

ЗАСТОСУВАННЯ ВЛАСНИХ ЧИСЕЛ І ВЛАСНИХ ВЕКТОРІВ У ПОБУДОВІ ШИФРІВ ІЗ ПЕРЕВІРКОЮ В ОБЧИСЛЮВАЛЬНОМУ СЕРЕДОВИЩІ

Вінницький національний технічний університет

Анотація

У цій роботі досліджуються методи шифрування і дешифрування текстових рядків за допомогою алгоритму RSA, а також елементи лінійної алгебри, зокрема власні числа та власні вектори. Проаналізовано можливості їх використання у створенні шифрів і перевірено точність обчислень у комп'ютерному середовищі.

Ключові слова: шифрування, дешифрування, вектор, RSA, шифр, алгоритм, відкритий закритий ключ.

Abstract

This paper investigates methods for encrypting and decrypting text strings using the RSA algorithm, as well as elements of linear algebra, including eigenvalues and eigenvectors. The possibilities of their use in the created ciphers are analyzed and the accuracy of calculations in a computer environment is verified.

Keywords: encryption, decryption, vector, RSA, cipher, algorithm, public-private key.

Вступ

Сьогодні одним з найбільш надійних і поширених методів шифрування є асиметричний метод, що використовує як відкриті, так і закриті ключі. Криптостійкість такого підходу визначається довжиною цих ключів. Одним із найвідоміших прикладів асиметричного шифрування є RSA.

Криптосистема RSA базується на використанні односторонньої функції, яка формує добуток двох великих простих цілих чисел. Розкладання такого великого числа на прості множники є значно складнішим процесом[1].

Метою роботи є показати алгоритм шифрування та дешифрування тексту на реальних прикладах за допомогою програми на мові C++.

Результати дослідження

Алгоритм RSA включає три основні етапи: створення ключів, процес шифрування і процедура дешифрування.

Генерація ключів

- 1) Вибираються два великі прості числа p та q .
- 2) Обчислюється модуль: $n = p \cdot q$.
- 3) Визначається функцію Ейлера: $\varphi(n) = (p-1)(q-1)$.
- 4) Підбирається відкрита експонента e , що задовольняє умовам $\gcd(e, \varphi(n)) = 1$.
- 5) Розраховується закрита експонента d за рівнянням: $e \cdot d \equiv 1 \pmod{\varphi(n)}$.
- 6) Формуються ключі:
 - відкритий ключ (e, n) ;
 - закритий ключ d .

Шифрування

- 1) Спочатку повідомлення перетворюється у числовий формат (ASCII).
- 2) Для кожного символу m обчислюється: $c = m^e \pmod{n}$.
- 3) В результаті отримується зашифрована послідовність чисел.

Дешифрування

- 1) Для кожного зашифрованого числа c виконується обчислення: $m = c^d \pmod{n}$.
- 2) Отримані результати перетворюються назад у символи.
- 3) Відновлюється початкове повідомлення.

Складність знаходження секретного ключа системи RSA визначається складністю розкладання числа n на прості множники. У зв'язку з цим необхідно вибирати числа p і q таким чином, щоб задача розкладання числа n була надто складною в обчислювальному плані[2].

Надійність системи шифрування залежить від довжини ключа. Ключі з довжиною 1024 біти вважаються недостатньо безпечними, тоді як ключі на 2048 біт і більше (3072, 4096 біт) є прийнятими стандартами для забезпечення високого рівня захисту.

Алгоритм RSA для конкретного прикладу.

Вхідні дані:

$p = 1000000007$

$q = 1000000009$

Повідомлення: "hi"

Генерація ключів

$n = 1000000016000000063$

$\phi(n) = 1000000014000000048$

$e = 65537$

$d = 648946405777194593$

Відкритий ключ: (65537, 1000000016000000063)

Закритий ключ: (648946405777194593, 1000000016000000063)

Приклад початкових вхідних даних та генерація ключів в програмі зображено на рис. 1.

```
Автор: Свитка Богдана Миколаївна
1EXKB-256

p = 1000000007
q = 1000000009

n = p * q = 1000000016000000063
phi(n) = (p-1)*(q-1) = 1000000014000000048

e = 65537
Перевірка: gcd(e, phi) = 1

Шукаємо d таке, що e*d = 1 (mod phi)
x = -351053608222805455, y = 23007

d = 648946405777194593

Відкритий ключ (e, n)
(65537, 1000000016000000063)

Закритий ключ (d, n)
(648946405777194593, 1000000016000000063)

Введіть рядок: hi
```

Рис. 1. Вхідні дані та генерація ключів в програмі на C++

Шифрування

'h' = 104 = 20060992951408779

'i' = 105 = 472811125885963763

Зашифрована послідовність:

