Лабораторна робота 5

Фб-11 Іван Кустов, Яцентюк Андрій

Вивчення криптосистеми RSA та алгоритму електронного підпису; ознайомлення з методами генерації параметрів для асиметричних криптосистем

Мета:

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.

Постанова задачі:

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) , (,) 1 n1 e та секретні d і d1

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.

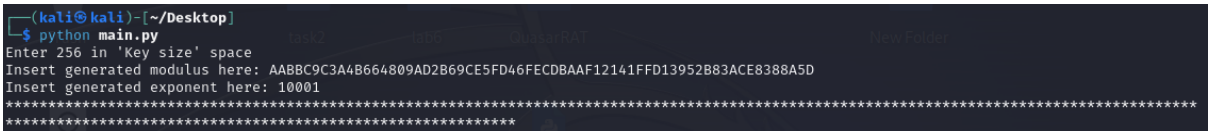
Хід роботи:

Було написано окремі необхідні функції для генерації, отримання, перевірки ключів\шифротекстів. Трохи пізніше, для полегшення орієнтування була написана одна велика функція - routine, в які імплементовано всі інші функції та всі виклики для роботи із сайтом

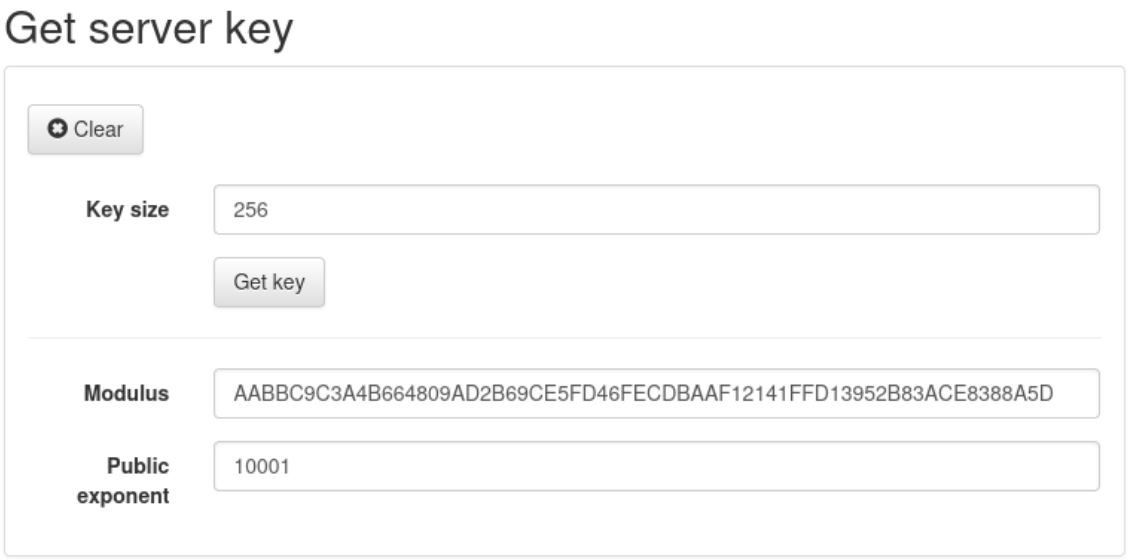
Опис труднощів:

Труднощі почалися із роботою з сайтом, бо сайт написано дегенератом. Спочатку в коді працювали із строками, потім строки переводили в числа(байти), а вже їх в hex, бо сайт хоче hex. Також сайт не вміє працювати із hex з маленькими літерами, йому потрібні великі, і він звісно про це ніде не написав. Потім сайт не видає приватний ключ, як ти його не проси, тому перевірити підпис неможливо. Це не говорячи про то, що сайт при генеруванні ключа добавляє до ключа 0, звісно не говорячи про це. А ну, і ще документація на сайті повна херня - робота з api, як сказано на сайті не працює, бо він просто очікує для передачі не те, що каже документація.

