Ім'я користувача:
приховано налаштуваннями конфіденційності

Дата перевірки:
02.06.2023 10:49:52 EEST

Дата звіту:
02.06.2023 10:55:08 EEST

Назва документа: Охотов_Богдан_ОК_41
ID перевірки: 1015386560

Тип перевірки: Doc vs Library

ID користувача: 100011372

Кількість сторінок: 35 Кількість слів: 6851 Кількість символів: 48818 Розмір файлу: 804.50 KB ID файлу: 1015051607

17.3%
Схожість

Найбільша схожість: 5.75% з джерелом з Бібліотеки (ID файлу: 1014968858)

Пошук збігів з Інтернетом не проводився

Вилучення цитат вимкнене

.....

...С..т..о..р..і.н..к.а...3..7.....

0% Цитат

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Назва документа: Охотов_Богдан_ОК_41 ID файлу: 1015051607

1 ОСНОВНІ ПОНЯТТЯ ПРО ШИФРУВАННЯ ВІДКРИТИХ ТЕКСТІВ 1.1

Основні

Зміна вигляду відкритого тексту так, щоб заховати його зміст, називається кодуванням або шифруванням. Шифроване повідомлення називається шифротекстом. Процес перетворення шифротексту у відкритий текст називається розшифруванням (дешифруванням) або криптоаналізом. Процесами шифрування і дешифрування повідомлень займається криптографія. Галузь, що охоплює криптографію і криптоаналіз, називається криптологією. а люди, які нею займаються, називаються криптологами.

Відправник хоче відправити своє повідомлення і бути впевненим, що його не зможе ніхто перехопити і прочитати. Тому текст повідомлення необхідно зашифрувати. Одержувач повідомлення повинен розшифрувати одержану інформацію. Схему шифрування і розшифрування інформації показано на рис. 1.1.

Рисунок 1.1 – Схема шифрування і дешифрування інформації

В ролі відкритого тексту може бути текстовий файл, потік бітів або бітове зображення. Відкритий текст створений для зберігання або передачі даних. Він повинен бути зашифрований.

Позначимо відкритий текст через M, а шифротекст – через C. Розмір шифротексту може співпадати з відкритим текстом, або бути різним. Якщо шифрування супроводжується стисненням, то розмір шифротексту C є меншим за розмір відкритого тексту M. Якщо шифрування є розширенням вхідного тексту, то розмір шифротексту C є більшим від розміру відкритого тексту M. Якщо

позначити через E функцію шифрування початкового тексту, а через D – функцію дешифрування зашифрованого тексту, то функція кодування E діє на M, створюючи шифротекст C. Математично цей процес виражається формулою:

$$E(M) = C \quad (1.1)$$

У зворотному процесі функція дешифрування D діє на шифротекст C, відновлюючи початковий текст M: Математично це записуєтьбся формулою:

$$D(C) = M \quad (1.2)$$

Оскільки метою шифрування і дешифрування повідомленнь є відновлення початкового відкритого тексту, то виконується рівність (1.3):

$$D(E(M)) = M \quad (1.3)$$

Криптографічні алгоритми або шифри є математичними функціями, які використовується для шифрування і дешифрування повідомлення. Згідно формули (1.3) ці функції пов’язані між собою.

Якщо безпека алгоритму ґрунтується тільки на збереженні в таємниці самого алгоритму, то це обмежений алгоритм. Велика група користувачів не може користуватися обмеженими алгоритмам, оскільки при покиданні групи конкретним користувачем, її члени повинні переходити на інший алгоритм. Алгоритм повинен бути змінений і тоді, коли розкрито секрет алгоритму.

Обмежені алгоритми не можуть забезпечити стандартизацію або якісний контроль, оскільки кожна група користувачів має свій унікальний алгоритм. Відповідно, зловмисник може купити такий продукт і розшифрувати секретне повідомлення. Обмежені алгоритми можуть використовуватися тільки для повідомлень з низьким рівнем безпеки.

1.2 Класифікація систем шифрування і розшифрування повідомлень

11

Для підвищення безпеки шифрування інформації з метою її захисту в сучасній криптографії вирішуються ці проблеми за допомогою використання ключа. Ключем може бути будь-який об’єкт, що належить конкретній множині, яка називається простором можливих ключів. Тоді процеси шифрування і дешифрування залежать від використовуваного ключа.

Якщо позначити процес шифрування відкритого тексту M на ключі K1

через $E_{K1}(M)$, то функція (1.1) матиме вигляд:

$$C = E_{K1}(M). \quad (1.4)$$

Процес розшифрування криптограми C за допомогою відомого ключа K2, математично можна записати так:

$$M = D_{K2}(C). \quad (1.5)$$

Якщо ключі, що використовується для шифрування і дешифрування повідомлень однакові, тобто K1=K21=K, то такі системи називаються симетричними. Симетричні системи використовують переважно для перетворення тексту тоді, коли використовуються шифри заміни, перестановки або комбінації

перестановок і замін. Тоді функція D_K повинна бути обернена до функції E_K, тобто виконується рівність:

Оскільки при ключі K отримуємо початковий текст M, то виконується наступна рівність:

$$D_K(E_K(M))=M. \quad (1.6)$$

Симетричну систему шифрування і розшифрування інформації з однаковими ключами показано на рис. 1.2.

исунок 1.2 – Симетрична система шифрування і розшифрування інформації

На передавальній стороні виконується шифрування відкритого повідомлення M за допомогою функції $E_K(M)$ на ключі K , результатом якого є криптограма C. Її передають відкритим каналом зв'язку. На приймальній стороні до отриманого зашифрованого повідомлення C застосовано той самий ключ K і обернене перетворення $D_K(C)$, результатом якого є відкрите повідомлення M . Розшифрування буде вірним, якщо повідомлення не було змінено під час передачі по каналу зв'язку. Симетричні алгоритми характеризуються можливістю швидкого шифрування великих потоків інформації. Симетричні алгоритми дають змогу використовувати одні і ті ж апаратні засоби для шифрування і дешифрування інформації.

Проблемою використання симетричних алгоритмів є зберігання і передача ключа розшифрування повідомлення, так як він є секретною частиною криптосистеми як на передавальній так і на приймальній сторонах. Також потрібна велика кількість ключів для кожного одержувача секретного повідомлення.

Для усунення недоліків симетричних систем були розроблені несиметричні (асиметричні) способи шифрування текстів. В асиметричних алгоритмах при шифруванні і дешифруванні використовуються різні ключі. Ключ для шифрування інформації є відкритим, його не потрібно приховувати. Цей ключ суттєво відрізняється від відповідного ключа дешифрування. Якщо ключ шифрування позначити через K_1, а ключ дешифрування – через K_2, то одержимо наступні співвідношення:

$$E_{K1}(M)=C \quad (1.7)$$

$$D_{K2}(C)=M \quad (1.8)$$

Тоді функція (1.3) матиме вигляд:

$$D_{K2}(E_{K1}(M))=M \quad (1.9)$$

Безпека асиметричних алгоритмів повністю ґрунтується на ключах. Це означає, що алгоритм може бути опублікований і проаналізований. Якщо конкретний ключ не відомий, то ніхто не зможе прочитати секретні повідомлення. Процес інформаційного обміну зображено схематично. Асиметричну систему шифрування і дешифрування текстів з двома різними ключами показано на рис. 1.3.

Рисунок 1.3 – Асиметрична система шифрування і дешифрування текстів

Асиметричні криптосистеми розв’язали основну проблему симетричних криптосистем, пов’язану з розповсюдженням ключів. Адже для за шифрування інформації багатьма користувачами потрібно мати лише одну пару ключів: відкритий для шифрування і секретний для розшифрування. Перевагою асиметричної системи є те, що для створення багатокористувацької системи обміну зашифрованою інформацією не потрібно великої кількості ключів.

Але асиметричні алгоритми набагато повільніші, ніж

симетричні. Повільність пояснюється складністю математичних перетворень, на яких ґрунтуються асиметричні криптосистеми, та великій довжині ключів. Такі задачі виконуються досить повільно на комп'ютері. Але асиметричний метод шифрування можна використати там, де симетричні алгоритми працювати не будуть, наприклад, для створення електронного цифрового підпису.

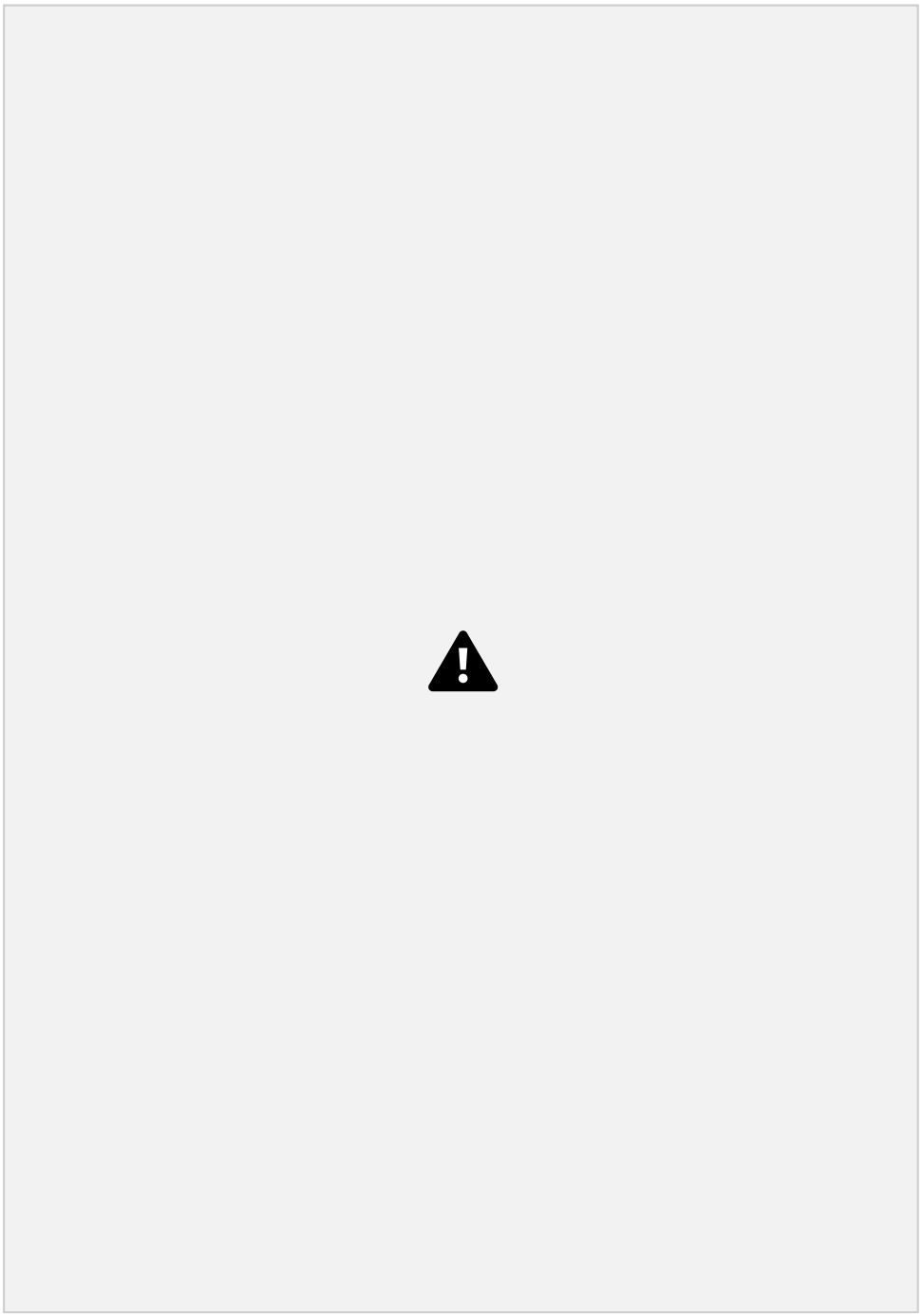
Симетричні алгоритми завдяки високій надійності та швидкодії найкраще підходять для захисту комп'ютерної інформації. Асиметричні алгоритми внаслідок малої швидкодії недоцільно застосовувати для шифрування великих потоків інформації. Асиметричну систему можна використати для обміну інформацією малих об'ємів, тобто можна нею шифрувати ключі для симетричних систем. Для успішної роботи учасники інформаційного секретного обміну повинні заздалегідь домовитися про алгоритм шифрування і розшифрування. Важливим фактором також є передача криптографічного ключа.

1.3 Змішані криптосистеми

На практиці широко використовуються комбіновані системи шифрування інформації. В комбінованих криптосистемах інформація шифрується за допомогою симетричних алгоритмів, а для передачі криптографічних симетричних ключів використовуються асиметричні алгоритми.

Алгоритми з відкритими ключами не замінюють симетричні алгоритми, а використовуються для шифрування ключів. Алгоритми з відкритими ключами працюють повільно. Оскільки спостерігається тенденція до збільшення об'єму передаваної інформації, то завжди потрібно шифрувати дані швидше, ніж це зможе зробити криптографія з відкритими ключами.

Криптосистеми з відкритими ключами уразливіші по відношенню до злому з вибраним відкритим текстом. Якщо C = E(M), де M – відкритий текст, то при n можливих відкритих текстів криптоаналітику потрібно тільки зашифрувати всі n текстів і порівняти результати з шифротекстом C. Він не зможе розкрити ключ



дешифрування, але він зможе визначити відкритий текст. Використання криптографії з відкритими ключами для шифрування ключів вирішує важливу проблему передачі ключів. Функціональну схему роботи комбінованої криптосистеми зображено на рис. 1.4.

Рисунок 1.4 – Функціональна схема роботи комбінованої криптосистеми

Передавальна сторона вибирає відкритий ключ асиметричної системи приймальної сторони і зашифровує ним ключ симетричного алгоритму. Зашифрований ключ симетричного алгоритму відправляється відкритим каналом зв'язку на приймальну сторону, де розшифровується секретним ключем. Відповідно, ключ шифрування відомий обом учасникам обміну інформацією. Комбіновані криптосистеми в даний час дуже популярні.

1.4 Криптографічні протоколи

Задача криптографії полягає у вирішенні проблем секретності, перевірки достовірності і цілісності переданої інформації. Для вирішення цієї задачі використовується протокол, який задає порядок дій двох або більше сторін, які беруть участь в обміні даними. Протокол виконується в певній послідовності, з

початку до кінця обміну. Для реалізації протоколу

потрібно принаймні дві людини, одна людина не зможе реалізувати протокол. Оскільки протокол призначений для вирішення певного завдання, то він повинен приводити до кінцевого результату. Протоколи мають наступні характеристики.

- Кожен учасник протоколу повинен знати протокол і послідовність його дій;
- Кожен учасник протоколу повинен погодитися виконувати протокол;
- Протокол повинен бути несуперечливим, кожна дія повинна бути однозначною, щоб не було нерозуміння між учасниками протоколу;
- Протокол повинен бути повним, кожній можливій ситуації повинно відповідати одне значення;

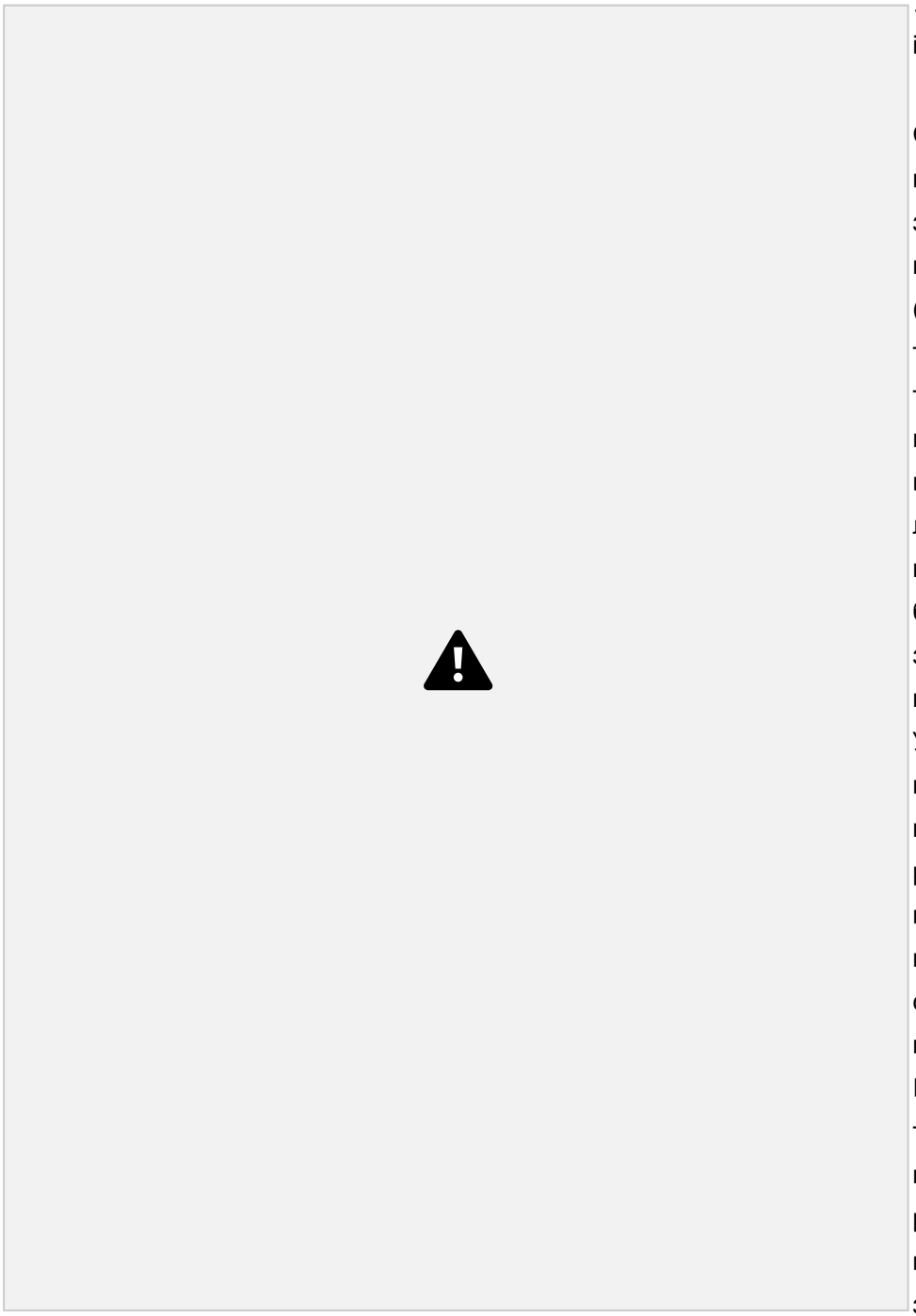
Виконання протоколу має лінійний характер Це означає, що до тих пір, поки не буде команди перейти до наступної дії, до неї переходити не можна. Криптографічний протокол включає криптографічний алгоритм. Учасники протоколу можуть захотіти поділитися секретом один з одним, спільно генерувати випадкову послідовність, підтвердити один одному свою справжність або підписати контракт в один і той же момент часу. Протоколу полягає в неможливості робити або дізнатися більше, ніж це визначено протоколом.

В протоколі беруть участь відправник і одержувач. Як правило, відправник посилає всі протоколи, а одержувач одержує. Виконує протокол незацікавлена третя сторона (посередник). У посередника немає зацікавленості в результаті роботи протоколу і схильності до однієї із сторін. Всі учасники протоколу зобов'язані приймати все, що скаже посередник.

Комп'ютерна мережа забезпечує підтримку посередника. Посередник повинен брати участь в кожному сеансі обміну інформацією. Оскільки кожен учасник інформаційного обміну в мережі повинен довіряти посередникові, то посередник є слабким місцем в мережі при спробі розкриття протоколу, так як він

може стати на будь-яку з сторін. Але не зважаючи на це, посередництво все ще активно використовується при передачі секретних повідомлень.

1.5 Безпека алгоритмів шифрування інформації\



Суть криптографії полягає а збереженні відкритого тексту (або ключа, або і того, і іншого) в таємниці при умові, що зловмисник може контролювати лінії Одержанням відкритого тексту без ключа займаються криптоаналітики. Успішно проведений криптоаналіз може розсекретити відкритий текст або ключ, виявити слабкі місця в криптосистемах. Існують такі основні типіи криптоаналітичного розкриття. Для кожного з них, звичайно, припускається, що криптоаналітик знає використовуваний алгоритм шифрування:

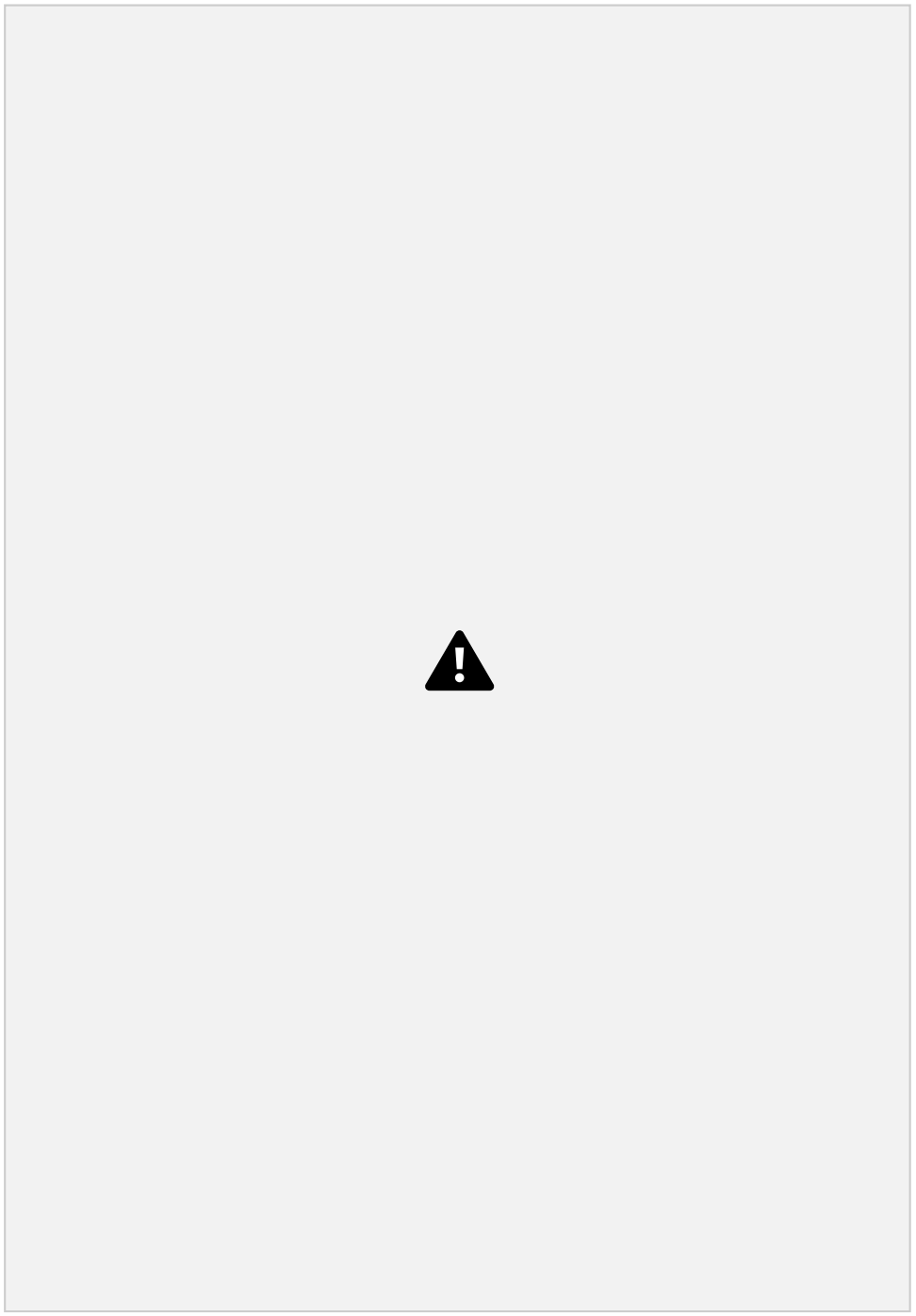
1 Розкриття з використанням тільки шифротексту. У криптоаналітика є шифротексти кількох повідомлень, зашифрованих одним і тим алгоритмом шифрування. Завдання криптоаналітика полягає в розкритті відкритого тексту як можна більшого числа повідомлень, для того щоб дешифрувати інші.

2. Розкриття з використанням відкритого тексту. У криптоаналітика є доступ не тільки до шифротекстів декількох повідомлень, але і до відкритого тексту цих повідомлень. Його завдання полягає в одержанні ключа (або ключів), використаного для шифрування інших повідомлень, зашифрованих тим ключем

або ключами.

3 Розкриття з використанням вибраного відкритого тексту. У криптоаналітика не тільки є доступ до шифротекстів і відкритих текстів декількох повідомлень, але і можливість вибирати відкритий текст для шифрування. Тут криптоаналітик може вибирати шифровані блоки відкритого тексту, які дають більше інформації про ключ шифрування повідомлень. Його завдання полягає в отриманні ключа (або ключів), який використовувався для шифрування повідомлень, або алгоритму, що дозволяє дешифрувати нові повідомлення, зашифровані тим самим ключем (або ключами).

4 Адаптивне розкриття з використанням відкритого тексту. Це окремий випадок розкриття з використанням вибраного відкритого тексту. Криптоаналітик не тільки може вибирати шифрований текст, але також може будувати свій подальший вибір на базі отриманих результатів шифрування. При розкритті з



використанням вибраного відкритого тексту криптоаналітик міг вибрати для шифрування тільки один великий блок відкритого тексту, при адаптивному розкритті з використанням вибраного відкритого тексту він може вибрати менший блок відкритого тексту, потім вибрати наступний блок, використовуючи результати першого вибору і т. д.

5. Розкриття з використанням вибраного шифротексту. Криптоаналітик може вибрати різні шифротексти для дешифрування. Він має доступ до відкритих текстів, що дешифруються. Наприклад, у криптоаналітика є доступ до "чорного ящика", який виконує автоматичне дешифрування. Його завдання полягає в отриманні ключа. Такий тип розкриття звичайно застосовується до алгоритмів з відкритим ключем. Розкриття з використання вибраного шифротексту є досить ефективним.

6 Розкриття з використанням вибраного ключа. Такий тип розкриття означає, що криптоаналітик може вибирати ключ, якщо у нього є деяка інформація про зв'язок між різними ключами.

Різні алгоритми мають різні ступені безпеки залежно від того, наскільки важко зламати алгоритм. Якщо час злому алгоритму більший, ніж час, протягом якого зашифровані дані повинні зберігатися в секреті, то такий алгоритм безпечний. Алгоритм є безумовно безпечним, якщо, незалежно від об'єму шифротекстів у криптоаналітика, він не має достатньої інформації для отримання відкритого тексту.

Алгоритм вважається обчислювально безпечним. якщо він не може бути зламаний з використанням доступних комп'ютерних ресурсів.

В той час, як складність розкриття залишається постійною, то обчислювальні потужності комп'ютерів зростають. Багато криптографічних зломів можна використати для паралельних комп'ютерів. При цьому завдання розбивається на мільярди маленьких кусочків, розв'язання яких не вимагає взаємодії між окремими процесорами. Тому ккриптосистеми повинні проектуватися стійкими до злому з урахуванням розвитку обчислювальних засобів.

Незалежно від того, які алгоритми використовує

криптосистема для шифрування повідомлень, вони повинні задовольняти наступні вимоги:

- Зашифрований текст можна прочитати лише за допомогою ключа;
- Складною повинна бути практична реалізація розкриття криптографічного ключа;
- Розшифрування криптографічного ключа повинно бути практично неможливим для сучасних обчислювальних комп'ютерних систем;
- Висока надійність системи захисту переданих повідомлень;
- Незначна зміна алгоритму шифрування повинна суттєво змінити зашифрований текст;
- В процесі шифрування необхідно забезпечити постійний контроль за даними, що шифруються;
- Довжина зашифрованого тексту не повинна бути набагато більшою, ніж довжина відкритого повідомлення.

Криптографічні алгоритми можуть реалізовуватися як апаратно так і програмно. В дипломному проєкті реалізовано алгоритми програмного захисту інформації при її передачі по каналах зв'язку.

2 ОПИС СИМЕТРИЧНИХ КРИПТОГРАФІЧНИХ МЕТОДІВ

2.1 Симетричні алгоритми для шифрування текстів

Створення надійного шифру є непростою задачею. Чим більша довговічність шифру, тим більша ймовірність його розкриття і читання зашифрованої секретної інформації. Якщо в шифрі є змінний ключ, то його заміна унеможливить швидке розкриття шифротексту і розроблені супротивником методи не дадуть бажаного ефекту. Відправник повідомлення повинен перед початком використання системи шифрування повідомити адресата секретний ключ, щоб той зміг розшифрувати зашифрований текст. Функціональну схему симетричної криптосистеми зображено на рис. 2.1.

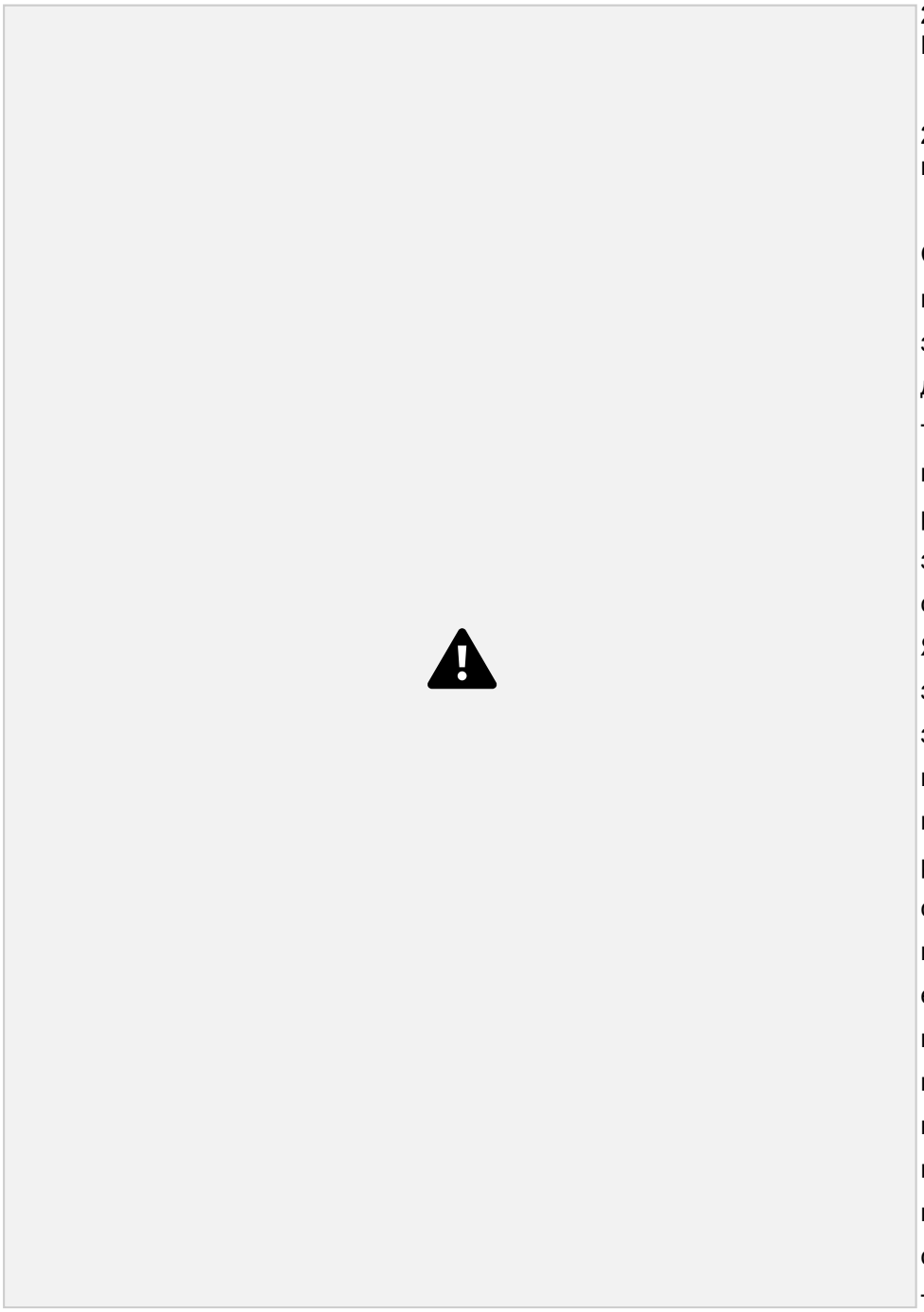


Рисунок 2.1 Функціональна схема симетричної криптосистеми

В симетричному алгоритмі ключ шифрування може бути розрахований по ключу дешифрування і навпаки. У більшості симетричних алгоритмів ключі шифрування і дешифрування одні і ті ж. Ці алгоритми, ще називаються з секретним ключем або алгоритмами з одним ключем. Безпека симетричного алгоритму залежить від ключа. При розкритті ключа може хто завгодно розшифрувати повідомлення.

Симетричні алгоритми діляться на потокові та блокові. При обробці відкритих текстів побітно або посимвольно використовують потокові алгоритмами

або потокові шифри. При роботі з групами бітів або символів відкритого тексту

використовують блокові алгоритми або блокові шифри.

До появи комп'ютерної техніки криптографія використовувала алгоритми на символьній основі. Різні криптографічні алгоритми замінювали одні символи іншими, або переставляли їх. Використовувалися також комбіновані алгоритми, тобто символи переставляли і замінювали порядок їх розміщення. В останній час алгоритми суттєво ускладнилися. Вони працюють з бітами, а не з символами.

2.2 Класифікація шифрів заміни

В шифрах заміни кожний символ відкритого тексту в шифротексті замінюється іншим символом. Одержувач інвертує заміну символу шифротексту, відновлюючи текст відправника. Існує чотири типи шифрів заміни:

– Простий шифр заміни;

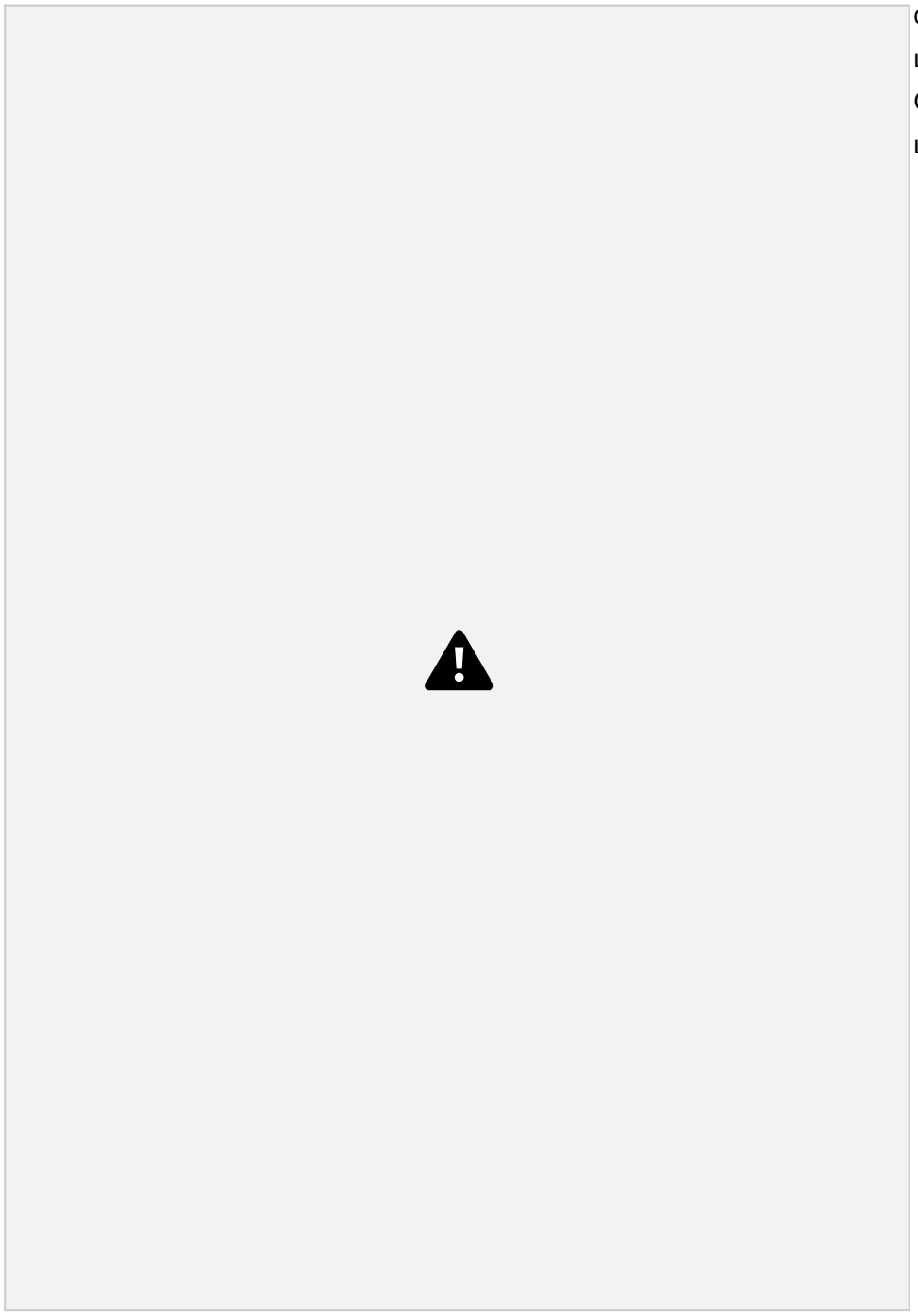
– Однозвучний шифр заміни;

– Блоковий шифр заміни;

– Багатоалфавітний шифр заміни.

