Plag

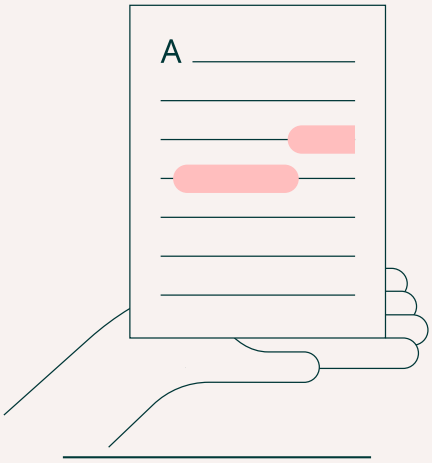
Звіт про оригінальність

● Оцінка схожості %38

● Ризик плагіату НАЙВИЩИЙ

Olga Kagalo 2025-06-19 22:55

Посилання на звіт: 10mC5 / Посилання користувача: qEAc



Ось вона –
Ваша звіт про оригінальність!

Ми раді повідомити, що перевірка вашого документа завершена, і результати вже готові! Наші алгоритми старанно працювали, щоб знайти збіги в наших базах даних.

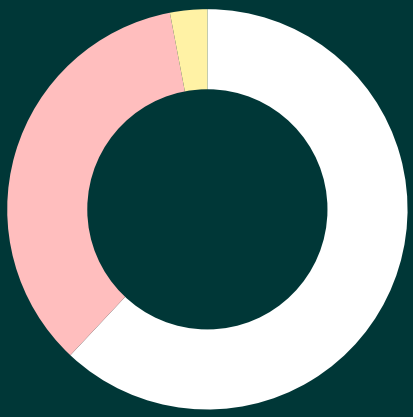
На наступних сторінках ви знайдете результати перевірки:

Бали
Збіги
Посилання

Ваш документ було перевірено за такими джерелами:

- ✓ База даних інтернет-джерел
- ✓ База даних наукових статей
- ○ Глибока перевірка (наш вдосконалений алгоритм)

Бали



Збіги тексту	35%
Перефразування	3%
Цитований текст	0%
Неправильне цитування	0%
Збігів не знайдено	62%

Ризик плагіату НАЙВИЩИЙ

Ризик плагіату вказує, як збіги тексту розподілені по документу. Вищий ризик виникає, коли збіги з'являються близько один до одного, наприклад, у тому самому абзаці або розділі.

Оцінка схожості 38%

Оцінка схожості показує, скільки слів або символів у вашому документі збігаються з текстами інших документів, включаючи перефразовані тексти або неправильні цитати.

Збіги

1 ОСНОВНІ АСПЕКТИ ЗАХИСТУ ІНФОРМАЦІЇ

1.1 Актуальність проблеми інформаційної безпеки

Актуальність вивчення 8 аспектів інформаційної безпеки пов’язана із входженням України в глобальні процеси, в яких постійно зростає значення інформації та 8 її перетворення 8 на найцінніший товар і продукт. В інформаційному суспільстві інформаційний вплив на суспільство 8 та громадянина є надзвичайно важливим. 8 Значення інформації зростає в міру зникнення національних кордонів між державами. Але суспільство повинно також турбувати проблема інформаційного перенасичення, 8 недостовірної та шкідливої інформації, загрози національній безпеці держави через інформаційну агресію іноземних держав. [12]

17 Інформаційна безпека – це захист життєво важливих інтересів суспільства, держави і особи від 17 заподіяння шкоди 17 через негативні наслідки функціонування інформаційних технологій або внаслідок розповсюдження інформації, забороненої для розповсюдження законами України. Основними характеристиками інформаційної безпеки є [20]:

Балансування на стику національної безпеки та інформаційної функції держави;

Не замикання на національних кордонах;

Протистояння між бажанням держави засекретити як найбільший масив інформації і невід’ємним правом громадянина мати вільний доступ до неї;

Державне 1 регулювання інформаційної сфери на правовій основі.

Інформаційна безпека має важливе значення для функціонування суспільства. Невід’ємною частиною загальнолюдських прав є інформаційні стосунки між особою, державою та суспільством [4].

Програмні засоби призначені для перетворення відкритих текстів до незрозумілого вигляду, шляхом розробки відповідного програмного забезпечення. Зміна вигляду

відкритого тексту для захования його змісту називається шифруванням. Відкритим текстом може бути текстовий файл або бітове зображення. Зашифроване повідомлення називається шифротекстом. Операція перетворення зашифровано тексту у початковий називається дешифруванням або розшифруванням. Шифруванням і дешифруванням текстів займається криптографія Розшифруванням шифротекстів називається криптоаналізом. Галузь, що охоплює криптографію і криптоаналіз, називається криптологією. а люди, які нею займаються, називаються криптологами.

1.2 Правовий 1 захист інформації в комп'ютерних 1 системах

1 Широке впровадження інформаційних технологій у сфері державної діяльності, економіки, фінансів зумовило підвищення вимог до забезпечення безпеки інформації. Особливо гостро це питання виникло з появою комп'ютерної техніки та автоматизованих систем опрацювання інформації. 1 Карний Кодекс України здійснює 1 правову 1 охорону інформації в автоматизованих 1 систем, а саме:

1 умисне втручання в роботу автоматизованих 1 систем, що приводить 1 до перекручення чи знищення інформації;

1 поширення програмних і технічних засобів, призначених для незаконного проникнення в автоматизовані системи, здатних до перекручення або знищення інформації [7].

Комп'ютерні системи здійснюють 20 автоматизовану обробку даних. До 20 її складу 20 входять технічні засоби обробки, а також методи, процедури та програмне забезпечення. 1 Захист інформації в автоматизованих системах - це сукупність організаційно-технічних заходів і правових норм для запобігання шкоди інтересам власників даних. 1 Право власності на інформацію встановлюється з урахуванням норм авторського права на підставі угоди між власником вхідної інформації і користувачем автоматизованих 1 систем. 1 Користувач може обробляти інформацію лише за згодою власника.

1 Власник 1 автоматизованих систем 1 повинен забезпечити захист інформації згідно угоди з власником інформації. Захист 1 інформації здійснюється шляхом:

1 дотримання суб'єктами правових відносин, норм, вимог та правил організаційно-технічного характеру щодо захисту інформації;

1 перевірки відповідності засобів автоматизованих 1 систем встановленим вимогам захисту інформації.

1 Інформація, що є власністю держави, повинна оброблятися в автоматизованих 1 систем, 1 що має відповідний сертифікат. Закон встановлює відповідальність за

порушення порядку і правил захисту 1 інформації, 1 механізм відшкодування нанесеної шкоди, гарантує забезпечення інформаційних прав.

