ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ
ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Donetsk National Technical University
Department of higher mathematics and physics

Ist INTERNATIONAL SCIENTIFIC AND PRACTICAL CONFERENCE

for Students and Young Scientists

"Mathematics and Mathematical Simulation in a Modern Technical University"

Abstracts of Ist International
Scientific and Practical Conference

Lutsk, Ukraine
November 30, 2022

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ»

ФАКУЛЬТЕТ МАШИНОБУДУВАННЯ, ЕЛЕКТРОІНЖЕНЕРІЇ ТА ХІМІЧНИХ ТЕХНОЛОГІЙ

КАФЕДРА ВИЩОЇ МАТЕМАТИКИ І ФІЗИКИ

І МІЖНАРОДНА НАУКОВО-ПРАКТИЧНА КОНФЕРЕНЦІЯ СТУДЕНТІВ ТА МОЛОДИХ ВЧЕНИХ «МАТЕМАТИКА ТА МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ У СУЧАСНОМУ ТЕХНІЧНОМУ УНІВЕРСИТЕТІ»

30 ЛИСТОПАДА 2022 РОКУ

ЛУЦЬК, 2022

ЗМІСТ

Section 1. Mathematical modeling and research of processes in modern technologies and techniques

АСИМЕТРИЧНИЙ МЕТОД ШИФРУВАННЯ. АЛГОРИТМ RSA, ВИКОРИСТАННЯ ПАКЕТУ WOLFRAM MATHEMATICA ДЛЯ ГЕНЕРУВАННЯ ВЕЛИКИХ ПРОСТИХ ЧИСЕЛ В КРИПТОГРАФІЇ

Є.Р. Терехов, студент ОП «Кібербезпека»
Т.В. Скрипник, старший викладач кафедри ПМІ
Донецький національний технічний університет, Луцьк, Україна

Анотація. Досліджується найпоширеніший алгоритм асиметричного шифрування RSA. Стійкість RSA базується на складності факторизації великих цілих чисел. RSA можна застосовувати як для шифрування/розшифрування, так і для генерації/перевірки електронного цифрового підпису (ЕЦП).
Ключові слова: шифрування, алгоритм RSA, відкритий ключ, просте число, дешифрування.

Вступ. На сьогодні асиметричне шифрування на основі відкритого ключа RSA (розшифровується, як Rivest, Shamir and Aldeman – творці алгоритму) використовує більшість продуктів на ринку інформаційної безпеки.

Його криптостійкість ґрунтується на складності розкладання на множники великих чисел, а саме - на винятковій важкості завдання визначити секретний ключ на основі відкритого, тому що для цього потрібно розв'язати задачу існування дільників цілого числа. Найбільш криптостійкі системи використовують 1024-бітові та великі числа.

Мета. Дослідити роботу алгоритму RSA за критерієм швидкодії, зокрема генерування простих чисел для формування ключів шифрування, засобами пакету Wolfram Mathematica.

Основна частина. З практичної точки зору алгоритм RSA можна поділити на два етапи, кожний з яких складається з кількох кроків. Це етапи генерування відкритого та секретного ключів і, власне, шифрування повідомлення.

Етап генерування ключів покроково виглядає так:

- Візьмемо два великі прості числа p та q порядку 2^{768} та 2^{1024} відповідно до кожного (рис. 2).
- Розрахуємо n, як результат добутку p і q ($n = p*q$) (рис. 3).
- Виберемо випадкове число, яке назвемо d. Це число має бути взаємно простим (тобто, не мати жодного спільного дільника, крім 1) із результатом добутку $(p-1)*(q-1)$.
- Визначимо таке число e, для якого є істинним наступне співвідношення $(e*d) \bmod ((p-1)*(q-1)) = 1$.
- Назвемо відкритим ключем числа e і n, а секретним - d і n.

Для процесу шифрування повідомлення за відкритим ключем $\{e, n\}$ слід виконати наступні дії:

- розбити текст, що шифрується, на блоки, кожен з яких може бути представлений у вигляді числа $M(i) = 0, 1, 2, ..., n-1$(рис. 1).
- зашифрувати текст, що розглядається як послідовність чисел $M(i)$ за формулою $C(i) =(M(i)^e) \bmod n$ (рис. 4).

Для того, щоб розшифрувати отриману криптограму, використовуючи секретний ключ $\{d, n\}$, необхідно виконати такі обчислення $M(i) = (C(i)^d) \bmod n$. В результаті чого буде отримано множину чисел $M(i)$, що відповідатиме при певній конвертації вхідному тексту. В якості вхідного повідомлення біло обрано ПІБ автора « Терехов Єгор Русланович».

Опис пропозиції. Алгоритми шифрування з відкритим ключем у середньому працюють у тисячі разів повільніше за симетричні алгоритми, а також вони вимогливі до пам'яті та обчислювальної потужності комп'ютера. Ця особливість була досліджена спробами рандомізації простих чисел особливо великих порядків для формування пари ключів засобами програмування на мовах C++ та Python. Оскільки програми працювали надзвичайно довго і не завжди видавали очікуваний результат, то було обрано рішення скористуватися потужними засобами програми «Wolfram Mathematica», що може опрацьовувати доволі великі числа.

Застосовуючи функціональні можливості означеної програми було представлено описаний метод шифрування.