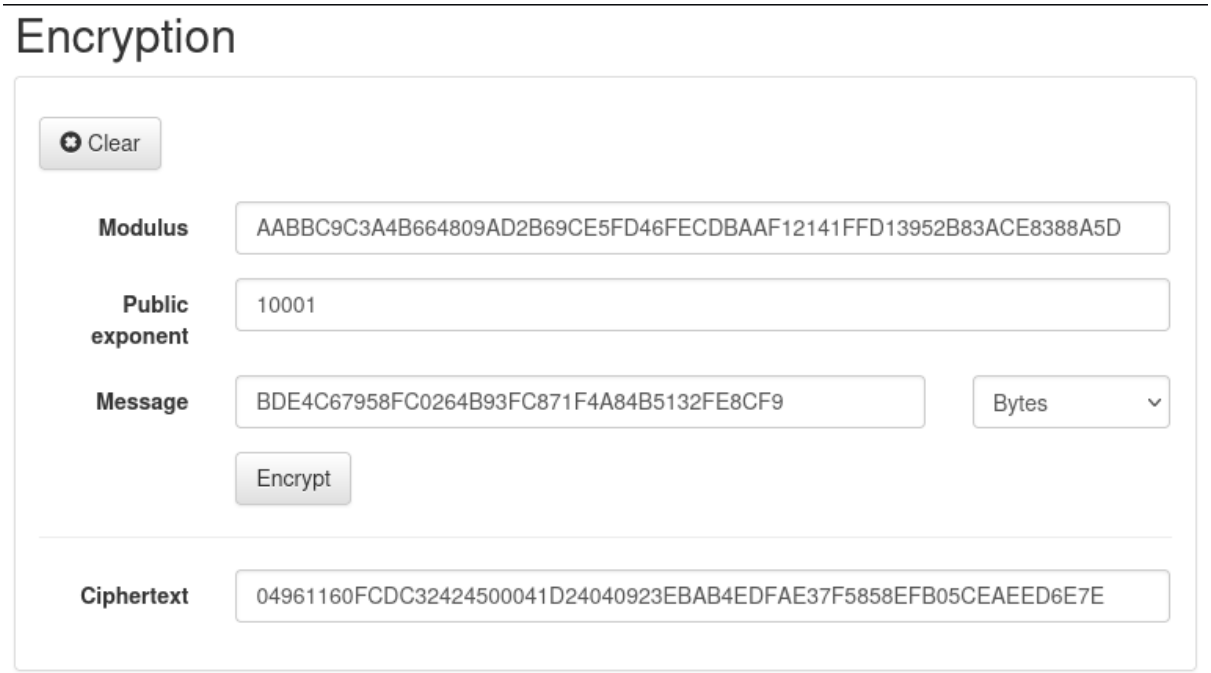
Робота кода:



отримуємо ключ

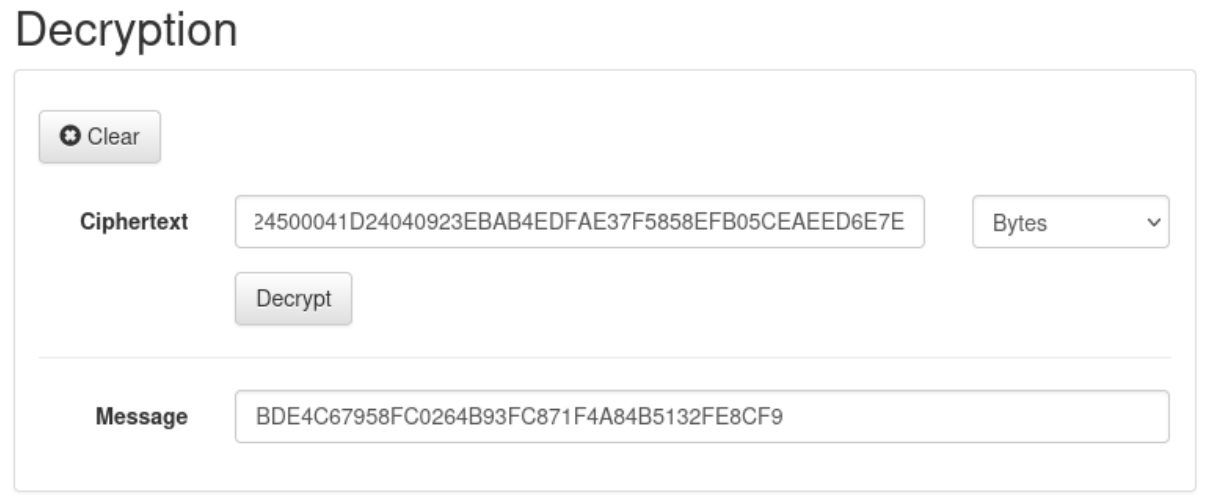


шифруємо повідомлення



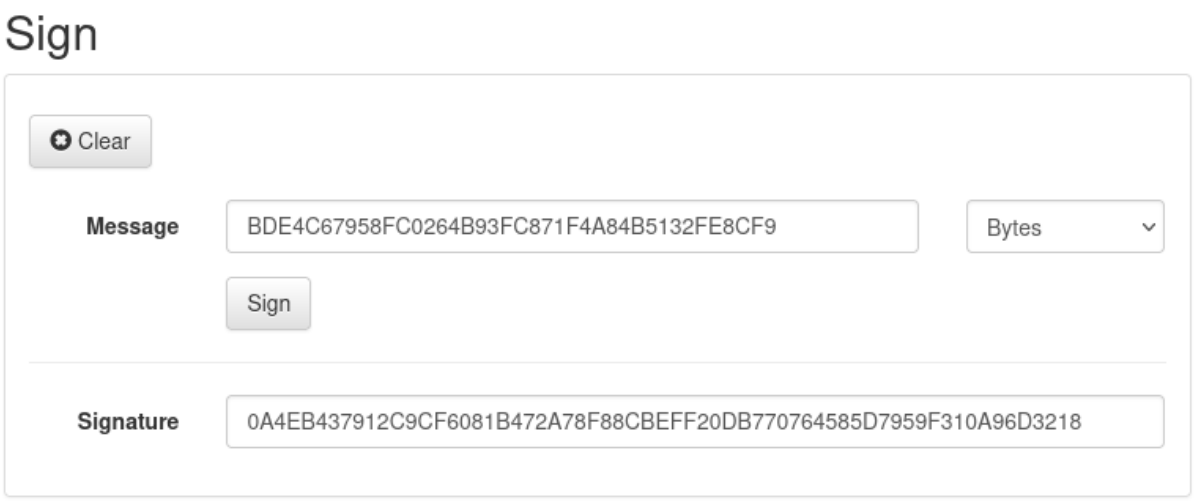

```
Press 'Decryption' button on the left side
Insert into 'Ciphertext' message that was encrypted locally
Insert here from 'Message' space: BDE4C67958FC0264B93FC871F4A84B5132FE8CF9
Are locally decoded and server decoded messages the same: True
************************************************************************************************************************************************************
************************************************************
```

