



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 III курсу ФТІ
групи ФБ-82
Сумовська Юлія та Руднік Анатолій
Перевірили:
Завадська Л.О.
Савчук М.М.
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 $pq \leq p_1q_1$; 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, 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 $0 < k < n$.

Опис роботи та основні труднощі:

Програма має такі функції:

- gcd - пошук найбільшого загального дільника;
- miller_rabin - тест Міллера-Рабіна з попередніми розподілом;
- choose_random_prime - функція для вибору випадкового простого числа з певного інтервалу;
- findModInverse - пошук зворотного по модулю;
- good_random - генерує хороші прості числа для кращої стійкості;
- gen_p_q - генерує q, p, q_1, p_1 , такі що $q * p \leq q_1 * p_1$;
- GenerateKeyPair - генерує відкритий і закритий ключ користувачам А і В;
- Encrypt — шифрування;
- Decrypt — розшифрування;
- Sign - цифровий підпис;

- Verify - перевірка цифрового підпису;
- SendKey - відправити ключ з підтвердженням справжності;
- ReceiveKey - отримати ключ і перевірити справжність відправника.

Труднощі виникли на моменті зашифрування, коли потрібно перетворювати рядок в число.
Проблема була вирішена взяттям значення ASCII кожного символу і конкатенацією цих значень.

Значення вибраних чисел p , q , q_1 , p_1 із зазначенням кандидатів, що не пройшли тест перевірки простоти і параметрів криптосистеми RSA для абонентів А і В:

- p користувача А:
0x2c33f23718c8f8f1908baef04604d800a29c0b5dc30ce6cd93262f1fc7b7b48ca0c32ec112e7e3950a30efdd49b20cebdf92f07a31f07d8fb659df050c0b9666f6103
- q користувача А:
0x3a5f81de55a38151223d908d3a089537160205aa4a2baa0948c1da8e7606994ab33dd77be3b403f5154c9063009c62bcfdf412bc6244aaf982800ddd380c253504d13
- n користувача А:
0xa144293eef3051393d455d68e72042da5762a1a13a871ad3915708adc04384fdcc4890a13f36c52d1c49d7983369db1620f1e0dcdd16980980051202eb5c3c0654d941d96704cf9c060c2ffd4d1204d2895fea151747702660ced8c1ed22ca532a4bba81e6881633eca5e8d49c192155ce253bfa1b2932174203b1ffd537e351a64621a39
- e користувача А: 0x10001
- d користувача А:
0x8b1d789340d5268c412dd059c9600c2bbc4c6a6fe2446040f854e338b7edc71105a3e7982227064b56e6e063fc9e298fda55ae0bf9711f30bace24ee7dc8704369f834980fecaea2cc6cc16420fee8d5fce7335ca0f817485d2a6966310ffea1ffd72e01a8fb4542e7cd438fb0870b5fe1035a9ae0a2bbce2afa7afdd781dd3ca211ac23d
- p користувача В:
0x272623c159e49e78e7a4f4ada10664abad563e4ab72b1a55b70d2e10e1daaed53c2e3405c319efca0d6ee99ee891ceed1f216ce844fc69674ca1af891392f77e35b67
- q користувача В:
0x4344e38e6fa674ef2892f578e20bd235ce76194872e7a66dca74b640bdfa942eb29af8f8e91b510d7abe0183c541b37627f438306065d6dac96ed691b204db3073f1b
- n користувача В:
0xa498449b3d1fc473270a8667bc9dc318e804e625bdd51d1ae26fb6602413d9b8475fcbf3a1f449ecd9a3a5a14a2675e0e6d4620e45bfc64bf89b60c685dd205da233b7729f65c9235b219105288da3124e4a7c36faf95fdc0448c6f22353dd4a3bc0dd73f9d02c492a47377d232548a708e80b4062f72bab7e2d84b0f1330549e1d49fcdd
- e користувача В: 0x10001

- d користувача В:
0xa388f2a9772ac80a4da10477af8ce41a2bf52a2217a1d6002f3925701288728ac8c7a2a197a95b85eb9bcfe618016c9f9f59276fdcd73513a897e0a87eba7cfceb3aa36f47299c41560a36b2362d4a3b2d0753b5a051de60f1284a1cc34ddb780be7d40e79e031ac1704b3a0357649149f93ec4037541002577642fd3a4ae68668cca3eb5

Чисельні значення прикладів ВТ, ШТ, цифрового підпису для А і В:

