Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”

КРИПТОГРАФІЯ КОМП’ЮТЕРНИЙ ПРАКТИКУМ №4

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

Варіант: 7

Виконали: студенти Оласюк Олександр
групи ФБ-32 та Гарбар Дар’я
групи ФБ-33

Київ 2025

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.

Постановка 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 p1, q1 довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб pq <p1q1; p і q –прості числа для побудови ключів абонента A, p1і 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) , (e1,n1) 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 $0 \ k \ n$.

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 а) зашифрувати власною реалізацією повідомлення для серверу та розшифрувати його на сервері, б) зашифрувати на сервері повідомлення для вашої реалізації та розшифрувати його локально.

Хід роботи:

Генерація пар ключів:

```
A keys:
pub_e=65537
pub_n=586191638480351374651588809021263047253060253322074698600807755012975565483854839954626349985555637758910292940038516129554630264134862141420861048877922
priv_d=221428509120337803544177068772302485585192481060184037086860808140458414317376864817884876927108905062341192262386182064621853303717550494256469361104416
priv_p=665261120305618131292522508127621233622366584078312219590028126964600370380 83
priv_q=881145193350631143546131914753679152507425523418968654483087202364061801705 69
B keys:
pub_e=65537
pub_n=608839051552391818340427456077251088545052559495550517765674882531444199644330220167514364849070297960848824790110455567217659011110860805464667008056923 9
priv_d=531666980794717240087624751076507618557965088117099594606551620111914667220716630068163328859428299692906263573063441494440784423414539239825619354915171 3
priv_p=961798345682953831190193389117095496418249679231268627823822054039519072099 77
priv_q=633021520867835705662288395577028127839319586365335336104023020731190709522 07
```

Шифрування та дешифрування повідомлення:

```
----------Encrypted by A and decrypted by B:-----------
message = 1234567890987654321
Encrypted by A: 37947205850602470672079254402857261512934126374681714234802583965710931094387338708865501683356134058830620230537310819830528418148065875980102917
Decrypted by B: 1234567890987654321
Successful? True

----------Encrypted by B and decrypted by A:-----------
message = 9876543212345678900
Encrypted by B: 14583046103451662842630323232763403261031279445003486295814498659733442157266406923905684428773238259399425010342525854147276286596333403011589664
Decrypted by A: 9876543212345678900
Successful? True
```

Цифровий підпис та верифікація цифрового підпису:

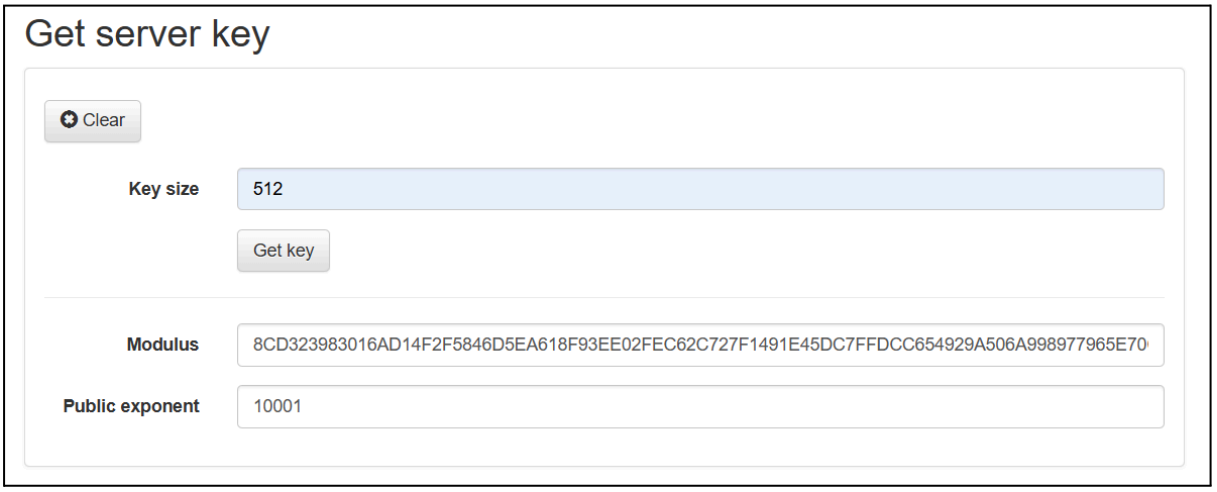
```
----------Signed by A and verified by B:-----------
message = baobab
Signed by A: 45769382560036961557753400773041940887904269102036661959703935754102534157039344395791823119737484844985737917420893816643901775267566543092078054
Verified by B: True
Successful? True

----------Signed by B and verified by A:-----------
message = pytsia
Signed by B: 41423299556821940594717597535768496204406307767999239534081121745526318333323562298000238479953217664836194016015448810886434412561172965651413560
Verified by A: True
Successful? True
```

