Міністерство освіти і науки України

Харківський національний університет ім. В. Н. Каразіна

Факультет комп’ютерних наук

Кафедра безпеки інформаційних систем і технологій

Лабораторна робота №6

*з навчальної дисципліни*

«Прикладна криптологія»

Виконав:

Студент групи КБ-31 Кравченко Є. М.

Перевірив:

Професор

Горбенко І. Д.

Старший викладач

Єсіна М. В.

Харків – 2020

**Лабораторна робота №6**

*на тему:*

**«Вивчення побудови та реалізації протоколів»**

**Програма роботи**

1. Демонстрація протоколу Діффі-Гелмана.

2. Демонстрації протоколів розділення таємниці.

2.1 Демонстрація протоколу розділення таємниці Шаміра.

2.2 Демонстрація протоколу розділення таємниці за допомогою КТЗ.

3. Протоколи мережевої автентифікації та атака посередині.

3.1 Протокол передачі паролю.

3.2 Протокол передачі паролю сеансу.

3.3 Протокол обміну (симетричний).

3.4 Протокол обміну (асиметричний).

3.5 Протокол взаємної автентифікації.

4. Протокол шифрування «Блокнот – RSA» та атака на реалізацію.

5. Побудова протоколу.

**Мета роботи:** ознайомлення з принципами побудови, використанням, атаками на протоколи. Отримання практичних навичок в розробці протоколів, використання та аналізу їх вразливостей.

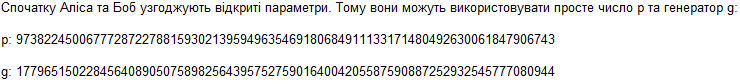
**Теоретичні положення**

*Протокол* – розподілений алгоритм сумісного розв’язку об’єктами та суб’єктами деякої задачі, кожен з яких досягає мети (розв’язує задачу) з використанням деяких алгоритмів, при цьому усі об’єкти та суб’єкти використовують однакову специфікацію даних, синхронізацій дій, організацію відновлення роботи після збоїв тощо.

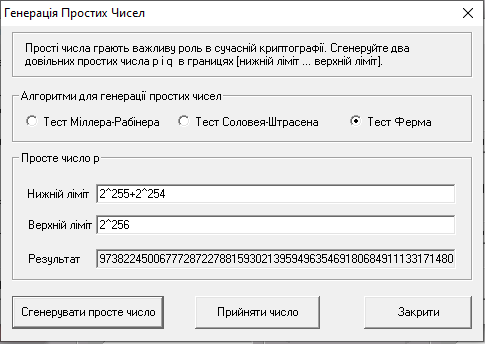
**Хід роботи**

# **1. Демонстрація протоколу Діффі-Гелмана.**

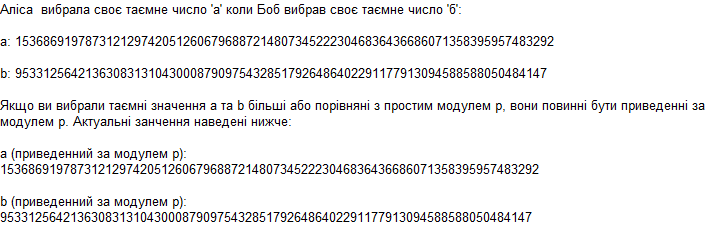
|  |  |
| --- | --- |
|  |  |



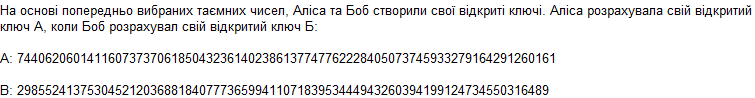
Генерація простих чисел відбувається за допомогою тесту Ферма:



