



Міністерство освіти і науки України
НТУУ «Київський політехнічний інститут»
Фізико-технічний інститут

Лабораторна робота № 4

з предмету «Криптографія»

на тему: «Вивчення криптосистеми RSA та алгоритму електронного
підпису; ознайомлення з методами генерації параметрів для
асиметричних криптосистем»

Варіант №15

Виконала:

Студентка III
курсу

ФТІ групи ФБ-84

Матвієнко В.С.

Перевірив:

Чорний О. М.

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 $pq \leq p_1 q_1$; p і q – прості числа для побудови ключів абонента А, p_1, 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 и 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 $0 < k < n$. 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()`.

Опис кроків протоколу:

1. Програма генерує 2 пари простих чисел. Для абонента Alice p, q та для Bob p_1, q_1 довжиною 256 біт($p \cdot q \leq p_1 \cdot q_1$)
2. Програма генерує ключові пари RSA для Alice і Bob, де (n, e) -open key, (d, p, q) -secret key.
3. Абонент Alice формує повідомлення, використовуючи функцію SendKey(), де ще використовуються функції Encrypt(), Sign().

$k_1 = 0x9b82c1937813fae77e9dea1b1af66ff23e949de5b9b3d849440356abe48a70ba3bc209fff23ade53cc26605ec58294bf34b8927b028cec3fab88e74e3d483a10$
 $S = 0x5f04e97213a60e061f12a43a724bf12d6d1300265282438c0c8ed684048595a2ee082c73f949f1df973bb22a9a6caf08ce8035570865e04624442fee8c2c6a71$
 $S_1 = 0x766908bd9f0066fcee3e037766511060ed264e38f26847443286603b2ba052339ca68920deea792e929fa7fd50b3c1ebf9b089aeb45fd5ade9dad4609bae7bf$

4. Абонент Bob приймає повідомлення і за допомогою свого таємного ключа перевіряє підпис Alice.

$k = 0x109f95664f4b50b1330d7ccbfcd2b02d1fb407f989406ad8373b0dac022d1d4bf653b5400e05a8b3faa244ea1d1419d7b579b7cfda1dd123bf50b416dee8e8a$
 $S = 0x5f04e97213a60e061f12a43a724bf12d6d1300265282438c0c8ed684048595a2ee082c73f949f1df973bb22a9a6caf08ce8035570865e04624442fee8c2c6a71$
 $S^e \bmod(n) = 0x109f95664f4b50b1330d7ccbfcd2b02d1fb407f989406ad8373b0dac022d1d4bf653b5400e05a8b3faa244ea1d1419d7b579b7cfda1dd123bf50b416dee8e8a$
 $S^e \bmod(n) = k$, отриманий підпис правильний.

Хід роботи:

Таблиці вибраних чисел p, q та кандидатів, що не пройшли тест перевірки простоти

Alice	
$p = 0xb34313196979640425de085073dc54deef7e7b2ae503ff956995d310b91478610x66e58c407986e08701280c8b0862c53c559994c299fd26fdadf42eb5a82de64f$ is not prime number $0x7b29e0db2deee9a74fe635c03e838d715c7ba6260e7efbeffbb72245f16c1ff$ is not prime number $0x848ff0e571b83203a25335c29c29a7e9098abc0672f173c61a286f768a4f0d5$ is not prime number $0xcb56dba19e0de535874ee41bac209791e4efcdd2607e69ce075cc20caea3f$ is not prime number $0xf1d4c8bb900d7cfc89ebd29bce54ef9a1e06adee1193a95b9b45afede8ae1b7$ is not prime number $0x338735679812af0de62f8eccbc7a5351c30280be3d63230db6682cbf7de4861$ is not prime number	$q = 0xabe62b2ead5987636852146855947b5b740f8cb36e77d56e70df5e9a8daa1f70xb5107a6bcc8b4e80959c217d76776dbfe503847ca7f91fed3963f455de4143f$ is not prime number $0x6a7eabdda3d69f920469283093bb68636af904b070d8fb4a9890113ebc3e5$ is not prime number $0xa9794fc97deef20e65444bdca66fe61d1914aa28fba873bbe91792da8800e581$ is not prime number $0x6031d1751fa6824cd2788a6c5088c099df26ad74a1fe54d5faf07b14859ea235$ is not prime number $0x7f40f8c67b4710525964716fb3382446a9192e85baabeed356f891a00a59b4a3$ is not prime number $0xda6ebb3f68c327b8ed6cbeea8cb2f7ccbcf84b871d21b48abeb5d543363c949b$ is not prime number.

