НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. ІГОРЯ СІКОРСЬКОГО”

КАФЕДРА АВТОМАТИЗОВАНИХ СИСТЕМ ОБРОБКИ ІНФОРМАЦІЇ І УПРАВЛІННЯ

Практична робота № 3

з дисципліни

“Основи захисту інформації”

Варіант 4

Виконала:

студентка групи ІС-71

Вознюк О. В.

Перевірив:

аспірант

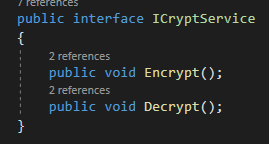
Ільїн К. І.

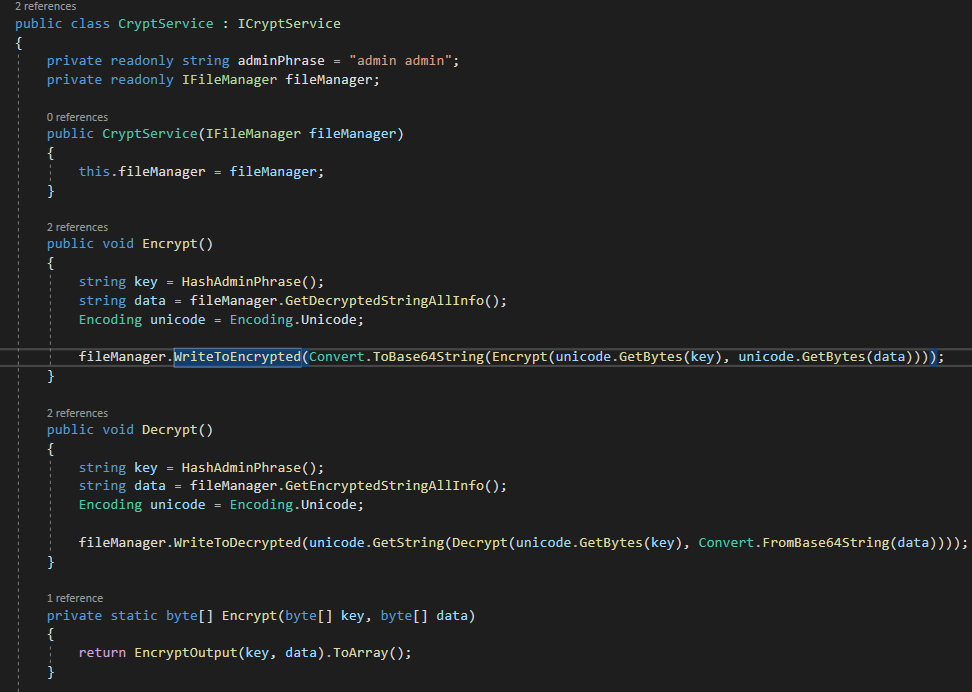
Київ-2020

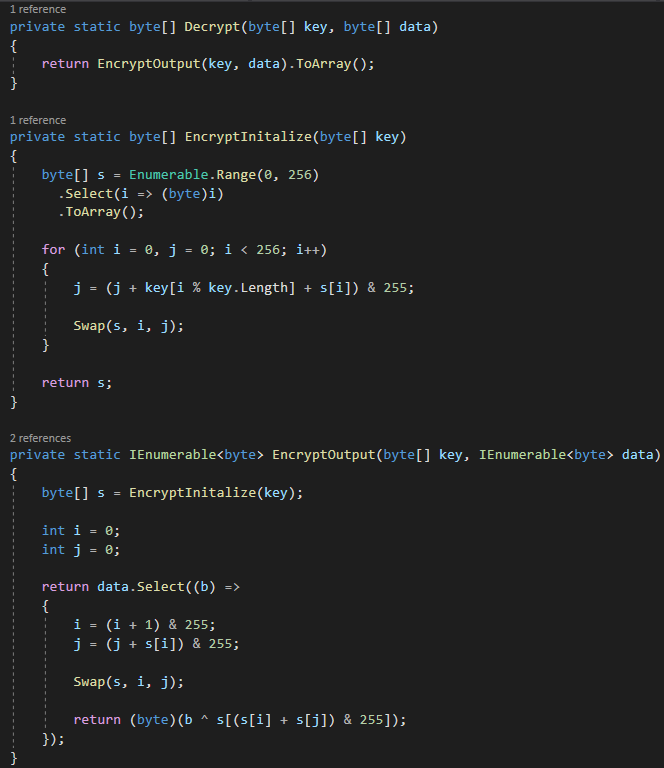
**Хід виконання роботи**

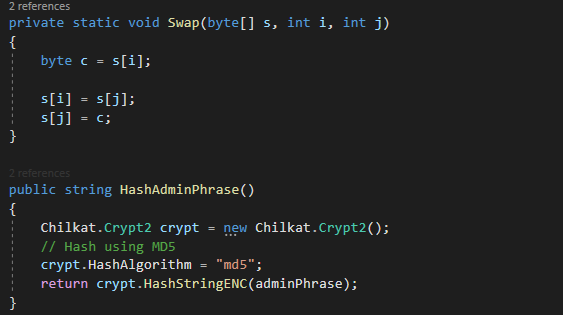
https://github.com/oleksandravozniuk/InformationSecurity/tree/master/SecurityProtectionBasis3

1. Був створений інтерфейс для для надання операцій шифрування та дешифрування та імплементація цього інтерфейсу у вигляді сервісу

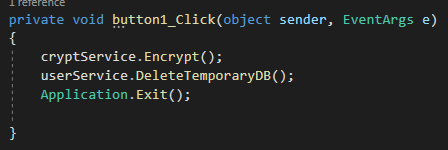




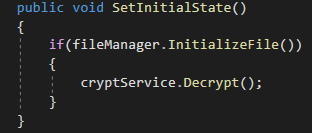




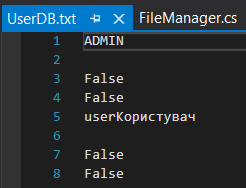
1. Метод шифрування та метод видалення тимчасової бд викликається коли ми закриваємо програму за допомогою кнопки “Exit”.



1. Далі дешифрування відбувається при запуску програми.



1. Вигляд тимчасової бд, де є записи про двох юзерів.



1. Вигляд зашифрованої бд



**Відповіді на теоретичні питання**

1. Які криптографічні прийоми (підстановки, перестановки,

перемішування, гамування) використовують режими СВС, ЕСВ в блочних шифрах?

ECB використовує підстановку :

Повідомлення ділиться на блоки однакового розміру. Розмір (довжина) блоку дорівнює n і вимірюється в бітах. В результаті виходить послідовність блоків P1,P2,..,Pn.Останній блок при необхідності доповнюється до довжини n.

Кожен блок Pi шифрується алгоритмом шифрування Ek з використанням ключа k: Ci=Ek(Pi,K), де:

i - номер блоку;

k - ключ;

Pi - блок повідомлення (відкритий текст);

Ci - зашифрований блок (шифротекст);

Ek - функція, що виконує блочне шифрування.

В результаті виходять зашифровані блоки C1,C2,…,Cn.

CBC використовує гамування:

Повідомлення розбивається на блоки однакового розміру. Розмір блоку дорівнює n і вимірюється в бітах. При необхідності останній блок доповнюється до довжини n. Шифрування чергового (i-го) блоку повідомлення Рі виконується з використанням попереднього зашифрованого ((i-1) -го) блоку (Сі-1). Для першого блоку (Р1) зашифрованого блоку (С0) не існує, тому перший блок шифрують з використанням вектора ініціалізації: С0 = IV (вектор ініціалізації - випадкове число).Розмір (довжина) IV дорівнює розміру блоку (n).

У функцію шифрування Ek передається сума по модулю 2 (гамування) поточного блоку повідомлення Pi і попереднього зашифрованого блоку Ci=Ek(PiCi-1,k), де:

i - номер блоку;

k - ключ;

IV - вектор ініціалізації (сінхропосилка);

Pi - блок повідомлення (відкритий текст);

Ci-1 - зашифрований блок (шифротекст), отриманий на попередньому кроці шифрування;

Ek - функція, що виконує блочне шифрування.

2. Проведіть порівняльний аналіз режимів «Проста заміна»,

«Гамування зі зворотним зв’язком», «Режим вироблення

імітовставки» в ДСТУ ГОСТ 28147:2009.

Проста заміна: застосування ДСТУ ГОСТ 28147:2009 в режимі простої заміни бажане лише для ключових даних, адже має наступні недоліки:

* Може застосовуватися тільки для шифрування відкритих текстів з довжиною, кратною 64 біт.
* При шифруванні однакових блоків відкритого тексту виходять однакові блоки шифротекста, що може дати певну інформацію криптоаналітику.