Простий шифр заміни ще називають моноалфавітним. Це шифр, в якому кожний символ відкритого тексту замінюється відповідним символом шифротекста. Прикладом шифру заміни є відомий шифр Цезаря, в якому кожен символ відкритого тексту замінюється символом, який знаходиться на три символи правіше по модулю 26 («A» замінюється на «D», «B» – на «E», ... «W» – на «Z», «X» – на «A», «Y» – на "B", «Z» – на "C"). Це простий шифр, оскільки алфавіт шифротексту є зміщеним на задану кількість позицій. Прості ифри заміни легко розкриваються, оскільки вони не ховають частоти використання різних символів у відкритому тексті. Щоб відновити відкритий текст, доброму криптоаналітику потрібно тільки знати 26 символів англійського алфавіту.

Однозвучний шифр заміни подібний на просту підстановочну криптосистему, в якій один символ відкритого тексту відображається на декілька символів шифротексту. Наприклад, символу «A» можуть відповідати декілька



символів шифротексту. Однозвучні шифри заміни використовувалися вже в 1401 році. Вони складніші для розкриття, хоча і не приховують всіх статистичних властивостей мови відкритого тексту. За допомогою розкриття з відомим відкритим текстом ці шифри розкриваються просто. Розкриття з використанням тільки шифротексту складніше, але і воно займає на комп'ютері небагато часу.

Блоковий шифр заміни – це шифр, при якому блоки символів шифруються по групах (блоках). Наприклад, блоку літер «ABC» може відповідати блок літер «RTQ», блоку літер «ABB» може відповідати блок літер «SLL» і т.д.. Фактично блоковий шифр це – система підстановки по алфавіту блоків. Вона може бути моноалфавітною або багатоалфавітною в залежності від режиму блокового шифру. Блокові шифри дуже поширеніші на практиці.

Багатоалфавітний шифр заміни складається з декількох простих підстановочних шифрів. Наприклад, можуть бути використані п'ять різних простих підстановочних шифрів; при якому кожний символ відкритого тексту замінюється шляхом використання одного конкретного шифру. Багатоалфавітний шифр був винайдений Ліном Баттістой в 1568 році. Він використовувався армією Сполучених Штатів в ході Громадянської війни в Америці. Не дивлячись на те, що вони легко могли бути розкриті, багато комп'ютерних програм використовує ці шифри для забезпечення безпеки даних.

В багатолфавітних шифрах використовуються множина однобуквенних ключів, кожний з яких використовується для шифрування одного символу відкритого тексту. Першим ключем шифрується перший символ відкритого тексту, другим ключем – другий символ і т. д. Після використання всіх ключів вони повторюються циклічно Якщо застосовується 20 однобуквених ключів, то кожна двадцята буква шифрується тим же ключем. Цей параметр називається періодом шифру. У криптографії шифри з довгим періодом важче розкрити, ніж шифри з коротким періодом. Використання комп'ютерів дозволяє легко розкрити шифри заміни з довгим періодом Часто використовують один текст для шифрування іншого тексту. Тоді період цього шифру рівний довжині тексту.

2.3 Алгоритм шифрування одноразовими блокнотами

Спосіб шифрування одноразовим блокнотами був винайдений в 1917 році Мейджором Джозефом і Гілбертом

Вернамом. У класичному розумінні одноразовий блокнот є неповторюваною послідовністю символів ключа, розподілених випадковим способом, які написані на шматочках паперу і приклеєних до листка блокнота. Відправник використовує кожний символ ключа

блокнота для шифрування тільки одного символу відкритого тексту. Шифруванням є додавання по модулю 26 (кількість букв алфавіту) числових кодів символів відкритого тексту і символів ключа з одноразового блокнота. Кожний символ ключа використовується тільки один раз і для одного повідомлення. Відправник шифрує повідомлення і знищує використані сторінки блокнота.

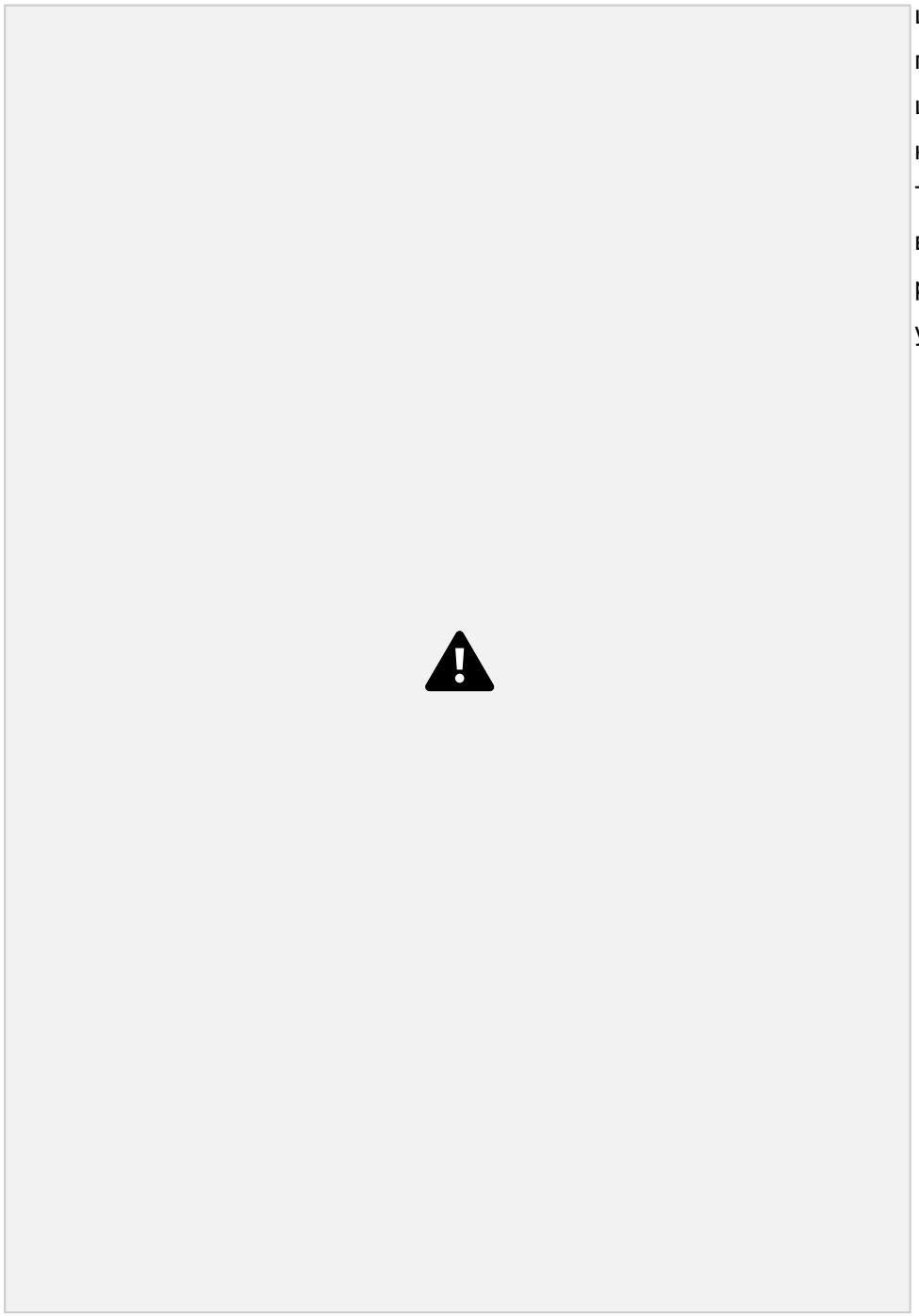
Одержувач, у свою чергу, використовуючи такий самий блокнот, розшифровує кожний символ зашифрованого тексту повідомлення. Розшифрувавши повідомлення, одержувач знищує відповідні сторінки блокнота. Нове повідомлення має новий ключ. Наприклад, повідомлення ONETIMEPAD, яке

шифрується ключовою послідовністю літер в блокноті TBFRGFARFM, матиме вигляд IPKLPSFHGQ, оскільки мають місце співвідношення:

$$Q+T \bmod 26=I,$$
$$N+B \bmod 26 =P,$$
$$E+ F \bmod 26 = K,$$
$$T+R \bmod 26 =L,$$
$$\dots$$

Якщо зловмисник не зможе дістати доступ до ключового слова в одноразовому блокноті, яке використовується для шифрування повідомлення, то цей метод шифрування абсолютно безпечний. Даний шифротекст на вигляд відповідає будь-якому відкритому повідомленню того ж розміру.

Оскільки всі ключові послідовності абсолютно однакові, оскільки символи ключа генеруються випадковим способом, то у супротивника відсутня інформація,



що дозволяє піддати шифротекст криптоаналізу. Так як всі відкриті тексти рівноймовірні, у криптоаналітика немає можливості визначити, який з відкритих текстів є правильним. Випадкова ключова послідовність літер, складена по модулю з невипадковим відкритим текстом, дає абсолютно випадковий шифротекст, і ніякі обчислення не зможуть його розкрити.

Важливим є той факт, що ключову послідовність ніколи не можна використовувати другий раз. Навіть якщо ви використовуєте блокнот розміром в декілька гігабайт, то якщо криптоаналітик отримає декілька текстів з ключами, що перекриваються, він зможе відновити відкритий текст . Він перевірить кожну пару шифротекстів і підрахує число збігів в кожній позиції. Одноразовий блокнот

зручний для декількох невеликих повідомлень, але його не можна використовувати для шифрування великих текстів при передачі по каналу зв'язку з малою пропускною здатністю.

При використанні одноразових блокнотів необхідно точно синхронізувати роботу відправника і одержувача повідомлення. Якщо одержувач пропустить символ або кілька символів пропадуть при передачі, то повідомлення втратить всякий сенс. Але, якщо декілька символів зміниться при передачі, то лише ці символи будуть розшифровані неправильно. Одноразовий блокнот не забезпечує перевірку достовірності. Одноразові блокноти використовуються на практиці в основному для передачі секретних ключів і надсекретних коротких повідомлень.

Простим різновидом одноразових блокнотів є потокові шифри. Потокові шифри перетворюють текст у шифрований послідовно по одному біту. Генератор ключової послідовності видає послідовність бітів $k_1, k_2, \ldots, k_i, \ldots$. Ця ключова послідовність додається за модулем два з послідовністю бітів відкритого тексту $p_1, p_2, \ldots, p_i, \ldots$ для отримання зашифрованого тексту $c_i = p_i \oplus k_i$.

Якщо ключова послідовність бітів має малий період, то надійність системи шифрування є невисокою. При нескінченній ключовій послідовності одержимо одноразовий блокнот з високою безпекою шифрування даних. Генератор ключової послідовності видає випадковий потік бітів, але насправді він є детермінованим і може бути визначеним на приймальній стороні. Чим більше в згенерованому

потоці випадковості, тим складніше

криптоаналітику розкрити зашифроване повідомлення.

