1. Цель работы

Реализовать криптографическую систему RSA. Система должна работать в двух режимах: шифрования и постановки подписи. При постановке подписи использовать хеш-функцию SHA1.

2. Описание алгоритма

RSA – криптографический алгоритм с открытым ключом, основывающийся на вычислительной сложности задачи факторизации больших целых чисел. Криптосистема RSA стала первой системой, пригодной и для шифрования, и для цифровой подписи.

Главной идеей RSA является использование односторонней функции с секретом. Под односторонней понимается такая функция $y=f(x)$, которая легко вычисляется при имеющемся x, но аргумент x при заданном значении функции вычислить сложно. Аналогично, односторонней функцией с секретом называется функция $y=f(x, k)$, которая легко вычисляется при заданном x, причём при заданном секрете k аргумент x по заданному y восстановить просто, а при неизвестном k – сложно.

Подобным свойством обладает операция возведения числа в степень по модулю:

$$c \equiv f(m) \equiv m^e mod\ n,$$
$$m \equiv f^{-1}(c) \equiv c^d mod\ n,$$
$$d \equiv e^{-1} mod\ \varphi(n)$$

Где $\varphi(n)$ – функция Эйлера числа n, c – зашифрованное сообщение m, (e, n) – открытый ключ, а (d, n) – закрытый ключ.

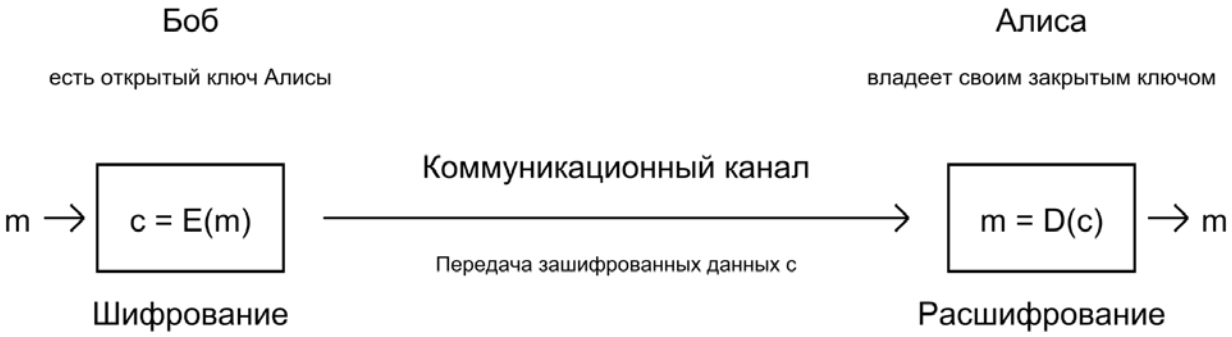


Рисунок 1. Схема передачи сообщения в RSA

Рисунок 2. Схема проверки электронной подписи в RSA

3. Описание реализации

Для работы с большими числами используется библиотека Arbitrary Precision.

В данном примере SIZE = 1024, но он может быть увеличен при необходимости.