Протокол конфіденційного розсилання ключів:

```
----------A send key / B receive key:-----------
key = 54004652910015626861835615402324504168891657880186309501976987489979639872395
Sent by A:
Signature: 490921173354993407703009663909532525157218857678948629416532221787576707340111021307100550250290629702800434309551582104715795187645423545654425978
Encrypted key: 47980973597590423170383216529968202403441221542452447657669811739389240824612519438918597501514249291509799650292046127024971403731124007137955
Received by B:
Received key: 54004652910015626861835615402324504168891657880186309501976987489979639872395
Successful? True
```

Перевірка сайтом:

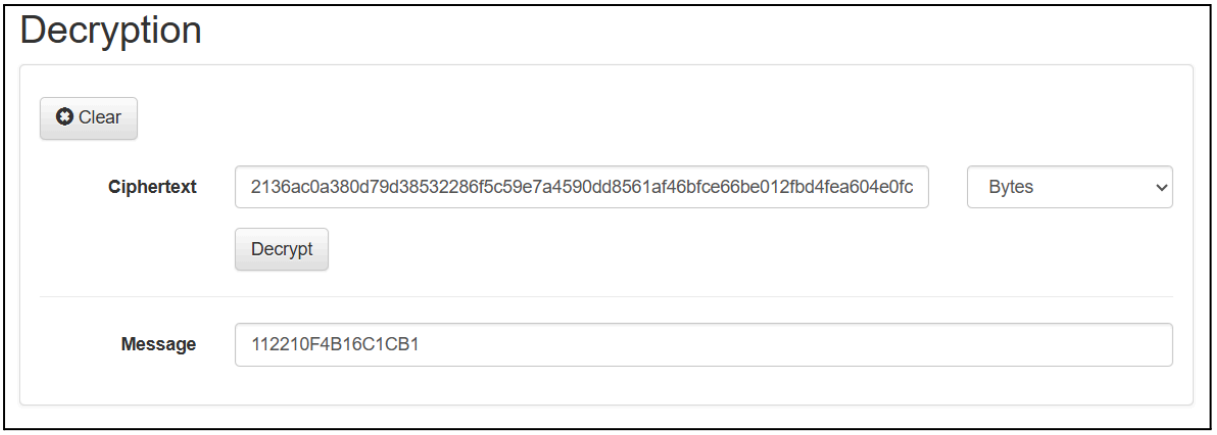
Генерація своїх ключів та ключів сайту:




```
My keys:
pub=(10001, 6842d681ccbd6c002b3ff5f0853943fb53b151b231e120d67f5ba6dd30a8b9586c1868a3a71d1271cf5ab383f395c346074e09f37862a01c0bcddd65da20d34b)
priv=(160217591707654081855473988777869039966801318621791231362744839579202836951173204623191935865472288660670432690351903241061189654596677822952480056825279
Site keys:
pub=(65537, 737559124628818664796379313891963610185492234031831644416514808493478767576548469352794026291392069102383080669884098464961362338044969239748064222315
```

Шифрування:

```
------------------Encrypt------------------
Message = 1234567890987654321
Encrypted: 2136ac0a380d79d38532286f5c59e7a4590dd8561af46bfce66be012fbd4fea604e0fc0d36efcebaf69b97e295534012ebb542e95736fa49b805b7d218570f69
Site decrypted: 1234567890987654321
Encrypt good? : True
```

