Міністерство освіти і науки, молоді та спорту України

Національний технічний університет України

“Київський політехнічний інститут”

Фізико-Технічний інститут

КРИПТОГРАФІЯ
КОМП’ЮТЕРНИЙ ПРАКТИКУМ №4

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

Виконали:

Студенти 3-го курсу ФТІ
групи ФБ-93
Тішков М.С та Папуча Н.В.

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 p_1 , q_1 довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб pq ? p_1q_1 ; p і q – прості числа для побудови ключів абонента A, p_1 і q_1 – 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 , n) та секретні d і d . 111

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().

Виконання практикуму:

Alice’s private keys

```
Generate Alice's private keys:
p:
627330056782172104552475541605136992513738416338105687171652492096540977253 39
q:
799263640805100714688268810096957389783898805480944748464472905107805201456 89
d:
484706885584517961718771301049194345852651528313669379353047882695842707025279160313120416187224926254625196275154724394346221505304408532000301950178733 5
Generate Alice's public keys:
e:
225021471833423270578864322834684654878404336026340646684309881920359267694910009284574839558284459470445869240763530093998483803602278871098664049850107 9
n:
501402105170189440451831187959896252803128170125407400187684934550942362465585349921493860716248634756894259518109274435825539676615792969476743978691357 1
```

Bob’s private keys

```
Generate Bob's private keys:
p:
864290435115578319629764900618481303259752660115048785278809998791098642010 19
q:
110842691559050068609762626554489056223506912439319425835344113066963498046411
d:
402108636649369146651779187667141653932331213418198532303214730738463624657410325892739799781504067380038526599921524905064376199048765246308158695042342 1
Generate Bob's public keys:
e:
949953632706605558229068139542115277953163694644716408557759540720305513682208387434952011028703982512828194447658397593673923167088341429511699707806010 1
n:
958002781169532240191469185522430718576272249985013671350813920168567438432646947255294893480813332098817413890393159066784343162592896974157531189549280 9
```

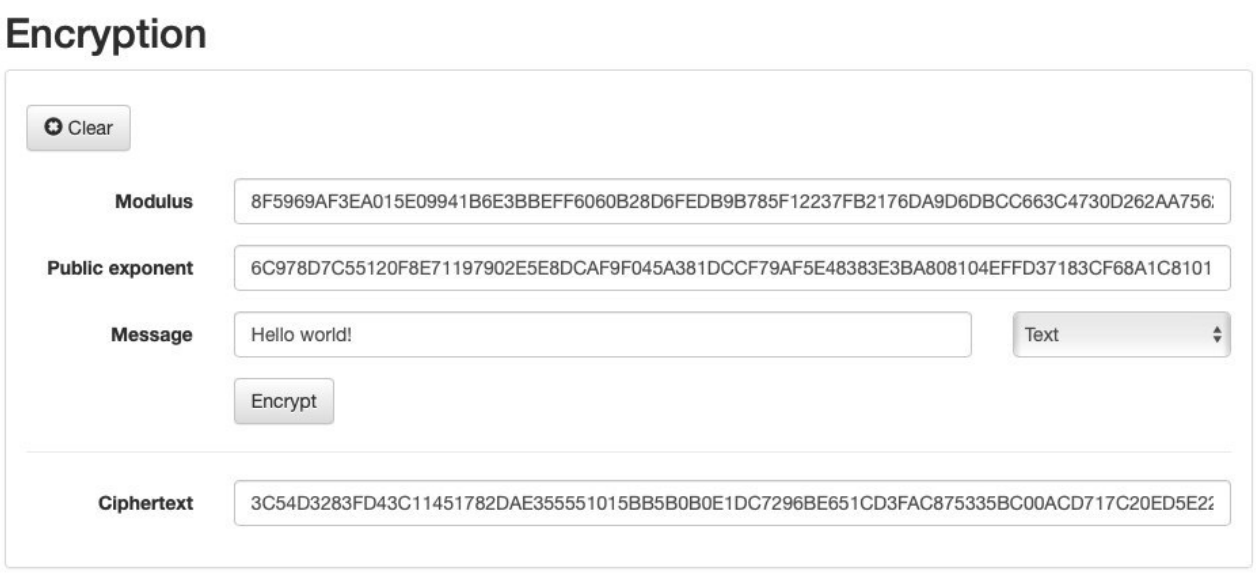
Authentication

```
Alice's key message: 20
Alice generate sign (s):
384384223812222708980201191568497956868447674394723622966126489131904492135763655814023158151101952758603634114414283938019785247683287608893095146799629 0
Alice generate k1:
115551533639129023279767148796028408255036934705880412607996887087828368806395551820793665743624009353830891008793848547755258748298613328435153071682465 6
Alice generate s1:
549447763221975220767294279122238253035419377078766350392318022037194072074049923127829863724602363968014259005784497066884445593348296253806238634132430 4
Alice return s1, k1
Bob earn s1:
549447763221975220767294279122238253035419377078766350392318022037194072074049923127829863724602363968014259005784497066884445593348296253806238634132430 4
Bob earn k1:
115551533639129023279767148796028408255036934705880412607996887087828368806395551820793665743624009353830891008793848547755258748298613328435153071682465 6
Bob generate s:
384384223812222708980201191568497956868447674394723622966126489131904492135763655814023158151101952758603634114414283938019785247683287608893095146799629 0
Bob generate key:
20
Bob return s
Alice checks (s) from other abonent
Authentication success!
Key is 20
```