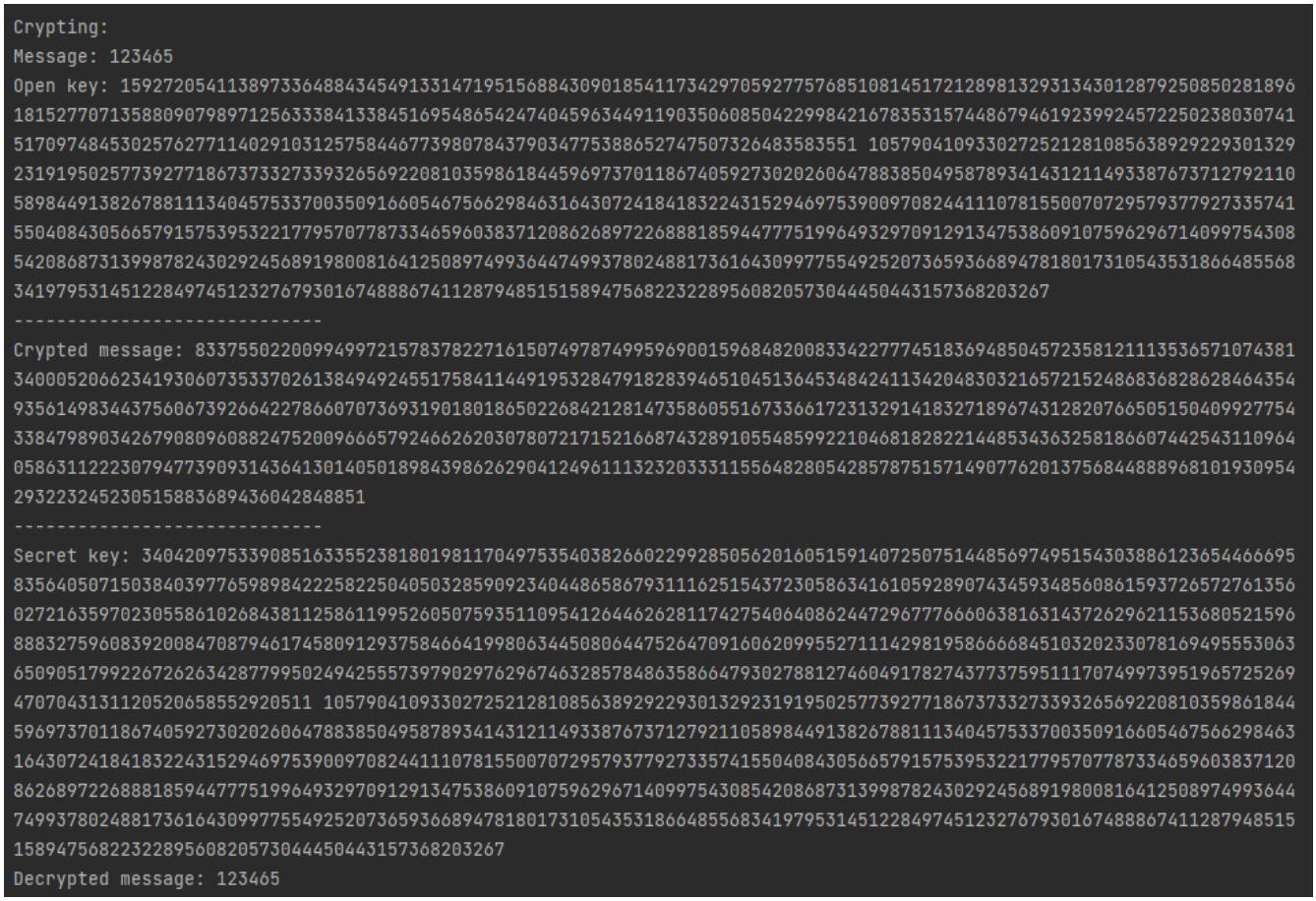
- RSA(ap_uint<SIZE> p, ap_uint<SIZE> q) – создание экземпляра класса RSA, во время которой задаются простые числа p и q, а также вычисляются число n = p * q и функция Эйлера, равная (p - 1) * (q-1);
- pair<ap_uint<4 * SIZE>, ap_uint<4 * SIZE>> createOpenKey(ap_uint<4 * SIZE> fi_n, ap_uint<4 * SIZE> n, size_t countOfBits) – создание пары (e, n), являющейся открытым ключом;
- pair<ap_uint<4 * SIZE>, ap_uint<4 * SIZE>> createSecretKey(ap_uint<4 * SIZE> e, ap_uint<4 * SIZE> n, ap_uint<4 * SIZE> fi_n) – создание пары (d, n), являющейся закрытым ключом;
- ap_uint<4 * SIZE> BinaryPower(ap_uint<4 * SIZE> a, ap_uint<4 * SIZE> fi_n, ap_uint<4 * SIZE> n) – бинарное возведение в степень;
- ap_uint<4 * SIZE> ExtendedEuclidian(ap_uint<4 * SIZE> e, ap_uint<4 * SIZE> n) – расширенный алгоритм Евклида;
- bool DigitalSignCheck(pair<ap_uint<4 * SIZE>, ap_uint<4 * SIZE>> digitalSign, pair<ap_uint<4 * SIZE>, ap_uint<4 * SIZE>> openKey) – проверка подлинности подписи;