Bob	
$p = 0xdb6858d928f8bb43f783cb6552f05b27d878b3d8d6044830b85b9494f81ee20b0xb402f0a9660de9b848c09b493e7853b5f4f549f8e5d84c4fd6b46b8bd9c7a81f$ is not prime number $0x443302e7b75f60a0d2632c60e24ea33ac92f105575b571a5cdf4b962b6c0d661$ is not prime number $0xb894c5ebf2a1f64dc6d19b08405e0f481e7e0dbfa0e269f0ce382f700801ac03$ is not prime number $0x7ce5491a5c41ae5cef1ecf5db1d38a29de1cef9a907832817244fda623035005$ is not prime number $0xe5e17feb3963acae35550e479e1ec376392f252620954639b38b1b83531d5b33$ is not prime number $0xed5ccac0c1f9ff632feee9847e4e35425e4e90cd918d030816dd0e7425999c17$ is not prime number.	$q = 0xabc6a8dd6d0d07777cec8f18cae440b1d76ba0a99c1811218961998a8d7cb42910x35a71c8f8dc61028acb13f421e54c0da2cf796ee84b184ea66330c1ca5cb5e93b$ is not prime number $0xbe6c43051e6e8fefd80779cc0bf4aaee75afc2c9a785e8f22b7c85d6cdb08989$ is not prime number $0x6f00713a6308336f2c819a0645528d67df2a018646b3bb0c99632ed600869199$ is not prime number $0xd5e9a75f8c7c1a48a7c594404a105cfcf98245f4dca900729427ac8ed50e107f$ is not prime number $0xdb29240f1d787f182f185147b5aa927b69fd933490e2d0c381fab379d783599$ is not prime number $0xae3aec330782019818d1e4f5dd7fae7557a640a5fec0966399f30a65b2817311$ is not prime number

Параметри криптосистеми RSA для абонентів Alice і Bob

Alice		Bob
Open key		
n	0x7862a171782bcb501cc82eb0b5d766237c5634fced632d6902d3c2b0491c3a65e2f6da61264cec93f05fc2ac4f5e9c115b09726ab98494417dc2993f9d0f2697	0xa17bf404d922f4f984be115c21bffd01ea372fa8192f4f9f7a5efcd4a4d9a4f433605872e75d7e88f42ca77d6e67484be4f486b4615cd1ba63e1fe4fb7dde3b
e	0x50ebb13e701f9a8ff567d7f75ce2e1f5648857e3a241273cb0f1fa844c5aef907d8311605dbec41d7332257745611a052d8ae754d707506441589b261c60b36f	0x17c80be95940b64a7ee818a057bf0677f3cc2fb9d12c0d3c0d9664eca8e746e2d3d982afe346fdc6b08ea4a9cb816c2ec09d42523684a7f25fef41b84e244f55
Secret key		
d	0x44aee938daa81a74b91ed7ed505d5f7b884948b4c4b10279d596dcdbd778f62aa00a0e6d2563e8ac3adbffe61c0606121a5ff765c28004b18ace867cc678b80f	0x6dffddd3fb30c189fe3c718a453e3fc7b516a58e9d0b4af46538e0357fca171fd91fa6ec27ae2eacab5d4568d8e4f16d17693ac70ba85fec69e30e6250095dfd
p	0xb34313196979640425de085073dc54deef7e7b2ae503ff956995d310b9147861	0xdb6858d928f8bb43f783cb6552f05b27d878b3d8d6044830b85b9494f81ee20b
q	0xabeb62b2ead5987636852146855947b5b740f8cb36e77d56e70df5e9a8daa1f7	0xbc6a8dd6d0d07777cec8f18cae440b1d76ba0a99c1811218961998a8d7cb4291

Чисельні значення прикладів ВТ, ШТ

ВТ	ШТ
0x109f95664f4b50b1330d7ccbfcd2b02d1fb407f989406ad8373b0dac022d1d4bf653b5400e05a8b3faa244ea1d1419d7b579b7cfda1dd123bf50b416dee8e8a	0x9b82c1937813fae77e9dea1b1af66ff23e949de5b9b3d849440356abe48a70ba3bc209fff23ade53cc26605ec58294bf34b8927b028cec3fab88e74e3d483a10