1 Особливість 1 регулювання інформаційних відносин в Інтернеті визначається особливістю фізичного представлення інформації в мережі в електронному вигляді. При передачі інформації відсутній носій інформації, що ускладнює оформлення і представлення документованої інформації у віртуальному середовищі. Виникає проблема закріплення і захисту правового режиму електронного документа, який би гарантував достовірність. Тому актуальною є проблема електронного підпису. Основними об'єктами, з приводу яких виникають інформаційні відносини в Інтернеті, є:

1 програмно-технічні комплекси, інформаційні системи;

1 інформація, інформаційні ресурси;

1 інформаційні права та свободи;

1 інтереси особистості, суспільства та держави;

1 інформаційний суверенітет держави;

1 інформаційна 1 безпека.

1 З поширенням електронних документів виникають 1 такі проблеми:

1 визначення поняття "електронний документ";

1 підтвердження юридичної сили електронного документа;

1 встановлення факту і дати введення документа в мережу;

1 ідентифікація змісту електронного документа з його власником;

1 доведення права авторства електронного документа.

1 З позицій інформаційної безпеки Інтернет може використовуватися зі злочинною метою. Тому 1 інформаційна безпека 1 в Інтернеті спрямована на захист:

1 національної безпеки;

людської гідності, репутації, прав неповнолітніх;

1 інформації (несанкціонований доступ);

1 таємниці особистого життя;

1 інтелектуальної власності (незаконне поширення творів, програмного забезпечення, музики тощо).

1 1.3 Засоби захисту інформації

Засоби захисту інформації діляться на технічні та програмні. 9 Вся сукупність технічних засобів ділиться 9 на фізичні й 9 апаратні. Фізичні засоби реалізуються у виді автономних пристроїв і систем і виконують функції загального захисту об'єктів, на яких опрацьовується інформація. До них відносяться пристрої захисту територій і будинків, де розміщена апаратура, грати на вікнах, електронно-механічне устаткування охоронної сигналізації. До апаратних технічних засобів відносяться 9 пристрої, що вбудовуються в обчислювальну техніку та 9 телекомунікаційну апаратуру.

3 Програмні засоби являють собою програмне забезпечення, призначене для виконання функцій захисту інформації. За допомогою програмних засобів дані перетворюються в незрозумілу форму для її передачі по каналах зв’язку.

3 Організаційні засоби захисту передбачають організаційно-технічні й організаційно-правові заходи, здійснювані в процесі створення й експлуатації апаратури телекомунікацій для забезпечення захисту інформації. Вони охоплюють усі структурні елементи на всіх етапах їх життєвого циклу (будівництво помешкань, проектування системи, монтаж і наладка устаткування).

3 Морально-етичні засоби захисту реалізуються у вигляді норм, що склалися традиційно в даній країні. Ці 3 норми зазвичай 3 не є обов'язковими, як законодавчі міри, але 3 їх недотримання веде 3 до втрати авторитету і престижу співробітника.

3 Законодавчі засоби захисту визначаються законодавчими актами країни, які 3 регламентують 3 правила використання, опрацювання і передачі інформації обмеженого доступу і встановлюють міри відповідальності за порушення цих правил. На рис. 1.1 наведена класифікація засобів захисту інформації.

Рисунок 1.1 – Класифікація засобів захисту інформації

3 Необхідно також відзначити, що всі розглянуті засоби захисту діляться на формальні, що виконують захисні функції строго по заздалегідь передбаченій процедурі без особистої участі людини, і неформальні, обумовлені цілеспрямованою діяльністю людини або регламентуючої цієї діяльності.

1 Технічний захист інформації - це сукупність організаційних структур, поєднаних цілями захисту інформації, нормативно-правової та матеріально-технічної бази і спрямована на забезпечення інженерно-технічними засобами конфіденційності, цілісності та доступності інформації. [4].

1 Технічний захист інформації спрямований на забезпечення інженерно-технічними засобами порядку доступу до інформації, яка становить державну та іншу таємницю. Витік інформації, яка становить державну 1 або конфіденційну таємницю є загрозою 1 національній безпеці України в інформаційній сфері.

1 Загрози інформаційній безпеці зумовлені наступними 1 факторами:

1 неефективністю державної політики в галузі інформаційних технологій;

1 діяльністю іноземних держав;

1 діяльністю політичних партій та окремих осіб у політичній боротьбі та конкуренції;

1 злочинною діяльністю, спрямованою на протизаконне отримання інформації;

1 Напрямки державної політики у 1 сфері 1 технічного захисту інформації:

1 нормативно-правове 1 забезпечення;

1 розробка нормативних актів захисту важливої відкритої інформації;

1 організація 1 забезпечення 1 технічного захисту інформації;

1 контроль 1 за імпортом технологій 1 технічного захисту інформації;

1 підготовка 1 кадрів у галузі 1 технічного захисту інформації;

1 розвиток 1 міжнародної співпраці у сфері 1 технічного захисту інформації.

1 1.4 Криптографічний захист інформації

2 Основу забезпечення інформаційної безпеки в інформаційно-телекомунікаційних системах складають криптографічні методи і засоби захисту інформації. Історично криптографія використовується 2 з метою збереження секретних даних. 2 Основними задачами, які вирішує криптографія, є забезпечення

2 конфіденційності, цілісності, достовірності, юридичної значущості та оперативності доступу до інформації.

2 Криптографія 1 вивчає методи, прийоми і системи шифрування та дешифрування текстових 1 повідомлень. Криптографічні проблеми 1 вимагають глибоких знань з історії, лінгвістики, філології, інформатики, психології 1 та 1 математики. Криптографічні методи мають давню історію, їх застосовували в Давньому Єгипті, Давній Греції та Римі, Київській Русі. Криптографія знаходила застосування в зовнішній

політиці, військовій справі, релігійних підпільних рухах, кримінальному та антиурядовому середовищі. Наприклад, кілька систем ручного шифрування винайшли декабристи: метод транспозиції (переміщення літер і слів у тексті, що мало наслідком зміну їх порядку при написанні), метод заміни літер тексту літерами іншого алфавіту [24].

2 Навіть сама писемність була свого роду шифруванням . У стародавньому Китаї тільки вищі шари суспільства могли навчатися читанню і листуванню, а перший досвід застосування криптографії в Єгипті відноситься до 1900 року до н. е. Автор напису користувався незвичайними ієрогліфами. Є і інші приклади: дощечки з Месопотамії, на яких зашифрована формула виготовлення керамічної глазурі (1500 рік до н. э.), єврейський шифр ATBASH (500-600 роки до н. е.), грецький «небесний лист» (486 рік до н. е.) і шифр простої підстановки Юлія Цезаря (50-60 рік до н. е.). Кама Сутра Ватс'яяни навіть ставить мистецтво тайнопису на 44-е, а мистецтво секретної розмови на 45-е місце в списку 64 мистецтв, якими повинні володіти чоловіки і жінки [13].

