ЛАБОРАТОРНАЯ РАБОТА 6

КРАЖЕВСКИЙ АЛЕКСЕЙ ИГОРЕВИЧ, 15 ГРУППА

ВАРИАНТ 8

Условие варианта:

8	706328544280902456686127198375891542636741400450633261471218641282896406727 21

Для выполнения всех заданий необходимо

использовать "длинную" арифметику. Разрешается использовать любую готовую библиотеку или написать свою.

Шаг 1. В условиях своего варианта для заданного числа q (см. Таблица 1)

необходимо выполнить алгоритм Gen.

Шаг 2. Для параметров, полученных на шаге 1, и сообщения вида «Я, Иван Иванов, люблю КМ», где «Иван Иванов» – ваше имя и фамилия соответственно, выполнить алгоритм Sign.

Шаг 3. Проверьте результаты подписи из Шага 2, выполнив алгоритм Verify.

Выполнение задания:

В процессе выполнения задания я написал функцию генерации ключа по указанному алгоритму

Генерация ключей (Gen)

Вход: q – простое число.

Шаги:

1. Выбрать такое четное число R, что $R < 4\,(q + 1)$.
2. Вычислить число $p = qR + 1$.
3. Если $2^{qR} \neq 1 \pmod p$ или $2^R = 1 \pmod p$, то возвратиться к шагу 1, иначе p – простое.
4. Случайным образом выбрать x из Z_p и вычислить $g = x^R \pmod p$.
5. Если $g = 1$, то возвратиться к шагу 4, иначе искомое g найдено.
6. Случайным образом выбрать личный ключ d из Z_q.
7. Вычислить открытый ключ $e = g^d \pmod p$.

Выход: (p, q, g) – параметры ЭЦП; e – открытый ключ; d – личный ключ.

Также были написаны функции подписи и проверки подписи по следующим алгоритмам:

Функция подписи (Sign)

Вход: (p, q, g) – параметры ЭЦП; d – личный ключ; M – подписываемое сообщение (в виде строки текста произвольной длины).

Шаги:

1. Вычислить хэш-значение m от сообщения M: $m = h(M)$ ($h()$ – хэш-функция).
2. Случайным образом выбрать одноразовый личный ключ k из $Z_q / \{0\}$.
3. Вычислить $r = g^k \pmod p$.
4. