

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
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 курсу
Групи ФБ-04
Осіпчук Антон
Подима Катерина

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 $pq \leq p_1q_1$; p і q – прості числа для побудови ключів абонента А, p_1 і q_1 – абонента В.
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 А і В – 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 А і В. 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 М і знайти криптограму для абонентів А і В, перевірити правильність розшифрування. Скласти для А і В 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

Ми написали функцію для пошуку простого числа за допомогою тесту Міллера-Рабіна. Згенерували пари ключів, реалізували функції шифрування та дешифрування повідомлень і їх верифікації.

```
--1 py main.py
Alice keys: {'public': [6182646950349545423711835293709161301031043786131991965325152185982576743585448715297139194023236389903816167888344032328283036695573983728630901624220519, 2565057299495413510095886716944662817759434331550160958295, 23044541714098179779400171794081476772251627105063896479247441209518047944280052756527192785851], 'private': [545215131670182283931327550671559865392001155814455640744443002757293204384809563456536158788268187141641863244832983860523679, 36305318122107273723484851, 8099853998114217890084754413958597751500337649191935563680046640254504968329, 7633835054453881745457891606306678249345482814436151835788341286859753984111]}

Bob keys: {'public': [9761833110977813217081505291405662783977885301907138138615032710918906079412235117493142391095297786088724925328281685774478865583992554666507478727166461, 831908479521207910293651765416525802675515661349660332791167, 6705609484199784038501174842938217930454166244649038388379851521209312459097360807079967457919], 'private': [55749519777214845357506254190239800611838828859213650302718281548177948246724776485870770276594882131014775954258418681051270372, 3578402010632543896367679, 110372522473004872696938267879220668307119227575789677885376853486201164555001, 88437166173901532206702166948282571259566184114327467071876849668155262311461]}

Alice encrypt message "1488148814881488" using Bob public key
Ciphertext: 3003913917969540232573473893604823904332226553799829836966708224397289141847838809794686076453432255006404857772948211786477861152829869804884114509884210

Alice create sign for ciphertext using private key
Sign: 1641615435355687709729257191826386374168821629661377996926916464772514317759109159725166342614319155310896547092783975929310272954860931652313642241598370

Bob verify sign using Alice public key
Sign: True

Bob decrypt Alice ciphertext using private key
Plaintext: 1488148814881488
```

Висновок

Під час виконання роботи ми ознайомились з принципами роботи RSA. Використали тест Міллера-Рабіна для пошуку простих чисел.