Цифровий підпис для Alice і Bob

Alice	Bob
0x5f04e97213a60e061f12a43a724bf12d6d1300265282438c0c8ed684048595a2ee082c73f949f1df973bb22a9a6caf08ce8035570865e04624442fee8c2c6a71	0x766908bd9f0066fcee3e037766511060ed264e38f26847443286603b2ba052339ca68920deea792e929fa7fd50b3c1ebf9b089aeb45fd5ade9dad4609bae7bf

Обмін ключами з сайтом

На сайті було згенеровано 512 бітне випадкове число-значення n-для Website, а також значення e.

Alice		Bob
Open key		
n	0x7f116c387731b412c4ec52f80ed37a696ca47f8830f5fb3d3a0b58a678c1e68e03cbb32b61e1235d6b79778cf82bb7e4ed296e362af3fcc709f0be4310d1368f	0x8B856D59DE4C42D743D2A1F90915A2B57BB93FB6A18C642F954F0011296E322185D566CBB6083E6E5CDC10143EF991960E7D689D4B4DD77C7C84878433575445
e	0xfa74848502db5fd417d9d153c40c8b633b1eae26647ddadc3399c00915df7ed7a05631d19c227ec08992172deaf16984418ca259e14294b8617a50bc4386427	0x10001
Secret key		
d	0xd40c351292618a9035bb41d2aa7e615778dc25bd459aa154fa1cac24b557d62fd78de911ae3cf36e6a72d627fd3311546a1253c527b4b0c92ab1448a5e63297	
p	0x953fa7873e14e1bda27ae479fb1b612e94d17bc464eb4133cc71a60053e574f1	
q	0xd9f45d8e697ca42837f3a9b2159d2769efdbf8c8d26323954c6d3aa105f3637f	

SendKey() повернуло значення ключів від Alice:

$k_1=7a7c3323220f74da1fade97587dff135d2d662bcfbefddf3c106f910e98c422f0fa7344d2955d74fcc556ebfcfc152fe42a8071a569876ebc76813468112aeaa$

$S_1=6d0c825f28fbe4e8dc238377b48c53eb6ab69e54f1351e139672ba994f67dd864bb121c8f7e9a5fdc72ba64f3ad8e83209e0366df60ad308522279b9cec3d706$

$n=7f116c387731b412c4ec52f80ed37a696ca47f8830f5fb3d3a0b58a678c1e68e03cbb32b61e1235d6b79778cf82bb7e4ed296e362af3fcc709f0be4310d1368f$

$e=fa74848502db5fd417d9d153c40c8b633b1eae26647ddadc3399c00915df7ed7a05631d19c227ec08992172deaf16984418ca259e14294b8617a50bc4386427$

ReceiveKey():

Receive key

Key	7a7c3323220f74da1fade97587dff135d2d662bcfbefddf3c106f910e98c422f0fa7344d2955d74fcc556ebfcfc152fe42a
Signature	6d0c825f28fbe4e8dc238377b48c53eb6ab69e54f1351e139672ba994f67dd864bb121c8f7e9a5fdc72ba64f3ad8e83
Modulus	7f116c387731b412c4ec52f80ed37a696ca47f8830f5fb3d3a0b58a678c1e68e03cbb32b61e1235d6b79778cf82bb7e
Public exponent	fa74848502db5fd417d9d153c40c8b633b1eae26647ddadc3399c00915df7ed7a05631d19c227ec08992172deaf16984418ca259e14294b8617a50bc4386427

Key	3D3B8F207DB7AC080F3A03DE79CEBFE4A462461486FAA0E90303F2E92A0405F57584A91148ED55CF4CD7C
Verification	true <input checked="" type="checkbox"/>

Висновки:

Під час виконання лабораторної роботи №4 я ознайомила з асиметричною криптосистемою RSA, генерацією ключів для цієї криптосистеми, а також з тестами перевірки чисел на простоту. У ході роботи реалізувала функції шифрування, розшифрування, підпису, перевірки підпису для RSA. RSA-алгоритм з відкритим ключем, що часто використовується в криптографічних застосунках. Складність задачі цього алгоритму полягає у великих цілих простих числах.