- Чисельне значення ВТ:
0x61747461636b206174206461776e
- Чисельне значення ШТ:
0x138093801a459194ab70982a7979a8c4b20ccc5a2c4d905ffbb352a703171330d8d1b290af67b8d5c5aed40533e5ee0933a61155291952cf2db23103f7cb09b294115f0532b606543050c16e1d3b8dc7994bc88fca9ddcd46804e34b5fb3fea46b2db7f82ce80fdfe7aff899bc0bd86e30275de4a98c19d3932066b562d20ad82333049b1
- Підпис користувача А:
(attack at dawn,
0x2805625a752e4f367b0acddb14d752919a331d7cc8a8f2a934a9cf1770919fcc557e007ca60a2cc80ebb0bb2a6fb0f91bf71ea57c1a62864f37e3150aac524115295206ecc5f1d1183721b58023dd7cd08e29fb36ffa7c1f557c9cc3a13645754793f320967b763e77b27da314009d99c4301e7f9a41767729cdda2a326df847f51137c0)
- Підпис користувача В:
(meet near the fountain,
0x35ceb38cf257c71daad294cea518fc6dd43d7d0927d8b548ea3a191aaeeb768e8e4bd510086baa97d3f24967892ef52463afb0d83abb960e5d68cd37370350e7cc24ea678dcd1045d7f1ab17d32c990474e7c62a8ae0ccff79df041f463b9a96bfabb326beb6a1918ff6fcc48a947fd8fb8746d3378110edbddee8fbc0eac441ba20752d)

Опис кроків протоколу конфіденційного розсилання ключів з підтвердженням справжності, чисельні значення характеристик на кожному кроці:

Процедура SendKey приймає 2 параметри: одержувач і відправник,

Спершу бере n, e відправника:

n=0xa144293eef3051393d455d68e72042da5762a1a13a871ad3915708adc04384fdcc4890a13f36c52d1c49d7983369db1620f1e0dcdd16980980051202eb5c3c0654d941d96704cf9c060c2ffd4d1204d2895fea151747702660ced8c1ed22ca532a4bba81e6881633eca5e8d49c192155ce253bfa1b2932174203b1ffd537e351a64621a39

e=0x10001

Потім генерує k:

k=0x164d039f14ae0283a625d16ceed70a88a86bf9f4ddac8db5f02cb07b3a43589e2dae2ef66ab79f68378719754d0c0e8ded4bf685f673f075afdd836408912f41f618c76df0d4aaae83f389941c4fd4863179092b5a6a97279d8f00845fcf9806f922fe94cd5252a2f4eaddc180ce2180203d93fbab7ceab45e786436d1909209548e9cb2

Потім бере n , e одержувача і зберігає в змінні n_1 , e_1 :

$n_1 = 0xa498449b3d1fc473270a8667bc9dc318e804e625bdd51d1ae26fb6602413d9b8475fcbf3a1f449ecd9a3a5a14a2675e0e6d4620e45bfc64bf89b60c685dd205da233b7729f65c9235b219105288da3124e4a7c36faf95fdc0448c6f22353dd4a3bc0dd73f9d02c492a47377d232548a708e80b4062f72bab7e2d84b0f1330549e1d49fcdd$

$e_1 = 0x10001$

Підносить $(k^{e_1} \bmod n_1)$ і зберігає в змінну k_1 :

$k_1 = 0x980c0120f11eae0f2dbf13967f442ab165fa0b5895beef4f21941c0bc9002cd06c502f1acae7f58b77a3bfd06ef61ada51d60b62e9f2fe663d2dbe3f1858be646d7401a0e15fe91eddf3d5ba9bcb2eda44fced38d4a7a3da96634ec471be8e91d2fdb9ee35583b6616c55cd1949f088682f95796fc9cefcfb588adaa5009c3e7006da1eb8f$

Бере закритий ключ відправника і зберігає в змінну d :

$d = 0x8b1d789340d5268c412dd059c9600c2bbc4c6a6fe2446040f854e338b7edc71105a3e7982227064b56e6e063fc9e298fda55ae0bf9711f30bace24ee7dc8704369f834980fecaea2cc6cc16420fee8d5fce7335ca0f817485d2a6966310ffea1ffd72e01a8fb4542e7cd438fb0870b5fe1035a9ae0a2bbce2afa7afdd781dd3ca211ac23d$

Підносить $(k^d \bmod n)$ і зберігає в змінну s :

$s = 0x1579ad0a4739bde119135a85d3ac8e1f22d5ceb67b33e4fc279ff8d5060962f8551c89ebf141d75aaad993aa62aeeb0cd561eb7eec1fae19bd4ff967651c36281d58f00b06cd5d0ad03b4737bbf2ec68035cef00a1b7e40ec3dcc5c1453855c892cddf44685351dc9d185fcce57b172a6b42716bb21b17e697b2b0d6bad228ee0df2dcae$

Підносить $(s^{e_1} \bmod n_1)$ і зберігає в змінну s_1 :

$s_1 = 0x5a13ee178411611e7dc515cc5f353fdedd99c3aa16383e582b97d9637c14f8c4d46e5f307dc8fe5dbd6db5e1c0b93fbf7b70953bbfe9f1243457d8fa6375ed14f7d40f47662fbb141e5b3aba5777e7a4bc20e1a98d6b3dd7e084996874ceeb7cd8cf5b9a96e4016f3bba711c7f9a5826962a26b97b83c21760dd78c646a479dc8b483c6e8$

Відправляє k_1 і s_1 одержувачу.