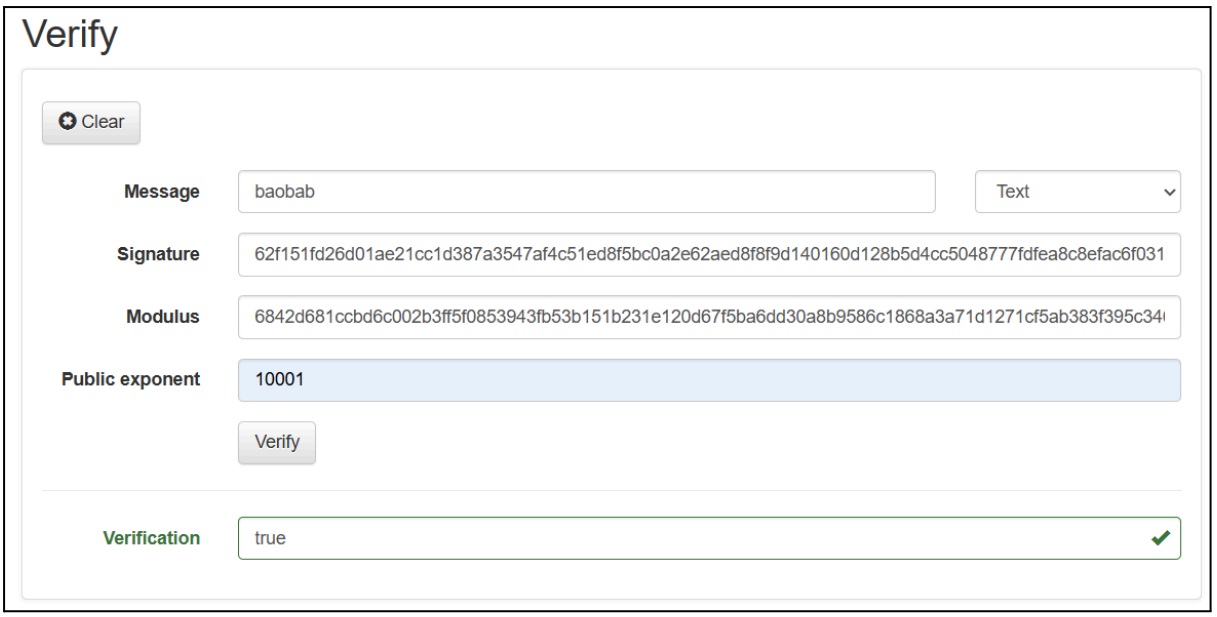


Розшифрування:


```
------------------Decrypt------------------
Message: 9876543212345678900
Message bytes: 891087b934a8dc34
Encrypted message: 380929403757534928948802940854386114282248649337992849286042655133656955193608852
Decrypted message: 9876543212345678900
Decrypt good? True
```

Цифровий підпис:

```
------------------Sign--------------------
Message = baobab
Signature: 62f151fd26d01ae21cc1d387a3547af4c51ed8f5bc0a2e62aed8f8f9d140160d128b5d4cc5048777fdfea8c8efac6f0317c773e708243b1bb0d49b9bb4170365
```

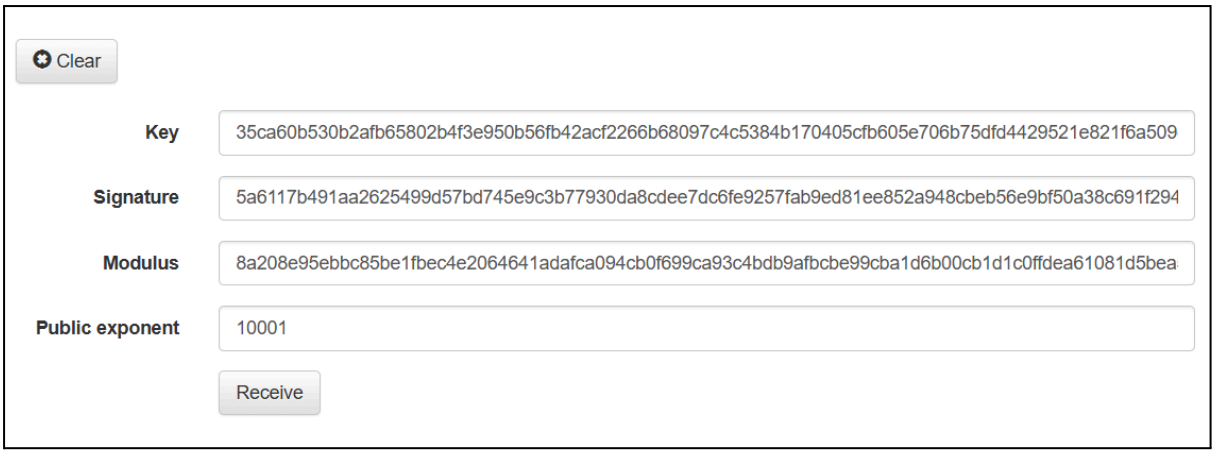


Верифікація цифрового підпису:


```
------------------Verify------------------
Message: pytsia
Signature: 28713352418080638828086182763459606875883157113912213923144164821729928815913866799034322599012073555
Verify good? True
```

Надіслати ключ:

```
------------------SendKey-----------------
Key: 123123123123
Sent key: 35ca60b530b2afb65802b4f3e950b56fb42acf2266b68097c4c5384b170405cfb605e706b75dfd4429521e821f6a509ccbf01e22c90a22d3db94cbdcaf0b2d07
Signature: 5a6117b491aa2625499d57bd745e9c3b77930da8cdee7dc6fe9257fab9ed81ee852a948cbeb56e9bf50a38c691f294ecc96a10123b534bd143465f1e499241ce
SendKey good? True
```



Отримати ключ:


```
------------------ReceiveKey---------------
Encrypted key: 406879357299738796742582412549275519386869165524320910926970400186878741117542492526575668518895105180712
Signature: 730281192396831958394424366589734796423672033316323788965681186041532032451591168218680577684195537820005887
Received key: 7269217576318821932
ReceiveKey good? True
```

Висновки:

У ході роботи в навчальних цілях було реалізовано криптосистему RSA. Отримані результати підтвердили коректність основних етапів алгоритму: генерації ключових пар, шифрування, розшифрування та підписання повідомлень. Проведені експерименти продемонстрували працездатність нашої реалізації та її відповідність теоретичним властивостям RSA.

Застосування створеної системи дало змогу краще зрозуміти принципи роботи сучасних криптографічних протоколів і оцінити вплив вибору параметрів на рівень безпеки та продуктивність.