Рядок повідомлення та множина числових блоків $M(i)$, відповідно до кожного символу вхідного тексту подано на рис. 1.

```
In[1]:= str = "Терехов Єгор Русланович"
        M = ToCharacterCode[str]

Out[1]= Терехов Єгор Русланович

Out[2]= {1058, 1077, 1088, 1077, 1093, 1086,
         1074, 32, 1028, 1075, 1086, 1088, 32, 1056, 1091,
         1089, 1083, 1072, 1085, 1086, 1074, 1080, 1095}
```

Рис. 1. Рядок відкритого тексту та кодування символів у числові блоки

Генерування простих чисел означених вище порядків наведено на рис. 2.

```
In[3]:= p = NextPrime[2^768]

Out[3]= 1552518092300708935148979488462502555256886017116696611 ↕
        1390520380260509526863768863308784088286464779504877 30 ↕
        6971310732061715800441148143914442872750411811392044 54 ↕
        9760208499055502652856315984448252629991937164687508 92 ↕
        846853816058039

In[4]:= q = NextPrime[2^1024]

Out[4]= 1797693134862315907729305190789024733617976978942306572 73 ↕
        4300811577326758055009631327084773224075360211201138 79 ↕
        8713933576587897688144166224928474306394741243777678 93 ↕
        4248654852763022196012460941194530829520850057688381 50 ↕
        6823424628814739131105408272371633505106845862982399 47 ↕
        2459384797163048353563296242241378 59
```

Рис. 2. Генерування простих чисел p і q

Формування параметрів системи RSA подано на рис. 3.

```
In[5]:= n = p * q

Out[5]= 279095111627852376407822673918065072905887935345660252615
        989519488029661278604994789701101367875859521849524793
        382568057369148405837577299984720398976429790087982805
        274893437406788716103454867635208144157749912668657006
        085226160294706625852843084369218264705363016445247708
        352165200176022381626250444607084964470181341019003003
        415598483967897657554394321071280340452051898833108957
        724846122456100123936170740252983449524382165752876265
        876496572523442181968085530164687457504983765791136126
        377958572067585002096597551704523428264564681198501

In[6]:= ϕ = (p - 1) (q - 1)

Out[6]= 279095111627852376407822673918065072905887935345660252615
        989519488029661278604994789701101367875859521849524793
        382568057369148405837577299984720398976429790087982805
        274893437406788716103454867635208144157749912668657006
        085226160294526856539356852778445334186284113971885910
        654270969518748951545092711931279461954530540240971660
        730597875391515230904150645164814914896377367959627365
        564344858357328290242038986000276117581156973093621733
        747375989293847212350581467551569891988135678722235350
        407740675382812493158924118930937179088086641002604
```

Рис. 3. Модуль перетворення n, як добуток p і q та функція $\phi(n)$

Обчислення криптограми відбувається в циклі, що подано на рис. 4.

```
In[12]:= CC = Table[0, {i, 1, Length[M]}];
         For[i = 1, i ≤ Length[M], i++,
           {CC[[i]] = PowerMod[M[[i]], e, n]}
          ];
         CC // MatrixForm

Out[14]//MatrixForm=
         834317622049652889673154264833047207803049865894914042287452
         276720968585444070141262568582971937931566952407487809793216
         132525465987745554388301175172966048827559230268739584624433
         276720968585444070141262568582971937931566952407487809793216
         250395406270577612099110976133527752251201352273862485411141
          806319133964596073554188764031718264290562214294015467488424
          515934393818502082487007269574980805879834658540874610742276
          482598084477935086483907739310067694451989739273648935308253
          969092122779389004393246428447663254608931628853131959354266
          468470433279446934546192485053606489952894961283499581304711
          806319133964596073554188764031718264290562214294015467488424
         132525465987745554388301175172966048827559230268739584624433
          482598084477935086483907739310067694451989739273648935308253
         181824940532759845755501119888251612761978718052485463868432
          549148053219939008524372073488825573624916028835433275592236
         191914503291132628498449704911311469747756061718403006972877
          170451759728400184651265813178971286988731284992867348005771
          386042714806938921365307156840411563274045129967707089882402
          997163944697234397179115549420245763505212793813917214124477
          806319133964596073554188764031718264290562214294015467488424
          515934393818502082487007269574980805879834658540874610742276
         150476505433508250100173952480711865417637843430987787109846
         232316967410062146442996977199005218981712768326335650085404
```

Рис. 4. Формування криптограми

Висновки. Серед найбільш відомих недоліків алгоритму RSA можна виділити некоректний вибір значень *p* і *q*. Числа *p* і *q* мають бути простими і не міститися в жодній з відомих таблиць простих чисел. Ці числа також не повинні бути близькими один до одного, бо це впливає на стійкість алгоритму шифрування.

Перелік посилань

1. Тарнавський Ю.А. Технології захисту інформації / Київ, КПІ ім. Ігоря Сікорського – 2018 – 161 с.
2. Барич, С. Г. Основи сучасної криптографії: навчань, курс / С. Г. Барич, В. В. Гончаров, Р. Є. Сєров. - М: Гаряча лінія - Телеком, 2011.
3. Рябко, Б. Я. Криптографічні методи захисту інформації / Б. Я. Рябко.

Наукове видання

«Математика та математичне моделювання у сучасному технічному університеті»

Збірник тез доповідей

І МІЖНАРОДНОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ СТУДЕНТІВ ТА МОЛОДИХ ВЧЕНИХ

30 листопада 2022 року, м. Луцьк

Технічна обробка, комп'ютерний набір, верстка: Гоголєва Наталія Федорівна

Оригінал-макет виготовлено на кафедрі вищої математики і фізики
ДВНЗ «Донецький національний технічний університет»

Формат А4