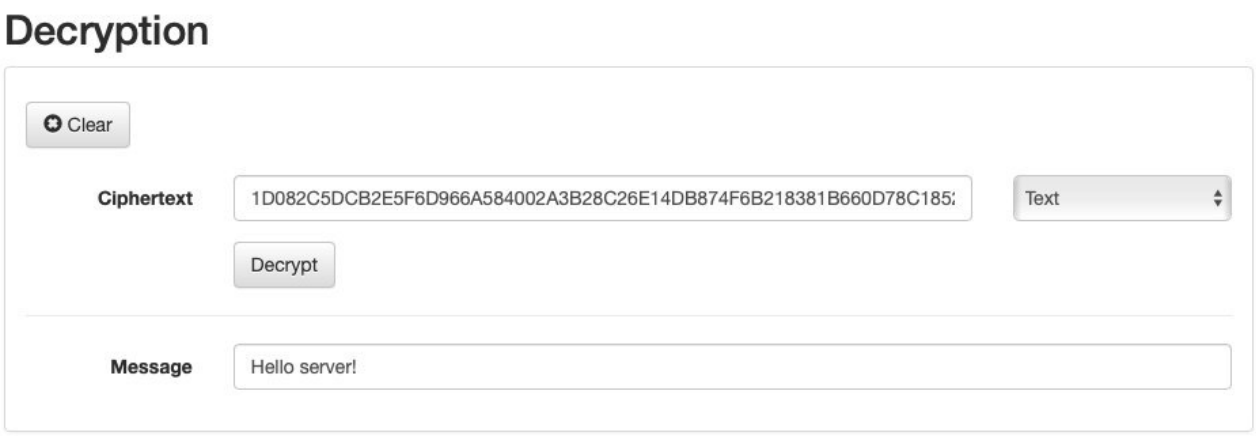
Перевірка з програмою:

```
A - e:
6C978D7C55120F8E71197902E5E8DCAF9F045A381DCCF79AF5E48383E3BA808104EFFD37183CF68A1C8101B4A8D19709883FF518FACD5E1B5327770B2CFD1E3F
A - n:
8F5969AF3EA015E09941B6E3BBEFF6060B28D6FEDB9B785F12237FB2176DA9D6DBCC663C4730D262AA756227026DE43649B48E809DD339EA92EDD85D0F68482D
Msg = Hello world!
Encrypted msg:
3C54D3283FD43C11451782DAE355551015BB5B0B0E1DC7296BE651CD3FAC875335BC00ACD717C20ED5E22FC67699F57220C9B8EC9E1997633F7CB6EDBF2579E9
Message for website = Hello server!
Encrypted msg:
1D082C5DCB2E5F6D966A584002A3B28C26E14DB874F6B218381B660D78C1852C
Generate sign:
Alice generate sign (s):
5E149FB3D16CF803AC23999B40A25A787AE2F3EF5CADBEEFA3413A60ADC83554D0EA9C027393DD596C8FE10B566DF7EB81A1A37EB97BB9D7C50E1BABEE7F2992
```