Процедура ReceiveKey також приймає два параметри: одержувач і відправник.

Бере k_1 і s_1

$k_1 = 0x980c0120f11eae0f2dbf13967f442ab165fa0b5895beef4f21941c0bc9002cd06c502f1acae7f58b77a3bfd06ef61ada51d60b62e9f2fe663d2dbe3f1858be646d7401a0e15fe91eddf3d5ba9bcb2eda44fced38d4a7a3da96634ec471be8e91d2fdb9ee35583b6616c55cd1949f088682f95796fc9cefcfb588adaa5009c3e7006da1eb8f$

$s_1 = 0x5a13ee178411611e7dc515cc5f353fdedd99c3aa16383e582b97d9637c14f8c4d46e5f307dc8fe5dbd6db5e1c0b93fbf7b70953bbfe9f1243457d8fa6375ed14f7d40f47662fbb141e5b3aba5777e7a4bc20e1a98d6b3dd7e084996874ceeb7cd8cf5b9a96e4016f3bba711c7f9a5826962a26b97b83c21760dd78c646a479dc8b483c6e8$, котрі відправив відправник, бере n одержувача та зберігає в змінну n_1 :

$n_1 = 0xa498449b3d1fc473270a8667bc9dc318e804e625bdd51d1ae26fb6602413d9b8475fcbf3a1f449ecd9a3a5a14a2675e0e6d4620e45bfc64bf89b60c685dd205da233b7729f65c9235b219105288da3124e4a7c36faf95fdc0448c6f22353dd4a3bc0dd73f9d02c492a47377d232548a708e80b4062f72bab7e2d84b0f1330549e1d49fcdd$

Бере закритий ключ одержувача та зберігає у змінну d_1 :

$d_1=0xa388f2a9772ac80a4da10477af8ce41a2bf52a2217a1d6002f3925701288728ac8c7a2a197a95b85eb9b$
 $cfe618016c9f9f59276fdcd73513a897e0a87eba7cfceb3aa36f47299c41560a36b2362d4a3b2d0753b5a051de$
 $60f1284a1cc34ddb780be7d40e79e031ac1704b3a0357649149f93ec4037541002577642fd3a4ae68668cca3e$
 $b5$

Бере відкритий ключ відправника і зберігає в змінні n та e :

$n=0xa144293eef3051393d455d68e72042da5762a1a13a871ad3915708adc04384fdcc4890a13f36c52d1c49$
 $d7983369db1620f1e0dcdd16980980051202eb5c3c0654d941d96704cf9c060c2ffd4d1204d2895fea1517477$
 $02660ced8c1ed22ca532a4bba81e6881633eca5e8d49c192155ce253bfa1b2932174203b1ffd537e351a64621$
 $a39$

$e=0x10001$

Підносить $(k_1^{d_1} \bmod n_1)$ і зберігає в змінну k :

$k=0x164d039f14ae0283a625d16ceed70a88a86bf9f4ddac8db5f02cb07b3a43589e2dae2ef66ab79f6837871$
 $9754d0c0e8ded4bf685f673f075afdd836408912f41f618c76df0d4aaae83f389941c4fd4863179092b5a6a972$
 $79d8f00845fcf9806f922fe94cd5252a2f4eaddc180ce2180203d93fbab7ceab45e786436d1909209548e9cb2$

Підносить $(s_1^{d_1} \bmod n_1)$ і зберігає в змінну s :

$s=0x1579ad0a4739bde119135a85d3ac8e1f22d5ceb67b33e4fc279ff8d5060962f8551c89ebf141d75aad99$
 $3aa62aeeb0cd561eb7eec1fae19bd4ff967651c36281d58f00b06cd5d0ad03b4737bbf2ec68035cef00a1b7e40$
 $ec3dcc5c1453855c892cddf44685351dc9d185fcce57b172a6b42716bb21b17e697b2b0d6bad228ee0df2dcae$

Якщо $k = s^e \bmod n$:

$(s^e \bmod n) = 0x164d039f14ae0283a625d16ceed70a88a86bf9f4ddac8db5f02cb07b3a43589e2dae2ef66ab79$
 $f68378719754d0c0e8ded4bf685f673f075afdd836408912f41f618c76df0d4aaae83f389941c4fd4863179092$
 $b5a6a97279d8f00845fcf9806f922fe94cd5252a2f4eaddc180ce2180203d93fbab7ceab45e786436d19092095$
 $48e9cb2$,

то автентифікація пройшла успішно!

Висновки:

В даному лабораторному практикумі ми ознайомились з тестами перевірки чисел на простоту і методами генерації ключів для асиметричної криптосистеми типу RSA. Також практично ознайомились з системою захисту інформації на основі криптосхеми RSA, організували з використанням цієї системи засекречений зв'язок і електронний підпис. А також вивчили протокол розсилання ключів. Цей практикум дуже важливий, тому що алгоритм RSA використовується у таких сучасних протоколах як: PGP, TLS/SLL, IPSEC та ін. і треба розуміти як це працює.