|  |  |
| --- | --- |
|  |  |
|  |  |



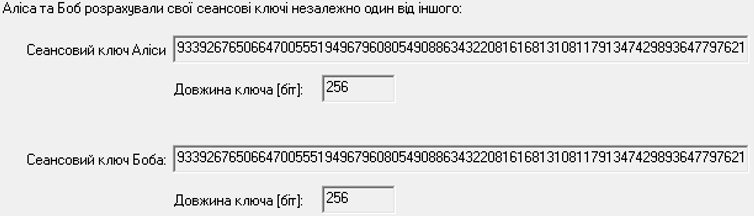
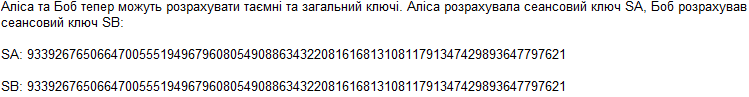
|  |  |
| --- | --- |
|  |  |
|  |  |



|  |  |
| --- | --- |
|  |  |

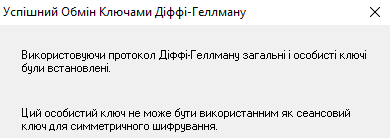


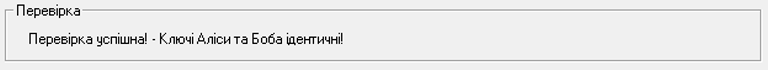
|  |  |
| --- | --- |
|  |  |

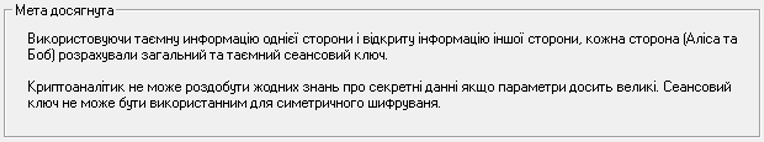


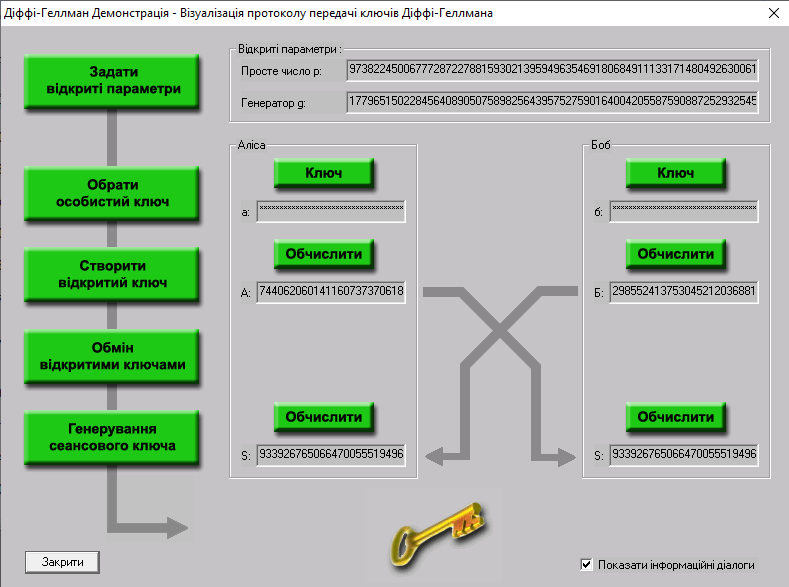


Результати:









Протокол Діффі-Гелмана не є безпечним, адже зовнішній порушник (криптоаналітик) може створювати хибні криптограми, модифікувати криптограми, передавати раніше передані криптограми, порушувати цілісність. Для захисту від цих загроз необхідно застосовувати замість компонент сеансу SA та SB сертифікати відкритих ключів.

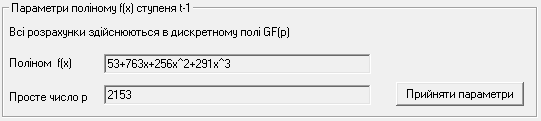
# **2. Демонстрації протоколу розділення таємниці.**

# **2.1. Демонстрація протоколу розділення таємниці Шаміра.**

Спочатку обираємо параметри, які визначають схему Шаміра:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

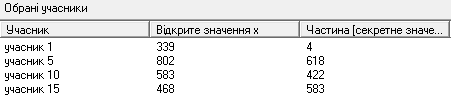
Формуємо поліном *f(x)*:



Значення учасників, обчислені за даними параметрами:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Реконструювання секрету:



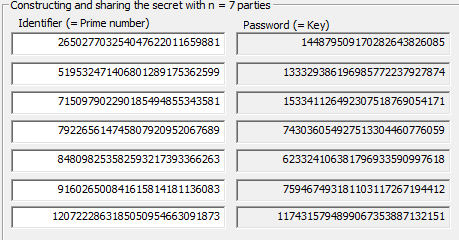


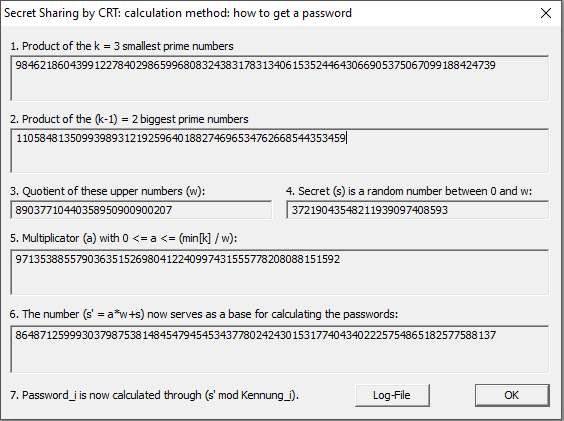
Реконструювання секрету за меншої кількості мінімального порогу неможливе:



# **2.2. Демонстрація протоколу розділення таємниці за допомогою КТЗ.**

Генеруємо прості числа за допомогою тесту Ферма і розраховуємо ключі.





Намагаємося відновити секрет за допомогою 3-х ідентифікаторів:

|  |  |  |
| --- | --- | --- |
|  | | |
|  |  |  |

Намагаємося відновити секрет за допомогою 2-х ідентифікаторів:

|  |  |  |
| --- | --- | --- |
|  | | |
|  |  |  |

Намагаємося відновити секрет за допомогою 3-х ідентифікаторів, один з яких є хибним.

|  |  |  |
| --- | --- | --- |
|  | | |
|  |  |  |

# **3. Протоколи мережевої автентифікації та атака посередині.**

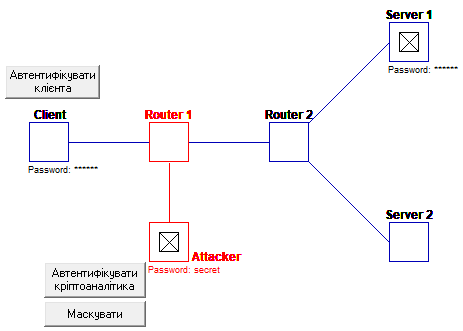
# **3.1. Протокол передачі паролю.**

У цьому сценарії клієнт автентифікує сам себе шляхом передачі паролю, який знає лише він і Server1. Криптоаналітику необхідно автентифікувати себе так, аби Server1 вважав його клієнтом.

Порядок дій:

|  |  |
| --- | --- |
| 1 |  |
| 2 |  |
| 3 | → |
| 4 |  |
| 5 |  |

***Загальний вигляд системи***



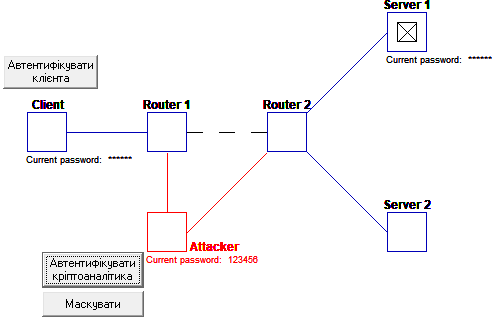
# **3.2. Протокол передачі паролю сеансу.**

Для того щоб вилучити можливість атаки, яка використовується в сценарії пароля, клієнт та сервер погодились змінювати пароль після кожної автентифікації. З цієї причини вони згенерували список сеансових паролів яким володіють тільки вони. Після автентифікації з дійсним паролем, цей пароль помічається як використаний, і він не може бути використаним ще раз. Для наступної автентифікації повинен бути використаний інший пароль зі списку. Криптоаналітику необхідно автентифікувати себе так, аби Server1 вважав його клієнтом.

