КРИПТОГРАФІЯ

КОМП’ЮТЕРНИЙ ПРАКТИКУМ №4
Вивчення криптосистеми RSA та алгоритму електронного підпису; ознайомлення з методами генерації параметрів для асиметричних криптосистем

Виконав: Кандила Микита ФБ-12 6 варіант

Мета роботи: Ознайомлення з тестами перевірки чисел на простоту і методами генерації ключів для асиметричної криптосистеми типу RSA; практичне ознайомлення з системою захисту інформації на основі криптосхеми RSA, організація з використанням цієї системи засекреченого зв'язку й електронного підпису, вивчення протоколу розсилання ключів.

Порядок виконання роботи:

1. Написати функцію пошуку випадкового простого числа з заданого інтервалу або заданої довжини, використовуючи датчик випадкових чисел та тести перевірки на простоту. В якості датчика випадкових чисел використовуйте вбудований генератор псевдовипадкових чисел вашої мови програмування. В якості тесту перевірки на простоту рекомендовано використовувати тест Міллера-Рабіна із попередніми пробними діленнями. Тести необхідно реалізовувати власноруч, використання готових реалізацій тестів не дозволяється.
2. За допомогою цієї функції згенерувати дві пари простих чисел p, q і 1 1 p , q довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб pq ≤ p1q1 ; p і q – прості числа для побудови ключів абонента A, 1 p і q1 – абонента B.
3. Написати функцію генерації ключових пар для RSA. Після генерування функція повинна повертати та/або зберігати секретний ключ (d, p, q) та відкритий ключ (n,e) . За допомогою цієї функції побудувати схеми RSA для абонентів A і B – тобто, створити та зберегти для подальшого використання відкриті ключі (e, n) , (e_1, n_1) та секретні d і d_1.
4. Написати програму шифрування, розшифрування і створення повідомлення з цифровим підписом для абонентів A і B. Кожна з

операцій (шифрування, розшифрування, створення цифрового підпису, перевірка цифрового підпису) повинна бути реалізована окремою процедурою, на вхід до якої повинні подаватись лише ті ключові дані, які необхідні для її виконання. За допомогою датчика випадкових чисел вибрати відкрите повідомлення M і знайти криптограму для абонентів A и B, перевірити правильність розшифрування. Скласти для A і B повідомлення з цифровим підписом і перевірити його.

5. За допомогою раніше написаних на попередніх етапах програм організувати роботу протоколу конфіденційного розсилання ключів з підтвердженням справжності по відкритому каналу за допомогою алгоритму RSA. Протоколи роботи кожного учасника (відправника та приймаючого) повинні бути реалізовані у вигляді окремих процедур, на вхід до яких повинні подаватись лише ті ключові дані, які необхідні для виконання. Перевірити роботу програм для випадково обраного ключа $0 < k < n$. Кожна з наведених операцій повинна бути реалізована у вигляді окремої процедури, інтерфейс якої повинен приймати лише ті дані, які необхідні для її роботи; наприклад, функція Encrypt(), яка шифрує повідомлення для абонента, повинна приймати на вхід повідомлення та відкритий ключ адресата (і тільки його), повертаючи в якості результату шифротекст.

Відповідно, програмний код повинен містити сім високорівневих процедур: GenerateKeyPair(), Encrypt(), Decrypt(), Sign(), Verify(), SendKey(), ReceiveKey(). Кожну операцію рекомендується перевіряти шляхом взаємодії із тестовим середовищем, розташованим за адресою http://asymcryptwebservice.appspot.com/?section=rsa

Наприклад, для перевірки коректності операції шифрування необхідно

a) зашифрувати власною реалізацією повідомлення для серверу та розшифрувати його на сервері,
b) зашифрувати на сервері повідомлення для вашої реалізації та розшифрувати його локально

1. Математичні функції

З минулої лабораторної роботи були взяті функції для знаходження НСД та оберненого елемента *extended_gcd()* і *find_inverse()*.

Для знаходження простого числа довжиною 256 біт була реалізована функція *find_prime()*, яка в свою чергу викликає функцію *is_prime_miller_rabin()*. Остання функція перевіряє, чи є число простим. Також в цій функції є виклик двох додаткових функцій, а саме *d_2s()* та *power_barret()*. Остання потрібна для швидкого піднесення числа до степеня і ділення за модулем

2. Реалізація RSA

2.1 Генерація ключів

Для створення пари ключів для абонента А та Б, була використані функції *find_p1q1p2q2()* та *generate_key_pair()*. Перша знаходить 2 пари чисел (p_1, q_1) та (p_2, q_2) такі, що $p_1 < p_2,\ q_1 < q_2$. Функція *generate_key_pair()* створює публічний та приватний ключ для абонентів.

2.2 Шифровка/розшифровка

Для шифрування була використана функція *encrypt()*, яка шифрує повідомлення за допомогою відкритого ключа одно із абонентів. Функція використовує *power_barret()* для швидкого піднесення до степеня і ділення за модулем. Функція розшифровки працює аналогічно, тільки використовується приватний ключ. Реалізація – *decrypt()*.