Одним із ключових критеріїв при розробці програмного забезпечення, що використовує криптографію, є застосування алгоритму шифрування. В даний час існує достатня кількість алгоритмів для реалізації методів шифрування.

Конфіденційність 25 – це 2 властивість інформації бути доступною тільки обмеженій 2 групі 2 осіб. Під цілісністю розуміється властивість інформації зберігати свою структуру і зміст в процесі зберігання і передачі. Достовірність інформації полягає 2 в строгій 2 приналежності об'єкту, який є її джерелом. Здатність інформації бути доступною для кінцевого користувача відповідно до його часових 2 вимог забезпечується оперативністю. 2 Юридична значущість означає, що документ володіє юридичною силою.

2 В основі криптографічних методів лежить поняття криптографічного перетворення інформації на основі певних математичних законів з метою виключити доступ до неї 2 сторонніх користувачів. 2 Криптографічне перетворення називається алгоритмом шифрування. Процес шифрування однозначно відображає множину відкритих повідомлень в множину криптограм.

Шифри повинні володіли наступними властивостями:

2 Законний одержувач зможе виконати зворотне перетворення і однозначно розшифрувати текст, 2 знаючи криптографічний алгоритм;

2 Криптоаналітик (зловмисник), що перехопив повідомлення, не зможе відновити по ньому початкове повідомлення без часових 2 витрат 2 і засобів, які зроблять інформацію непридатною.

Для коректної передачі секретної інформації по каналах зв'язку з використанням криптографічного алгоритму сторони інформаційного обміну повинні дотримуватись певної послідовності дій, що називається криптографічним протоколом. 2 В основі криптографічного протоколу лежить шифр. Криптографічні протоколи є 2 важливою складовою частиною криптографічної системи. Через 2 наявність слабих 2 місць в протоколі можливі ситуації, коли завдання забезпечення безпеки інформації не розв'язуються.

2 Кожна 2 дія криптопротоколу 2 є або обчисленнями, що виконуються діючими суб'єктами протоколу, або розсилкою повідомлень між ними. Атаки на протоколи з боку противника можуть бути направлені як проти криптографічних алгоритмів, використовуваних в протоколах, так і проти самих протоколів.

2 При 2 пасивній атаці противник 2 обмежується спостереженням за діями сторін протоколу і намагається витягнути 2 із спостережень корисну для себе інформацію, не втручаючись в реалізацію протоколу. При активній атаці на криптографічний протокол противник 2 видозмінює протокол 2 в своїх інтересах. Це може привести до введення в протокол нових повідомлень, підміни 2 одних повідомлень іншими, видалення з протоколу реальних даних, виводу з ладу каналу зв'язку або пам'яті, 2 в якій зберігається інформація.

2 Основними 2 завданнями 2 забезпечення інформаційної безпеки за допомогою криптографічних протоколів є:

Обмін алгоритмами з подальшим захистом обміну даними;

2 Аутентифікація сторін, що встановлюють зв'язок;

2 Авторизація користувачів при доступі до телекомунікаційних і інформаційних служб.

2 1.5 Класифікація алгоритмів захисту інформації

Шифрування даних є одним з важливих рішень проблеми криптографічного захисту. 12 Зашифровані дані стають доступними тільки для того, хто знає, як їх розшифрувати, і тому викрадення зашифрованих даних пов’язане 12 з великими труднощами 12 для несанкціонованих користувачів. Шифри використовувались задовго до появи комп’ютерної техніки. При шифруванні використовуються алгоритми і ключі. 12 Алгоритм дозволяє використати порівняно короткий ключ для шифрування наскільки завгодно великого тексту [13, 12 24].

10 Криптографічний захист – це захист даних з допомогою криптографічного перетворення, під яким розуміється перетворення даних шифруванням. Шифруванням

даних називається 16 процес перетворення відкритих даних на зашифровані з 10 допомогою шифру, а розшифруванням 16 даних – процес перетворення закритих даних на 10 відкриті з допомогою шифру. Ключ – це секретний параметр 16 алгоритму криптографічного перетворення даних, що забезпечує вибір одного варіанту із множини 10 для даного алгоритму.

10 Крипостійкість 10 шифру визначається його стійкістю 10 до дешифрування. Звичайно ця характеристика 16 визначається періодом часу, необхідним для 10 дешифрування. Важливим критерієм при розробці програмного забезпечення для 10 криптографічного захисту інформації є вибір алгоритму шифрування. 18 В даний час існує велика кількість алгоритмів для реалізації методів шифрування.

У 7 криптографія з ключем алгоритми шифрування переданих даних можуть бути відомі 7 усім стороннім особам, але вони 7 ще 7 залежить від деякого параметра – "ключа", яким володіють лише відправник і одержувач повідомлення. В залежності від кількості ключів, які застосовуються в алгоритмі, криптографічні 18 алгоритми поділяються на 3 категорії:

Безключові алгоритми, що не використовують ключів при шифруванні;

З одним ключем, що використовують для шифрування та дешифрування один ключ;

З двома ключами, один для шифрування інформації, – інший для дешифрування. Зазвичай один ключ є секретним, а інший відкритим;

На рисунку 1.2 наведена 7 класифікація криптографічних алгоритмів за 7 кількістю ключів.

7 Рисунок 1.2 – 7 Класифікація криптографічних алгоритмів за кількістю ключів

В залежності від використання кількості ключів та їх типів, криптогалгоритми поділяються на симетричні та асиметричні. В симетричних алгоритмах для шифрування та дешифрування повідомлень 7 використовується один і той ключ. В асиметричних алгоритмах 18 для шифрування 7 повідомлення використовується один відкритий ключ, що відомий усім бажаючим, а для дешифрування 7 – другий 7 закритий, який існує тільки в одержувача зашифрованого шифру. На рисунку 13. наведена класифікація криптосистем за кількістю та типом використання ключів.

Рисунок 1.3 – Класифікація криптосистем за кількістю та типом ключів

Системи, в котрих для 18 шифрування і дешифрування повідомлень використовується однаковий ключ, називаються симетричними. Симетричні системи використовують переважно для перетворення тексту, який є комбінацією перестановок і замін.

Симетричні алгоритми по конструктивному принципу 18 діляться на потокові та блочні 18 шифри.

18 В потокових шифрах одиницею 18 кодування є один біт. Результат кодування не залежить від попереднього вхідного потоку. Потокова 6 схема застосовується в системах передачі потоків інформації, тобто в тих випадках, коли передача інформації починається і закінчується в довільні моменти часу. Поширеними 13 представниками потокових 6 шифрів є скремблери.