20060992951408779 472811125885963763

Приклад шифрування літер в програмі зображено на рис. 2.

```
=== ШИФРУВАННЯ ===  
  
Символ: 'h' -> 104  
  
Обчислюємо: 104^65537 mod 1000000016000000063  
result = 104  
result = 20060992951408779  
Зашифроване число: 20060992951408779  
  
Символ: 'i' -> 105  
  
Обчислюємо: 105^65537 mod 1000000016000000063  
result = 105  
result = 472811125885963763  
Зашифроване число: 472811125885963763  
  
Зашифрована послідовність:  
20060992951408779 472811125885963763
```

Рис. 2. Наглядний метод шифрування в програмі на C++

Дешифрування

20060992951408779 = 104 = 'h'

472811125885963763 = 105 = 'i'

Результат:

hi

Перевірка правильності розшифрування за допомогою програмного кода зображено на рис. 3. та рис. 4.

```
=== ДЕШИФРУВАННЯ ===  
  
Число: 20060992951408779  
  
Обчислюємо: 20060992951408779^648946405777194593 mod 1000000016000000063  
result = 20060992951408779  
result = 398742675832701857  
result = 844594351391296620  
result = 57725570531691864  
result = 462071539079272112  
result = 862952033255751275  
result = 498398979170266423  
result = 795894127655326489  
result = 301960058725888829  
result = 455049899528407923  
result = 371699480044680898  
result = 151876845589751921  
result = 180707084113757989  
result = 796224669078702292  
result = 607916061037993531  
result = 568593068721379632  
result = 864377191330731195  
result = 408846564590345133  
result = 549630998988827206  
result = 256981103683374578  
result = 412122851674229626  
result = 602067966030975162  
result = 524564030654086079  
result = 9495174184968123  
result = 645289879129919787  
result = 619501406213266875  
result = 104  
Розшифроване число: 104 -> 'h'
```

Рис. 3. Розшифрування першого символу тексту в програмі на C++

```

Число: 472811125885963763

Обчислюємо: 472811125885963763^648946405777194593 mod 1000000016000000063
result = 472811125885963763
result = 592412965912014728
result = 887629591613876793
result = 701792316866607810
result = 708305811815089835
result = 152785843561665765
result = 784358206096135951
result = 939701377622848453
result = 369059277210353754
result = 26898981584041358
result = 223712512541640612
result = 767197099061695070
result = 914183469252816238
result = 231585593622503744
result = 836878886273178051
result = 425280878418677181
result = 122527467677259603
result = 561765394642485728
result = 214141196547511864
result = 437526648872658814
result = 490164162845882110
result = 90452161408105060
result = 898091427114285569
result = 870734129190502142
result = 449943982636253373
result = 812141307552712953
result = 105
Розшифроване число: 105 -> 'i'

Кінцевий розшифрований рядок:
hi

```

Рис. 4. Розшифрування другого символу тексту в програмі на C++

Висновки

В ході роботи було показано, як працює асиметричний алгоритм шифрування на прикладі RSA з використанням власних чисел та значень, також паралельно було продемонстровано виконання дій шифрування та дешифрування тексту в обчислювальному середовищі.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Практикум з прикладної криптології: Навчальний посібник з дисципліни «Прикладна криптологія» для здобувачів першого (бакалаврського) рівня вищої освіти зі спеціальності 125 «Кібербезпека та захист інформації» усіх форм навчання [Електронний ресурс] / [Автори В.В.Палагін, О.А.Палагіна, О.В.Івченко] – ; М-во освіти і науки України, Черкас. держ. технол. ун-т. – Черкаси: ЧДТУ, РВЦ, 2023. – Режим доступу: http://er.chdtu.edu.ua/bitstream/ChSTU/5554/1/Посібник_Прикладна_криптологія.pdf
2. Самойлов І. В, Матійко А. А, Сторчак А. С Криптографія: навчальний посібник. Київ: ІСЗІ КПІ ім. Ігоря Сікорського, 2023. 372 с. – Режим доступу: <https://ela.kpi.ua/items/8c2f071a-0585-4157-bdac-461cbefb2e98>

Свитка Богдана Миколаївна — студентка групи ІЕКБ-25б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: bogdanasvutka@gmail.com

Науковий керівник: **Клеопі Ірина Анатоліївна** — доцент, рНд, Вінницький національний технічний університет, м. Вінниця, e-mail: pakeka08@vntu.edu.ua

Svitka Bohdana Mykolaivna — student of group ІЕКБ-25b, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: bogdanasvutka@gmail.com

Scientific advisor: **Klieopa Iryna Anatoliivna** — docent, рНd, Vinnytsia National Technical University, Vinnytsia, e-mail: pakeka08@vntu.edu.ua