей та властивостей різних алгоритмів шифрування і їхнього використання в конкретних сценаріях. Важливо враховувати вимоги безпеки та специфіку конкретного застосування при виборі алгоритму шифрування.

Розглянемо блочне шифрування.

Блочне шифрування — це метод криптографічного шифрування, при якому дані розбиваються на фіксовані блоки (зазвичай 64 або 128 біт), які обробляються незалежно один від одного. Кожен блок обробляється індивідуально, і результат обробки залежить від тексту блоку і ключа шифрування. Блочне шифрування може бути використане як для шифрування, так і для розшифрування.

Основні принципи блочного шифрування:

Режими Роботи:

Режими роботи визначають, які блоки даних впливають на один одного і як вони об'єднуються. Деякі популярні режими включають ECB (Electronic Codebook), CBC (Cipher Block Chaining), OFB (Output Feedback), CFB (Cipher Feedback), CTR (Counter) та інші.

Ключ: Блочні шифри використовують ключ для визначення того, як кожен блок даних буде оброблятися. Ключ визначає конфігурацію шифрування та розшифрування.

Перестановка та Заміна:

Блоки даних піддаються перестановці (зміні місцями) та заміні (заміні символів) відповідно до ключа. Ці операції забезпечують конфузію та дифузію, що роблять шифрування більш безпечним.

Раунди:

Багато блочних шифрів використовують концепцію раундів, де кожен раунд включає в себе послідовні операції над блоком даних з використанням ключа. Це додає складність шифруванню та робить його більш стійким.

Функції Захисту Ключа:

Блочні шифри зазвичай включають в себе механізми для захисту ключа, такі як ключові розклади або підтвердження цілісності ключа.

Деякі популярні блочні шифри включають:

AES (Advanced Encryption Standard): Застосовується широко та визнаний як стандарт у багатьох сферах.

DES (Data Encryption Standard): Хоча застарілий, був популярним протягом деякого часу.

Triple DES (3DES): Розширена версія DES для покращення безпеки.

Blowfish та Twofish: Асиметричні шифри, що надають високу швидкодію та безпеку.

Блочне шифрування використовується в різних застосунках, таких як захист конфіденційності даних, створення цифрових підписів та інші області криптографії.

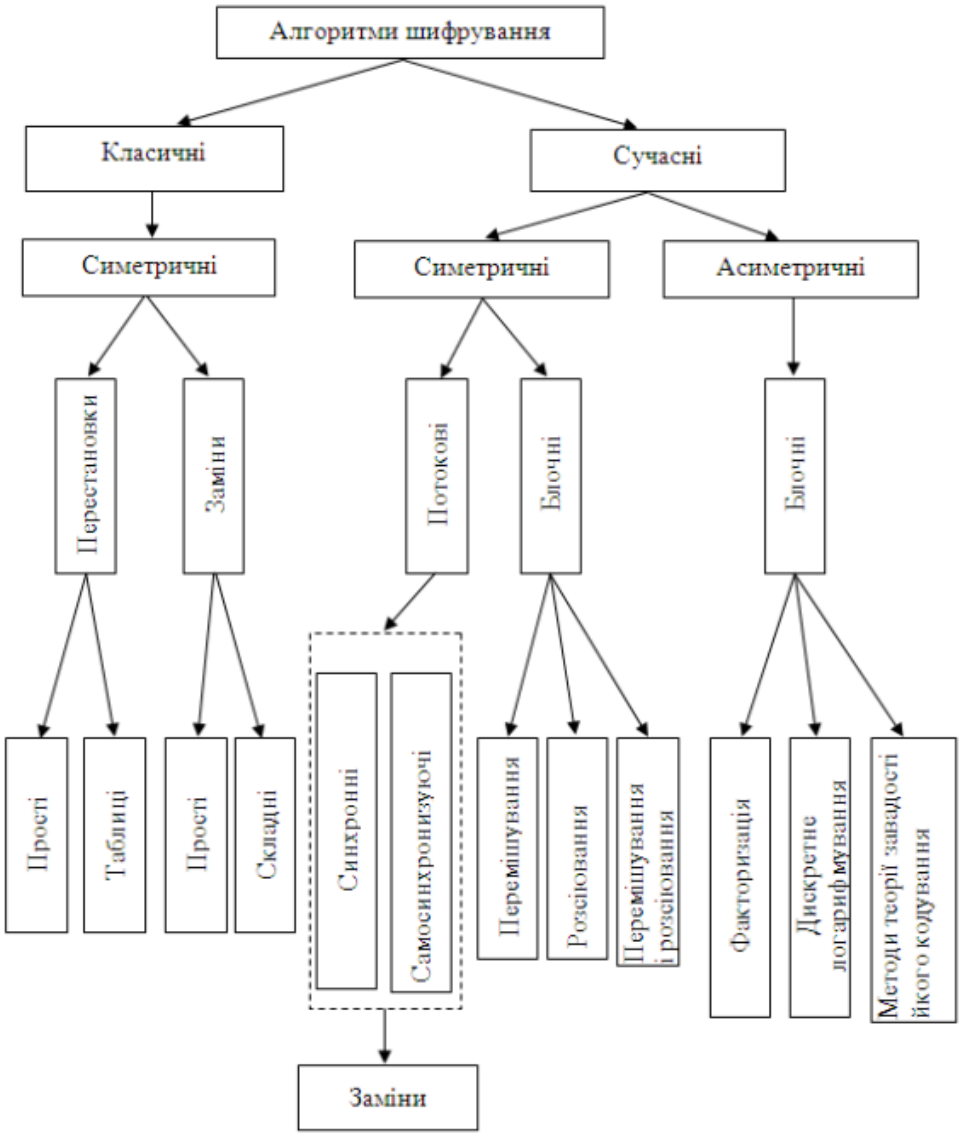


Рисунок 1.1 – Класифікація алгоритмів шифрування

На рисунку 1.1 наведена структурна схема класифікації алгоритмів шифрування.

### 1.3 Переваги блочного шифрування

Блочне шифрування має кілька переваг, які роблять його ефективним та широко використовуваним в різних областях криптографії та інформаційної безпеки. Основні переваги блочного шифрування включають:

Ефективність:

Блочні шифри зазвичай працюють дуже швидко, особливо на спеціалізованому обладнанні. Це дозволяє їх використовувати для шифрування та розшифрування великої кількості даних.

Стійкість:

З правильним вибором і використанням блочних шифрів, їх можна зробити дуже стійкими до різноманітних атак, включаючи криптоаналіз та злам з використанням сучасних обчислювальних ресурсів.

Гнучкість:

Багато блочних шифрів можна конфігурувати за допомогою параметрів, таких як розмір ключа та режим роботи. Це дозволяє вибирати оптимальні налаштування для конкретних вимог безпеки.

Використання Режимів Роботи:

Блочне шифрування може використовувати різні режими роботи (ECB, CBC, CFB, OFB, CTR), що дозволяє налаштовувати процес шифрування залежно від конкретних вимог додатка чи системи.

Використання Різних Розмірів Блоків:

Деякі блочні шифри дозволяють використовувати блоки різного розміру в залежності від потреб застосування.

Велика Опірність до Зміни Даних:

Зміни в одному біті вхідних даних приведуть до значних змін у виході шифру (ефект лавинного ефекту), що робить його стійким до найменших змін вхідних даних.

Додаткові Заходи Безпеки:

За допомогою додаткових методів, таких як раунди шифрування та функції захисту ключа, блочне шифрування може надавати високий рівень безпеки.

Загалом, блочне шифрування залишається однією з основних технологій для забезпечення конфіденційності даних та безпеки інформації.

### 1.4 Поняття алгоритм Blowfish

BLOWFISH - це алгоритм блочного шифрування, розроблений Брюсом Шнайером у 1993 році. Цей алгоритм входить до числа симетричних шифрів, що використовують один ключ як для шифрування, так і для дешифрування даних. BLOWFISH став відомим своєю швидкодією та здатністю до ефективної роботи з різними розмірами ключів.

Основні характеристики BLOWFISH:

Блочний Шифр: BLOWFISH шифрує дані блоками фіксованого розміру. В оригінальній версії це 64 біта, але алгоритм може працювати з більшими блоками.

Ключова Довжина: BLOWFISH приймає ключі довжиною від 32 до 448 бітів. Розмір ключа може бути будь-яким кратним 8-ми бітам.

Раунди та Підключі: Алгоритм використовує ітеративний процес, що включає в себе серію раундів, кожен з яких використовує свій підключ для обробки даних.

Замкнена Структура: BLOWFISH використовує замкнуту структуру, що означає, що в процесі шифрування і дешифрування використовується один і той же алгоритм.

BLOWFISH використовується в різних системах для шифрування даних, таких як віртуальні приватні мережі (VPN), системи електронного банківського обслуговування та інші додатки, де забезпечення конфіденційності даних є важливим завданням.

## 1.5 Висновок до першого розділу

В першому розділі кваліфікаційної роботи освітнього рівня «Магістр» описано канали зв’язку, переваги блочного шифрування, загрози безпеці інформації, коротко описаний алгоритм BLOWFISH.

# 2 МЕТОДИ Й ЗАСОБИ ЗАХИСТУ ІНФОРМАЦІ. BLOWFISH

Існує багато методів та засобів для захисту інформації. Забезпечення безпеки даних — це комплексний підхід, який включає в себе технічні, організаційні та правові заходи. Ось деякі з найважливіших методів та засобів захисту інформації:

1. Шифрування Даних:

Опис: Процес перетворення чіткого тексту в нечитабельний код (шифр) за допомогою ключа. Забезпечує конфіденційність даних.

Застосування: Блочне шифрування, потокове шифрування, асиметричне шифрування (RSA, ECC), симетричне шифрування (AES, DES).

2. Мережеві Засоби Захисту:

Опис: Використання брандмауерів, вторгнення з детекторами, VPN (віртуальні приватні мережі), контроль доступу до мережевих ресурсів.

Застосування: Firewalls, IDS (Intrusion Detection System), IPS (Intrusion Prevention System).

3. Контроль Доступу:

Опис: Визначення та обмеження прав доступу користувачів та систем до різних ресурсів.

Застосування: Управління ідентифікацією та аутентифікацією, ролевий доступ, біометричні системи.

4. Антивірусне та Антималware Захист:

Опис: Виявлення, блокування та видалення вірусів, троянських програм, шпигунського програмного забезпечення та іншого шкідливого ПЗ.

Застосування: Антивірусні програми, антималware рішення.

5. Фізичний Захист:

Опис: Заходи, спрямовані на фізичне обмеження доступу до інформаційних ресурсів.

Застосування: Замки, камери відеоспостереження, контроль доступу до приміщень.

6. Резервне Копіювання та Відновлення Даних:

Опис: Регулярне створення резервних копій даних та можливість їхнього відновлення в разі втрати або пошкодження.

Застосування: Архівування, системи резервного копіювання.

7. Системи Виявлення та Реагування на Інциденти (SIEM):

Опис: Моніторинг та аналіз подій в інформаційній системі для виявлення можливих інцидентів безпеки.

Застосування: Системи SIEM, виявлення аномалій.

8. Спеціалізовані Протоколи та Стандарти:

Опис: Використання стандартів та протоколів, які гарантують безпеку даних (SSL/TLS для захищеної передачі даних по мережі, HTTPS, IPsec).

Застосування: Захищені мережеві комунікації, безпечні протоколи.

9. Організаційні та Правові Заходи:

Опис: Визначення політик безпеки, тренінги з безпеки для персоналу, дотримання відповідних норм та законодавства.

Застосування: Політики безпеки, навчання та свідомість персоналу, дотримання стандартів безпеки.