6 В 6 блочних шифрах 6 одиницею кодування є блок з декількох байтів. Результат кодування залежить від усіх байтів цього блоку. Схема застосовується при пакетній 13 передачі інформації та кодуванні 21 файлів. Блочні 11 шифри оперують з блоками відкритого тексту, на 11 які накладаються 11 наступні вимоги:

11 Достатня криптостійкість;

11 Простота процедур шифрування і 11 дешифрування;

Висока надійність.

11 В залежності від криптографічних перетворень, що здійснюються над даними, симетричні алгоритми діляться на шифри 4 заміни та перестановки.

4 При 4 використанні шифру заміни кожний 4 елемент початкового тексту взаємно-однозначно замінюється одним, або декількома знаками деякого алфавіту. Шифр простої заміни замінює кожний знак вхідного алфавіту на деякий знак з того ж алфавіту, Результат заміни 6 не залежить від розташування 4 знаку у відкритому тексті. Ключами для шифрів заміни є таблиці. В алгоритмах заміни змінюється порядок блоків інформації за законами криптосистеми. 6 Переважна більшість сучасних алгоритмів належить цій групі.

6 В 6 перестановочних алгоритмах 6 блоки інформації (байти, біти, або 6 інші 6 одиниці) не змінюються, але змінюється їх порядок проходження, що робить шифровану інформацію недоступною сторонній особі.

Шифри перестановки відрізняються 4 від шифрів заміни тим, 4 що при шифруванні буква 4 відкритого тексту переходить не у фіксований знак алфавіту, а в іншу букву того ж відкритого тексту, внаслідок чого букви розташовуються на нових місцях, тобто переставляються. Ключем для даного шифру також служить таблиця заміни, тільки не букв алфавіту, а їх індексів в тексті, який підлягає шифруванню. Розмір таблиці заміни дорівнює довжині відкритого тексту.

4 Симетричні алгоритми характеризуються можливістю швидкого шифрування

великих потоків інформації в каналах зв’язку. Вони забезпечують високу ступінь секретності. Симетричні алгоритми дають змогу використовувати одні і ті ж апаратні засоби для шифрування і дешифрування інформації [13, 22].

Схематично процес шифрування і дешифрування інформації показано на рис. 1.4.

Рисунок 1.4 – Схематичне представлення процесу шифрування і дешифрування інформації

На передавальній стороні виконується шифрування відкритого повідомлення за допомогою алгоритму шифрування з використанням ключів. В результаті отримуємо криптограму, яка передається відкритим каналом зв’язку. На приймальній стороні до отриманого зашифрованого повідомлення розшифровується. Розшифрування буде вірним, якщо криптограма не була змінена під час передачі по каналу зв’язку.

1.6 Основні вимоги до алгоритмів шифрування

Процес криптографічного захисту даних може здійснюватися як програмно, так і апаратно. Апаратна реалізація відрізняється істотно більшою вартістю. Але їй притаманні і висока продуктивність, простота, захищеність і т.д. Програмна реалізація є практичнішою, допускає гнучкість у використанні [24]. Для сучасних криптографічних систем захисту інформації ставляться наступні вимоги [5]:

зашифроване повідомлення повинно піддаватися читанню тільки при наявності ключа;

число операцій для визначення ключа шифрування за фрагментом шифрованого повідомлення і відповідного йому відкритого тексту, має бути не менше загального числа можливих ключів;

число операцій, необхідних для дешифрування інформації шляхом перебору різноманітних ключів, повинно виходити за межі можливостей сучасних комп’ютерів;

знання алгоритму шифрування не впливають на надійність захисту;

незначна зміна ключа повинна приводити до істотної зміни виду зашифрованого повідомлення;

незначна зміна вихідного тексту повинно приводити до істотної зміни виду зашифрованого повідомлення навіть при використанні одного і того ж ключа;

5 структурні елементи алгоритму шифрування повинні бути незмінними;

5 додаткові 15 дані, 5 що вводяться в повідомлення в процесі шифрування, повинен 24 бути повністю та 5 надійно сховані в зашифрованому 24 тексті;

5 не повинно бути простих і легко встановлюваних залежностей між ключами, що 5 послідовно використовуються 5 в процесі шифрування;

5 будь-який 15 ключ із 5 множини можливих повинен забезпечувати надійний захист інформації.

5 Процес 5 обміну інформацією здійснюється таким способом:

14 одержувач обчислює відкритий і секретний ключі, секретний ключ зберігає в таємниці;

14 відправник, 14 відкритим ключем 14 одержувача, зашифровує сеансовий ключ, який пересилається одержувачу по незахищеному каналу;

14 одержувач отримує сеансовий ключ і розшифровує його, використовуючи свій секретний ключ.

14 2 КРИПТОГРАФІЧНА СИСТЕМА ЗАХИСТУ 21 ДАНИХ НА ОСНОВІ МАТРИЧНОГО ШИФРУ ПЕРЕСТАНОВОК

2.1 Класифікація шифрів перестановок

Основна ідея шифру перестановки полягає в перестановці символів в початковому тексті так, щоб без знання правил цієї перестановки, неможливо прочитати шифротекст. Є різні способи організації шифру перестановок:

Числова перестановка;

Матрична перестановка;

Блочна перестановка шифрування.

Будь-який спосіб задання ключа можна представити в вигляді перестановки. Тому кількість можливих ключів для тексту довжини n рівно n!.

При шифрі перестановок початковий текст 4 розбивається на блоки довжиною k, де k – довжина ключа. Після цього букви в кожному блоці переставляються згідно перестановці символів ключа. Якщо в останньому блоці не вистачає букв, то додаються випадкові, щоб довжина останнього блоку була рівна довжині ключа перестановки.

В шифрах перестановок змінюється порядок блоків інформації, що робить шифровану інформацію недоступною сторонній особі. Шифри діляться на:

Шифр частоколу;

Матричний шифр перестановок;

Шифр Першої світової війни (ADFGVX-шифр).

Одним із найпростіших 19 шифрів перестановки є шифр частоколу. Він дуже схожий на матричний шифр. Він характеризується висотою частоколу. Наприклад, шифрування слова “студент” шифром 19 із висотою частоколу 2. Для цього запишемо його так:

т

д

н

с

у

е

т

Далі зчитуємо спочатку верхній рядок «тдн», 19 а потім нижній «сует». В результаті одержимо зашифроване повідомлення «тднсует». Для частоколу висотою 3 отримаємо такий зашифрований текст:

у

н

т

е

с

д

т

19 Тепер зчитуємо спочатку верхній рядок «ун», 19 потім – другий «те», 19 а потім

нижній “сдт”. В результаті одержимо зашифроване повідомлення «унтесдт».