Якщо ключова послідовність буде весь час однаковою, то дуже легко буде розкрити зашифроване повідомлення. Тому для безпеки шифрування використовується ключ, який робить неможливим простий криптоаналіз зашифрованого повідомлення. Потокові шифри використовуються для шифрування неперервних потоків інформації у мережах передачі даних.

Ідея одноразового блокнота легко поширюється на двійкові дані. Замість одноразового блокнота, який складається з букв, використовується одноразовий блокнот з бітів. При цьому використовується побітова операція XOR. Для дешифрування застосовується операція XOR до шифротексту з тим же одноразовим блокнотом.

2.4 Опис системи симетричного шифрування методом заміни Цезаря

При шифруванні методом заміни символи відкритого тексту шифруються символами того ж алфавіту за встановленими правилами заміни. Основне, щоб адресат одержаного повідомлення знав довжину і напрям зсуву літер. Кожний символ тексту замінюється іншими символами протягом усього тексту.

Для розробки програмного забезпечення шифрування відкритого тексту з метою його захисту використано шифр Цезаря. Система Цезаря полягає в перетворенні відкритого тексту так, що кожна літера замінюється третьою після неї літерою алфавіту. Порядок літер йде по колу, тобто за літерою «я» слідує літера «а». Особливістю цієї системи є можливість використовувати зміщення літер на будь-яке число та їх заміни.

В методі заміни, що базується на шифрі Цезаря, вибирається деяке число m, $0 \blacksquare m \blacksquare n$, де n - кількість символів алфавіту, на якому написано текст. Для українського алфавіту ця кількість рівна 32. В табл. 2.1 наведено співвідношення між літерами українського алфавіту та їх числовими кодами.

Таблиця 2.1 – Співвідношення між літерами та їх числовими кодами

6 7 8 9 10 11 12 13 14 15 Літер

аА Б В Г Д Е Є Ж З И І Ї Й К Л М Код 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 Літер

аН О П Р С Т У Ф Х Ц Ч Ш Щ Ь Ю Я

При застосуванні шифру Цезаря (m ●3) співвідношення між кодами літер до початкового алфавіту наведено в табл. 2.2.

Таблиця 2.2 – Заміна літер при використанні шифру Цезаря при величині зсуву 3

Код 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 Літера А Б В Г Д Е Є Ж З И І Ї Й К Л М Заміна Г Д Е Є Ж З И І Ї Й К Л М Н О П Код 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 Літера Н О П Р С Т У Ф Х Ц Ч Ш Щ Ь Ю Я Заміна Р С Т У Ф Х Ц Ч Ш Щ Ь Ю Я А Б В

На основі одержаної підстановки табл. 2.2 кожну літеру повідомлення ДИПЛОМНИЙ ПРОЄКТ замінимо на літеру, згідно табл. 3.2. Відповідно, одержимо зашифрований текст ЖЙТОСПРЙМ ТУСИНХ.

В табл. 2.3 показано заміну літер основного алфавіту для шифрування при величині зсуву вліво на 10 позицій (m ● ᵡ 10).

Таблиця 2.3 – Заміна літер шифру Цезаря при величині зсуву -10

Код 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 Літера А Б В Г Д Е Є Ж З И І Ї Й К Л М Заміна У Ф Х Ц Ч Ш Щ Ь Ю Я А Б В Г Д Е
Код 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 Літера Н О П Р С Т У Ф Х Ц Ч Ш Щ Ь Ю Я Заміна Є Ж З И І Ї Й К Л М Н О П Р С Т На основі одержаної підстановки кожну літеру повідомлення ДИПЛОМНИЙ ПРОЄКТ замінимо на літеру, згідно табл. 2.3. Відповідно, одержимо зашифрований текст ЧЯЗДЖЕЄЯВ ЗИЖЩГЇ.

2.5 Опис системи шифрування методом заміни з ключовим словом

Чим більша тривалість шифру, тим більша ймовірність його розкриття і читання зашифрованих повідомлень. Для безпеки розкриття шифрованої інформації використовують ключі. Такий шифр унеможливить швидке розкриття

шифротексту і розроблені супротивником методи не дадуть бажаного ефекту. Для розробки програмного забезпечення шифрування відкритого тексту використано систему підстановок Цезаря з ключовим словом. Особливістю цієї системи є можливість використання ключового слова для зміщення та заміни порядку символів у алфавіті підстановки.

В методі шифрування за допомогою підстановок Цезаря з ключем вибирається деяке число зсуву k , 0 ■ k ■ n , де n – кількість символів алфавіту, вибраного для написання текстів, і ключове слово або ключову фразу. Для підвищення безпеки шифрування бажано, щоб усі літери ключового слова були різними. Ключове слово записується під буквами алфавіту, починаючи з літери, код якої співпадає з числом зсуву k .

Нехай вибрано ключове слово ДИПЛОМ та число k ● 10 .Слово ДИПЛОМ починається підписувати під літерою, код якої співпадає з вибраним ключовим числом k ● 10 . Співвідношення між числовими кодами літер вибраного алфавіту для шифрування відкритого тексту наведено в табл. 2.1. Решту літер записуємо в алфавітному порядку після ключового слова за винятком літер, які наявні в слові ДИПЛОМ. Літери, які залишилися, записуються спочатку таблиці.

В таблиці 2.4 наведено підписування ключового слова ДИПЛОМ, починаючи з заданої позиції k ● 10 .

Таблиця 2.4 – Підписування ключового слова ДИПЛОМ з позиції k ● 10

Код	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Літера	А	Б	В	Г	Д	Е	Є	Ж	З	И	І	Ї	Й	К	Л	М
Заміна	Д	И	П	Л	О	М										

Код	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Літера	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ю	Я

В таблиці 2.5 показано підстановку літер до початкового алфавіту з

ключовим словом ДИПЛОМ та числом k ● 10 .

Таблиця 2.5 – Записування літер після ключового слова ДИПЛОМ

	Код 0 1 2 3 4 5 6 7 8 9 10 11 12 12 14 15 Літера А Б В Г Д Е Є Ж З И І Ї Й К Л М Заміна У Ф Х Ц Ч Ш Щ Ь Ю Я Д И П Л О М Код 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 Літера Н О П Р С Т У Ф Х Ц Ч Ш Щ Ь Ю Я Заміна А Б В Г Е Є Ж З І Ї Й К Н Р С Т

На основі табл. 2.5 кожна літера тексту ДИПЛОМНИЙ ПРОЄКТ замінена на іншу літеру. Одержино шифрований текст ЧЯВОБМАЯП ВГБЩЛЄ. Цей метод можна використати для криптографічного захисту база даних членів організації, де необхідно надати

доступ до бази не всім членам організації. Можна зашифрувати базу даних так, щоб отримати адресу однієї людини було легко, а витягувати список поштових адрес всіх членів складно. Цього можна досягти, якщо в ролі ключа вибрати прізвище співробітника організації.

Поле, в якому зберігається ім'я і адреса, шифрується за допомогою ключа-прізвища. При невідомому прізвищу не можна розшифрувати поле даних співробітника. При пошуку конкретного прізвища воно беруться ролі ключа, і розшифровуються дані в базі. Наявність декількох збігів означає, що база даних містить інформацію про декількох співробітників з однаковими прізвищами. 2.6 Передача інформації з використанням симетричної криптографії

Для безпечного обміну інформацією двома сторонами її шифрують, дотримуючись певних правил шифрування:

- Вибір системи шифрування;
- Вибір ключа для шифрування;
- Шифрування відправником відкритого тексту з використанням вибраного алгоритму і ключа;
- Відправлення шифрованого повідомлення одержувачу;
- Дешифрування одержувачем шифротексту використанням алгоритму дешифрування і ключа, одержання відкритого тексту повідомлення.

У добрій

криптосистемі безпека шифрування повністю залежить від знання ключа і не залежить від знання алгоритму. Саме тому управління ключами так важливо в криптографії. Ключ повинен залишатися в секреті перед, після і протягом роботи протоколу шифрування. Інакше повідомлення буде розкрито. Розподіл ключів повинен проводитися в секреті. Ключі такі ж важливі, як і повідомлення, зашифровані цими ключами, оскільки знання ключа дозволяє розкрити зміст повідомлення. Якщо ключ розкрито, то можна розшифрувати всі повідомлення, зашифровані цим ключем.

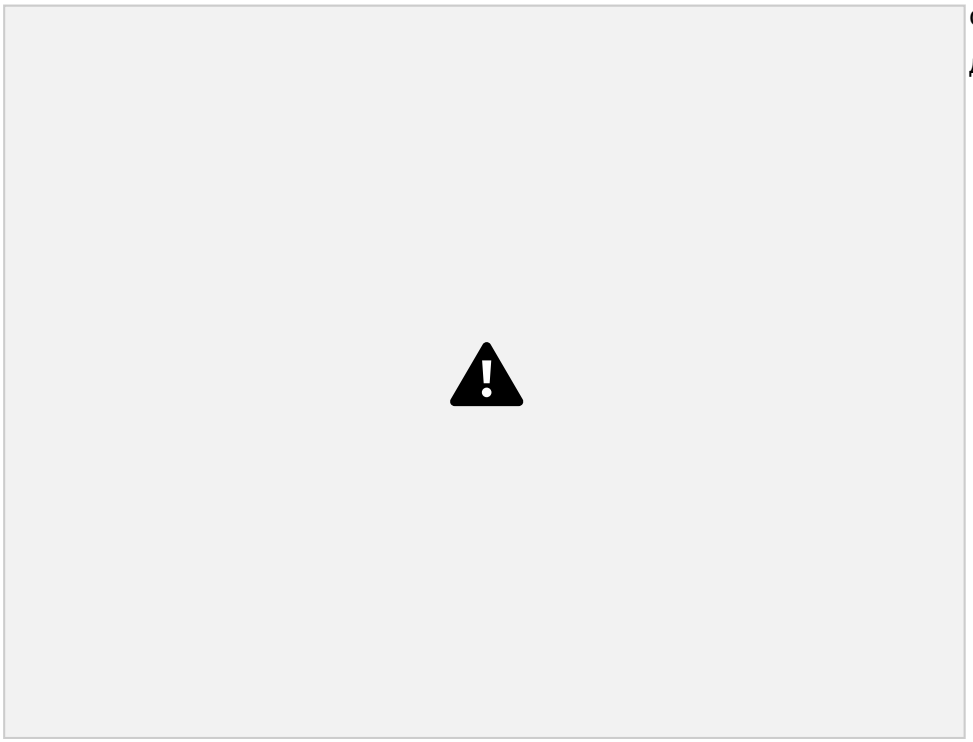
При передачі секретної інформації з використанням симетричних алгоритмів важливою задачею є обмін ключами. Загальноприйнятою криптографічною технікою є шфрування кожного індивідуального обміну повідомленнями окремим ключем. Такий ключ називається сеансовим, оскільки він використовується для єдиного окремого сеансу обміну інформацією. Сеансові ключі корисні, оскільки час їх існування визначається тривалістю сеансу зв'язку. Передача цього загального сеансового ключа до рук тих, що обмінюються інформацією, є складною проблемою .

Обмін ключами за допомогою симетричної криптографії повинен здійснюватися перед початком обміну даними. Ці ключі повинні бути у користувачів. При обміні ключами слід дотримуватися наступних правил:

- Генерується випадковий сеансовий ключ. Він зашифровує дві копії ключа: одну для відправника повідомлення, а іншу – для його одержувача;
- Відправник розшифровує свою копію сеансового ключа;
- Відправник посилає одержувачу ого копію сеансового ключа.
- Одержувач розшифровує свою копію сеансового ключа.
- Відправник і одержувач повідомлення використовують цей сеансовий ключ для безпечного обміну інформацією.

Проблема виділення надійного сеансового ключа для пари комп’ютерів (або людей) в мережі настільки важлива, що вимагає додаткових досліджень.

Необхідно проводити перевірку достовірності обміну ключами. З цією метою створюються різні засоби перевірки (комп’ютерні програми), розроблені



спеціально для аналізу криптографічних протоколів. Розробляються експертні системи, що дозволяють розробляти і досліджувати різні сценарії втрати чи не втрати ключа для розкриття змісту повідомлення.