Для зашифровування в цьому режимі 64-бітний блок відкритого тексту спочатку розбивається на дві половини: Tо = (A0, B0). На i-му циклі використовується з'єднання Xі:

* + Ai+1=Bif(Ai,Xi)
  + Bi+1=Ai

Для генерації підключів вихідний 256-бітний ключ розбивається на вісім 32-бітових чисел: K0 ... K7. Підключі X0 ... X23 є циклічним повторенням K0 ... K7. Підключи X24 ... X31 є K7 ... K0. Результатом виконання всіх 32 раундів алгоритму є 64-бітний блок шифртекста: tш = (A32, B32).

Розшифрування здійснюється за тим же алгоритмом, що і зашифрування, з тією зміною, що інвертується порядок підключів: X0...X7 є K0...K7, а X8... X31 є циклічним повторенням K7... K0.

Функція f(Ai,Xi) обчислюється таким чином:

Ai і Xi складаються по модулю 232.

Результат розбивається на вісім 4-бітових підпослідовностей, кожна з яких надходить на вхід свого вузла таблиці замін (в порядку зростання старшинства бітів), званого нижче S-блоком. Загальна кількість S-блоків стандарту - вісім, тобто стільки ж, скільки і підпослідовностей. Кожен S-блок являє собою перестановку чисел від 0 до 15 (конкретний вид S-блоків в стандарті не визначено). Перша 4-бітна підпослідовність потрапляє на вхід першого S-блоку, друга - на вхід другого і т. д.

Якщо вузол S-блоку виглядає так: 1, 15, 13, 0, 5, 7, 10, 4, 9, 2, 3, 14, 6, 11, 8, 12

і на вході S-блоку 0, то на виході буде 1, якщо 4, то на виході буде 5, якщо на вході 12, то на виході 6 і т. д

Виходи всіх восьми S-блоків об'єднуються в 32-бітове слово, потім все слово циклічно зсувається вліво (до старшим розрядам) на 11 бітів.

Гамування зі зворотним зв’язком: При використанні даного режиму слід мати на увазі, що синхропосилку не можна використовувати повторно (наприклад, при шифруванні логічно роздільних блоків інформації - мережевих пакетів, секторів жорсткого диска і т. п). Це обумовлено тим, що перший блок шифр-тексту отримано всього лише складанням по модулю два з зашифрованою синхропосилкою; таким чином, знання всього лише 8 перших байт вихідного і шифрованого тексту дозволяють читати перші 8 байт будь-якого іншого шифр-тексту після повторного використання синхропосилки.

1. Синхропосилка заноситься в регістри N1 і N2
2. Вміст регістрів N1 і N2 шифрується відповідно до алгоритму простої заміни. Отриманий результат є 64-бітовим блоком гамми.
3. Блок гами побітно складається по модулю 2 з блоком відкритого тексту. Отриманий шифротекст заноситься в регістри N1 і N2
4. Операції 2-3 виконуються для решти блоків вимагає шифрування тексту

При зміні одного біта шифротекста, отриманого з використанням алгоритму гамування зі зворотним зв'язком, у відповідному блоці розшифрованого тексту змінюється тільки один біт, так само зачіпається наступний блок відкритого тексту. При цьому всі інші блоки залишаються незмінними.

Режим вироблення імітовставки: Цей режим не є в загальноприйнятому сенсі режимом шифрування. При роботі в режимі вироблення імітовставки створюється певний додатковий блок, що залежить від усього тексту і ключових даних. Даний блок використовується для перевірки того, що в шифротекст випадково або навмисно не були внесені спотворення.

Імітовставка виробляється для M ≥ 2 блоків відкритого тексту по 64 біт.

1. Блок відкритих даних записується в регістри N1 і N2, після чого піддається перетворенню, що відповідає першим 16 циклам шифрування в режимі простої заміни
2. До отриманого результату побітно по модулю 2 додається наступний блок відкритих даних. Останній блок при необхідності доповнюється нулями. Сума також шифрується відповідно до пункту 1.
3. Після додавання і шифрування останнього блоку з результату вибирається імітоставка довжиною L біт: з біта номер 32-L до 32 (відлік починається з 1). Стандарт рекомендує вибирати L виходячи з того, що ймовірність нав'язування помилкових даних дорівнює 2-L. Імітовставка передається по каналу зв'язку після зашифрованих блоків.

3. Сфера застосування блокових та поточних шифрів.

Сфера застосування потокових шифрів - військові, мережеві, телефонні та інші системи, де необхідно перетворення мовної інформації в цифрову форму і надійне шифрування даних. Причина популярності - простота реалізації і конструювання генераторів, надійність шифрування, відсутність помилок в потоковому шифрі.