Порядок дій:

|  |  |
| --- | --- |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 | → |
| 5 |  |
| 6 |  |
| 7 |  |

***Загальний вигляд системи***



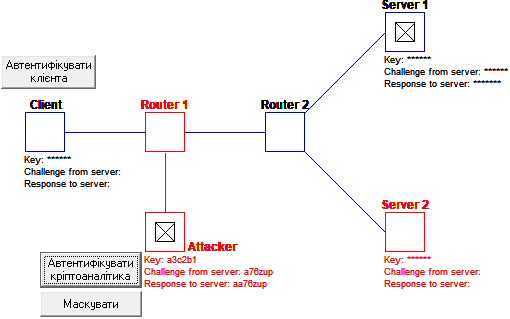
# **3.3. Протокол обміну (симетричний).**

Протокол починає Server1, що відправляє випадкове число клієнтові як виклик. Клієнт отримавши виклик, зашифровує цей номер з секретним ключем і повертає результат як відповідь до Server1, який також знає секретний ключ клієнта. Server1 розшифровує відповідь і порівнює її з викликом. Якщо вони збігаються, ідентичність клієнта прийнята і він автентифікований. Криптоаналітику необхідно автентифікувати себе так, аби Server1 вважав його клієнтом.

Порядок дій:

|  |  |
| --- | --- |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |

***Загальний вигляд системи***



Оскільки сервери та клієнт розділяють один ключ, можна атакувати Сервер №2, щоб здобути його, а потім виконати атаку на один з маршрутизаторів та аутентифікуватися.

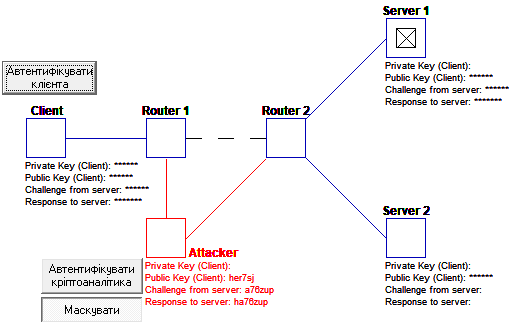
# **3.4. Протокол обміну (асиметричний).**

Процедура протоколу обміну виповняється з асиметричним шифруванням. Кожна сторона має свою пару ключів. Знову, для автентифікації сервер посилає запит клієнту, який підписує його своїм особистим ключем і тоді посилає результат серверу. Тепер усі, хто знає відкритий ключ клієнта, включаючи сервера, може перевірити відгук. Не зважаючи на це криптоаналітик може аутентифікувати себе на сервері як клієнт.

Порядок дій:

|  |  |
| --- | --- |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 | →→  → |

***Загальний вигляд системи***

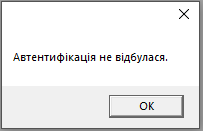


Криптоаналітик повинен розірвати зв’язок між двома марштрутизаторами та підключитись до кожного з них, після чого активувати маскування. В результаті цих дій сервер сприйме криптоаналітика як клієнта і автентифікує його.

# **3.5. Протокол взаємної автентифікації.**

Цей сценарій також побудований на асиметричній криптографії. Але тепер він використовується з двох сторін. Сервер посилає запит клієнту, який клієнт підписує и вертає разом зі своїм запитом. Тепер сервер може перевірити використовуючи відкритий ключ клієнту, який підписав запит, який може буде підписаний тільки клієнтом, тому що тільки він володіє своїм таємним ключем. Для того щоб довести свою ідентичність, сервер підписує запит клієнта своїм особистим ключем і повертає його клієнту. Тепер, клієнт може перевірити автентичність серверу. Обидва можуть бути впевнені в тому, що спілкуються зі справжнім партнером. Проте вони повинні вести зашифровані переговори, щоб виповнити вимогу цілісності.

*Атака на цьому рівні неможлива.* Жодна с попередніх атак не змогла зламати протокол.



