



**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-81
Столярчук Владислав,
Шаруєв Олександр.

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.

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 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 M 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 $0 < k < n$.

Хід роботи:

Кандидати, що не пройшли перевірку та значення вибраних чисел p , q , $p1$, $q1$:

Not prime number: 193c3f535d1605ae564060bf77628225b99a1a93888d7608209180577a1e39cb

Not prime number: 1e4d6dc3e828b4ed3cb2892b9f0755eb661c78b4a809c948fe008ced08414601f

Not prime number: 19081f2c26e416f6b1709c1b66c72316f223ff18431694281eb2d7bcb148de1

Not prime number: 1eab3a56955d2fb0b684fecc9011e7784bd61d85b54305600004cb64f849d5d3f

Not prime number: 1bb345706e2e53d8239d8a9ef73b6965168fe9a92250dfb4a16f973524315ea0d

Not prime number: 1b1854000636c0f9ebe8e25fb5e7972111d20a4c1179628496a111dedc86667c7

Not prime number: 1effc5237166b7321b1ed6b4d544606eb45de3050cce9fb864cf4e6b0f7429161

Not prime number: 10a6de537ac081bfaeef63a5cb0dc14ce2ababa067c02671f3d026f995d9e4799

Not prime number: 19a9210f2ea3de4de1fae955ab1f7b720391355f46192dc745ec93530f453b2dd

Not prime number: 165a183176f2291c7322b9ee0a7ec51825d9aa2a5ffe1b787b28c590d2aa600cf

Not prime number: 1295508d89873bd015aaea17a7bb1a4d20dbb7856e35244731ce3b6ab9e5a0163

Not prime number: 1248a51ee88a928bea5db18d52dfd2f1f158cecd4d58e79b39e238d3409db039d

Not prime number: 11abe3c3aa2010cfb59efa3e3302ab0f7ac44deaaf8b58f219c7b7cfc041e8735

Not prime number: 13134bf7092d3e01f327d38bfd778212740f20f3ad97ecb9581d7c3d492259cbf

Not prime number: 1d4ad327f0af6443b195b5226082b3f9159f7de889073cb28ba23d14b49c1a573

Not prime number: 16f3a4cee7f53732622a6d5ec534c051bab659e73f95ea4e7e4971b2a105964d7

Not prime number: 1aa8c00278906f9bf49c034600ab909983594c0ccd074ebbf0a7a47f001991207

...

Alice:

$p = 16311999af15d6d1b5120a7f6c21e3b92205f80395a6230d46dc9fc9f0d5c8f4b$

$q = 1e1a352e5fbf93e64c7e48f5ce6858f0af1f5801b7afb4fe1ed79d1a79565103d$

$n =$

$29c0698cb8c9635a9ff06f28d5c5e40ee151d0962bf97100e750a9e25ac80a9eed3850f66a25506bb03fe19db7630bcda42f40de44bd187317842fc942799d4df$

$e =$

$23f9c24895b3222eb666dab61174405dda512ddb31badf20586b7a06b48d2fd7b8a1b86fcabac88a96a675f6d8d1c234d8babc7a5c23ef8e5d99368886c0ef54d$

$d =$

$29f7280f8033e5bc0fc703f2ba5d9bc8fed87b636e0f8ca38195c41be657e311cd04671a8ba5ea093b5346ef4a997d1b3617af99e924763c6aa3af9163911c3d$

Bob:

$p = 17d10efebab9a3086f9947d5921a460e37b540651d0cbb74a7fb69b595c0cd9df$

$q = 1ab9476c821e671ff7b2390f02e32d24328e527b819f748fba54674b2560ce277$

$n =$

$27c783ec881c1ab2207143b07a225a839eb2796fd4c577989c1f019eece798c067e7122744269ba9d54ca701feec9057df4d94aabfe10988d01c644a642c424a9$