Ці заходи можуть використовуватися окремо чи в поєднанні для створення комплексної системи захисту інформації. Важливо регулярно оновлювати та адаптувати заходи безпеки відповідно до змін у загрозах та технологіях.

## 2.1 Захист інформації в каналах зв'язку

Захист інформації в каналах зв'язку вельми важливий для забезпечення конфіденційності, цілісності та доступності передачі даних. Враховуючи сучасний характер комунікацій, особливо в мережевих середовищах, існує кілька ключових аспектів захисту інформації в каналах зв'язку:

1. Шифрування Трафіку:

Використання криптографічних алгоритмів для захисту вмісту передачі даних. TLS/SSL для захищеної передачі даних через Інтернет, IPsec для захисту мережевого трафіку.

2. Віртуальні Приватні Мережі (VPN):

Встановлення безпечного тунелю між двома або більше вузлами, щоб зашифрувати весь мережевий трафік між ними. VPN дозволяють забезпечити конфіденційність інформації, навіть у відкритих мережах.

3. Фізичний Захист Каналів:

Захист фізичних ліній передачі даних від несанкціонованого доступу або перехоплення. Використання захищених кабелів, шифрування фізичного середовища передачі даних.

4. Контроль Доступу:

Встановлення обмежень на доступ до мережевих ресурсів та комунікаційних засобів. Захищені паролі, ідентифікація та автентифікація для забезпечення тільки санкціонованого доступу.

5. Захист від Вторгнень:

Використання систем виявлення та запобігання вторгненням (IDS/IPS) для виявлення та блокування небажаних або зловмисних дій у мережі.

6. Криптографічні Протоколи та Стандарти:

Використання надійних криптографічних протоколів та стандартів для захисту та обміну ключами, наприклад, для TLS/SSL або IPsec.

7. Аутентифікація та Авторизація:

Забезпечення віртуального підтвердження ідентичності користувачів та пристроїв, а також контроль доступу до різних ресурсів.

8. Фільтрація Трафіку:

Використання засобів фільтрації трафіку для блокування шкідливих або небажаних пакетів даних.

9. Фізичний Контроль Пристроїв:

Обмеження доступу до пристроїв зв'язку та мережевого обладнання, фізичний контроль за їхнім розташуванням та доступом.

10. Навчання та Свідомість Користувачів:

Навчання користувачів щодо засобів захисту, підвищення свідомості щодо потенційних загроз та правил безпеки в області зв'язку.

Загалом, в комплексі ці заходи дозволяють ефективно захищати інформацію в каналах зв'язку від різних загроз та атак.

## 2.2 Захист інформації в месенджерах

Захист інформації в сучасних месенджерах є критично важливим, оскільки вони часто використовуються для обміну особистими повідомленнями та конфіденційною інформацією. Нижче подано деякі ключові аспекти захисту інформації в месенджерах:

1. Шифрування Кінцевий-до-Кінця:

Сучасні месенджери повинні використовувати протоколи шифрування кінцевий-до-кінця (End-to-End Encryption, E2EE). Це означає, що дані шифруються на пристрої відправника і розшифровуються лише на пристрої отримувача. Це гарантує, що навіть сервіс месенджера не може прочитати ваші повідомлення.

2. Аутентифікація та Авторизація:

Важливо впевнитися, що тільки правомірні користувачі мають доступ до месенджера. Використання сильних методів аутентифікації, таких як двофакторна аутентифікація (2FA), може підвищити безпеку входу.

3. Захист Даних на Прискорених Серверах:

Якщо месенджер використовує хмарні сервіси для зберігання повідомлень, важливо, щоб дані також були зашифровані на сервері. Це забезпечить додатковий захист від можливих атак або витоків даних.

4. Аудіт та Контроль Доступу:

Реалізація системи аудиту та контролю доступу для виявлення несанкціонованого доступу до системи та моніторингу активності користувачів.

5. Знищення Повідомлень та Даних:

Можливість автоматичного або ручного видалення повідомлень з пристроїв учасників чату, а також з серверів, щоб забезпечити приватність та уникнути можливості витоку даних.

6. Відсутність Зберігання Ключів:

Найкращі практики передбачають відсутність зберігання ключів шифрування на серверах месенджера. Ключі повинні залишатися тільки на пристроях користувачів.

7. Оновлення та Патчі Безпеки:

Регулярні оновлення та вчасні випуски патчів для виправлення виявлених безпекових уразливостей та забезпечення безпеки програмного забезпечення.

8. Сповіщення про Нові Входження:

Система сповіщення про нові входження або незвичайну активність може виявити можливі вторгнення або несанкціонований доступ.

9. Політика Безпеки та Практики Користувачів:

Забезпечення введення ефективної політики безпеки, а також навчання користувачів стосовно практик безпеки в месенджерах.

10. Анонімізація та Псевдонімізація:

Забезпечення можливості користувачам залишатися анонімними чи використовувати псевдоніми для додаткового рівня конфіденційності.

Загальною метою є створення механізмів, що забезпечують приватність та безпеку для користувачів, навіть у віртуальних середовищах месенджерів.

## 2.3 Застосування Blowfish для захисту інформації

Шифр BLOWFISH може використовуватися у месенджерах для шифрування текстових повідомлень та інших конфіденційних даних, забезпечуючи таким чином конфіденційність інформації, що обмінюється між користувачами. Це допомагає у захисті особистої інформації та забезпеченні приватності під час взаємодії у месенджерах. Важливо зазначити, що, незважаючи на ефективність BLOWFISH, існують інші сучасні шифри, такі як AES (Advanced Encryption Standard), які також можуть бути використані для забезпечення безпеки даних.

Основні етапи використання BLOWFISH у месенджерах включають:

Генерація Ключа:

Ключ для шифрування та розшифрування повідомлень генерується на кожному з пристроїв учасників чату. Генерація ключа може відбуватися за допомогою вибору випадкових бітів або інших методів генерації ключа.

Шифрування Повідомлень:

Текстові повідомлення шифруються за допомогою алгоритму BLOWFISH перед відправкою. Ключ, який використовується для шифрування, відомий тільки учасникам чату.

Передача Зашифрованих Повідомлень:

Зашифровані повідомлення передаються через мережу. Тільки коректно налаштовані пристрої, які мають правильний ключ, можуть розшифрувати та прочитати ці повідомлення.

Розшифрування на Пристрої Отримувача:

Пристрій отримувача, який має вірний ключ, розшифровує отримане зашифроване повідомлення, забезпечуючи можливість прочитати оригінальний текст.

Збереження Ключів Локально:

Ключі шифрування зберігаються локально на пристроях користувачів, і вони не передаються чи не зберігаються на серверах месенджера.

Управління Ключами:

Існує потреба у впровадженні ефективного управління ключами, включаючи їхню безпеку, оновлення та ротацію.

Шифрування, включаючи використання BLOWFISH, грає важливу роль у захисті конфіденційності в месенджерах, забезпечуючи безпечний обмін інформацією між користувачами.

## 2.4 Висновок до другого розділу

В другому розділі кваліфікаційної роботи описано підходи до захисту інформації в незахищенеих каналаз зв’язку, описано захист інформації в сучасних месенджерах, та описано застосування Blowfish для задачі захисту інформації в незахищених каналах зв’язку.

# 3 ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМУ ЗАХИСТУ ІНФОРМАЦІЇ

## 3.1 Мережі Фейстела

Більшість блочних алгоритмів є *мережами Фейстела* (Felstel networks). Ця ідея датується початком 70-х років. Візьмемо блок довжиною *n* і розділимо його на дві половини довжиною *n/2*: *L* і *R*. Можна визначити ітеративний блочний шифр, у якому результат *j*-го етапу визначається результатом попереднього етапу:

$$L_i = R_{i-1}$$

$$R_{i-1} = L_{i-1} \oplus f(R_{i-1}, K_i) \tag{3.1}$$

$K_i$ - це підключ, який використовується на *j*-му етапі, а *f* - це довільна функція етапу.

Цю концепцію можна побачити в DES, Lucifer, FEAL, Khufu, Khafre, LOKI, COST, CAST, Blowfish і інших алгоритмах. Чому це так важливо? Гарантується, що ця функція є оборотною. Через те, що для об'єднання лівої половини з результатом функції етапу використовується XOR, наступний вираз обов'язково є істинним:

$$R_{i-1} = L_{i-1} \oplus f(R_{i-1}, K_i) \oplus f(R_{i-1}, K_i) = L_{i-1} \tag{3.2}$$

Гарантується, що шифр, який використовує таку конструкцію, оборотний, якщо можна відновити вихідні дані *f* на кожному етапі. Сама функція *f* неважлива, вона не обов'язково має бути оборотною. Ми можемо спроектувати *f* настільки складною, наскільки захочемо. Не потрібно реалізовувати два різних алгоритми - один для шифрування, а інший для дешифрування. Структура мережі Фейстела автоматично подбає про це.

## 3.2 Опис Blowfish

Blowfish являє собою 64-бітовий блочний шифр із ключем змінної довжини. Алгоритм складається із двох частин: розгортання ключа й шифрування даних. Розгортання ключа перетворить ключ довжиною до 448 біт у кілька масивів підключів, загальним об'ємом 4168 байт.

Шифрування даних складається з простої функції, яка послідовно виконується 16 разів. Кожний етап складається із залежної від ключа перестановки й залежної від ключа й даних підстановок. Використовуються тільки додавання й XOR 32-бітових слів. Єдиними додатковими операціями на кожному етапі є чотири витяги даних з індексованого масиву.

В Blowfish використовується багато підключів. Ці підключі повинні бути розраховані до початку шифрування або дешифрування даних.

P-Масив складається з 18  32-бітових підключів: P1,P2, .. ., P18

Кожен із чотирьох 32-бітових S-блоків містить 256 елементів:

$S_{1,0}$, $S_{1,1}$, …, $S_{1,255}$

$S_{2,0}$, $S_{2,1}$, …, $S_{2,255}$

$S_{3,0}$, $S_{3,1}$, …, $S_{3,255}$

$S_{4,0}$, $S_{4,1}$, …, $S_{4,255}$

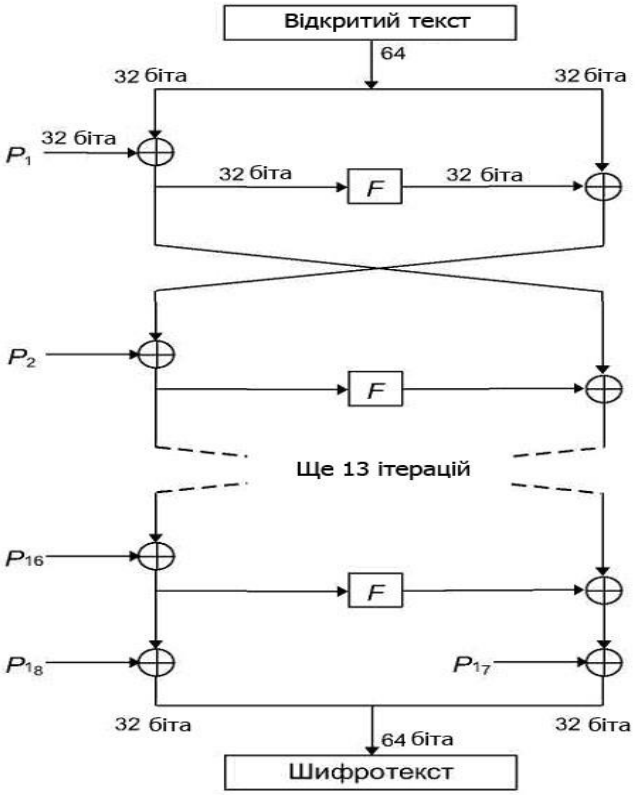


Рисунок 3.1 – Blowfish

Blowfish є мережею Фейстела, що складається з 16 етапів. На вхід подається 64-бітовий елемент даних *x*.

Для шифрування:

Розіб'ємо *x* на дві 32-бітових половини: $x_L, x_R$

Для i= 1 по 16:

$x_L = x_L \oplus P_{18}$

$x_R = F(x_L) \oplus x_R$

Переставити $x_L$ і $x_R$ (крім останнього етапу.)

$x_R = x_R \oplus P_{17}$

$x_L = x_L \oplus P_{18}$

Об'єднати $x_L$ і $x_R$

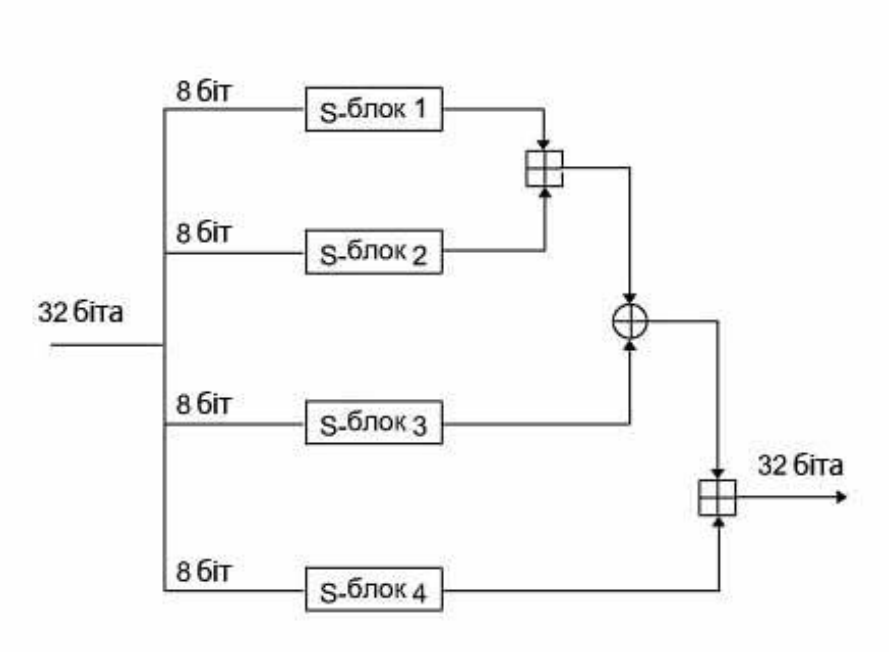


Рисунок 3.2 – Функція F

Функція F являє собою наступне (див. рис. 3.2):

Розділити *x* на чотири 8-бітових частини: *a, b, c і d*

$$F(x_L) = ((S_{1,a} + S_{2,b} \bmod 2^{32}) \oplus S_{3,c}) + S_{4,d} \bmod 2^{32} \quad (3.3)$$

Дешифрування виконується точно так, як і шифрування, але $P_1, P_2, ..., P_{18}$ використовуються у зворотному порядку.

У реалізаціях Blowfish, для яких потрібно дуже велика швидкість, цикл повинен бути розгорнутий, а всі ключі повинні зберігатися в кеші.

Підключі розраховуються за допомогою спеціального алгоритму. От яка точна послідовність дій.

(1) Спочатку P-масив, а потім чотири S-блоки один по одному ініціалізуются фіксованим рядком. Цей рядок складається із шістнадцяткових цифр $\pi$.

(2) Виконується XOR $P_1$ з першими 32 бітами ключа, XOR $P_2$ із другими 32 бітами ключа, і так далі для всіх бітів ключа (до $P_{18}$). Використовується циклічно, поки для всього P-масиву не буде виконана операція XOR з бітами ключа.

(3) Використовуючи підключі, отримані на етапах (1) і (2), алгоритмом Blowfish шифрується рядок з одних нулів.

(4) $P_1$ і $P_2$ заміняються результатом етапу (3).

(5) Результат етапу (3) шифрується за допомогою алгоритму Blowfish і змінених підключів.

(6) $P_3$ і $P_4$ заміняються результатом етапу (5).

(7) Далі в ході процесу всі елементи P-масиву й потім один по одному всі чотири S-блоки заміняються виходом алгоритму Blowfish, який постійно змінюється.

Усього для генерації всіх необхідних підключів потрібна 521 ітерація. Додатки можуть зберігати підключі - немає необхідності виконувати процес їхнього одержання багаторазово.

### 3.3 Безпека Blowfish

Серж Водене (Serge Vaudenay) досліджував Blowfish з відомими *S*-блоками й *r* етапами, диференціальний криптоаналіз може розкрити *P*-масив за допомогою $2^{8r+1}$ обраних відкритих текстів. Для деяких слабких ключів, які генерують погані S-блоки (імовірність вибору такого ключа становить 1 до $2^{14}$), ця ж атака розкриває P-масив за допомогою всього $2^{4r+1}$ обраних відкритих текстів. При невідомих S-блоках це розкриття може виявити використання слабкого ключа, але не може визначити сам ключ (ні S-блоки, ні P-масив). Це розкриття ефективне тільки проти варіантів зі зменшеним числом етапів і зовсім безплідне проти 16-етапного Blowfish.

Звичайно, важливе й розкриття слабких ключів, навіть якщо вони швидше за все не будуть використовуватися. Слабким є ключ, для якого два елементи даного S-блоку ідентичні. До виконання розгортання ключа неможливо визначити, чи він є слабким. Для цього доведеться виконати розгортання ключа й перевірити, чи немає в S однакових елементів.

На сьогоднішній день невідомо про успішний криптоаналіз Blowfish.

Kent Marsh Ltd. вмонтувала Blowfish у свій продукт забезпечення безпеки FolderBolt, призначений для Microsoft Windows і Macintosh. Алгоритм також входить в Nautilus і PGPfone.

### 3.4 Реалізація програмного забезпечення

Програма написана мовою програмування Objekt Pascal. Спочатку планувалося реалізувати алгоритм шифрування довільних файлів для чого заради економії системних ресурсів і збільшення швидкодії краще було б створити консольний додаток, але оскільки було вирішено реалізувати текстовий редактор з шифруванням даних, програму було розроблено в середовищі Borland Delphi.

Оскільки дуже важливою є доставка конфіденційних повідомлень по незахищених каналах зв’язку таким чином щоб зловмиснику не вдалось їх прочитати, то було вирішено створити простенький текстовий редактор, який можна використовувати замість текстового редактора Notepad (Блокнот), який застосовується в операційній системі Windows. На відміну від Блокнота при зберіганні тексту він шифрується. При виборі алгоритму кодування перевагу було віддано Blowfish в першу чергу через те, що він незапатентований і являється алгоритмом з вільним ліцензуванням. Крім того порівняно з такими загальновідомими алгоритмами шифрування як DES і IDEA він працює набагато швидше (Blowfish шифрує дані на 32-бітових мікропроцесорах із швидкістю 26 тактів на байт).

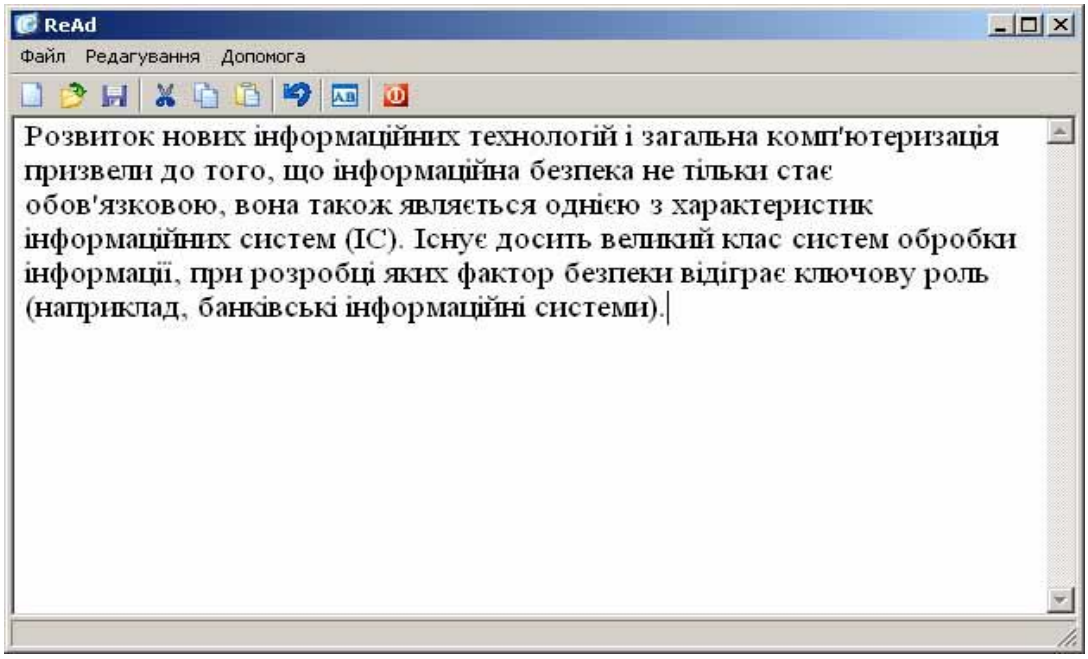


Рисунок 3.3 – Вікно текстового редактора ReAd при наборі тексту.

Даний редактор дає можливість виконувати елементарні операції над набраним текстом серед яких: вибір шрифту і його кольору, способу накреслення.

Редактор складається з робочої області для вводу тексту, меню команд і панелі інструментів. Крім того виклик команд для виконання операцій можна здійснювати за допомогою комбінацій клавіш.

Набравши текст, зазвичай потрібно його зберегти. Якраз тут ховається захист від „лінивих” користувачів – без введення ключової фрази (Рис. 5.2) інформація не збережеться, допоки вимога задати ключову фразу (Рис 5.3) не буде виконана.

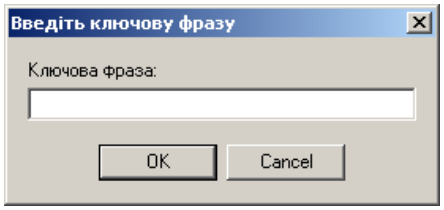


Рисунок 3.4 – Ввід ключової фрази

Після вводу ключа файл буде збережено в зашифрованому вигляді.

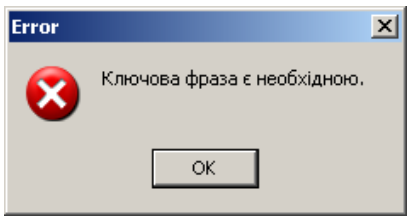


Рисунок 3.5 – Повідомлення про відсутність ключової фрази.

Особа, яка не знає ключової фрази не зможе прочитати повідомлення. Зловмисник, ввівши неправильний ключ відкриє файл, але повідомлення постане перед ним у вигляді „ієрогліфів”(Рис. 3.6.)

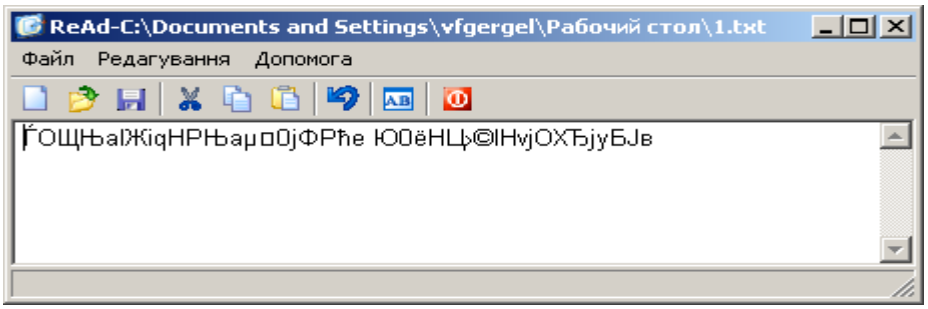


Рисунок 3.6 – Текст, показаний особі, яка не знає ключової фрази

Таким чином можна не перейматися, що хтось прочитає конфіденційне повідомлення. Такі текстові файли можна спокійно залишати на загальнодоступних ресурсах і ніхто без знання ключа не зможе прочитати їх.

### 3.5 Висновок до третього розділу

В третьому розділі кваліфікаційної роботи описано мережі Фейстела, алгоритм  Спроєктовано архітектуру Blowfish, описано реалізацію програмного забезпечення.

# 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

## 4.1 Джерела іонізуючого, електромагнітного та віброакустичного випромінювання

*Електромагнітні випромінювання* розрізняють за частотою коливань або довжиною хвилі. Найдовші хвилі – це коливання промислової або іншої звукової частоти, а також ультразвукові. Вони мають довжину хвилі понад 10 км (або частоту менш як 30 кГц), довгі і середні радіохвилі (від 10 км до 100 м або до 3 МГц) застосовують не тільки в радіотехніці, а й для плавлення металу, гартування деталей, сушіння деревини та ін. У промисловій електротермії для нагрівання діелектриків використовують також короткі радіохвилі (завдовжки 100—10 м або до 30 МГц), що, як і ультракороткі (10–1 м або до 300 МГц), належать до коливань ультрависокої частоти (УВЧ).

При промисловій частоті спеціальні заходи захисту від дії електричних полів доводиться застосовувати тільки під час обслуговування електроустановок напругою 330—500 кВ і вище. Тоді використовують спеціальні костюми і взуття, які дають можливість навідним зарядам стікати в землю без неприємних для людини відчуттів, а також екрануючі металеві козирки над робочими місцями (приводами роз'єднувачів та ін.). Використовувати ці козирки і костюми (так звані індивідуальні екрануючі комплекти) обов'язково тільки в розподільних пристроях напругою 750 кВ, під час робіт на опорах ЛЕП – 330–750 кВ або ж при напругах понад 5 кВ/м, коли перебування у такому електричному полі повинно тривати більше за гігієнічно допустимий час (понад 3 год при 5–10 кВ/м, 1,5 год при 10–15 кВ/м, 10 хв при 15–20 кВ/м і 5 хв при 20–25 кВ/м).

Тривале перебування на землі під ЛЕП теж шкідливе. Під крайньою фазою в середині прольоту на ЛЕП напругою 330 кВ напруга становить 6 кВ/м, а на ЛЕП-500 – 14 кВ/м. Тому під час польових робіт під ЛЕП напругою 330 кВ і вище треба враховувати цю обставину і краще використовувати трактори та інші машини з металевою кабіною або з встановленими зверху і з боків екранами, які виготовлені з металевої сітки.

Автомашини і трактори на пневматичних шинах заряджаються в електричному полі ЛЕП зарядами хоч і малого значення, але напругою, що становить кілька кіловольт. Дотик до них людини, яка стоїть на землі, не смертельний, але спричиняє болісний удар розрядним струмом, що може призвести до мимовільних рухів, а отже, і до механічних травм від дотику до рухомих частин та ін. Тому бажано не залишати машину під ЛЕП, якщо треба зупинитися, то до виходу з кабіни заземлити машину спеціальним заземлювачем (у вигляді гирі з штирем), прикріпленим до машини гнучким проводом. Заземлення може бути постійним у вигляді диска або сошника. Електроогорожі під ЛЕП 330–750 кВ краще взагалі не робити, бо в протяжних металевих частинах наводяться такі електрорушійні сили (ерс), що, наприклад, електроогорожа завдовжки 300 м навіть під ЛЕП напругою 220 кВ може при замиканні на опір 1000 Ом (людина) створити струм 10 мА, а на опір 500 Ом (корова) – 30 мА. Провід для виноградників, оскільки він не ізолюється спеціально від землі, порівняно безпечний, особливо при розташуванні перпендикулярно до траси ЛЕП і заземленні на кінцях.

Для захисту робітників від випромінювання високої частоти (ВЧ) і УВЧ застосовують екранування листовим металом високої електропровідності завтовшки не менш як 0,5 мм. Отвори в екрані для штурвалів і кнопок екранують металевою сіткою з вічками не більш як 4 х 4 мм. Екрани заземлюють. Максимально допустима напруженість електромагнітного поля випромінювання ВЧ і УВЧ на робочих місцях, згідно з ГОСТ 12.1.006-76, для частот 60 кГц дорівнює 50 В/м – 3 МГц, 20 В/м для частот 3–30 МГц, 10 В/м для частот 30–50 МГц і 5 В/м для частот 50–300 МГц. Тільки для індукційних плавильних печей і нагрівальних індикаторів тимчасово допускають 10 В/м через технічні труднощі повного екранування їх.

Напруженість магнітного поля не має перевищувати 5 А/м для частот 60 кГц – 1,5 МГц і 0,3 А/м для 30–50 МГц.

Тривалий вплив електромагнітних полів ВЧ і УВЧ з напругою, більшою за допустиму, призводить до функціональних змін у печінці, селезінці та особливо в центральній нервовій системі, які виявляються в головному болі, підвищеній

втомлюваності, порушенні сну, дратівливості, в уповільненні пульсу, зниженні кров'яного тиску. При дії випромінювань УВЧ також підвищується температура тіла. Коливання, які мають довжину хвилі від 1 м до 1 мм (частотою до 300 тис МГц), називаються надвисокочастотними (НВЧ), їх використовують у радіолокації і для деяких приладів. Розроблявся, наприклад, прилад для вимірювання жирності молока, який використовував НВЧ випромінювання. Гігієнічні норми НВЧ випромінювань визначаються в одиницях густини потоку потужності (вектора Пойнтінга) і залежить від тривалості впливу на людину: 0,1 Вт/м$^2$ при опроміненні протягом усього робочого дня; 10 Вт/м2 при опроміненні протягом 20 хв. на день. Але при цьому треба працювати в захисних окулярах, які зроблені з мідної сітки та екранують очі. Без цих окулярів уражується кришталик ока (утворюється катаракта).

Екрануванням захищаються і від інфрачервоних (теплових) променів (з довжиною хвилі 100–0,76 мкм).

Видиме світло має довжину хвилі від 0,76–0,38 мкм, а ультрафіолетове проміння – від 0,38 до 0,005 мкм, тобто до 5 нм. Ці промені виникають, наприклад, при електрозварюванні і можуть уражати очі (електроофтальмія) або спричинити запалення шкіри відкритих частин тіла. Для захисту очей і шкіри обличчя застосовують щитки зі світлофільтрами, а для захисту шкіри рук – рукавиці.

Рентгенівські промені (від 5 до 0,004 нм) використовують в установках промислової рентгеноскопії. Вони випромінюються і під час випробування кабелів та електроустаткування випрямленим струмом високої напруги. Застосований тут високовольтний кенотрон є джерелом м'якого рентгенівського випромінювання (тобто довжина хвилі понад 0,01 нм) і має бути екранований. Для екрана досить мати залізний лист завтовшки 0,5–1 мм. У промисловій рентгеноскопії застосовують також фартухи, рукавиці, шапочки з просвинцьованої гуми. Дозу рентгенівського або будь-якого іншого іонізуючого випромінювання, поглинутого тканинами опроміненого тіла, вимірюють кількістю поглинутої тілом енергії в джоулях на 1 кг речовини. Вживають також поняття: величина дози рентгенівського випромінювання (А/кг). Наприклад, при

36-годинному робочому тижні в осіб, зайнятих випробуванням електроустановок з використанням кенотронів, величина дози рентгенівського випромінювання у будь-якій точці на відстані 5–10 см від захисного кожуха кенотрона або всього випробувального пристрою не має перевищувати 20,6 * 1012 А/кг. Останнім часом почали застосовувати замість кенотронів високовольтні напівпровідникові випрямлячі, які усувають появу рентгенівського випромінювання.

Порушення санітарно-гігієнічних норм призводить до зміни складу крові і функціональних порушень центральної нервової системи, які виявляються в дратівливості, сонливості або безсонні, пітливості, головних болях, ослабленні пам'яті, загальній слабості. Порушується робота серцево-судинної системи. При великих дозах може виникнути променева хвороба, тобто порушення нормального кровотворення, розлад нервової системи, травлення, що супроводжуються загальною слабістю, болями і зниженням опірності проти інфекції. М'яке рентгенівське випромінювання призводить насамперед до місцевого впливу на опромінені ділянки тіла; може мутніти кришталик ока (катаракта), випадати волосся.

Гамма-промені випромінюються радіоактивною речовиною. Вони мають довжину хвилі від 4 до 0,1 нм. Як і два інших види ядерних випромінювань (альфа- і бета-випромінювання, які є вже не потоком електромагнітних хвиль, а потоком заряджених частинок), гамма-випромінювання дедалі ширше застосовують у науці і техніці, зокрема в гамма-дефектоскопії та в автоматиці. Гамма-випромінювання використовують також і для передпосівного опромінювання насіння, знищення комах-шкідників, опромінювання харчових продуктів, щоб подовжити строки зберігання та для знешкодження сільськогосподарської сировини.

Альфа-випромінювання мають дуже малу проникну здатність і при зовнішньому опромінюванні затримуються зовнішнім шаром шкіри без помітної шкідливої дії. Проте потрапляння альфа-частинок всередину організму з повітрям або їжею дуже небезпечне.

Бета-промені мають невелику проникну здатність, але шкідливо діють на шкіру й очі. Проникна здатність гамма-променів набагато більша. Це випромінювання може спричинити променеву хворобу. Однак додержуючись санітарних правил роботи з радіоактивними речовинами та джерелами іонізуючих випромінювань, можна тривалий час працювати без шкоди для здоров'я.

Нижче наведено максимально допустимі поглинуті тілом дози рентгенівського, гамма- і бета-випромінювань. Для альфа-випромінювань вони в 10 разів вищі. Норми різні для персоналу, що обслуговує установки і апарати, які створюють випромінювання, і для окремих осіб, що не зв'язані з обслуговуванням цих установок, але зазнають дії випромінювання. Для першої групи тканин (червоний кістковий мозок, статеві залози або взагалі усе тіло) допускається для персоналу не більш як 30 мДж/кг на 13 тижнів (квартал) і не більш як 50 мДж/кг на рік, а для інших осіб – 5 мДж/кг на рік; для другої групи тканин і органів (це будь-який орган тіла, крім зазначених в інших групах) – 80 мДж/кг на квартал і 150 мДж/кг на рік для персоналу або 15 мДж/кг для інших осіб; для третьої групи тканин (щитовидна залоза, кісткова тканина і шкіра, крім частин тіла, які належать до зазначених у наступній групі) – 150 мДж/кг на квартал і 300 мДж/кг на рік для персоналу або 30 мДж/кг для інших осіб віком старше 16 років (для тих, хто ще не досяг 16 років – 15 мДж/кг на рік); для четвертої групи органів (повністю кисті, передпліччя, кісточки і ступні) – 400 мДж/кг на квартал і 750 мДж/кг на рік для персоналу або 75 мДж/кг для інших осіб. Сумарна доза опромінення за ряд років органів першої групи у персоналу, що обслуговує установки і апарати, які створюють іонізуюче випромінювання, не має перевищувати Б = 50 (М – 18), де N – вік (років).