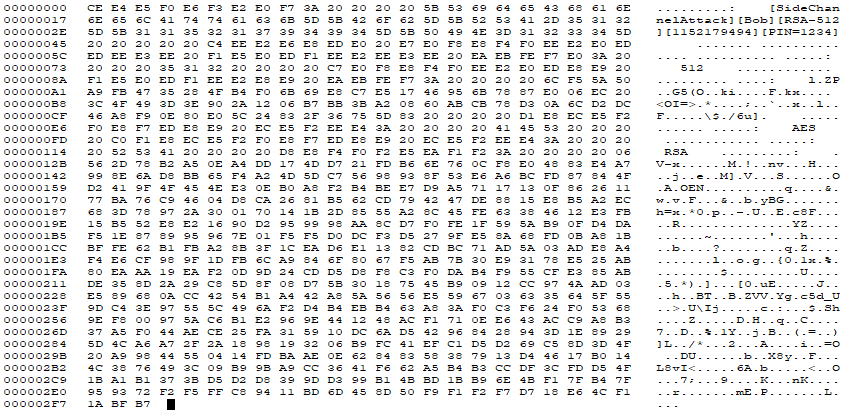
«Атака посередині» − це тип атаки, коли криптоаналітик намагається зайняти положення між двома сторонами, та намагатися прослуховувати обидві сторони. При цьому криптоаналітик виступає в мережі певним маршрутизатором, який може симулювати іншу сторону до кожної сторони, для того, щоб отримати доступ до інформації.

Криптоаналітик – це спеціаліст, який використовує різні методи для здійснення атак на криптоалгоритми та мережі.

Найбільш надійний спосіб захисту від «атаки посередині» – використання протоколу взаємної автентифікації, при якому схема Виклик-Відповідь застосовується в обох напрямках.

# **4. Протокол шифрування “Блокнот – RSA” та атака на реалізацію.**

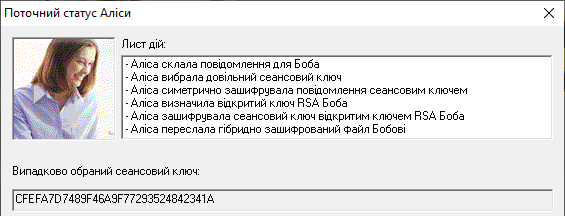
Гібридно зашифрований текст:

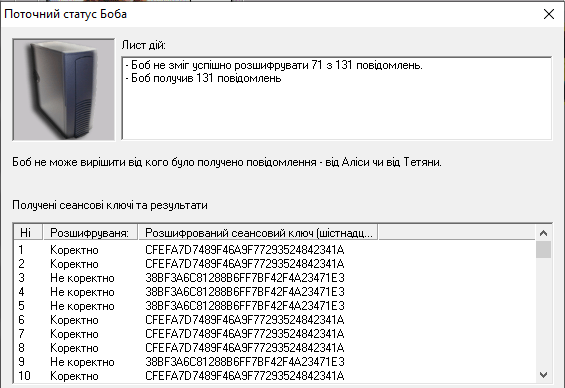


Звіт атаки:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

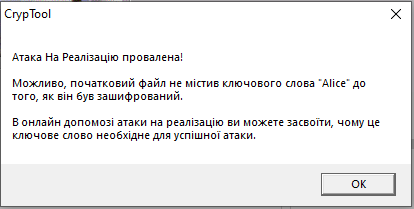
Статуси учасників:







Атака провалиться у випадку, якщо документ не буде містити слово “Alice”.



Атака на реалізацію – це спеціальний тип атаки, при якому криптоаналітик не використовує слабкості в математиці криптографічних перетворень, що використовуються, замість цього використовуються помилки в реалізації або використанні протоколу.

Типи атак на криптопротоколи: атака з відомим ключем, повторна передача, підміна боку інформаційного обміну, атака зі словником, підміна повідомлень.

Аби захиститися від атаки на реалізацію необхідер виконувати вимоги, що вказані в нормативних документах до алгоритму шифрування, або протоколу та не порушувати умови експлуатації.

Успішний криптоаналіз протоколу був здійснений завдяки знанню вразливості в системі.