Також синтезовано багато формальних методів, які ґрунтуються на записі властивостей криптографічних систем за допомогою функціональних залежностей. Протоколи моделюються у вигляді алгебраїчної систему.

Слабким місцем симетричних алгоритмів є проблема розподілу ключів, оскільки вони є секретною частиною як на передавальній, так і на приймальній сторонах. Для безпечного обміну інформацією відправник повинен згенерувати ключ і конфіденційно передати його одержувачу повідомлення. Компрометація ключа загрожує всій системі шифрування при передачі даних по каналах зв’язку.

У припущенні, що кожна пара користувачів мережі використовує окремий ключ, загальне число ключів швидко зростає із зростанням числа користувачів. Мережа з n користувачів вимагає n{n - l)/2 ключів. Наприклад, для спілкування 10 користувачів між собою потрібно 45 різних ключів, для 100 користувачів буде потрібно 4950 ключів.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЗАХИСТУ ІНФОРМАЦІЇ НА ОСНОВІ МЕТОДІВ ЗАМІНИ

3.1 Постановка завдання

Розробити алгоритми і написати програмні коди для шифрування відкритих текстів методом заміни на основі шифру Цезаря з ключами і без ключів, а також розробити алгоритмі і програмні коди для розшифрування відповідних шифротекстів.

Програма шифрування повинна виконувати наступні функції:

1. Читати заданий текст з файлу на диску.

2. Зашифровувати текст шифром Цезаря без ключа за заданою величиною зсуву символів, введеною з клавіатури. Зсув вправо задається додатним цілим числом, а зсув вліво – від'ємним цілим числом.

3. Записати шифротекст в інший файл.

4. Вивести на екран шифротекст з файлу.

Програма дешифрування повинна виконувати такі функції:

1. Читати з файлу зашифровану інформацію.

2. На основі відомої величини зсуву кодів літер розшифрувати зашифрований текст. Одержувач повідомлення для розшифрування тексту повинен знати величину зсуву.

3. Записати розшифровану інформацію в новий файл.

4. Вивести розшифрований текст на екран.

При написанні програм передбачити повідомлення про помилки при відкритті файлів й виконанні операцій вводу-виводу.

Програмне забезпечення розробити в програмному середовищі мови С. Головною особливістю мови є компактність програмного коду. Це, в свою чергу, відіграє велику роль при програмній реалізації великих за обсягом проєктів, оскільки на даний час проблема економії часу при розробці великих за обсягом

проєктів є одною з найактуальніших задач, з якою стикаються фахівці при шифровані і розшифрування повідомлень.

При проєктуванні програм шифрування і розшифрування повідомлень використовувався принцип розділення їх на окремі функціональні компоненти, котрі в процесі виконання програми взаємодіють між собою. Таким підхід в використовується для розробки багатофункціонального програмного забезпечення, що сприяє кращій структуризації коду.

3.2 Опис програмного забезпечення для шифрування методом заміни на основі шифру Цезаря

Програма призначена для шифрування відкритого тексту за допомогою симетричного алгоритму заміни, що ґрунтується на шифрі Цезаря.

Алгоритм програми складається з наступних функціональних компонентів: ✗ Підключення бібліотек і опис змінних;

- Проведення синтаксичного аналіз даних у файлі, представлених у вигляді масиву символів;
- Шифрування (заміна) символів.

Операції синтаксичного аналізу полягають у визначені формату в якому зберігаються дані. Ідентифікація тих чи інших даних здійснюється із використанням певних критеріїв. У випадку з синтаксичним аналізом шляхів до файлів спочатку здійснюється синтаксичний розбір шляху на окремі складові частини. Якщо в процесі аналізу були отримані всі складові частини, то значення запису містить дані, які являють собою шлях до файлу.

На рис. 3.1 зображена структурна схема всіх компонентів, що входять до складу програми шифрування повідомлень методом заміни на основі шифру Цезаря.

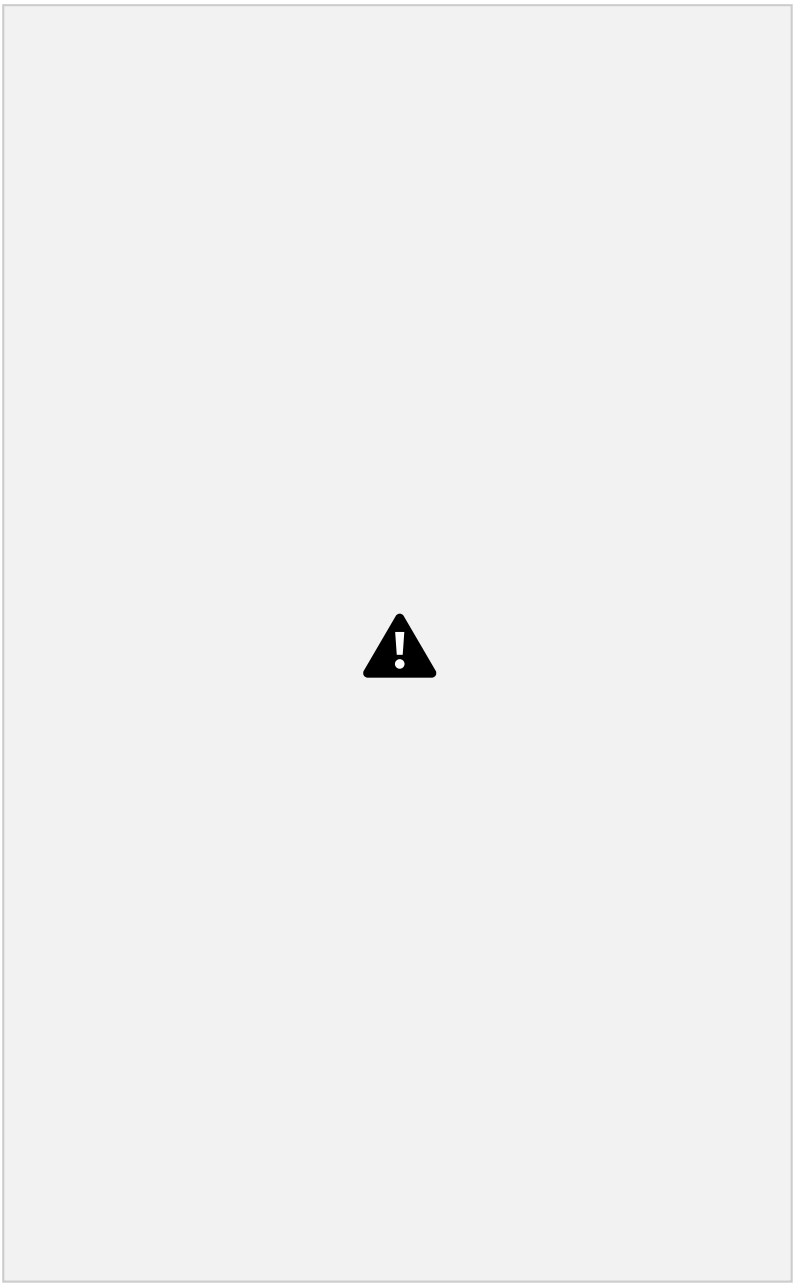


Рисунок 3.1 Структурна схема алгоритму програми шифрування відкритого тексту на основі шифру Цезаря

Програма складається з наступних елементів.

1. Підключення бібліотечних файлів, які містять прототипи функцій файлового вводу-виводу та опрацювання символьних даних.

2. Опис вказівників на файлові змінні структурного типу FILE та задання імен файлів для зберігання вхідного і зашифрованого текстів:

```
FILE *fp1,*fp2;
char filename1[30]="c:\\file_input.txt";
char filename2[30]="c:\\file_text.txt";
```

3. Відкриття файлу c:\\file_input.txt для читання відкритого тексту і файлу c:\\file_text.txt для запису шифрованого тексту. Перевірка правильності відкриття файлів.

4. Задання літер українського алфавіту, якими задається і шифрується текст (таб. 3.1).

Таблиця 3.1 ŠКоди літер українського алфавіту

```
KOD[0]='А'; KOD[1]='Б'; KOD[2]='В'; KOD[3]='Г';
KOD[4]='Д'; KOD[5]='Е'; KOD[6]='Є'; KOD[7]='Ж';
KOD[8]='З'; KOD[9]='И'; KOD[10]='І'; KOD[11]='Ї';
KOD[12]='Й'; KOD[13]='К'; KOD[14]='Л'; KOD[15]='М';
KOD[16]='Н'; KOD[17]='О'; KOD[18]='П'; KOD[19]='Р';
KOD[20]='С'; KOD[21]='Т'; KOD[22]='У'; KOD[23]='Ф';
KOD[24]='Х'; KOD[25]='Ц'; KOD[26]='Ч'; KOD[27]='Ш';
KOD[28]='Щ'; KOD[29]='Ь'; KOD[30]='Ю'; KOD[31]='Я';
```

5. Введення з клавіатури величини і напряму зсуву літер при шифруванні. При зсуві літер вліво знак зсуву додатний, а при зсуві вправо – від’ємний. 6. Читання вхідного тексту з файлу file_input.txt в пам’ять. Перевірка кінця файлу.

7. Заміна прочитаних літер згідно розробленого алгоритму і запис зашифрованого повідомлення в файл file_text.txt.

Результатом роботи програми є створення файлу file_text.txt, в якому знаходиться зашифроване повідомлення. Повний текст програми на мові С для шифрування повідомлень наведено в додатку А.

3.3 Опис програми розшифрування на основі шифру Цезаря

Програма призначена для розшифрування шифротексту, зашифрованого методом заміни, що ґрунтується на шифрі Цезаря. Програма замінює літери шифротексту на літери відкритого тексту згідно заданої величини зсуву (ключа). Програма складається з наступних елементів:

1. Підключення бібліотечних файлів, які містять прототипи функцій файлового вводу-виводу та опрацювання символьних даних.

2. Опис вказівників на файлові змінні структурного типу FILE та задання імен файлів для зберігання шифротексту і розшифрованого тексту:

```
FILE *fp2,*fp3;
char filename1[30]= "c:\\file_text.txt";
char filename2[30]="c:\\file_output.txt";
```

3. Відкриття файлу c:\\file_text.txt для читання шифроексту і файлу c:\\file_output.txt для запису розшифрованого тексту. Перевірка правильності відкриття файлів.

4. Задання літер українського алфавіту, якими задається і шифрується текст (таб. 3.1).

5. Введення з клавіатури величини і напряму зсуву для розшифрування. При зсуві літер вліво знак зсуву додатний, а при зсуві вправо – від'ємний. 6. Читання шифротексту з файлу c:\\file_text.txt в пам'ять.

7. Заміна прочитаних літер згідно розробленого алгоритму дешифрування і запис розшифрованого повідомлення в файл c:\\file_output.txt.

На рис. 3.2 зображено структурну схему алгоритму програми розшифрування повідомлень на основі шифру Цезаря.