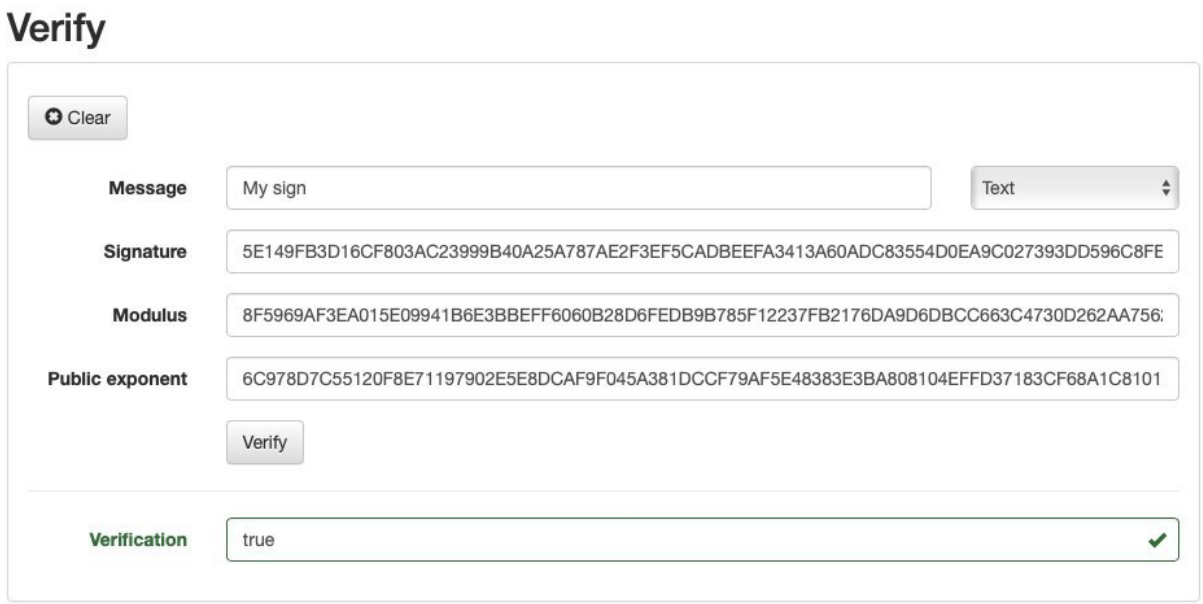
```
Msg = Hello world!
Encrypted msg:
3C54D3283FD43C11451782DAE355551015BB5B0B0E1DC7296BE651CD3FAC875335BC00ACD717C20ED5E22FC67699F57220C9B8EC9E1997633F7CB6EDBF2579E9
```



```
Message for website = Hello server!
Encrypted msg:
1D082C5DCB2E5F6D966A584002A3B28C26E14DB874F6B218381B660D78C1852C
```




```
Generate sign:
Alice generate sign (s):
5E149FB3D16CF803AC23999B40A25A787AE2F3EF5CADBEEFA3413A60ADC83554D0EA9C027393DD596C8FE10B566DF7EB81A1A37EB97BB9D7C50E1BABEE7F2992
```



Висновок: роблячи дану лабораторну роботу, ми ознайомлення з тестами перевірки чисел на простоту і методами генерації ключів для асиметричної криптосистеми типу RSA. Зрозуміли як данний протокол дає змогу безпечної комунікації.