Захист від радіоактивних випромінювань полягає в застосуванні захисних кожухів або екранів, спецодягу, індивідуальних захисних засобів. Важливу роль відіграє також дозиметричний і лікарський контроль.

## 4.2 Соціальні небезпеки

В основу визначення соціальних небезпек, що викликані низьким духовним рівнем, кладуться цінності і компоненти суспільства та людини.

Існують два ціннісні компоненти, співвідношення між якими характеризує стан суспільного життя.

Перший ціннісній компонент — цінності культури суспільства. Другий ціннісний компонент — ціннісна орієнтація особистості. Зв'язок між цими двома крайніми компонентами культури — найважливіший цементуючий і стимулюючий початок всього суспільного життя. І навпаки — порушення цього зв'язку визначає глибоку духовну кризу. В сучасному суспільстві у поєднанні будь-яких складових суспільного життя і в усвідомленості його цілісності велика роль належить інтелігенції. Вона виконує роль духовного і інтелектуального посередника в системі суспільних зв'язків. Але інтелігенція може справитися з цією роллю за умови, коли її усвідомлення не порушено, коли воно само засновується на відповідних посиланнях світоглядного характеру. Одна з цих особливостей сучасної духовної кризи інтелігенції є різкий поворот від атеїстичного світоусвідомлення до релігійного. І тому криза в світоглядній орієнтації інтелігенції теж є небезпекою соціального стану суспільства.

На фоні змінених орієнтирів суспільство потерпає від соціальних небезпек, які викликали зміни і втрати загальнолюдських цінностей і орієнтацій значної кількості населення.

Результатом зміни світу цінностей і орієнтирів частини суспільства є бродяжництво, проституція, п'янство, алкоголізм, тютюнопалін-ня, наркоманія, захворювання на СНІД.

Визначені соціальні небезпеки формують в людському середовищі "групи ризику". "Групи ризику" впливають на стан суспільства шляхом підвищення чисельності кримінальних злочинів, втягування в свої лави все нових представників здорової частини суспільства, впливу на стан здоров'я оточуючих їх людей, погіршують генофонд нації.

Вищезгадані негативні явища в суспільстві (бродяжництво, проституція та ін.) створюють негативне коло, причини якого, в більшості випадків, пов'язані між собою.

Насамперед слід зазначити наявність ряду моральних факторів, які розподіляють суспільство на робітників комерційних структур, сумісних підприємств та робітників державного сектору економіки. Неспіввідношення платні в обставинах низьких заробітків і нерегулярних виплат викликають незадоволення роботою у працівників державного сектору.

Загостреність обставин викликають також незадовільний стан умов праці, проживання і побуту. Вся сукупність обставин збільшує ступінь соціального напруження.

Це може стати передумовою виникнення страйку, а за розвитком відповідних негативних явищ і умов піти в своєму розвитку аж до повстання чи революції.

Такий розвиток подій завжди, навіть в ретельно організованих суспільних змінах, супроводжується виникненням стихійних угруповань, які здійснюють свої плани наживи за рахунок мародерства і вандалізму серед населення, що в своєму подальшому розвитку закінчується тяжкими наслідками актів тероризму.

Необхідність усвідомлених знань розвитку соціальних небезпечних факторів пов'язана з розумінням — куди може привести сукупний їх розвиток і як може потерпіти населення від необачливих дій стихійного характеру.

Найбільше, від чого може потерпіти людина, це від помилок розуміння свого становища в суспільстві. Щоб орієнтуватися в цьому світі, щоб його розуміти, людина повинна визначити, які її дії будуть мати підтримку і схвалення, а які, навпаки, — будуть викликати недовіру і непорозуміння з боку суспільства. Визначена для себе система спілкування з суспільством становить передумови подальшого прогнозу під час взаємодій з суспільством. Це вже робиться за набутим досвідом і становить систему самооцінки людини до своїх дій. Розуміння критеріїв самооцінки й оцінки вчинків інших людей дається завдяки усвідомленню ціннісного і нормативного змісту культури.

Найбільш суттєва характеристика системи цінностей складається з того, що якраз тут сконцентровані уявлення людей про смисл їх життя. Неадекватність системи оцінок, що застосовує людина, призводить до конфліктної ситуації її з суспільством. Неадекватність оцінок завжди викликає відповідну реакцію з боку людини — морального та психологічного походження. Накопичення таких реакцій у людини сприяє відчуттю відповідного дискомфорту і не-прогнозованій поведінці.

.

### 4.3 Висновок до четвертого розділу

В четвертому розділі кваліфікаційної роботи описано в підрозділі лхорони праці джерела іонізуючого, електромагнітного та віброакустичного випромінювання. А в підрозділі який стосується безпеки в надзвичайних ситуаціях види можливих соціальних небезпек.

# ВИСНОВКИ

В результаті написання кваліфікаційної роботи магістра було розроблено програму для шифрування інформації, яка може бути поширена в незахищених каналах зв’язку.

Створена програма дозволяє надійно шифрувати інформацію на основі алгоритму blowfish.

- В роботі був проведений огляд різновидів загроз безпеці інформації в не захищених каналах зв’язку.

- Проведено огляд та порівняння методів і засобів захисту інформації.

- Реалізовано програмне забезпечення для захисту інформації, яка передається незахищеними каналами зв’язку.

- Написана у даній роботі програма може значно полегшити забезпечення безпеки при зберіганні текстової інформації. В подальших дослідженнях даний програмний продукт можна розширити, застосувавши в ньому шифрування за допомогою інших методів наприклад, методу RSA.

# ПЕРЕЛІК ДЖЕРЕЛ

1 Ємець В., Мельник А., Попович Р. Сучасна криптографія. Основні поняття. – Львів: БаК, 2003. – 145 с.

2 Бабичев С. Г., Гончаров В.В., Серов Р.Е. Основы современной криптографии. М., 2001. – 153 с.

3 Бассар Ж. Современная криптографія – М.: Полимед, 1999. – 178 с.

4 Пасічник В. В. Глобальні інформаційні системи та технології (моделі ефективного аналізу, опрацювання та захисту даних) / В.В. Пасічник, П.І. Жежнич, Р.Б. Кравець та ін. – Львів : Вид-во Національного університету «Львівська політехніка», 2006.- 350 с.

5 Богуш В. Інформаційна безпека держави/ Володимир Богуш, Олександр Юдін,; Гол. ред. Ю. О. Шпак. -К.: "МК-Прес", 2005. -432 с.

6 Бойченко О. В. Політика інформаційної безпеки в системі інформаційного забезпечення ОВС України / О. В. Бойченко // Форум права. - 2009. - № 1. - С. 50-55

7 Братель О. Поняття та зміст доктрини інформаційної безпеки// Право України.- К., 2006.- 5.- С.36-41.

8 Остапов С.Е. Основи криптографії: Навчальний посібник / С.Е. Остапов, Л.О. Валь. – Чернівці: Книги – XXI, 2008. – 188 с.

9 Задірака В.К. Методи захисту фінансової інформації / В.К. Задірака, О.С. Олексюк. – К.: Вища школа, 2009. – 460 с.

10 Асосков А.В. Поточные шифры / А.В. Асосков, М.А. Иванов, А.А. Мирский, А.В. Рузин, А.В. Славин, А.Н. Тютвин. – М.: КУДИЦ – ОБРАЗ, 2008. – 336 с.

11 Столлингс Вильям. Криптография и защита сетей: принципы и практика / Вильям Столлингс. – М.: Издательский дом “Вильямс”, 2008.– 672 с.

12 Бабаш А.В. Криптография / А.В. Бабаш, Г.П. Шанкин. – М.: Соломон – Р, 2007. – 512 с.

13 Чмора А. Современная прикладная криптография / А. Чмора. – М.: “Гелиос АРВ”, 2007.– 256 с.

14 Алферов А.П. Основы криптографии / А.П. Алферов, А.Ю. Зубов, А.С. Кузьмин, А.В. Черемушкин. – М.: “Гелиос АРВ”, 2006.– 480 с.

15 Петров А.А. Компьютерная безопасность. Криптографические методы защиты / А.А. Петров. – М.: ДМК, 2000. – 448с.

16 Нечаев В.И. Элементы криптографии (основы защиты информации) / В.И. Нечаев. – Москва: Высшая школа, 1999. – 109с.

17 Ященко В.В. Введение в криптографию / В.В. Ященко. – М.: МЦНМО: ЧеРо, 1999. – 272 с.

18 Масленников М. Практическая криптография / М. Масленников. – СПб. : БХВ, 2003. – 464 с.

19 Молдовян Н.А. Ввведение в криптосистемы с открытым ключом / Н.А. Молдовян, А.А. Молдовян. – СПб. : БХВ-Петербург, 2005. – 288 с.

20 Вербіцький О.В. Вступ до криптології / О.В. Вербіцький. – Львів: Науково-техн.літ., 1998. – 248с.

21 Месси Дж.Л. Введение в современную криптологию / Дж.Л. Месси // ТИИЭР. – 1988. – Т.76, №5. – С.24-42

22 Симмонс Г.Дж. Обзор методов аутентификации информации / Г.Дж. Симмонс // ТИИЭР – 1988. – Т.76.№5 – С. 105–125.

23 Черемушкин А.В. Лекции по арифметическим алгоритмам в криптографии / А.В. Черемушкин. – М. : МЦНМО, 2002. – 103 с.

24 Гэри М. Вычислительные машины и труднорешаемые задачи / М. Гэри, Д. Джонсон. – М.: Мир, 1982. – 416с.

25 Диффи У. Защищенность и имитостойкость / У. Диффи, М. Хеллман. // ТИИЭР. – 1979. – Т.67, №3. – С. 71–109.

26 Саломаа А. Криптография с открытым ключом / А. Саломаа. – М.: Мир, 1996. – 318 с.

27 Шеннон К.Э. Теория связи в секретных системах / К.Э. Шеннон. // В кн.: Шеннон К.Э. Работы по теории информации и кибернетике. – М.: ИЛ, 1963. – С. 243–332

28 В.О. Семенюк, Я.В. Литвиненко. Огляд методів захисту текстової інформації // Матеріали XI науково-технічної конференції «інформаційні моделі, системи та технології». Тернопіль, 2023, 112 с.

# ДОДАТКИ

**Додаток А**

# Скріншот опублікованих тез конференції

**УДК 681.518.3**
**В.О. Семенюк, д.т.н., проф.; Я.В. Литвиненко**
(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

## ОГЛЯД МЕТОДІВ ЗАХИСТУ ТЕКСТОВОЇ ІНФОРМАЦІЇ

**V.O. Semenyuk, Dr., Prof.; Ia.V. Lytvynenko**

## OVERVIEW OF TEXT INFORMATION PROTECTION METHODS

Захист текстової інформації є важливою задачею, особливо в сучасному цифровому середовищі. На даний час існує велика кількість розроблених методів захисту інформації. Щоб зорієнтуватись які найкраще застосувати для нашої задачі проведемо огляд існуючих на практиці методів.

Дана доповідь стосується огляду відомих методів, які можна застосувати для захисту текстової інформації.

Існує кілька методів та стратегій для захисту текстової інформації від несанкціонованого доступу та збереження конфіденційності. Розглянемо деякі з них:

- Шифрування: Симетричне шифрування: Використовує один ключ для як шифрування, так і розшифрування тексту. Приклади - AES, DES.
- Асиметричне (або публічне) шифрування: Використовує пару ключів (приватний і публічний) для шифрування та розшифрування. Приклади - RSA, ECC.
- Хешування: Використання хеш-функцій: Дозволяє перетворити вхідні дані в унікальний хеш-код фіксованої довжини. Використовується для перевірки цілісності даних.
- Цифровий підпис: RSA, DSA, ECDSA: Дозволяє встановити автентичність та цілісність даних за допомогою підпису, який генерується приватним ключем.