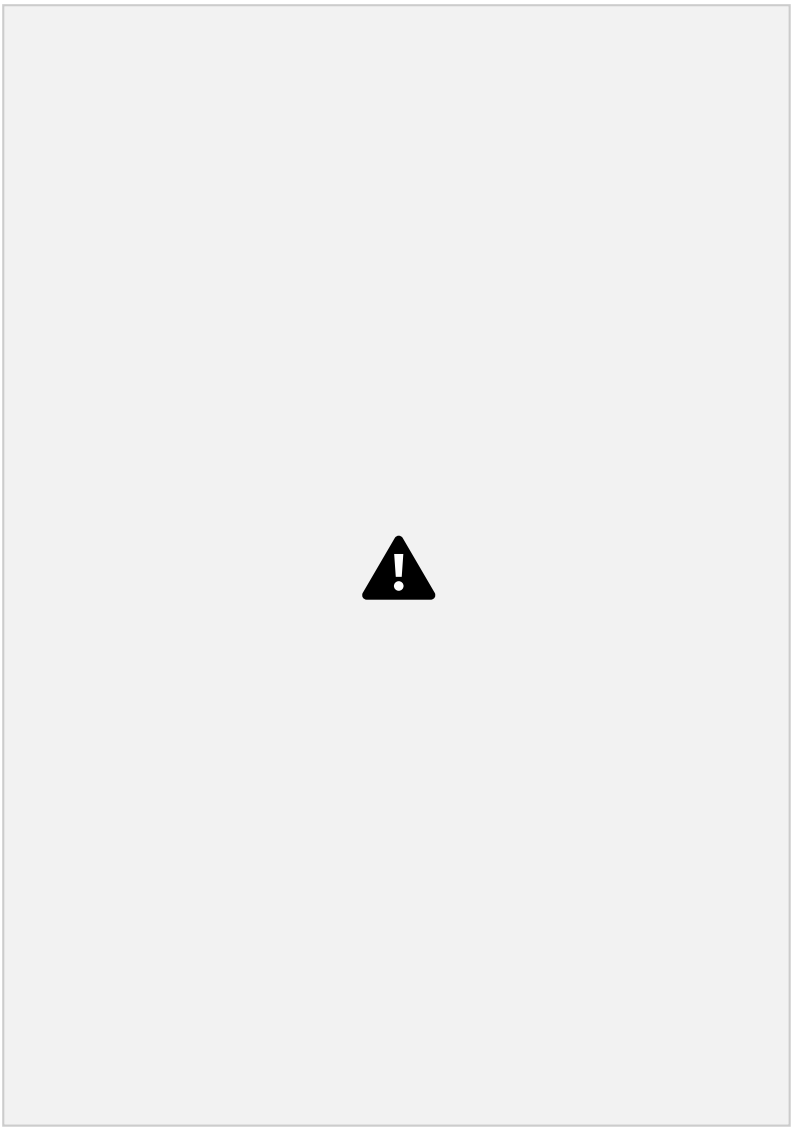


Рисунок 3.2 Структурна схема алгоритму програми розшифрування зашифрованого тексту на основі шифру Цезаря

Результатом роботи програми для розшифрування шифротексту на основі шифру Цезаря є створення файлу file_output.txt, в якому знаходиться

розшифроване повідомлення. Повний текст програми на мові С наведено в додатку Б.

3.4 Результати роботи програм шифрування та дешифрування текстів на основі шифру Цезаря

Програма Zaxuct_KOD.c шифрує відкритий текст на основі шифру Цезаря. Вхідними даними для програми є відкритий текст для шифрування (файлі file_input.txt). Текст наведено на рис. 3.3.

Рисунок 3.3 – Відкритий текст для шифрування на основі шифру Цезаря

Для шифрування вибрано додатний напрям зсуву, величина якого рівна «8». Результатом роботи програми є шифротекст, який записується у файл file_text.txt. На рис. 3.4 наведено шифротекст зашифрованого відкритого тексту з заданим напрямом і величиною зсуву.

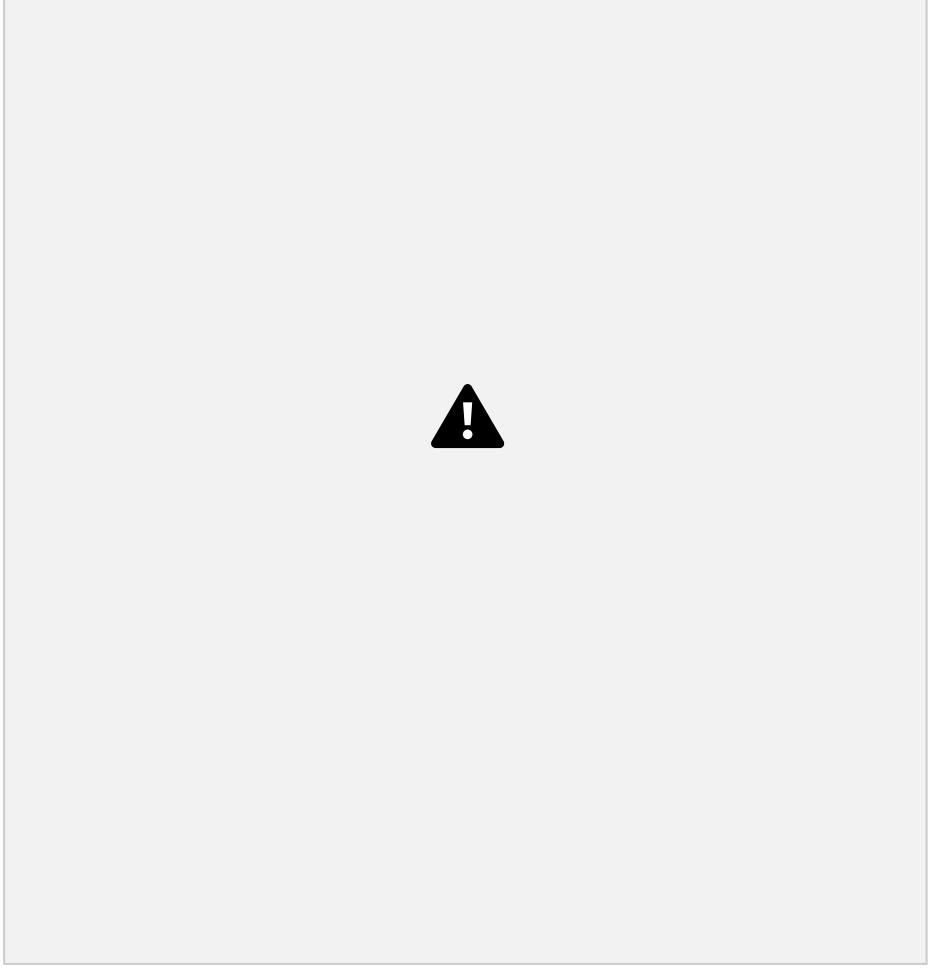


Рисунок 3.4 – Шифротекст перетвореного відкритого тексту

Для дешифрування шифротекстів, зашифрованих шифром Цезаря, використовується програмний код Zaxuct_DEKOD.c. Вхідними даними для програми є зашифрований текст для розшифрування у файлі file_text.txt (рис. 3.4). Результатом роботи програми є розшифрований текст, який записується у файл file_output.txt. На рис. 3.5 наведено початковий відкритий текст.

Рисунок 3.5 – Початковий текст розшифрованого шифротексту

Аналіз одержаних результатів показує, що після проведеної операції шифрування відкритого тексту, наведеного на рис. 3.3, та операції розшифрування зашифрованого тексту, одержано початковий текст, представлений на рис. 3.5, тобто початковий текст розшифровано правильно.

3.5 Опис схеми шифрування і дешифрування текстів на основі системи Цезаря з ключем

Описана в дипломній роботі програма здійснює шифрування повідомлень на системі Цезаря з ключем, який задається деяким числом і довільним ключовим словом. Програма написана на мові С [10]. Головною перевагою використання ключів полягає в тому, що вони забезпечують високу ймовірність безпеки збереження тексту в секреті за рахунок зміни ключів.

При шифруванні системою Цезаря з ключем символи відкритого тексту шифруються символами того ж алфавіту за встановленими правилами заміни. Кожний символ тексту замінюється іншими символами протягом усього тексту з використанням довільного ключового слова і довільної позиції його підписування.

Відкритий текст для шифрування розміщений в текстовому файлі file_input_k.txt у символьного форматі. Прочитані літери шифруємо згідно таблиці 2.5 і записуємо в файл file_text_K.txt. Програма шифрування виконує наступні функції:

– Читає текст, призначений для шифрування;

– Шифрує текст за допомогою системи Цезаря з ключем, який задається деяким числом і ключовим словом;

– Записує зашшифрований текст в файл file_text_k.txt

При розшифруванні шифротексту операції проводяться в зворотному порядку з використанням цієї ж табл.2.5. Символи шифротексту шукаємо в рядках таблиці з назвою «Заміна», а символи відкритого тексту Šв рядках «Літера»

Структурну схему алгоритму шифрування і розшифрування відкритого тексту на основі системи Цезаря з ключем показано на рис. 3.6.

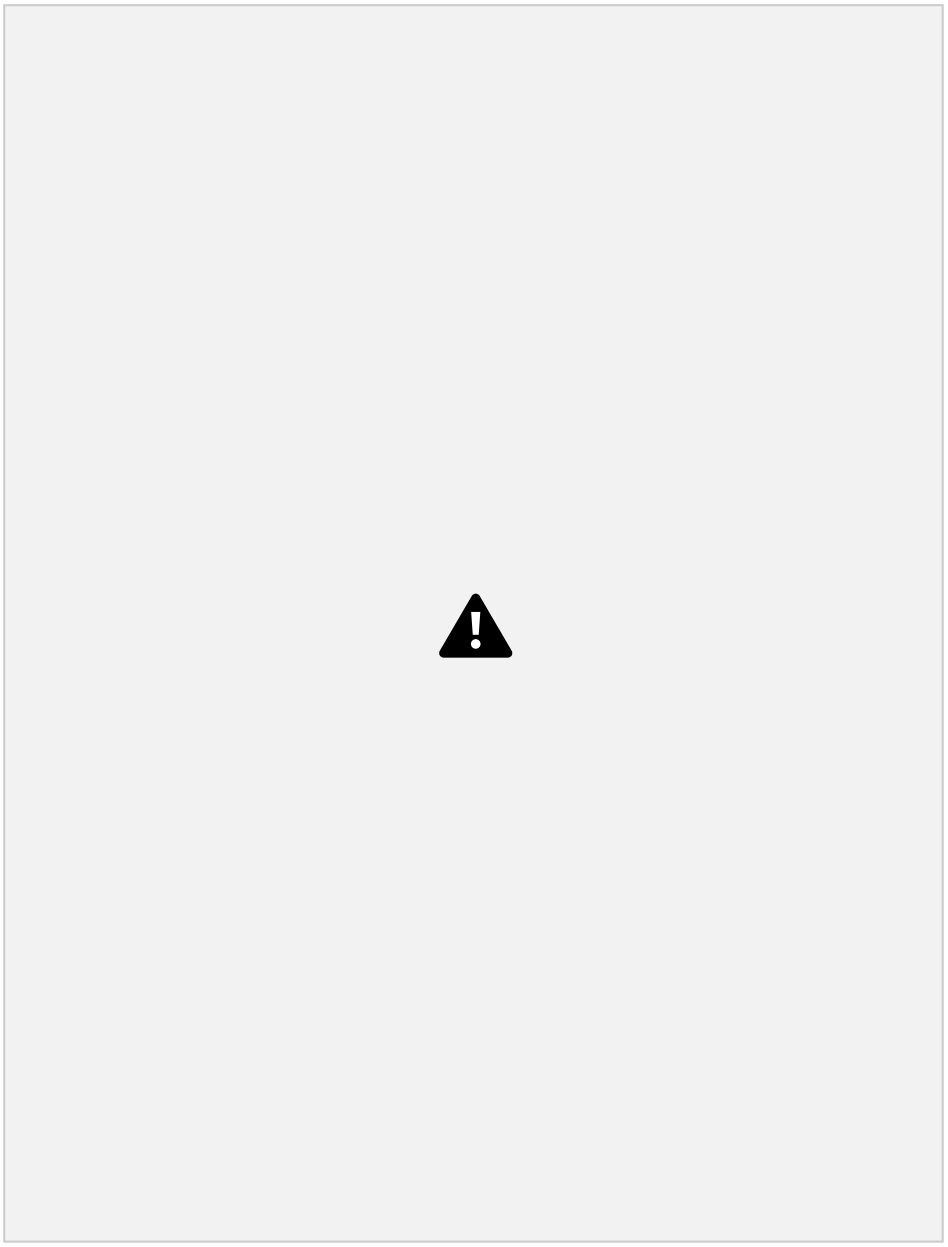


Рисунок 3.6 ŠСтруктурна схема алгоритму програми шифрування і дешифрування текстів на основі системи Цезаря з ключем