дешифруємо повідомлення



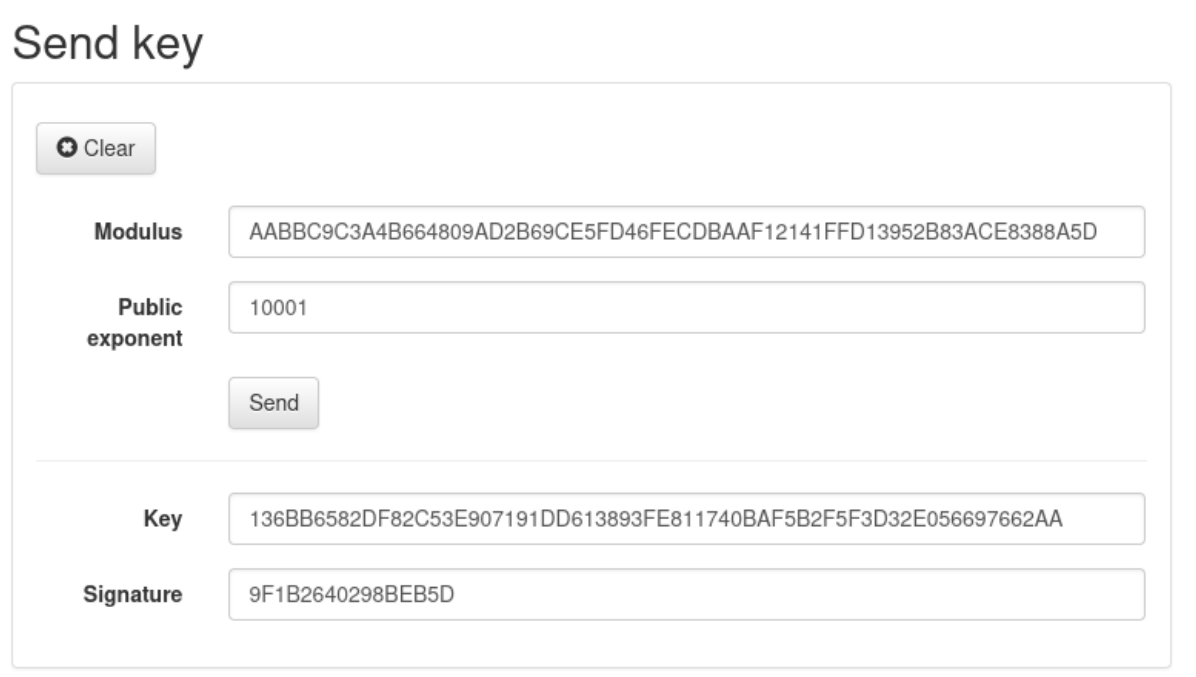

```
Press 'Signature' button on the left side
Insert this translated test message into 'Message': BDE4C67958FC0264B93FC871F4A84B5132FE8CF9
Insert from 'Signature' space here: 0A4EB437912C9CF6081B472A78F88CBEFF20DB770764585D7959F310A96D3218
Locally checked signature result: True
************************************************************************************************************************************************************
************************************************************
```

створюємо підпис




```
Press 'Verification' button on the left side
Locally generated keys:
Modulus : 123384CEE56F35E19F711A1EA5157AEA903387B9A40D0195C80503B7A02C1155573E87AD0B8223D88FEA29CA55FEED8A6302554C53C91857D7131289B0089043
Public exponent: AC6C1652A594B7F74C787BCEE40C5BEC0980C4F917258AA335558411B31F90A26E90846BAC5AA5BC5A34894CAB042BCC0115A7F20B5E3CEC0C4DDE19EDC
6667
Private key: 11356D23F082D4214D15BD5048B74B5938354A37D0B6F4CF68106557910833 5E388CEF3196C6258C954F9D81D2001FAED20EC3898B57D8C060DDEE690C8F81F
7
Translated message: BDE4C67958FC0264B93FC871F4A84B5132FE8CF9
Generating a signature...
Locally generated Signature: 57234F5A4DA698224C0DF784E3CBCF2764FC8C5B92D97327BA34013309EE187D17B8A032E895C567656BD6FA96352DAECD19CCBEA17518A
5206E63F6ADA1A51
Insert Translated message into 'Message' space
Insert Locally generated Signature into 'Signature' space
Insert Modulus into 'Modulus' space
Insert Public exponent into 'Public exponent' space
**********
**********
```

```
Press 'Send key' button on the left side
Insert this modulus into 'Modulus' space: 123384CEE56F35E19F711A1EA5157AEA903387B9A40D0195C80503B7A02C1155573E87AD0B8223D88FEA29CA55FEED8A63
02554C53C91857D7131289B0089043
Insert this public exponent into 'Public exponent': AC6C1652A594B7F74C787BCEE40C5BEC0980C4F917258AA335558411B31F90A26E90846BAC5AA5BC5A34894C
AB042BCC0115A7F20B5E3CEC0C4DDE19EDC6667
Insert here from 'Key' space: 136BB6582DF82C53E907191DD613893FE811740BAF5B2F5F3D32E056697662AA
Insert here from 'Signature' space: 9F1B2640298BEB5D
Is key received succ: False
```



Висновок:

Ми навчилися працювати із криптосистемою RSA, та створбвати цифрові підписи