$e =$

$19e2322dc573b8258b5a3166c86f70741b4e625972c22401acfed71b60062acfca07b5371dcb9b0e3a295ebd5c9e87c0e4e37b4b08e351dc0e1dadd22d58ccc4b$

$d =$

$20a14a8f539c33c4ca051d8a569da24c482f8cac344fe90966865f4a3ac605bcb213b35b7cabe6fa2a0e9cf5977516c92d5539dd480c5d0e7a08d87f982015cef$

Чисельні значення прикладів ВТ, ШТ та цифрового підпису:

Write a message: 1234567890

Message: 499602d2

Encrypted message:

253b17ba482f31a7375dac2ab691512b1853b1e84665f60d656b9f5003b58b717d18075d328aaa5f4b958942b80c1883dfed1193bb
a0d7e283f02cf25caafc56a

Signature:

33ab5722302e378a3e17624f885c9a1c4df896a1a3f483069e8fc0c3f809e3cc8a5951502268639c6b1cce836da92ea398e5c01b03e2
488e178be43ead912f

Decrypted message: 499602d2

Signature verification: True

Протокол конфіденційного розсилання ключів з підтвердженням справжності

1) Генерація ключів для відправника та одержувача за допомогою функції `generate_key_pair()` :

Sender:

p = 1d8eb993ecf03a5bf2c3da6a92e9e85717060f96ed26ef0f09cb95571f516eb5b

q = 152c0ad14f38ca2a1026ed40d11b0b17b4fc95ce21a13049b378626fd6deee97d

n =

271cb00c5c750ee1d4f93181aaec26071a96fc7f1ec4ba41b61562cb6508c14efa15fbee1ca6dd144eddac4d5db11a38a42fba8f3d58c07987c8292a1600be6f

e =

1a15c3b554f3b5b62d1b0982286f12a1ba663742a14b5eff2a9d944634a193320dd3d4b4a43756aa2ba21b66029289f027fbe7e9add6c25116b428a14df6cf171

d =

114561b35ece08ea52775e6ba632cdcf85e6442f87da9ecfd66180bdb3b5640d9393ca47daf4785649f36451273d785d42ceb2ed26aad4158a3ebdfd1a7055369

Recipient:

p = 1d45249712fc67df67506d59ea702bbf95835e59a60db76409dab5a91cc9a316b

q = 179c27fadd379e99b0f4337dc0d7bcb884462b7a7d99e7bc117c9dbffb9c4063

n =

2b310fe16280b24e3b832980e592065765b085b88c8542e7c519724551c429b6d501807459c37573f61a1572b13d2fa90463d71e0fa7eead4e848f4f29e2fdc61

e =

20af1c2e36794620f4d88c5fcb161d4cfa943d4c9b47f2d5c47e155e16d49dc0a0d41ce1d5d1b825f651c4af4686e8dcdb8b006e74c6e29c84ae9aed2c4818713

d =

cbf3f99b07a6a0dc22f9ad0ac904cd2eb57f3a1aa5713f6e8e832aad82167f72a053facb8ad0ed379540d0a999996cd0079c23e776505f721111e5da6f5eb883

2) Обирається випадкове число k

Message: 345fd0590e782f9dd795634c9bafdbe0db8d5862231bcab982e7cd45889f8fbb

3) Повідомлення шифрується відкритим ключем одержувача, додається підпис та надсилається одержувачу за допомогою функції `send_key()` :

Sent message:

Encrypted message:

275c066932cec2d085dd3203aa86a4e5d0fa1994c4142e9fabb0aa9e9a83266c1d608f4ee30643ebe0ae96f0313c62bbaf6678c60aa2d8908dc9423dbce777df5

Encrypted signature:

1f5c1a8dce0fa8b4e93c864403e23b36bf87e15e61c24badacc0f8f932f633cfb372231336ec86a5f62eb9ba0f3be8503b0616eddbc5c7217afd10de8aa851acb

4) Одержувач за допомогою функції `receive_key()` розшифровує повідомлення та перевіряє підпис:

Received message:

Message: 345fd0590e782f9dd795634c9bafdbe0db8d5862231bcab982e7cd45889f8fbb

Verification: True

Перевірка протоколу конфіденційного розсилання ключів з підтвердженням справжності за допомогою сайту - <http://asymcryptwebservice.appspot.com/>

1) Створюємо ключі відправника:

Sender:

p = 19ced807bdda965af3d18ccf89f653f5d438f211253dd4d5289dfe7ed7dfc3a35

q = 1b6f22287863a00f2f015b51403061c6de2c9c94b0f9918be810b965b50fa8e3f

n =
2c404ea098728737623e21a5fd21a3684b4091cfd70343acb1317419388c15dd5556012b1d58f01d490543d373610d1229dc4a17b
21ebf53169441129983d390b
e =
2470d737904f913bd52bcd3c4c18ff7099ba4ed53be392d3e427108f2ac496ff40ccee5530c37c16ff4249cec6f0166e98ca47f7e45da
efcce756e9e0d40df3e5
d =
f50d71f5e8d211e98c7e15d0e432e0ddcefd432d7eddbbaf3f47162b6c03d2ef2980ee3aa75632bc4dd49c62f517ce130d3a9f332c8
68fbc474a5f05b286ef5

2) Створюємо ключі на сервері:

Recipient:

Get server key

Clear

Key size

512

Get key

Modulus

CC470BADA243EF8C68876D4A98071C3555AAFAE6CAD334C401BFCA5534B57B859099615FCC2852DC0221

Public exponent

10001

3) Створюємо повідомлення:

Message: aa9c3599802f117179267df6bcbca3d88ae2ae87a8c7a1c07906deca96e523

4) Формуємо повідомлення для сервера

Sent message:

Encrypted message:

3e5fccc0769e4f89fe9b0b0fa47b350a6f547097fcec40f7f893c1f095c8d2f6b46181a213197d6053f4ced00bc4457c52fa0a468ca1f
fb0c712bdfd8245551

Encrypted signature:

4eb7be639a5fce5613490976fc17287d1b6137a3f90c42782657513d4c4f7b1ca938148ab9d8f5d6dcd5619816dbbd22476c1abac86
57408bc217b84de1d1c7e

Modulus:

2c404ea098728737623e21a5fd21a3684b4091cfd70343acb1317419388c15dd5556012b1d58f01d490543d373610d1229dc4a17b
21ebf53169441129983d390b

Public exponent:

2470d737904f913bd52bcd3c4c18ff7099ba4ed53be392d3e427108f2ac496ff40ccee5530c37c16ff4249cec6f0166e98ca47f7e45da
efcce756e9e0d40df3e5

5) Розшифровуємо повідомлення та перевіряємо підпис на сервері:

Received message:

Receive key

Key

3e5fccc0769e4f89fe9b0fa47b350a6f547097fcea40f7f893c1f095c8d2f6b46181a213197d6053f4ced00bc4457cf

Signature

4eb7be639a5fce5613490976fc17287d1b6137a3f90c42782657513d4c4f7b1ca938148ab9d8f5d6dcd5619816dbbd

Modulus

2c404ea098728737623e21a5fd21a3684b4091cfd70343acb1317419388c15dd5556012b1d58f01d490543d373610

Public exponent

2470d737904f913bd52bcd3c4c18ff7099ba4ed53be392d3e427108f2ac496ff40ccee5530c37c16ff4249cec6f0166e9

Key

0AA9C3599802F117179267DF6BCBBA3D88AE2AE87A8C7A1C07906DECA96E523

Verification

true

✓

Висновки:

Під час виконання комп'ютерного практикуму ми ознайомилися з тестами перевірки чисел на простоту і методами генерації ключів для асиметричної криптосистеми типу RSA; практично ознайомилися з системою захисту інформації на основі криптосхеми RSA, організували з використанням цієї системи засекреченого зв'язку й електронного підпису, вивчили протокол розсилання ключів.