Прочитані літери шифруємо згідно таблиці 2.5 і записуємо в файл

file_text_K.txt. Це здійснює наступний фрагмент програми:

```
switch (KOD1)
{case 0: KODS[0]='У'; fprintf(fp2, "%c" ,KODS[0]); break;
case 1: KODS[1]='Ф';fprintf(fp2, "%c" ,KODS[1]); break;
case 2: KODS[2]='Х';fprintf(fp2, "%c" ,KODS[2]); break;
case 3: KODS[3]='Ц'; fprintf(fp2, "%c" ,KODS[3]); break;
case 4: KODS[4]='Ч'; fprintf(fp2, "%c" ,KODS[4]); break;
case 5: KODS[5]='Ш'; fprintf(fp2, "%c" ,KODS[5]); break;
case 6: KODS[6]='Щ'; fprintf(fp2, "%c" ,KODS[6]); break;
case 7: KODS[7]='Ь'; fprintf(fp2, "%c" ,KODS[7]); break;
case 8: KODS[8]='Ю'; fprintf(fp2, "%c" ,KODS[8]); break;
case 9: KODS[9]='Я'; fprintf(fp2, "%c" ,KODS[9]); break;
        case 10: KODS[10]='Д'; fprintf(fp2, "%c" ,KODS[10]); break;
         case 11: KODS[11]='И';fprintf(fp2, "%c" ,KODS[11]); break;
         case 12: KODS[12]='П';fprintf(fp2, "%c" ,KODS[12]); break;
        case 13: KODS[13]='Л';fprintf(fp2, "%c" , KODS[13]); break;
         case 14: KODS[14]='О'; fprintf(fp2, "%c" ,KODS[14]); break;
         case 15: KODS[15]='М';fprintf(fp2, "%c" ,KODS[15]); break;
case 16: KODS[16]='А'; fprintf(fp2, "%c" ,KODS[16]); break;
case 17: KODS[17]='Б';fprintf(fp2, "%c" ,KODS[17]); break;
case 18: KODS[18]='В'; fprintf(fp2, "%c" ,KODS[18]); break;
case 19: KODS[19]='Г'; fprintf(fp2, "%c" ,KODS[19]); break;
case 20: KODS[20]='Е'; fprintf(fp2, "%c" ,KODS[20]); break;
case 21: KODS[21]='Є'; fprintf(fp2, "%c" ,KODS[21]); break;
case 22: KODS[22]='Ж'; fprintf(fp2, "%c" ,KODS[22]); break;
case 23: KODS[23]='З'; fprintf(fp2, "%c" ,KODS[23]); break;
case 24: KODS[24]='І'; fprintf(fp2, "%c" ,KODS[24]); break;
case 25: KODS[25]='Ї'; fprintf(fp2, "%c" ,KODS[25]); break;
case 26: KODS[26]='Й'; fprintf(fp2, "%c" ,KODS[26]); break;
case 27: KODS[27]='К'; fprintf(fp2, "%c" ,KODS[27]); break;
case 28: KODS[28]='Н'; fprintf(fp2, "%c" ,KODS[28]); break;
case 29: KODS[29]='Р'; fprintf(fp2, "%c" ,KODS[29]); break;
case 30: KODS[30]='С'; fprintf(fp2, "%c" ,KODS[30]); break;
case 31: KODS[31]='Т'; fprintf(fp2, "%c" ,KODS[31]); break;
/*case 32: KODS[32]=' '; fprintf(fp2, "%c" ,KOD[32]); break;*/
default: fprintf(fp2, "%c" ,cumvol1); }
```

З 10 позиції записується ключ. Після ключа записуються символи алфавіту, які відсутні в ключовому слові. Решту символів алфавіту записуються паред ключовим словом. Результатом роботи програми є створення файлу file_text_K.txt,

в якому знаходиться зашифроване повідомлення. Повний текст програми на мові С, наведено в додатку В.

Результатом роботи програми дешифрування є створення файлу file_output_K.txt, в якому знаходиться відкритий текст. Повний текст програми на мові С для дешифрування шифротексту, наведено в додатку Д.

3.6 Результати роботи програм шифрування та дешифрування текстів на основі шифру Цезаря з ключем

Програма Zaxuct_KOD_K.c шифрує відкритий текст на основі шифру Цезаря з ключовим слово «ДИПЛОМ» і позицією підписування «10». Вхідними даними для програми є відкритий текст для шифрування (файлі file_input_K.txt). Початковий текст наведено на рис. 3.7.

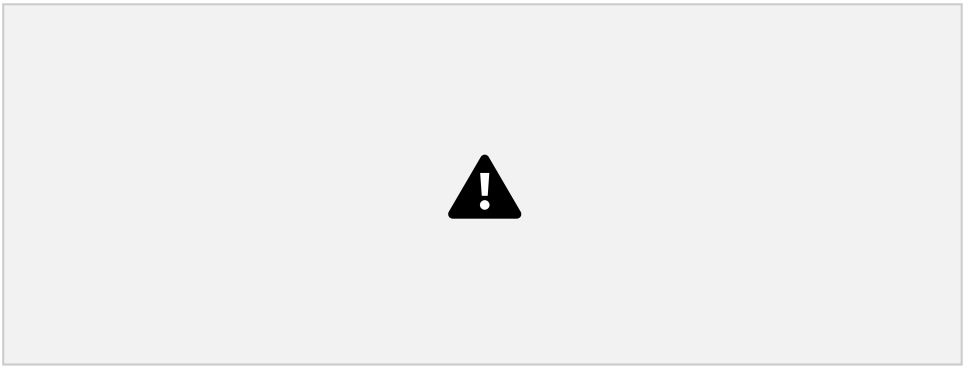


Рисунок 3.7 – Відкритий текст для шифрування на основі шифру Цезаря з ключем

Для шифрування вибрано ключове слово «ДИПЛОМ», позиція підписування рівна «10». Результатом роботи програми є шифротекст, який записується у файл file_text_K.txt. На рис. 3.8 наведено шифротекст зашифрованого відкритого тексту з заданим ключем і номером позиції підписування ключового слова.

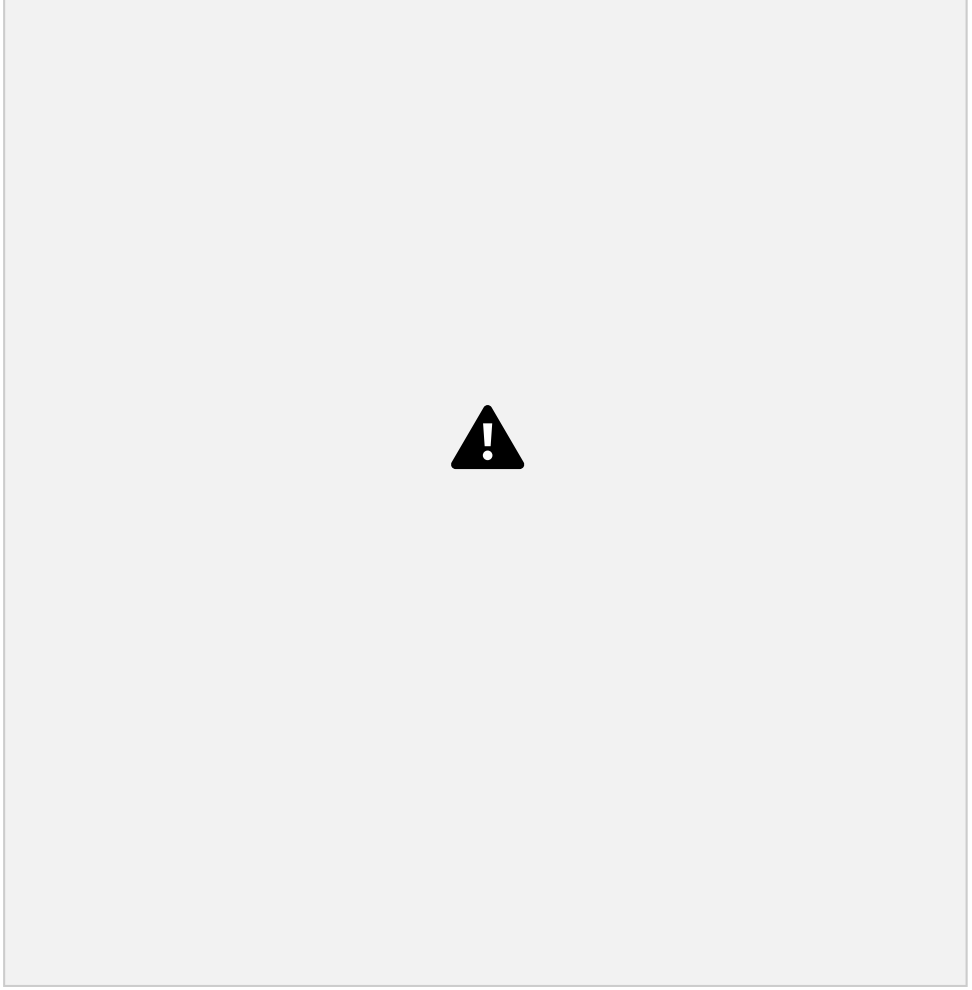


Рисунок 3.8 – Шифротекст перетвореного відкритого тексту з ключем

Для дешифрування шифротекстів, зашифрованих шифром Цезаря з ключем, використовується програмний код Zaxuct_DEKOD_K.c. Вхідними даними для програми є зашифрований текст, розміщений у файлі file_text_K.txt (рис. 3.8).

Результатом роботи програми є розшифрований текст, який записується у файл file_output.txt. На рис. 3.9 наведено початковий відкритий текст.

Рисунок 3.9 – Початковий текст розшифрованого шифротексту з ключем

Аналіз одержаних результатів показує, що після проведеної операції шифрування відкритого тексту шифром Цезаря з ключем (рис. 3.7) та операції розшифрування зашифрованого тексту, одержано початковий текст, представлений на рис. 3.9, тобто початковий текст розшифровано правильно.

Схожість

Схожість Цитати Посилання Вилучений текст Підміна символів Коментарі

Джерела на цій сторінці: 1, 3

Назва документа: Охотов_Богдан_ОК_41 ID файлу: 1015051607

Схожість

1 **Студентська робота ID файлу: 1014968858 Навчальний заклад: Lviv Polytechnic National University** 5.75%

36 Джерело

2 **Студентська робота ID файлу: 1000041791 Навчальний заклад: Izmail State University of Humanities** 4.57%

2 Джерело

3 **Студентська робота ID файлу: 1000097208 Навчальний заклад: Lviv Polytechnic National University** 3.62%

44 Джерело

4 **Студентська робота ID файлу: 1000783257 Навчальний заклад: National Technical University of Ukraine "Kyi...** 3.33%

39 Джерело

5 **Студентська робота ID файлу: 1000042884 Навчальний заклад: National University of Life and Environmenta...** 2.15%

39 Джерело

6 **Студентська робота ID файлу: 1000048061 Навчальний заклад: National Technical University of Ukraine "Kyi...** 1.97%

18 Джерело

7 **Студентська робота ID файлу: 1000045457 Навчальний заклад: National Technical University of Ukraine "Kyi...** 1.82%

7 Джерело

8 **Студентська робота ID файлу: 1008333086 Навчальний заклад: National Aviation University** 1.74%

2 Джерело

9 **Студентська робота ID файлу: 1000941620 Навчальний заклад: National Aviation University** 1.24% 10 **Студентська робота ID файлу: 1009715634 Навчальний заклад: National University of Water Management an** ... 0.7% 11 **Студентська робота ID файлу: 1008165189 Навчальний заклад: National University Ostroh Academy** 0.45% 12 **Студентська робота ID файлу: 1006761187 Навчальний заклад: National Aviation University** 0.44% 13 **Студентська робота ID файлу: 6001142 Навчальний заклад: National University of Water Management and N** ... 0.34% 14 **Студентська робота ID файлу: 1003109124 Навчальний заклад: Taras Shevchenko National University of Kyiv** 0.34%

4 Джерело

15 **Студентська робота ID файлу: 2042714 Навчальний заклад: Lviv Polytechnic National University** 0.32% 16 **Студентська робота ID файлу: 1013096738 Навчальний заклад: National University of Water Management an** ... 0.26% 17 **Студентська робота ID файлу: 1000760756 Навчальний заклад: National Technical University of Ukraine "Kyi** ... 0.25% 18 **Студентська**

робота ID файлу: 1014968874 Навчальний заклад: Lviv Polytechnic National University 0.25%

5 Джерело
19 Студентська робота ID файлу: 1008261378 Навчальний заклад: National Aviation University 0.15%

31 Джерело
20 Студентська робота ID файлу: 1004270445 Навчальний заклад: Poltava National Technical Yuri Kondratyuk U... 0.15%

Назва документа: Охотов_Богдан_ОК_41 ID файлу: 1015051607

21 Студентська робота ID файлу: 1014867823 Навчальний заклад: Lviv Polytechnic National University 0.13% 22 Студентська робота ID файлу: 1005943301 Навчальний заклад: Cherkasy State Technological Universit 6 Джерело 0.12%