До інших методів і стратегій захисту інформації можна віднести:

- Використання VPN: Віртуальні приватні мережі: Забезпечують безпечний тунель для передачі даних через незахищені мережі, що дозволяє шифрувати текстову інформацію.
- Керування доступом: Ролева модель: Визначення різних рівнів доступу до інформації на основі ролей користувачів.
- Механізми аутентифікації: Використання паролів, біометричних даних чи двофакторної аутентифікації.
- Фільтрація та моніторинг: Системи виявлення вторгнень (IDS) та системи захисту від вторгнень (IPS): Сприймають та реагують на аномалії або несподівані події в мережі.
- Зберігання та обробка: Шифрування на рівні файлів або дисків: Захищає дані, коли вони зберігаються або обробляються на пристроях.
- Оновлення та патчі: Регулярні оновлення програмного забезпечення: Забезпечують виправлення виявлених уразливостей.

Ці методи та стратегії захисту часто використовуються в поєднанні для створення комплексних систем безпеки, оскільки один захисний захід не завжди достатньо ефективний.

**Література**

1. Писарчук О.О. Основи захисту інформації : навчальний посібник / О.О. Писарчук, Ю. Г. Даник, С. Г. Вдовенко та ін. – Житомир : ЖВІ ДУТ, 2015. 226 с.
2. Хорошко В.А. Методи й засоби захисту інформації / ВА Хорошко, АА Чекатков К.: ЮНІОР, 2003.

**Додаток Б**

## Текст програми

Текст програми

```
unit UnitMain;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, ToolWin, ComCtrls, StdCtrls, XPMan,INIFiles, ImgList;

type
  TForm1 = class(TForm)
    S1: TStatusBar;
    MainMenu1: TMainMenu;
    File1: TMenuItem;
    New1: TMenuItem;
    Open1: TMenuItem;
    Save1: TMenuItem;
    SaveAs1: TMenuItem;
    Exit1: TMenuItem;
    SaveDialog1: TSaveDialog;
    OpenDialog1: TOpenDialog;
    Memo1: TMemo;
    Memo2: TMemo;
    N2: TMenuItem;
    XPManifest1: TXPManifest;
    Help1: TMenuItem;
    About1: TMenuItem;
    Edit1: TMenuItem;
    Undo1: TMenuItem;
    N3: TMenuItem;
    Cut1: TMenuItem;
    Copy1: TMenuItem;
    Paste1: TMenuItem;
    N4: TMenuItem;
    Font1: TMenuItem;
    F1: TFontDialog;
    C1: TColorDialog;
    ToolBar1: TToolBar;
    ToolButton1: TToolButton;
    ToolButton2: TToolButton;
    ToolButton3: TToolButton;
    ToolButton4: TToolButton;
    ToolButton5: TToolButton;
    ToolButton6: TToolButton;
    ToolButton7: TToolButton;
    ToolButton8: TToolButton;
```

```
    ToolButton9: TToolButton;
    ImageList1: TImageList;
    N5: TMenuItem;
    ToolButton10: TToolButton;
    ToolButton11: TToolButton;
    ToolButton12: TToolButton;
    ToolButton13: TToolButton;
    procedure SaveAs1Click(Sender: TObject);
    procedure Open1Click(Sender: TObject);
    procedure Exit1Click(Sender: TObject);
    procedure New1Click(Sender: TObject);
    procedure Save1Click(Sender: TObject);
    procedure About1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Undo1Click(Sender: TObject);
    procedure Cut1Click(Sender: TObject);
    procedure Copy1Click(Sender: TObject);
    procedure Paste1Click(Sender: TObject);
    procedure Font1Click(Sender: TObject);
    procedure Memo1DblClick(Sender: TObject);
    procedure FormDestroy(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

  procedure Openpar;

implementation

{$R *.dfm}
uses
  LbCipher, LbString, UnitAbout;

const inifile='read.ini';
      IniSection='Main';
var
  Key256          : TKey128;
  PlainText       : string;
  CipherText       : string;
  FileName        :string='noname';

procedure RefreshKeys(passphrase:string);
begin
  GenerateLMDKey(Key256, SizeOf(Key256),passphrase);
  Form1.S1.SimpleText:='Generate keys for encrypt';
end;

procedure ReadSettings;
var path:string;
```

```
    ini:TINIFile;
begin
 path:=sysutils.ExtractFilePath(application.ExeName)+inifile;
 if fileexists(path)then
 begin
   ini:=TINIFile.Create(path);
   Form1.Memo1.Font.Name:=ini.ReadString(INISection,'MFName','Arial');
   Form1.Memo1.Font.Size:=ini.ReadInteger(INISection,'MFSize',10);
   Form1.Memo1.Font.Color:=ini.ReadInteger(INISection,'MFColor',0);
   Form1.Memo1.Color:=ini.ReadInteger(INISection,'MColor',0);
   ini.Free;
 end;
end;

procedure WriteSettings;
var path:string;
    ini:TINIFile;
begin
 path:=sysutils.ExtractFilePath(application.ExeName)+inifile;
 ini:=TINIFile.Create(path);
 ini.WriteString(INISection,'MFName',Form1.Memo1.Font.Name);
 ini.WriteInteger(INISection,'MFSize',Form1.Memo1.Font.Size);
 ini.WriteInteger(INISection,'MFColor',Form1.Memo1.Font.Color);
 ini.WriteInteger(INISection,'MColor',Form1.Memo1.Color);
 ini.Free;
end;

procedure TForm1.SaveAs1Click(Sender: TObject);
var p:string;
begin
p:= InputBox('Введіть ключову фразу', 'Ключова фраза:', '');
if p=''then
begin
MessageDlg('Ключова фраза є необхідною.',mterror,[mbok],0);
Form1.S1.SimpleText:='Відмінено';
exit;
end;
if savedialog1.Execute then
 begin
  RefreshKeys(p);
  PlainText := memo1.Text;
  Form1.S1.SimpleText:='Шифрування...';
  CipherText := BFEncryptStringEx(PlainText, Key256, True);
  memo2.Text := CipherText;
  memo2.Lines.SaveToFile(savedialog1.FileName);
  Form1.S1.SimpleText:='Готово';
 end;
end;

procedure TForm1.Open1Click(Sender: TObject);
var p:string;
begin
if messagedlg('Відкрити зашифрований файл?',mtconfirmation,[mbYes,mbno],0)=mrYes then
 begin
```

```
   if opendialog1.Execute then
   begin
   FileName:=opendialog1.FileName;
   Form1.Caption:='ReAd-'+Filename;
   p:= InputBox('Введіть ключову фразу', 'Ключова фраза:', '');
   if p=''then
    begin
    MessageDlg('Ключова фраза є необхідною.',mterror,[mbok],0);
    Form1.S1.SimpleText:='Відмінено';
    exit;
    end;
   RefreshKeys(p);
   memo2.Lines.LoadFromFile(opendialog1.FileName);
   CipherText := memo2.Text;
   Form1.S1.SimpleText:='Розшифрування...';
   PlainText := BFEncryptStringEx(CipherText, Key256, False);
   memo1.Text:=PlainText;
   Form1.S1.SimpleText:='Готово';
   end;
 end else
 if opendialog1.Execute then
 begin
 filename:=opendialog1.FileName;
 Form1.Caption:='ReAd-'+Filename;
 memo1.Lines.LoadFromFile(opendialog1.FileName);
 end;
end;

procedure Openpar;
var p:string;
begin
if messagedlg('Відкрити зашифрований файл?',mtconfirmation,[mbYes,mbno],0)=mrYes then
 begin
   FileName:=paramstr(1);
   Form1.Caption:='ReAd-'+Filename;
   p:= InputBox('Введіть ключову фразу', 'Ключова фраза:', '');
   if p=''then
    begin
    MessageDlg('Ключова фраза є необхідною.',mterror,[mbok],0);
    exit;
    end;
   RefreshKeys(p);
   Form1.memo2.Lines.LoadFromFile(FileName);
   CipherText := Form1.memo2.Text;
   Form1.S1.SimpleText:='Розшифрування...';
   PlainText := BFEncryptStringEx(CipherText, Key256, False);
   Form1.memo1.Text:=PlainText;
   Form1.S1.SimpleText:='Готово';
 end else
 begin
 filename:=paramstr(1);
 Form1.Caption:='ReAd-'+Filename;
 Form1.memo1.Lines.LoadFromFile(FileName);
 end;
```

```
 end;

procedure TForm1.Exit1Click(Sender: TObject);
begin
close;
end;

procedure TForm1.New1Click(Sender: TObject);
begin
Filename:='noname.txt';
Form1.Caption:='ReAd-'+Filename;
memo1.Clear;
end;

procedure TForm1.Save1Click(Sender: TObject);
var p:string;
begin
p:= InputBox('Введіть ключову фразу', 'Ключова фраза:', '');
 if p="then
  begin
   MessageDlg('Ключова фраза є необхідною.',mterror,[mbok],0);
   Form1.S1.SimpleText:='Відмінено';
   exit;
  end;
  RefreshKeys(p);
  PlainText := memo1.Text;
  Form1.S1.SimpleText:='Шифрування...';
  CipherText := BFEncryptStringEx(PlainText, Key256, True);
  memo2.Text := CipherText;
  memo2.Lines.SaveToFile(FileName);
  Form1.S1.SimpleText:='Готово.';
end;

procedure TForm1.About1Click(Sender: TObject);
begin
MessageDlg('ReAd - текстовий редактор з підтримкою Blowfish 256 bit шифруванням.'
+#13#10
+#13#10+'Автор:'
+#13#10+'студент 5-го курсу факультету інформатики ЗакДУ Гергель Володимир'
+#13#10+'mailto:gwfwork@yahoo.com',mtinformation,[mbok],0);
end;

procedure TForm1.FormCreate(Sender: TObject);
var NewBigSize:int64;
begin
NewBigSize:=2000000000;
SendMessage(Memo1.Handle, EM_LIMITTEXT, 0, NewBigSize);
SendMessage(Memo2.Handle, EM_LIMITTEXT, 0, NewBigSize);
ReadSettings;
end;

procedure TForm1.Undo1Click(Sender: TObject);
begin
memo1.Undo;
```

```
end;

procedure TForm1.Cut1Click(Sender: TObject);
begin
memo1.CutToClipboard;
end;

procedure TForm1.Copy1Click(Sender: TObject);
begin
memo1.CopyToClipboard;
end;

procedure TForm1.Paste1Click(Sender: TObject);
begin
memo1.PasteFromClipboard;
end;

procedure TForm1.Font1Click(Sender: TObject);
begin
f1.Font:=memo1.Font;
if f1.Execute then memo1.Font:=f1.Font;
end;

procedure TForm1.Memo1DblClick(Sender: TObject);
begin
if c1.Execute then memo1.Color:=c1.Color;
end;

procedure TForm1.FormDestroy(Sender: TObject);
begin
WriteSettings;
end;

end.
{$I LockBox.inc}

unit LbCipher;
  {-private key encryption/decryption primitives}

interface

uses
{$IFDEF MSWINDOWS}
  Windows,
{$ENDIF}
{$IFDEF UsingCLX}
  Types,
{$ENDIF}
  Classes;
const
  { largest structure that can be created }
  MaxStructSize = 1024 * 2000000; {2G}
```

```
{ TLbBase - used to force this unit to be added to uses clause }
type
  TLBBase = class(TComponent)
  end;


{ general structures }
type
  pLongIntArray = ^TLongIntArray;
  TLongIntArray = array [0..MaxStructSize div SizeOf(LongInt) - 1] of LongInt;

  TLongIntRec = packed record
    case Byte of
      1: (Lo: Word;
          Hi: Word);
      2: (LoLo: Byte;
          LoHi: Byte;
          HiLo: Byte;
          HiHi: Byte);
  end;

  TInt64 = packed record
    case Byte of
      0: (Lo: LongInt;
          Hi: LongInt);
      1: (LoLo: Word;
          LoHi: Word;
          HiLo: Word;
          HiHi: Word);
      2: (LoLoLo: Byte;
          LoLoHi: Byte;
          LoHiLo: Byte;
          LoHiHi: Byte;
          HiLoLo: Byte;
          HiLoHi: Byte;
          HiHiLo: Byte;
          HiHiHi: Byte);
  end;

  TRDLVector = record
    case Byte of
      0 : (dw : DWord);
      1 : (bt : array[0..3] of Byte);
    end;


{ encryption key types }
type
  PKey64  = ^TKey64;                                {!!.03}
  TKey64  = array [0..7] of Byte;

  PKey128 = ^TKey128;                               {!!.03}
  TKey128 = array [0..15] of Byte;
```

```
  PKey192 = ^TKey192;                                {!!.03}
  TKey192 = array [0..23] of Byte;

  PKey256 = ^TKey256;                                {!!.03}
  TKey256 = array [0..31] of Byte;


{ encryption block types }
type
  PLBCBlock  = ^TLBCBlock;
  TBFBlock   = array[0..1] of LongInt;     { BlowFish }

{ context type constants }
const
  BFRounds = 16;      { 16 blowfish rounds }

{ block cipher context types }
type
  { Blowfish }
  TBFContext = packed record
    PBox    : array[0..(BFRounds+1)] of LongInt;
    SBox    : array[0..3, 0..255] of LongInt;
  end;

{ Blowfish Cipher }
procedure InitEncryptBF(Key : TKey128;
          var Context : TBFContext);
procedure EncryptBF(const Context : TBFContext;
          var Block : TBFBlock; Encrypt : Boolean);
procedure EncryptBFCBC(const Context : TBFContext;
          const Prev : TBFBlock; var Block : TBFBlock; Encrypt : Boolean);

{ Random Number Cipher }
procedure InitEncryptRNG64(KeyHi, KeyLo : LongInt;
          var Context : TRNG64Context);
procedure EncryptRNG32(var Context : TRNG32Context;
          var Buf; BufSize : LongInt);
procedure EncryptRNG64(var Context : TRNG64Context;
          var Buf; BufSize : LongInt);
procedure InitEncryptRNG32(Key : LongInt;
          var Context : TRNG32Context);

{ Miscellaneous hash algorithms }
procedure HashELF(var Digest : LongInt;
          const Buf;  BufSize : LongInt);
procedure HashMix128(var Digest : LongInt;
          const Buf;  BufSize : LongInt);

{ String hashing }
procedure StringHashELF(var Digest : LongInt;
          const Str : string);
procedure StringHashLMD(var Digest; DigestSize : LongInt;
          const Str : string);
procedure StringHashMD5(var Digest : TMD5Digest;
```

```
        const Str : string);
procedure StringHashMix128(var Digest : LongInt;
        const Str : string);
procedure StringHashSHA1(var Digest : TSHA1Digest;
        const Str : string);

{ Key generation }
procedure GenerateLMDKey(var Key; KeySize : Integer;
        const Str : string);
procedure GenerateMD5Key(var Key : TKey128;
        const Str : string);
procedure GenerateRandomKey(var Key; KeySize : Integer);

{ Misc public utilities }
function Ran01(var Seed : LongInt) : LongInt;
function Ran02(var Seed : LongInt) : LongInt;
function Ran03(var Seed : LongInt) : LongInt;
function Random32Byte(var Seed : LongInt) : Byte;
function Random64Byte(var Seed : TInt64) : Byte;
procedure ShrinkDESKey(var Key : TKey64);
procedure XorMem(var Mem1;  const Mem2;  Count : Cardinal);
function RolX(I, C : DWord) : DWord; register;

implementation

uses
  LbUtils, SysUtils;


{first 2048 bits of Pi in hexadecimal, low to high, without the leading "3"}
const
  Pi2048: array [0..255] of Byte = (
    $24, $3F, $6A, $88, $85, $A3, $08, $D3, $13, $19, $8A, $2E, $03, $70, $73, $44,
    $A4, $09, $38, $22, $29, $9F, $31, $D0, $08, $2E, $FA, $98, $EC, $4E, $6C, $89,
    $45, $28, $21, $E6, $38, $D0, $13, $77, $BE, $54, $66, $CF, $34, $E9, $0C, $6C,
    $C0, $AC, $29, $B7, $C9, $7C, $50, $DD, $3F, $84, $D5, $B5, $B5, $47, $09, $17,
    $92, $16, $D5, $D9, $89, $79, $FB, $1B, $D1, $31, $0B, $A6, $98, $DF, $B5, $AC,
    $2F, $FD, $72, $DB, $D0, $1A, $DF, $B7, $B8, $E1, $AF, $ED, $6A, $26, $7E, $96,
    $BA, $7C, $90, $45, $F1, $2C, $7F, $99, $24, $A1, $99, $47, $B3, $91, $6C, $F7,
    $08, $01, $F2, $E2, $85, $8E, $FC, $16, $63, $69, $20, $D8, $71, $57, $4E, $69,
    $A4, $58, $FE, $A3, $F4, $93, $3D, $7E, $0D, $95, $74, $8F, $72, $8E, $B6, $58,
    $71, $8B, $CD, $58, $82, $15, $4A, $EE, $7B, $54, $A4, $1D, $C2, $5A, $59, $B5,
    $9C, $30, $D5, $39, $2A, $F2, $60, $13, $C5, $D1, $B0, $23, $28, $60, $85, $F0,
    $CA, $41, $79, $18, $B8, $DB, $38, $EF, $8E, $79, $DC, $B0, $60, $3A, $18, $0E,
    $6C, $9E, $0E, $8B, $B0, $1E, $8A, $3E, $D7, $15, $77, $C1, $BD, $31, $4B, $27,
    $78, $AF, $2F, $DA, $55, $60, $5C, $60, $E6, $55, $25, $F3, $AA, $55, $AB, $94,
    $57, $48, $98, $62, $63, $E8, $14, $40, $55, $CA, $39, $6A, $2A, $AB, $10, $B6,
    $B4, $CC, $5C, $34, $11, $41, $E8, $CE, $A1, $54, $86, $AF, $7C, $72, $E9, $93);

type
  pMD5ContextEx = ^TMD5ContextEx;
  TMD5ContextEx = packed record
    Count : array [0..1] of DWord;  {number of bits handled mod 2^64}
    State : array [0..3] of DWord;  {scratch buffer}
```

```
    Buf   : array [0..63] of Byte;    {input buffer}
  end;

  TLMDContextEx = packed record
    DigestIndex : LongInt;
    Digest      : array [0..255] of Byte;
    KeyIndex    : LongInt;
    case Byte of
      0: (KeyInts : array [0..3] of LongInt);
      1: (Key     : TKey128);
  end;
  TBlock2048 = array [0..255] of Byte;

type
  {bit mixing types}
  T128Bit     = array [0..3] of DWord;
  T256Bit     = array [0..7] of DWord;

const
  BCSalts: array [0..3] of DWord =
    ($55555555, $AAAAAAAA, $33333333, $CCCCCCCC);

type
  TBCHalfBlock = array [0..1] of LongInt;

  TBFBlockEx = packed record
    Xl : array[0..3] of Byte;
    Xr : array[0..3] of Byte;
  end;

{ Blowfish tables }
{$I LbBF.inc }                                    {!!.01}

{
=============================================================================
=== }

procedure InitEncryptBF(Key : TKey128; var Context : TBFContext);
var
  I     : Integer;
  J     : Integer;
  K     : Integer;
  Data  : LongInt;
  Block : TBFBlock;
begin
  {initialize PArray}
  Move(bf_P, Context.PBox, SizeOf(Context.PBox));
  {initialize SBox}
  Move(bf_S, Context.SBox, SizeOf(Context.SBox));

  {update PArray with the key bits}
  J := 0;
  for I := 0 to (BFRounds+1) do begin
    Data := 0;
```

```
    for K := 0 to 3 do begin
      Data := (Data shl 8) or Key[J];
      Inc(J);
      if J >= SizeOf(Key) then
        J := 0;
    end;
    Context.PBox[I] := Context.PBox[I] xor Data;
  end;

  {encrypt an all-zero string using the Blowfish algorithm and}
  {replace the elements of the P-array with the output of this process}

  Block[0] := 0;
  Block[1] := 0;
  I := 0;
  repeat
    EncryptBF(Context, Block, True);
    Context.PBox[I] := Block[0];
    Context.PBox[I+1] := Block[1];
    Inc(I, 2);
  until I > BFRounds+1;

  {continue the process, replacing the elements of the four S-boxes in}
  {order, with the output of the continuously changing Blowfish algorithm}

  for J := 0 to 3 do begin
    I := 0;
    repeat
      EncryptBF(Context, Block, True);
      Context.SBox[J, I] := Block[0];
      Context.SBox[J, I+1] := Block[1];
      Inc(I, 2);
    until I > 255;
  end;

  {in total, 521 iterations are required to generate all required subkeys. }
end;
{ ---------------------------------------------------------------------- }
procedure EncryptBF(const Context : TBFContext;
  var Block : TBFBlock; Encrypt : Boolean);
var
  I : Integer;
  TmpBlock : TBFBlockEx;                                {!!.01}
begin
  Move(Block, TmpBlock, SizeOf(TmpBlock));                    {!!.01}
  if Encrypt then begin
    Block[0] := Block[0] xor Context.PBox[0];

    {16 Rounds to go (8 double rounds to avoid swaps)}
    I := 1;
    repeat
      {first half round }
      Block[1] := Block[1] xor Context.PBox[I] xor (((
              Context.SBox[0, TmpBlock.Xl[3]] + Context.SBox[1, TmpBlock.Xl[2]])
```

```
            xor Context.SBox[2, TmpBlock.Xl[1]]) + Context.SBox[3, TmpBlock.Xl[0]]);
     {second half round }
     Block[0] := Block[0] xor Context.PBox[I+1] xor (((
            Context.SBox[0, TmpBlock.Xr[3]] + Context.SBox[1, TmpBlock.Xr[2]])
            xor Context.SBox[2, TmpBlock.Xr[1]]) + Context.SBox[3, TmpBlock.Xr[0]]);
     Inc(I, 2);
   until I > BFRounds;
   Block[1] := Block[1] xor Context.PBox[(BFRounds+1)];
  end else begin
   Block[1] := Block[1] xor Context.PBox[(BFRounds+1)];

   {16 Rounds to go (8 double rounds to avoid swaps)}
   I := BFRounds;
   repeat
    {first half round }
    Block[0] := Block[0] xor Context.PBox[I] xor (((
            Context.SBox[0, TmpBlock.Xr[3]] + Context.SBox[1, TmpBlock.Xr[2]])
            xor Context.SBox[2, TmpBlock.Xr[1]]) + Context.SBox[3, TmpBlock.Xr[0]]);
     {second half round }
     Block[1] := Block[1] xor Context.PBox[i-1] xor (((
            Context.SBox[0, TmpBlock.Xl[3]] + Context.SBox[1, TmpBlock.Xl[2]])
            xor Context.SBox[2, TmpBlock.Xl[1]]) + Context.SBox[3, TmpBlock.Xl[0]]);
      Dec (I, 2);
    until I < 1;
    Block[0] := Block[0] xor Context.PBox[0];
  end;
end;
{ ---------------------------------------------------------------------------- }
procedure EncryptBFCBC(const Context : TBFContext; const Prev : TBFBlock;
  var Block : TBFBlock; Encrypt : Boolean);
begin
  if Encrypt then begin
    XorMem(Block, Prev, SizeOf(Block));
    EncryptBF(Context, Block, Encrypt);
  end else begin
    EncryptBF(Context, Block, Encrypt);
    XorMem(Block, Prev, SizeOf(Block));
  end;
end;
{ ---------------------------------------------------------------------------- }

procedure GenerateRandomKey(var Key; KeySize : Integer);
var
  I     : Integer;
begin
  Randomize;
  for I := 0 to KeySize - 1 do
    TByteArray(Key)[I] := System.Random(256);                    {!!.01}
end;
procedure Mix128(var X : T128Bit);
var
  AA, BB, CC, DD : LongInt;
begin
  AA := X[0];  BB := X[1];  CC := X[2];  DD := X[3];
```

```
  AA := AA + DD;  DD := DD + AA;  AA := AA xor (AA shr 7);
  BB := BB + AA;  AA := AA + BB;  BB := BB xor (BB shl 13);
  CC := CC + BB;  BB := BB + CC;  CC := CC xor (CC shr 17);
  DD := DD + CC;  CC := CC + DD;  DD := DD xor (DD shl 9);
  AA := AA + DD;  DD := DD + AA;  AA := AA xor (AA shr 3);
  BB := BB + AA;  AA := AA + BB;  BB := BB xor (BB shl 7);
  CC := CC + BB;  BB := BB + CC;  CC := CC xor (DD shr 15);
  DD := DD + CC;  CC := CC + DD;  DD := DD xor (DD shl 11);

  X[0] := AA;  X[1] := BB;  X[2] := CC;  X[3] := DD;
end;
{ ---------------------------------------------------------------------------- }
procedure HashELF(var Digest : LongInt; const Buf;  BufSize : LongInt);
var
  I, X  : LongInt;
begin
  Digest := 0;
  for I := 0 to BufSize - 1 do begin
    Digest := (Digest shl 4) + TByteArray(Buf)[I];                {!!.01}
    X := Digest and $F0000000;
    if (X <> 0) then
      Digest := Digest xor (X shr 24);
    Digest := Digest and (not X);
  end;
end;
{ ---------------------------------------------------------------------------- }
procedure StringHashELF(var Digest : LongInt; const Str : string);
begin
  HashELF(Digest, Str[1], Length(Str));
end;
{ ---------------------------------------------------------------------------- }
function RolX(I, C : DWord) : DWord; register;
asm
  mov  ecx, edx        {get count to cl}
  rol  eax, cl         {rotate eax by cl}
end;
{ ---------------------------------------------------------------------------- }
procedure Transform(var Buffer : array of DWord;  const InBuf : array of DWord);
const
  S11 = 7;
  S12 = 12;
  S13 = 17;
  S14 = 22;
  S21 = 5;
  S22 = 9;
  S23 = 14;
  S24 = 20;
  S31 = 4;
  S32 = 11;
  S33 = 16;
  S34 = 23;
  S41 = 6;
  S42 = 10;
```

```
  S43 = 15;
  S44 = 21;
var
  Buf : array [0..3] of DWord;                    {!!.01}
  InA : array [0..15] of DWord;                   {!!.01}
var
  A   : DWord;
  B   : DWord;
  C   : DWord;
  D   : DWord;

  procedure FF(var A : DWord;  B, C, D, X, S, AC : DWord);
  begin
    A := RolX(A + ((B and C) or (not B and D)) + X + AC, S) + B;
  end;

  procedure GG(var A : DWord;  B, C, D, X, S, AC : DWord);
  begin
    A := RolX(A + ((B and D) or (C and not D)) + X + AC, S) + B;
  end;

  procedure HH(var A : DWord;  B, C, D, X, S, AC : DWord);
  begin
    A := RolX(A + (B xor C xor D) + X + AC, S) + B;
  end;

  procedure II(var A : DWord;  B, C, D, X, S, AC : DWord);
  begin
    A := RolX(A + (C xor (B or not D)) + X + AC, S) + B;
  end;

begin
  Move(Buffer, Buf, SizeOf(Buf));                 {!!.01}
  Move(InBuf, InA, SizeOf(InA));                  {!!.01}
  A := Buf [0];
  B := Buf [1];
  C := Buf [2];
  D := Buf [3];


  {round 1}
  FF(A, B, C, D, InA [ 0], S11, $D76AA478);  { 1 }
  FF(D, A, B, C, InA [ 1], S12, $E8C7B756);  { 2 }
  FF(C, D, A, B, InA [ 2], S13, $242070DB);  { 3 }
  FF(B, C, D, A, InA [ 3], S14, $C1BDCEEE);  { 4 }
  FF(A, B, C, D, InA [ 4], S11, $F57C0FAF);  { 5 }
  FF(D, A, B, C, InA [ 5], S12, $4787C62A);  { 6 }
  FF(C, D, A, B, InA [ 6], S13, $A8304613);  { 7 }
  FF(B, C, D, A, InA [ 7], S14, $FD469501);  { 8 }
  FF(A, B, C, D, InA [ 8], S11, $698098D8);  { 9 }
  FF(D, A, B, C, InA [ 9], S12, $8B44F7AF);  { 10 }
  FF(C, D, A, B, InA [10], S13, $FFFF5BB1);  { 11 }
  FF(B, C, D, A, InA [11], S14, $895CD7BE);  { 12 }
  FF(A, B, C, D, InA [12], S11, $6B901122);  { 13 }
```

```
FF(D, A, B, C, InA [13], S12, $FD987193);  { 14 }
FF(C, D, A, B, InA [14], S13, $A679438E);  { 15 }
FF(B, C, D, A, InA [15], S14, $49B40821);  { 16 }

{round 2}
GG(A, B, C, D, InA [ 1], S21, $F61E2562);  { 17 }
GG(D, A, B, C, InA [ 6], S22, $C040B340);  { 18 }
GG(C, D, A, B, InA [11], S23, $265E5A51);  { 19 }
GG(B, C, D, A, InA [ 0], S24, $E9B6C7AA);  { 20 }
GG(A, B, C, D, InA [ 5], S21, $D62F105D);  { 21 }
GG(D, A, B, C, InA [10], S22, $02441453);  { 22 }
GG(C, D, A, B, InA [15], S23, $D8A1E681);  { 23 }
GG(B, C, D, A, InA [ 4], S24, $E7D3FBC8);  { 24 }
GG(A, B, C, D, InA [ 9], S21, $21E1CDE6);  { 25 }
GG(D, A, B, C, InA [14], S22, $C33707D6);  { 26 }
GG(C, D, A, B, InA [ 3], S23, $F4D50D87);  { 27 }
GG(B, C, D, A, InA [ 8], S24, $455A14ED);  { 28 }
GG(A, B, C, D, InA [13], S21, $A9E3E905);  { 29 }
GG(D, A, B, C, InA [ 2], S22, $FCEFA3F8);  { 30 }
GG(C, D, A, B, InA [ 7], S23, $676F02D9);  { 31 }
GG(B, C, D, A, InA [12], S24, $8D2A4C8A);  { 32 }

{round 3}
HH(A, B, C, D, InA [ 5], S31, $FFFA3942);  { 33 }
HH(D, A, B, C, InA [ 8], S32, $8771F681);  { 34 }
HH(C, D, A, B, InA [11], S33, $6D9D6122);  { 35 }
HH(B, C, D, A, InA [14], S34, $FDE5380C);  { 36 }
HH(A, B, C, D, InA [ 1], S31, $A4BEEA44);  { 37 }
HH(D, A, B, C, InA [ 4], S32, $4BDECFA9);  { 38 }
HH(C, D, A, B, InA [ 7], S33, $F6BB4B60);  { 39 }
HH(B, C, D, A, InA [10], S34, $BEBFBC70);  { 40 }
HH(A, B, C, D, InA [13], S31, $289B7EC6);  { 41 }
HH(D, A, B, C, InA [ 0], S32, $EAA127FA);  { 42 }
HH(C, D, A, B, InA [ 3], S33, $D4EF3085);  { 43 }
HH(B, C, D, A, InA [ 6], S34,  $4881D05);  { 44 }
HH(A, B, C, D, InA [ 9], S31, $D9D4D039);  { 45 }
HH(D, A, B, C, InA [12], S32, $E6DB99E5);  { 46 }
HH(C, D, A, B, InA [15], S33, $1FA27CF8);  { 47 }
HH(B, C, D, A, InA [ 2], S34, $C4AC5665);  { 48 }

{round 4}
II(A, B, C, D, InA [ 0], S41, $F4292244);  { 49 }
II(D, A, B, C, InA [ 7], S42, $432AFF97);  { 50 }
II(C, D, A, B, InA [14], S43, $AB9423A7);  { 51 }
II(B, C, D, A, InA [ 5], S44, $FC93A039);  { 52 }
II(A, B, C, D, InA [12], S41, $655B59C3);  { 53 }
II(D, A, B, C, InA [ 3], S42, $8F0CCC92);  { 54 }
II(C, D, A, B, InA [10], S43, $FFEFF47D);  { 55 }
II(B, C, D, A, InA [ 1], S44, $85845DD1);  { 56 }
II(A, B, C, D, InA [ 8], S41, $6FA87E4F);  { 57 }
II(D, A, B, C, InA [15], S42, $FE2CE6E0);  { 58 }
II(C, D, A, B, InA [ 6], S43, $A3014314);  { 59 }
II(B, C, D, A, InA [13], S44, $4E0811A1);  { 60 }
II(A, B, C, D, InA [ 4], S41, $F7537E82);  { 61 }
```

```
  II(D, A, B, C, InA [11], S42, $BD3AF235);  { 62 }
  II(C, D, A, B, InA [ 2], S43, $2AD7D2BB);  { 63 }
  II(B, C, D, A, InA [ 9], S44, $EB86D391);  { 64 }

  Inc(Buf [0], A);
  Inc(Buf [1], B);
  Inc(Buf [2], C);
  Inc(Buf [3], D);

  Move(Buf, Buffer, SizeOf(Buffer));                    {!!.01}
end;
procedure XorMemPrim(var Mem1;  const Mem2;  Count : Cardinal); register;
asm
  push esi
  push edi

  mov  esi, eax        //esi = Mem1
  mov  edi, edx        //edi = Mem2

  push ecx             //save byte count
  shr  ecx, 2          //convert to dwords
  jz   @Continue

  cld
@Loop1:                //xor dwords at a time
  mov  eax, [edi]
  xor  [esi], eax
  add  esi, 4
  add  edi, 4
  dec  ecx
  jnz  @Loop1

@Continue:             //handle remaining bytes (3 or less)
  pop  ecx
  and  ecx, 3
  jz   @Done

@Loop2:                //xor remaining bytes
  mov  al, [edi]
  xor  [esi], al
  inc  esi
  inc  edi
  dec  ecx
  jnz  @Loop2

@Done:
  pop  edi
  pop  esi
end;
{ ---------------------------------------------------------------------- }
procedure XorMem(var Mem1;  const Mem2;  Count : Cardinal);
begin
  XorMemPrim(Mem1, Mem2, Count);
end;
```

```
{ == SHA-1 hashing routines ================================================ }
procedure SHA1Clear( var Context : TSHA1Context );
begin
  fillchar( Context, SizeOf( Context ), $00 );
end;
{ ---------------------------------------------------------------------------- }
end.
```