2.2. Опис матричного шифру перестановок

Існує багато криптографічних алгоритмів та протоколів, що використовуються 21 для захисту інформації [23], але більшість із них орієнтовані на послідовну обробку скалярних даних. Але в локальних та глобальних мережах, системах зв'язку та телекомунікацій поширені при передачі двовимірні масиви та зображення. Так як при послідовній обробці та шифруванні скалярних даних використовуються одні і ті ключі, то послідовні алгоритми є не дуже стійкими до криптоаналізу. Тому актуальною є проблема модифікації відомих алгоритмів та протоколів для криптошифрувань на матричні, коли дані, що шифруються і дешифруються, представляються у вигляді багатовимірних, наприклад, матричних масивів.

Зашифруємо цей текст на ключових словах «слово» та «група». Ключове слово «слово» записуємо згори від матриці тексту, а ключове слово «група»—зліва від матриці тексту. Далі переставляємо рядки згідно з позицією кожної літери слова «група» у алфавіті (а, г, п, р, у). Матриця з відкритим текстом «матричний шифр перестановок» перетвориться в матрицю, показану на рис. 2.1.

Рисунок 2. 1 – Переставляння рядків матриці згідно з позицією літер кдючового слова у алфавіті

Далі переставляємо стовпчики згідно з позицією кожної літери слово «слово» у алвавіті (в, л, о, о, с). Матриця на рис. 2.1 (справа) перетвориться в матрицю, показану на рис. 2.2.

Рисунок 2.2 – Переставляння стовпчиків матриці згідно з позицією літер ключового слова у алфавіті

В результаті проведених перестановок рядків і стовпчиків одержано матрицю, показану на рис. 2.2. Прочитавши інформацію з матриці по стовпцях, отримаємо шифротекст:

«оовкнратимтесарйнишчпфреи».

Розшифрувати шифтекст можна, записавши слова в матрицю по стовпчиках згідно з порядком розташування літер ключів по абетці, а потім, переставивши стовпчики та рядки так, щоби ключі утворили зв'язані слова, зчитуємо розшифрований текст по рядках [22].

Для дешифрування шифротекстів пари літер замінюємо літерою, яка стоїть на перетині відповідного рядка та стовпця побудованої матриці. Аналогічно можна створити схожі

шифри для української мови та інших мов.

2.3 Опис алгоритму блочної перестановки шифрування та дешифрування

При використанні алгоритму блочної перестановки відкритий текст поділяється на блоки. Довжина блоків рівна довжині ключового слова або фрази. Якщо останній блок відкритого тексту коротший від ключового слова, то при необхідності він доповнюється довільними символами до розміру ключа.

Далі записуються номери букв в ключовому слові по зростанню їх появи в алфавіті. Послідовність номерів записуються під кожним блоком. Потім кожна літера блоку записується в порядку номерів літер ключа. Шифрування полягає в записуванні символів в блок на нові позиції. Відповідно, переставляються символи в кожному блоці, на які розбивається початкове повідомлення.

Для шифрування використовується алфавіт з українських букв і символу пропуску. В ролі ключа вибрано слово «диплом». Записуємо номери букв в ключовому слові по зростанню їх появи в алфавіті.

д

и

п

л

о

м

1

2

6

3

5

4

Текст для шифрування відкритого повідомлення «матричний шифр перестановок» розіб’ємо на блоки по 6 символів, так як довжина ключового слова рівна шести.

Останній блок доповнюємо символами «а» (три символи). Пронумеруємо символи в блоках. Під кожним блоком підпишемо нумерацію символів ключа. Процес шифрування блочною перестановкою показано на рис. 2.3.

Рисунок 2.3 – Процес шифрування блочною перестановкою

Для початкового тексту «матричний шифр перестановок» шифротекст матиме вигляд:

«марчитни ишйфрпре есаонтвоааак»

Порядок 4 шифрування складається з наступних кроків:

Відкритий текст доповнюється будь-якими символами, так щоб його довжина стала кратною довжині ключового слова або фрази;

Символи використовуваного ключа нумеруються згідно порядку їх появи в алфавіті;

Номери ключа використовуються для шифрування тексту.

Для дешифрування символи з блоку шифрограми виписуються згідно ключу “диплом”. На рис. 2.4 показано обернений процес дешифрування.

Рисунок 2.4 – Процес дешифрування блочною перестановко

3 РОЗРОБКА АЛГОРИТМІВ І ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ШИФРУВАННЯ ТА ДЕШИФРУВАННЯ ТЕКСТІВ НА ОСНОВІ МАТРИЧНОГО ШИФРУ ПЕРЕСТАНОВОК

3.1 Опис системи шифрування текстів на основі матричного шифру перестановок

При використанні матричного шифру перестановок для захисту даних відкритий текст 4 представляється у вигляді прямокутної матриці. Метод шифрування полягає у послідовному записуванні символів відкритого тексту рядок за рядком у матрицю. Для одержання шифротексту символи виписуються з цієї матриці по стовпчиках.

Для підсилення крипостійкості матричного алгоритму при шифруванні відкритого повідомлення використовують ключі. В ролі ключа може бути ключове слово або фраза. Довжина ключа рівна кількості символів у рядку матриці. Якщо довжина повідомлення не кратна довжині ключа, то повідомлення доповнюється будь-якими символами. Процес шифрування здійснюється за допомогою перестановки символів в ключовій фразі.

4 В першому рядку матриці записуються літери ключового слова. Далі послідовно записуються символи відкритого тексту рядок за рядком у матрицю. Після чого стовпці матриці переставляємо згідно номера позиції кожної літери ключа в алфавіті.

Результат шифрування зчитуємо з матриці по стовпцях. Така перестановка суттєво ускладнює процес дешифрування. Але це не означає, що цей шифр не можна дешифрувати за допомогою аналізу частот появи різних літер та їх блоків у зашифрованому тексті [22].

Процес шифрування повідомлень на основі матричного шифру перестановок складається з наступних кроків:

Читання з файлу ключового слова;

Запис номерів літер ключазгідно номерів позиції їх в алфавіті;

Читання з файлу 4 відкритого тексту і запис його в рядки матриці блоками, довжина яких рівна довжині ключа;

Стовпці матриці переставляються згідно номерів літер ключа.

