

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:

$$\begin{aligned}c &\equiv f(m) \equiv m^e \bmod n, \\m &\equiv f^{-1}(c) \equiv c^d \bmod n, \\d &\equiv e^{-1} \bmod \varphi(n)\end{aligned}$$

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

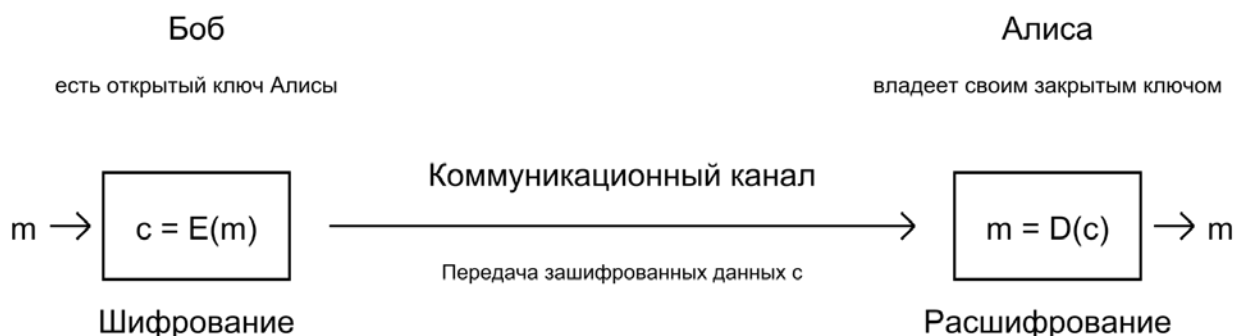


Рисунок 1. Схема передачи сообщения в RSA



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 = 10643589889866264446591633395719574664150088355183128734021422668986727815804583542119706998030129563923092302363897
788961758979080079056762525412204018937825763082039562561256105995978160405439660348747526709073652030216900028042580906
7205473682891372597100144916533121596200732986867834417888420975078993057
q = 99393542994126483743975165847411960208524265810838520264168837212070735539861932935581029723307611290040281384222048
242466107580772760211030233532351750918039161193768242370613043523698915569761429288203438482610049968351459632482496117
202035324929456714972601273230200867051512996864317988622630973417770531
n = 10579041093302725212810856389292293013292319195025773927718673733273393265692208103598618445969737011867405927302026
064788385049587893414312114933876737127921105898449138267881113404575337003509166054675662984631643072418418322431529469
753900970824411107815500707295793779273357415504084305665791575395322177957077873346596038371208626897226888185944777519
964932970912913475386091075962967140997543085420868731399878243029245689198008164125089749936447499378024881736164309977
554925207365936689478180173105435318664855683419795314512284974512327679301674888674112879485151589475682232289560820573
044450443157368203267
Euler f(n) = (q - 1) * (p - 1) = 105790410933027252128108563892922930132923191950257739277186737332733932656922081035986
184459697370118674059273020260647883850495878934143121149338767371279211058984491382678811134045753370035091660546756629
846316430724184183224315294697539009708244111078155007072957937792733574155040843056657915753953221777512484314538069101
613171270926191813359196281572951253665298495734480773780551987842194433818119618021286738351682195571143107925515389712
809598449860845854393722958136869420332624532089585560150726596399592823368495247940523720662071432717926670678448008067
38961826153218980043577088420637939391208871439680
```

```

Crypting:
Message: 123465
Open key: 15927205411389733648843454913314719515688430901854117342970592775768510814517212898132931343012879250850281896
181527707135880907989712563338413384516954865424740459634491190350608504229984216783531574486794619239924572250238030741
5170974845302576277114029103125758446773980784379034775388652747507326483583551 1057904109330272521281085638929229301329
231919502577392771867373327339326569220810359861844596973701186740592730202606478838504958789341431211493387673712792110
589844913826788111340457533700350916605467566298463164307241841832243152946975390097082441110781550070729579377927335741
550408430566579157539532217795707787334659603837120862689722688818594477751996493297091291347538609107596296714099754308
542086873139987824302924568919800816412508974993644749937802488173616430997755492520736593668947818017310543531866485568
3419795314512284974512327679301674888674112879485151589475682232289560820573044450443157368203267
-----
Crypted message: 8337550220099499721578378227161507497874995969001596848200833422777451836948504572358121113536571074381
340005206623419306073533702613849492455175841144919532847918283946510451364534842411342048303216572152486836828628464354
935614983443756067392664227866070736931901801865022684212814735860551673366172313291418327189674312820766505150409927754
338479890342679080960882475200966657924662620307807217152166874328910554859922104681828221448534363258186607442543110964
058631122230794773909314364130140501898439862629041249611132320333115564828054285787515714907762013756844888968101930954
293223245230515883689436042848851
-----
Secret key: 340420975339085163355238180198117049753540382660229928505620160515914072507514485697495154303886123654466695
835640507150384039776598984222582250405032859092340448658679311162515437230586341610592890743459348560861593726572761356
027216359702305586102684381125861199526050759351109541264462628117427540640862447296777666063816314372629621153680521596
888327596083920084708794617458091293758466419980634450806447526470916062099552711142981958666684510320233078169495553063
650905179922672626342877995024942555739790297629674632857848635866479302788127460491782743773759511170749973951965725269
4707043131120520658552920511 1057904109330272521281085638929229301329231919502577392771867373327339326569220810359861844
596973701186740592730202606478838504958789341431211493387673712792110589844913826788111340457533700350916605467566298463
164307241841832243152946975390097082441110781550070729579377927335741550408430566579157539532217795707787334659603837120
862689722688818594477751996493297091291347538609107596296714099754308542086873139987824302924568919800816412508974993644
749937802488173616430997755492520736593668947818017310543531866485568341979531451228497451232767930167488867411287948515
1589475682232289560820573044450443157368203267
Decrypted message: 123465

```

```

Signature:
Hashed message: 210a28f50a8e9a0986df287ac9ae224de95b8978
Correct digital signature: 0x252B39E9BD968098A6A781E31E0D2046476A7B666AC87619CBFE8FCA1D9ADBC4F990240DDDD0ED60BFD3A8307B5
FA0816567097CF005DA7FAFF1D5532E050ECE5B726D4615272C526B6DB9BC708C27AE2A40FDB42F43ABFCFEB4E8FFB5775F266E9546887DEA39AB053
52C2508CB365A561D239AC34240C53EF25BB6962396ADC0AAA42526B214D0BA0FC8E979F3B76F5665AB8B2849E9DD5EBA20BE1A6091614C3F37EF83E
D0A4A7C1790C834A0981B150FBFF120AD754CD9FF0D673A75D44DBFF42A864B245473BC966D4882152CDD0D33CB8E735993FB708BC02A4D0BCA39FA41
19BE779E4F10103EC15381C9622E6500110334A065261958A914225F13ECA
It's correct signature

```

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.