Блочні:

* Електронна кодова книга (ЄСВ)

Передача окремих значень (коротких текстів, ключової інформації)

* Зчеплення блоків шифру (СВС)

Поблочну передача даних загального призначення. аутентифікація

* Зворотній зв'язок по шифротексту (CFB)

Передача потокового даних загального призначення. аутентифікація

4. Засоби генерації ключових даних для симетричних алгоритмів

шифрування: генератори псевдовипадкових чисел та датчики

псевдовипадкових чисел.

Криптографічні додатки використовують для генерації випадкових чисел детерміновані алгоритми, отже, генерують послідовність чисел, яка теоретично не може бути статистично випадковою. У той же час, якщо вибрати хороший алгоритм, отримана чисельна послідовність - псевдовипадкових чисел - буде проходити більшість тестів на випадковість. Однією з характеристик такої послідовності є великий період повторення.

Прикладами відомих криптостійкі ГПВЧ є RC4, ISAAC, SEAL, SNOW, дуже повільний теоретичний алгоритм Блюм - Блюма - Шуба, а також лічильники з криптографічними хеш-функціями або крипостійкість блоковими шифрами замість функції виведення.

Приклад: Генератор псевдовипадкових чисел ANSI X9.17

* Вхід: генератором керують два псевдовипадкових входи. Один є 64-бітовим представленням поточних дати і часу, які змінюються кожного разу при створенні числа. Інший є 64-бітовим початковим значенням; воно ініціалізується деяким довільним значенням і змінюється в ході генерації послідовності псевдовипадкових чисел.
* Ключі: генератор використовує три модулі потрійного DES. Всі три використовують одну і ту ж пару 56-бітних ключів, яка повинна триматися в секреті і застосовуватися тільки для генерації псевдовипадкового числа.
* Вихід: вихід складається з 64-бітного псевдовипадкового числа і 64-бітного значення, яке буде використовуватися в якості початкового значення при створенні наступного числа.

DTi - значення дати і часу на початок i-ой стадії генерації.

Vi - початкове значення для i-ой стадії генерації.

Ri - псевдовипадкове число, створене на i-ої стадії генерації.

K1, K2 - ключі, які використовуються на кожній стадії.



Схема включає використання 112-бітного ключа і трьох EDE-шифрування. На вхід подаються два псевдовипадкових значення: значення дати і часу і початкове значення черговий ітерації, на виході створюються початкове значення для наступної ітерації і чергове псевдовипадкове значення. Навіть якщо псевдовипадкове число Ri буде скомпрометовано, обчислити Vi + 1 з Ri неможливо, і, отже, наступне псевдовипадкове значення Ri + 1, так як для отримання Vi + 1 додатково виконуються три операції EDE.

Датчики псевдовипадкових чисел:

Зазвичай датчики випадкових чисел містять m генераторів описаного типу, що працюють незалежно, так що датчиком видається наближене випадкове число записане в формі m-розрядного двійкового дробу. Для випадкових чисел відведена спеціальна комірка в накопичувачі, і швидкість генерування їх настільки велика, що на кожному такті роботи ЕОМ в цьому осередку виходить нове випадкове число. Застосування датчиків випадкових не має тих недоліків, які перешкоджають широкому застосуванню таблиць: не потрібно місця у внутрішньому накопичувачі і запас чисел практично необмежений.

5. Яким чином можна оцінити надійність ключа криптографічного

алгоритма?

Для сучасних симетричних алгоритмів (AES, CAST5, IDEA, Blowfish, Twofish) основною характеристикою криптостійкості є довжина ключа. Шифрування з ключами довжиною 128 біт і вище вважається сильним, так як для розшифровки інформації без ключа потрібні роки роботи найпотужніших суперкомп'ютерів. Для асиметричних алгоритмів, заснованих на проблемах теорії чисел (проблема факторизації - RSA, проблема дискретного логарифма - Elgamal) в силу їх особливостей мінімальна надійна довжина ключа в даний час - 1024 біт.

Для асиметричних алгоритмів, заснованих на використанні теорії еліптичних кривих (ECDSA, ГОСТ Р 34.10-2001, ДСТУ 4145-2002), мінімальної надійної довжиною ключа вважається 163 біт, але рекомендуються довжини від 191 біт і вище.

**Висновок**

Були вивчені функції графічного криптографічного інтерфейсу ОС Windows. Були досліджені типи симетричного шифрування, режими шифрування та алгоритми хешування.