4 Нехай треба зашифрувати повідомлення «КОМПЮТЕРНА МОДЕЛЬ» за допомогою ключового слова «КОМА». Довжина даного повідомлення має 17 символів і не кратна довжині ключа. Тому повідомлення доповнюємо до 20-ти символів, наприклад, трьома символами 'А'. Далі записуємо номери літер в ключовому слові по зростанню їх появи в алфавіті. На рис. 3.1 показано систему шифрування тексту «КОМПЮТЕРНА МОДЕЛЬ» на основі матричного шифр

Рисунок 3.1 − Шифрування тексту «КОМПЮТЕРНА МОДЕЛЬ» на основі матричного шифру

Шифротекст зчитуємо з матриці по стовпцях. Тому після зашифрування повідомлення "КОМПЮТЕРНА МОДЕЛЬ", отримаємо таку криптограму:

«ПРМЛАКЮНОЬМЕ ЕАОТАДА»

4 Нехай треба зашифрувати повідомлення

11 «ІНФОРМАЦІЙНА БЕЗПЕКА 11 МАЄ ВАЖЛИВЕ ЗНАЧЕННЯ ДЛЯ ФУНКЦІОНУВАННЯ СУСПІЛЬСТВА»

за допомогою ключового слова "ДИПЛОМ". На рис. 3.2 показано криптографічну систему шифрування повідомлення на основі матричного шифру.

Рисунок 3.2 − Криптографічна система шифрування повідомлення на основі матричного шифру перестановок

Після шифрування повідомлення отримаємо наступну криптограму:

«ІА К ВЧДНУ ЛНЦБАВЕЕЛКВСЬОЙЗМЖЗН ІНСТМАЕЄИА УНЯІАРНПАЛНЯФОНПВФІЕ А НЯЦАУС»

3.2 Опис алгоритму для шифрування текстів на основі матричного шифру перестановок

Розробити алгоритм для шифрування заданого тексту з використанням матричного методу і реалізувати його у вигляді програмного забезпечення. Такі алгоритми є стійкими до впливу завад та різних спотворень, що виникають при передачі документів по каналах зв’язку. На рис. 3.3 зображено структурну схему алгоритму шифрування текстів на основі матричного шифру перестановок.

Рисунок 3.3 − Структурна схема алгоритму шифрування текстів на основі матричного шифру перестановок

При використанні матричного шифру літерам алфавіту присвоюються числові коди. В табл. 3.1 наведено відповідність між числовими кодами і літерами алфавіту, вибраного для шифрування тексту.

Таблиця 3.1 – Відповідність між числовими кодами і літерами

Код

0

13 1

13 2

13 3

13 4

13 5

13 6

13 7

13 8

13 9

13 10

13 11

13 12

13 13

13 14

13 15

13 Літера

А

Б

В

Г

Д

Е

Є

Ж

З

И

І

Ї

Й

К

Л

М

Код

13 16

13 17

13 18

13 19

13 20

13 21

13 22

13 23

13 24

13 25

13 26

13 27

13 28

13 29

13 30

13 31

13 Літера

Н

О

П

Р

С

Т

У

Ф

Х

Ц

Ч

Ш

Щ

Ь

Ю

Я

Алгоритм шифрування матричним шифром виконує наступні функції:

Читання ключа шифрування з файлу file_key.txt;

Кодування літер ключового слова на основі табл. 3.1;

Читання тексту для шифрування з файлу file_input_text.txt;

Шифрування 4 тексту за допомогою матричного шифру з ключем, який задається відправником повідомлення і тримається в секреті;

Запис зашифрованої інформації в формі матриці в файл file_matr.txt та вивід її на екран;

Перетворення зашифрованої інформації з матриці в текст і запис його в файл file_shyf_text.txt.

3.3 Опис програми для шифрування текстів матричним шифром

11 На основі алгоритму розроблено програмне забезпечення. При розробці програмного забезпечення було використано принцип розділення його на окремі функціональні компоненти. В процесі роботи програми ці компоненти взаємодіють між собою. Таким підхід в більшості випадків використовується для розробки багатофункціонального програмного забезпечення.

Проблема економії часу при розробці великих проектів є одною з актуальних задач, з якою зустрічаються фахівці при шифруванні та розшифруванні повідомлень. Мову програмування С часто використовують при програмній реалізації складних та великих за обсягом проектів.

Описана в дипломному проєкті програма демонструє основні прийоми роботи з шифруванням повідомлень на основі матричного шифру. Шифрування проводиться з

ключем, який задається словом. Відкритий текст для шифрування розміщений в текстовому файлі file_key.txt в форматі символьного масиву.

Програма повинна виконувати наступні функції:

Читати задані тексти з файлів.

Читати ключ з файлу.

Зашифровувати текст матричним шифром із заданим ключем.

Записувати шифротексти в файли.

Виводити на екран одержані шифротексти.

Читати з файлу зашифровану інформацію.

Програма складається з таких елементів:

1 Підключення бібліотечних файлів, які містять прототипи стандартних функцій файлового вводу-виводу, функцій обробки символьної інформації:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

2 Опис вказівників на змінні структурного типу FILE, які асоціюють фізичні файли з потоками вводу-виводу:

FILE *fp1, *fp2, *fp3, *fp4;

Потік fp1 зв’язаний з файлом, що містить відкритий текст, fp2 – з файлом для зберігання шифротексту. Потік fp3 асоціюється з файлом для зберігання ключа, а потік fp4 зв’язаний з файлом для запису проміжних матриць.

3 Задання імен використовуваних файлів:

char filename130]=" file_input_text.txt"; /* fp1 */

char filename230]=" \file_shyf_text.txt"; /* fp2 */

char filename330]=" file_key.txt"; /* fp3 */

char filename430]=" file_matr.txt"; /* fp4 */

4 Відкриття файлу file_input_text.txt для читання відкритого тексту:

fp1=fopen(filename1,mode_r);

if (fp1!=NULL) {printf("file %s open mode %s\n",filename1, mode_r); }

else { printf("file %s not open mode %s\n", filename1, mode_r); exit(1);}

5 Відкриття файлу file_shyf_text.txt" для запису шифротексту:

fp2=fopen(filename2,mode_w);

if (fp2!=NULL) {printf("file %s open mode %s\n",filename2, mode_w); }

else {printf("file %s not open mode 3 %s\n",filename2,mode_w); exit(2);}

6 Відкриття файлу file_key.txt, для читання ключа:

fp3=fopen(filename3,mode_r);

if (fp3!=NULL){printf("file %s open mode %s\n",filename3, mode_r);}

else { printf("file %s not open mode %s\n",filename3, mode_r); exit(3);}

7 Відкриття файлу file_matr.txt для запису матриці тексту.

fp4=fopen(filename4,mode_w);

if (fp4!=NULL){printf("file %s open mode %s\n",filename4, mode_w);}

else {printf("file %s not open mode %s\n",filename4, mode_w); exit(4);}

8 Задання літер алфавіту для написання тексту наведено в таблиці 3.2.