# **5. Побудова протоколу.**

Система RSA відноситься до криптосистем з відкритими ключами. В цій системі ключі , причому один з них має бути особистим, а другий – відкритим. Наприклад, – особистий, а  – відкритий, якщо вони використовуються для ЕЦП і навпаки, якщо використовуються для направленого шифрування.

Усі параметри  також поділяються на 2 класи:  – відкритий,  – конфіденційні (секретні).

Сутність забезпечення моделі взаємної недовіри – кожен користувач генерує ключі сам собі. Особистий ключ залишає в себе і забезпечує його сувору конфіденційність. Відкритий ключ розсилає всім користувачам, з якими він зв'язаний. Користувач також забезпечує цілісність і дійсність відкритих ключів.

 – мають вибиратися з повної множини випадково, порівняно ймовірно і незалежно, мають забезпечувати однозначну оборотність прямого та зворотного перетворення. Відповідним чином засвідчений відкритий ключ є сертифікатом.

Значення для практичних використань мають задовольняти умову

|  |  |
| --- | --- |
| , де | (1) |

Порівняння (1) можна звести до діафантового рівняння:

|  |  |
| --- | --- |
|  | (2) |

Це діафантове рівняння – нормоване, тому що справа коефіцієнт дорівнює 1; a, b – цілочисельні коефіцієнти, х, у – невідомі. Порівняння (1) можна подати у вигляді:

|  |  |
| --- | --- |
|  | (3) |

де k – деяке невідоме число.

Діафантове рівняння (2) має цілочисельне розв’язання, якщо a і b цілочисленні, і a ≥ b , a і b взаємно прості. Подавши (3) у вигляді

|  |  |
| --- | --- |
|  | (4) |

отримаємо .

Якщо Ek сформувати випадково, то а та b – відомі числа, а х та y – невідомі, що підлягають визначенню. Найбільш швидке розв’язання (4) дає застосування ланцюгових дробів, які дозволяють визначити х та у як

|  |  |
| --- | --- |
|  | (5) |

 – порядок ланцюгового дробу, a і b – параметри ланцюгового дробу. Знаходимо параметри:

 подається у вигляді ланцюгового дробу

|  |  |
| --- | --- |
|  | (6) |

 – порядок ланцюгового дробу, перший коефіцієнт, у якого залишок дорівнює 0.

Значення  та визначаються як

|  |  |
| --- | --- |
|  |  |

Значення , і т.д. визначаються рекурентно відповідно до правил

|  |  |
| --- | --- |
|  | (7) |

Середнє число ітерацій в (6), тобто , можна визначити як

|  |  |
| --- | --- |
|  |  |

Таким чином, для розв’язку порівняння (4) необхідно подати  у вигляді ланцюгового дробу, визначивши при цьому значення  та значення . Потім згідно з (7) визначити  і  , а згідно з (5) – x і y.

Для заданого дійсного набору параметрів ***еліптичної кривої*** особистий ключ і відповідний відкритий ключ можуть бути генеровані таким чином:

Вибирається випадкове або псевдовипадкове ціле  на відрізку , яке має бути захищене від несанкціонованого розкриття й бути непередбачуваним. Обчислюється точка , де – секретний сеансовий ключ,  – базова точка.

У деяких застосуваннях відкритий ключ обчислюється як , за умови, що .

Існують деякі практичні вимоги при генерації ключів RSA ЕЦП:

1. Співмножники RSA-модуля  повинні обиратися випадково і бути однакової довжини.

2. Величина секретного ключа повинна бути порівнянна з розміром модуля .

3. Довжина відкритого ключа повинна бути строго більше 16 бітів.

Звідси випливає наступний важливий аспект реалізації – обчислювальний. Адже доводиться використовувати апарат арифметики великих чисел.

При генерації ключів на еліптичній кривій основною є алгоритмічна проблема. Найбільш трудомістким етапом обчислення є визначення простого числа , яке є дільником кількості елементів в групі еліптичної кривої. Стійкість же ключів заснована на складності задачі дискретного логарифмування на кривій.

***Висновки***

На даній лабораторній роботі ми ознайомилися з принципами побудови, використанням, атаками на протоколи та отримали практичні навички в розробці протоколів, використанні та аналізу їх вразливих сторін.