4. Пример работы программы

```
p = 3557
q = 2579
n = 9173503
Euler f(n) = (q - 1) * (p - 1) = 9167368

Open key (e, n): 3, 9173503
Secret key (d, n): 6111579, 9173503

Message: 111111
NOD(m, n) = 1
Crypting: 4051753
Decrypting: 111111

Hashed message: 3d4f2bf07dc1be38b20cd6e46949a1071f9d0e3d
Correct digital signature: 0x194C5A
It's correct signature
```

Рисунок 3. Пример 1

```
p = 10643589889866264446591633395719574666415008835518312873402142266898672781580458354211970699803012956392309230236389778896175897908007905676252541220401893782576308203956256125610599597816040543966034874752670907365203021690002804258090672054736828913725971001449165331215962007329868678344178884209750789930570
q = 99393542994126483743975165847411960208524265810838520264168837212070735539861932935581029723307611290040281384222048242466107580772760211030233523517509180391611937682423706130435236989155697614292882034384826100499683514596324824961172020353249294567149726012732302008670515129968643179886226309734177705310
n = 10579041093302725212810856389292293013292319195025773927718673733273393265692208103598618445969737011867405927302026064788385049587893414312114933876737127921105898449138267881113404575337003509166054675662984631643072418418322431529469753900970824411107815500707295793779273357415504084305665791575395322177957077873346596038371208626897226888185944777519964932970912913475386091075962967140997543085420868731399878243029245689198008164125089749936447499378024881736164309977554925207365936689478180173105435318664855683419795314512284974512327679301674888674112879485151589475682232289560820573044450443157368203267
Euler f(n) = (q - 1) * (p - 1) = 10579041093302725212810856389292293013292319195025773927718673733273393265692208103598618445969737011867405927302026064788385049587893414312114933876737127921105898449138267881113404575337003509166054675662984631643072418418322431529469753900970824411107815500707295793779273357415504084305665791575395322177957077873346596038371208626897226888185944777519964932970912913475386091075962967140997543085420868731399878243029245689198008164125089749936447499378024881736164309977554925207365936689478180173105435318664855683419795314512284974512327679301674888674112879485151589475682232289560820573044450443157368203267
```



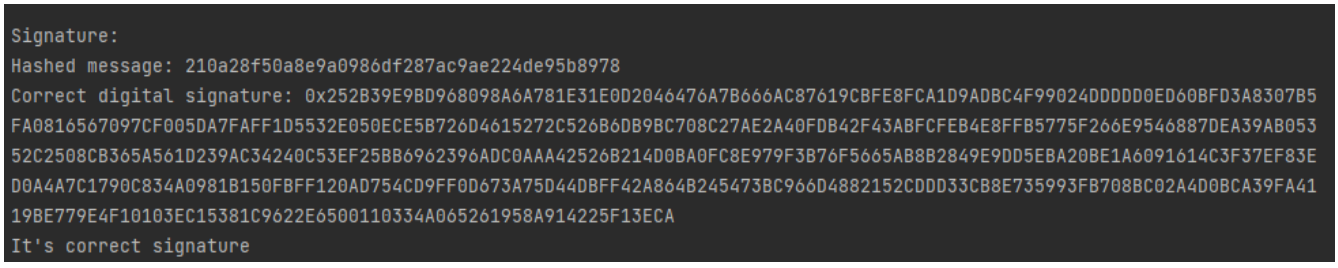


Рисунок 4. Пример 2

5. Вывод

В ходе выполнения данной лабораторной работы была реализована криптосистема RSA, а именно шифрование и дешифрование сообщения, а также электронная подпись с использованием алгоритма хеширования SHA1. Система RSA используется для защиты программного обеспечения и в схемах цифровой подписи. Также она используется в открытой системе шифрования PGP и иных системах шифрования в сочетании с симметричными алгоритмами.