Таблиця 3.2 − Задання літер алфавіту для написання відкритого тексту

KOD0]='А';

KOD1]='Б';

KOD2]='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';

9 Читання літер ключа та визначення його довжини:

i=0; char_keyi]=fgetc(fp3);

while(char_keyi]!=EOF) { i++; char_key[i]=fgetc(fp3);}

l_key=strlen(char_key)-1; /* Довжина ключа */

10 Кодування літер ключового слова та формування динамічних масивів key_MAS та key_pos для зберігання числових кодів літер ключа та їх номерів в алфавіті. Оскільки розмірність цих масивів залежать від довжини ключа і наперед невідома, то оперативна пам’ять для них виділяється динамічно за допомогою стандартної функції calloc():

key_MAS=(int *)calloc(l_key, sizeof(int));

key_pos=(int *)calloc(l_key, sizeof(int));

for (i=0; i<l_key; i++)

{key_MAS[i]=kodyvannja (char_key[i]); }

11 Читання тексту повідомлення з файлу c:\k\file_input_text.txt в оперативну пам’ять в масив text_mas та визначення його довжини:

i=0; text_mas[i]=fgetc(fp1);

while(text_mas[i]!=EOF)

{ i++; text_mas[i]=fgetc(fp1);}

l_text=strlen(text_mas)-5; /* Довжина тексту */

12 Визначення кількості рядків kr_m матриці text_matr запис у неї прочитаного з файлу тексту:

if (l_text%l_key==0) {kr_m= l_text/l_key;}

else {kr_m= l_text/l_key+1;}

k=0; for (i=0; i<kr_m; i++)

for (j=0; j<l_key; j++)

{text_matr[i][j]= text_mas[k];k++ ;}

13 Запис матриці в файл c:\k\file_matr.txt:

for (i=0; i<kr_m; i++)

{for (j=0; j<l_key; j++)

fprintf(fp4,"%c \t",text_matr[i][j]);

fprintf(fp4,"\n");}

14 Шифрування тексту і запис шифроматриці в файл c:\k\file_matr.txt :

for (i=0; i<kr_m; i++)

for (j=0; j<l_key; j++)

text_matr_shyf[i][key_pos[j]-1]=text_matr[i][j];

for (i=0; i<kr_m; i++)

{for (j=0; j<l_key; j++)

fprintf(fp4,"%c\t",text_matr_shyf[i][j]); fprintf(fp4,"\n");}

15 Перетворення шифроматриці в текст і запис його у файл c:\k\file_shyf_text.txt:

k=0; for (j=0; j<l_key; j++) for (i=0; i<kr_m; i++)

{text_mas_shyf[k]=text_matr_shyf[i][j];

fprintf(fp2,"%c",text_mas_shyf[k]);k++ ;}

16 Закриття файлів:

fclose(fp1); fclose(fp2);

fclose(fp3); fclose(fp4);

Присвоєння числового значення прочитаній літері тексту здійснює функція kodyvannja (), прототип якої має вигляд:

int kodyvannja(char cum);

Аргументом функції є прочитаний символ, код якого повертає функція.

Результатом роботи програми є створення файлу c:\k\file_shyf_text.txt зашифрованого повідомлення. Повні тексти програми і функції kodyvannja() наведено в додатку А.

3.4 Результати роботи програми шифрування текстів

Вхідними даними для програми є текст для шифрування, який знаходиться у файлі file_input_text.txt та ключ, який знаходиться у файлі file_key.txt. На рис. 3.4 показано текст для шифрування (файл file_input_text.txt).

Рисунок 3.4 – Текст для шифрування (файл file_input_text.txt)

На рис. 3.5 показано файл file_key.txt ключа для шифрування.

Рисунок 3.5 – Файл file_key.txt ключа для шифрування

Відкритий текст записується у вигляді матриці по рядках у файл file_matr.txt. На рис. 3.6 показано вигляд матриці відкритого тексту.

Рисунок 3.6 – Вигляд матриці відкритого тексту у файлі file_matr.txt

Відкритий текст шифрується за допомогою матричного шифру з ключем «КОЛЕДЖ» і записується у вигляді шифрованої матриці у файл file_deshyf_matr.txt. На рис. 3.7 показано вигляд шифрованої матриці.

Рисунок 3.7 – Вигляд матриці шифротексту

Результатом роботи програми є зашифрований текст, записаний у файл file_shyf_text.txt по стовпцях. На рис. 3.8 показано файл зашифрованого тексту.

Рисунок 3.8 – Файл зашифрованого тексту

3.5 Опис алгоритму для дешифрування шифротекстів

На рис. 3.9 зображено структурну схему алгоритму дешифрування повідомлень на основі матричного шифру перестановок.

Рисунок 3.9 − Структурна схема алгоритму дешифрування текстів на основі матричного шифру перестановок

Алгоритм розшифрування текстів, зашифрованих за допомогою матричного шифру з використанням ключа, складається з таких кроків:

Читання з файлу ключового слова;

Запис номерів літер ключа згідно номерів позиції їх в алфавіті;

Читання з файлу шифротексту і запис його в стовпці матриці блоками, довжина яких рівна результату від ділення довжини шифротексту на довжину ключа;

Стовпці матриці переставляються згідно номерів літер ключа.

Одержувач повідомлення для розшифрування тексту повинен знати ключ. Записати розшифровану інформацію в новий файл. Розшифрування здійснюється за певними правилами підстановок і ключового слова. При цьому кожний символ шифротексту замінюється символом початкового тексту.

3.6 Опис програми для дешифрування шифротекстів

В 23 дипломному проєкті розроблено програмне забезпечення для дешифрування зашифрованих повідомлень матричним шифром. Програма демонструє процес розшифрування повідомлень, зашифрованих матричним шифром з ключем. Програма перетворює зашифрований текст в початковий. Зашифрований текст розміщений в текстовому файлі file_shyf_text.txt в символьному форматі. Програма складається з наступних елементів.

1 Підключення бібліотечних файлів, які містять прототипи функцій файлового вводу-виводу, обробки символьної інформації та роботи системи:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

2 Опис вказівників на змінні структурного типу FILE, які асоціюють фізичні файли на диску з потоками вводу-виводу:

FILE *fp1,*fp2, *fp3, *fp4;

Потік fp1 зв’язаний з файлом, що містить шифротекст, fp2 – з файлом для зберігання відкритого тексту. Потік fp3 асоціюється з файлом для зберігання ключа, а потік fp4 зв’язаний з файлом для запису проміжних матриць.