2.3 Підпис

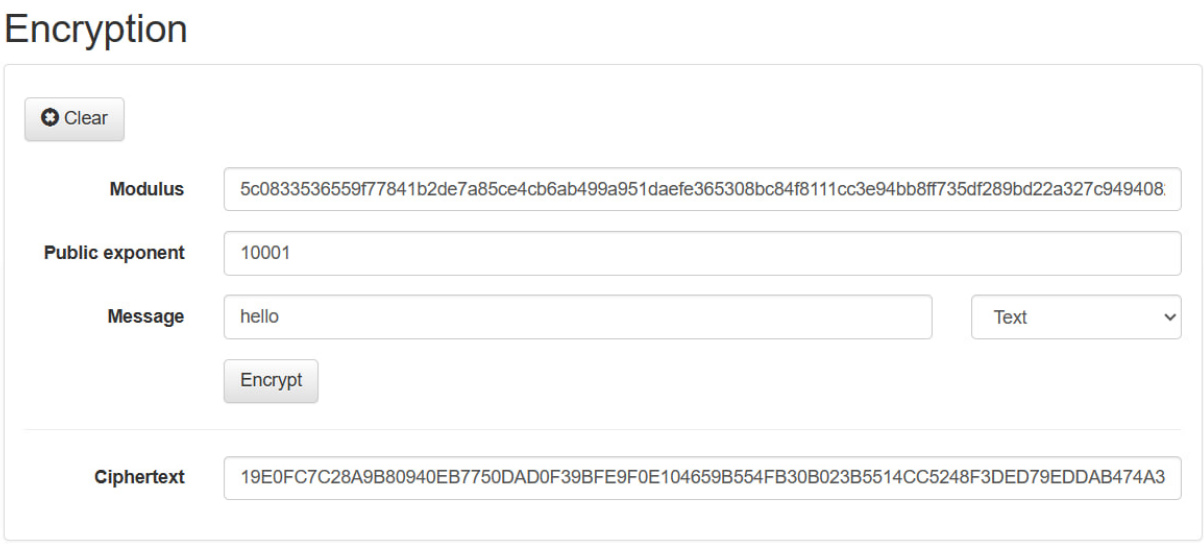
Функція для підпису повідомлення спочатку знаходить хеш від повідомлення, потім шифрує хеш, використовую приватний. Функція для перевірки підпису використовує відкритий ключ та перевіряє, чи сходиться надісланий хеш та хеш, отриманий після розшифровки. Реалізація – *sign()*, *verify()*.

2.4 Відправка даних

Реалізація – *send()*, *receive()*. В функції *send()* абонент А шифрує повідомлення публічним ключем користувача Б та підписує повідомлення своїм приватним ключем і після цього шифрує підпис публічним ключем Б. В функції *receive()* абонент Б розшифровує повідомлення своїм приватним ключем та перевіряє підпис теж публічним ключем А. Якщо все сходиться, то надіслене повідлення істинне.

3. Перевірка на сайті

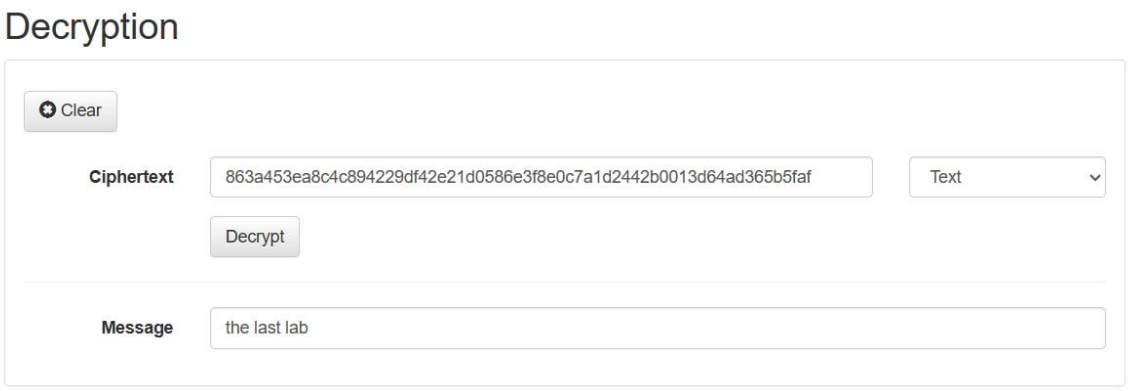
3.1 Шифровка




```
Testing RSA on website
Encrypted message: 19E0FC7C28A9B80940EB7750DAD0F39BFE9F0E104659B554FB30B023B5514CC5248F3DED79EDDAB474A3EC7A3750DDD89C7F62F048D73DD1DC050A0D54696F3F
Decryption...
Decrypted message: 'hello'
```

3.2 Розшифровка

```
Message to encrypt: 'the last lab'
e on website = 65537
n on website= 75112827691346006410283400653877159053688882729312595859850448591634012058997
Encrypted message: 0x863a453ea8c4c894229df42e21d0586e3f8e0c7a1d2442b0013d64ad365b5faf
Result - 'the last lab'
```



Висновки: в даній лабораторній роботі я ознайомився з криптосистемою RSA, як реалізовувати шифровку за розшифровку повідомлень, користуючись цією системою, а також як працює та реалізацію цифрового підпису.