

RSA ALGORITHM

Лабораторна робота

Мета: Дослідити і реалізувати механізм симетричного алгоритму шифрування RSA на основі згенерованих ключів.

Індивідуальне завдання:

Розробити додаток обміну таємними посиланнями між двома клієнтами за допомогою алгоритму шифрування RSA.

ХІД РОБОТИ

RSA (аббревіатура від прізвищ Rivest, Shamir та Adleman) — криптографічний алгоритм з відкритим ключем, що базується на обчислювальній складності задачі факторизації великих цілих чисел.

RSA став першим алгоритмом такого типу, придатним і для шифрування, і для цифрового підпису. Алгоритм застосовується до великої кількості криптографічних застосунків.

Алгоритм RSA складається з 4 етапів: генерації ключів, шифрування, розшифрування та розповсюдження ключів.

Безпека алгоритму RSA побудована на принципі складності факторизації цілих чисел. Алгоритм використовує два ключі — відкритий (public) і секретний (private), разом відкритий і відповідний йому секретний ключі утворюють пари ключів (keypair). Відкритий ключ не потрібно зберігати в таємниці, він використовується для шифрування даних. Якщо повідомлення було зашифровано відкритим ключем, то розшифрувати його можна тільки відповідним секретним ключем.

Залежно від структури використовуваних ключів методи шифрування поділяються на:

- симетричне шифрування: стороннім особам може бути відомий алгоритм шифрування, але невідома невелика порція секретної

інформації - ключа, однакового для відправника і одержувача повідомлення; Приклади: DES, 3DES, AES, Blowfish, Twofish, ГОСТ 28147-89

- асиметричне шифрування: стороннім особам може бути відомий алгоритм шифрування, і, можливо відкритий ключ, але невідомий закритий ключ, відомий тільки одержувачу. Криптографічні системи з відкритим ключем в даний час широко застосовуються в різних мережних протоколах, зокрема, в протоколах TLS і його попереднику SSL (що лежать в основі HTTPS), а так же SSH, PGP, S / MIME і т.

На даний момент 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 виняткової складності завдання визначити секретний ключ на підставі відкритого, так як для цього буде потрібно вирішити задачу про існування дільників цілого числа. Найбільш криптостійкі системи використовують 1024-бітові і великі числа.

Лістинг шифратора:

```
public byte[] encrypt(byte[] source, boolean isLastBlock) {
    int[] blocks = CipherUtils.bytesToIntsWithPad(source);
    for (int i = 0; i < blocks.length; i += 2) {
        encryptBlock(blocks, i, i + 1);
    }
    byte[] bytes = CipherUtils.convertIntsToBytes(blocks);
    if (isLastBlock) {
        int padCount = CipherUtils.paddingSize(source.length);
        byte[] tmp = new byte[bytes.length + 1];
```

```

System.arraycopy(bytes, 0, tmp, 0, bytes.length);

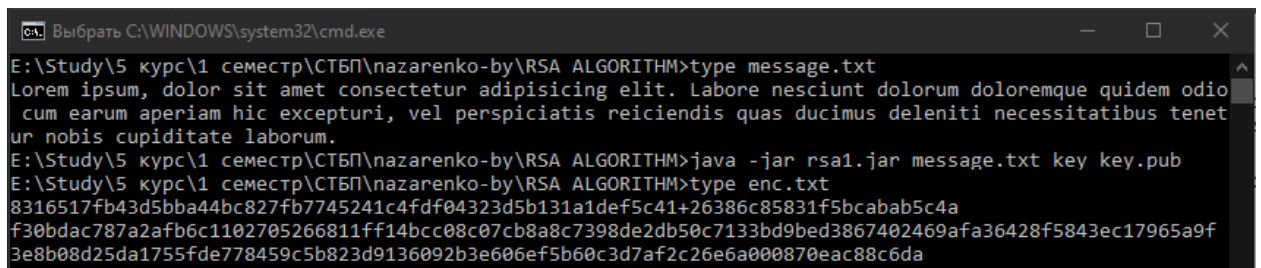
tmp[tmp.length - 1] = (byte)padCount;

bytes = tmp;
}

return bytes;
}

```

Шифруємо на клієнті №1 повідомлення, результат наведено на рис.1.



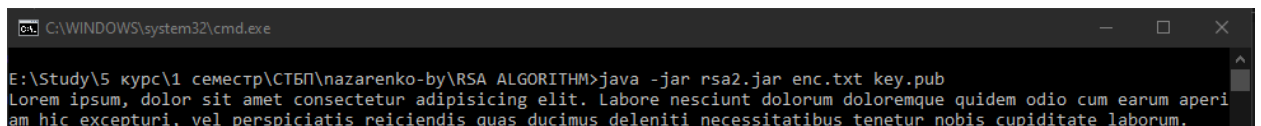
```

C:\WINDOWS\system32\cmd.exe
E:\Study\5 курс\1 семестр\СТБП\nazarenko-by\RSA ALGORITHM>type message.txt
Lorem ipsum, dolor sit amet consectetur adipisicing elit. Labore nesciunt dolorum doloremque quidem odio
cum earum aperiam hic excepturi, vel perspiciatis reiciendis quas ducimus deleniti necessitatibus tenet
ur nobis cupiditate laborum.
E:\Study\5 курс\1 семестр\СТБП\nazarenko-by\RSA ALGORITHM>java -jar rsa1.jar message.txt key key.pub
E:\Study\5 курс\1 семестр\СТБП\nazarenko-by\RSA ALGORITHM>type enc.txt
8316517fb43d5bba44bc827fb7745241c4fdf04323d5b131a1def5c41+26386c85831f5bcabab5c4a
f30bdac787a2afb6c1102705266811ff14bcc08c07cb8a8c7398de2db50c7133bd9bed3867402469afa36428f5843ec17965a9f
3e8b08d25da1755fde778459c5b823d9136092b3e606ef5b60c3d7af2c26e6a000870eac88c6da

```

Рисунок 1 – Результат

Розшифровуємо на клієнті №2 повідомлення, результат наведено на рис.1.



```

C:\WINDOWS\system32\cmd.exe
E:\Study\5 курс\1 семестр\СТБП\nazarenko-by\RSA ALGORITHM>java -jar rsa2.jar enc.txt key.pub
Lorem ipsum, dolor sit amet consectetur adipisicing elit. Labore nesciunt dolorum doloremque quidem odio cum earum aperi
am hic excepturi, vel perspiciatis reiciendis quas ducimus deleniti necessitatibus tenetur nobis cupiditate laborum.

```

Рисунок 2 – Результат

Висновки: в ході лабораторної роботи було досліджено механізм симетричного алгоритму шифрування RSA на основі згенерованих ключів.