3 Задання імен використовуваних файлів:

char filename1[30]=" file_shyf_text.txt"; /* fp1 */

char filename2[30]=" File_output_text.txt";/* fp2 */

char filename3[30]=" File_key.txt"; /* fp3 */

char filename4[30]=" File_deshyf_matr.txt";/* fp4 */

4 Відкриття файлу file_shyf_text.txt, який містить шифротекст:

fp1=fopen(filename1,mode_r);

if (fp1!=NULL) {printf("file %s open mode %s\n",filename1, mode_r);}

else { printf("file %s not open mode %s\n", filename1, mode_r); exit(1);}

5 Відкриття файлу File_output_text.txt, для запису розшифрованого тексту:

fp2=fopen(filename2,mode_w);

if (fp2!=NULL) {printf("file %s open mode %s\n",filename2, mode_w);}

else { printf("file %s not open mode %s\n", filename2, mode_w); exit(2);}

6 Відкриття файлу file_key.txt, для читання ключа:

fp3=fopen(filename3,mode_r);

if (fp3!=NULL){printf("file %s open mode %s\n",filename3, mode_r);}

else { printf("file %s not open mode %s\n",filename3, mode_r); exit(3);}

7 Відкриття файлу File_deshyf_matr.txt для запису шифроматриці:

fp4=fopen(filename4,mode_w);

if (fp4!=NULL){printf("file %s open mode %s\n",filename4, mode_w);}

else { printf("file %s not open mode %s\n",filename4, mode_w); exit(4);}

При відкритті всі файли перевіряються на правильність їх відкриття. При неможливості відкрити файли виводяться відповідні повідомлення і програма закінчує роботу, так як неможливо прочитати повідомлення для шифрування або записати зашифроване повідомлення.

8 Читання та кодування літер ключа, формування динамічних масивів key_MAS та key_pos для зберігання числових кодів літер ключа та їх номерів в алфавіті, визначення довжини ключа:

i=0; char_key[i]=fgetc(fp3);

while(char_key[i]!=EOF) { i++; char_key[i]=fgetc(fp3);}

l_key=strlen(char_key)-1; /* Довжина ключа */

key_MAS=(int *)calloc(l_key, sizeof(int));

key_pos=(int *)calloc(l_key, sizeof(int));

for (i=0; i<l_key; i++)

{key_MAS[i]=kodyvannja (char_key[i]); }

9 Читання шифротексту повідомлення з файлу file_shyf_text.txt і запис його в матрицю shyf_matr[kr_m][l_key], де kr_m кількість рядків матриці shyf_matr та визначення його довжини:

i=0; shyf_mas[i]=fgetc(fp1);

while(shyf_mas[i]!=EOF)

{ i++; shyf_mas[i]=fgetc(fp1);}

l_text=strlen(shyf_mas)-1; /* Довжина шифротексту */

10 Визначення кількості рядків kr_m матриці kr_m запис у неї прочитаного штфротексту, запис матриці в файл:

kr_m= l_text/l_key;

k=0; for (j=0; j<l_key; j++)

for (i=0; i<kr_m; i++)

{shyf_matr[i][j]= shyf_mas[k];k++ ;}

for (i=0; i<kr_m; i++)

{for (j=0; j<l_key; j++)

fprintf(fp4,"%c\t",shyf_matr[i][j]); fprintf(fp4,"\n");}

11 Дешифрування шифротексту і запис матриці в файл File_deshyf_matr.txt:

for (i=0; i<kr_m; i++)

```
for (j=0; j<l_key; j++)

shyf_matr_text[i][j]=shyf_matr[i][key_pos[j]-1];

for (i=0; i<kr_m; i++)

{for (j=0; j<l_key; j++)

fprintf(fp4,"%c\t",shyf_matr_text[i][j]); fprintf(fp4,"\n");}
```

12 Перетворення матриці в розшифрований текст і запис його у файл File_output_text.txt:

```
k=0; for (i=0; i<kr_m; i++)

for (j=0; j<l_key; j++)

{shyf_mas_text[k]=shyf_matr_text[i][j];

fprintf(fp2,"%c",shyf_mas_text[k]);k++ ;}
```

13 Закриття файлів:

```
fclose(fp1);

fclose(fp2);

fclose(fp3);

fclose(fp4);
```

Результатом роботи програми є створення файлу c:\k\File_output_text.txt, в якому знаходиться початковий текст. Повний текст програми наведено в додатку Б.

3.7 Результати роботи програми для дешифрування шифротекстів

Вхідними даними для програми є шифротекст, який знаходиться у файлі file_shyf_text.txt(рис. 3.8). Зашифрований текст записується у вигляді матриці по стовпцях у файл File_deshyf_matr.txt., показаний на рис. 3.7.

Шифротекст розшифровується за допомогою матричного шифру з ключем «КОЛЕДЖ» і записується у вигляді розшифрованої матриці у файл File_shyf_matr.txt. На рис. 3.10 показано вигляд розшифрованої матриці.

Рисунок 3.10 – Вигляд розшифрованої матриці

Зчитуючи текст матриці по рядках, одержимо початковий текст, який записується у файл File_output_text.txt. На рис. 3.11 показано 22 файл розшифрованого тексту.

22 Рисунок 22 3.11 22 – Файл розшифрованого тексту

22 Аналіз одержаних результатів показує, що після проведеної операції шифрування відкритого тексту, наведеного на рис. 3.4, та операції розшифрування зашифрованого тексту, одержано початковий текст, представлений на рис. 3.11.

Посилання

Це джерела виділених збігів у вашому документі. Кожен збіг позначено темно-зеленим числом, яке відповідає вказаному тут джерелу. Джерела впорядковані за схожістю — чим вищий бал, тим сильніше збіг.

#	Джерело	%
1	nmetau.edu.ua	9.3%
2	skachatvs.com	6.8%
3	files.znu.edu.ua	2.5%
4	dspace.uzhnu.edu.ua	2.4%
5	metod.onat.edu.ua	2.3%
6	essuir.sumdu.edu.ua	1.4%
7	wiki.tntu.edu.ua	1.1%
8	info.dgu.edu.ua	1.0%
9	dspace.lvduvs.edu.ua	0.9%
10	antibotan.com	0.8%
11	dspace.wunu.edu.ua	0.7%
12	studfile.net	0.7%
13	teteryakv12.wordpress.com	0.7%
14	stick-king.ru	0.6%
15	xemttc.at.ua	0.6%
16	kananikol.ru	0.5%
17	applaw.net	0.5%
18	ela.kpi.ua	0.5%
19	ekt.elit.sumdu.edu.ua	0.5%
20	ufin.com.ua	0.2%
21	repository.rshu.edu.ua	0.2%
22	elartu.tntu.edu.ua	0.2%
23	ela.kpi.ua	0.1%

#	Джерело	%
24	essuir.sumdu.edu.ua	0.1%
25	uabs.edu.ua	0.0%

Plag

Дякуємо, що перевірили
свій документ за допомогою
Plag!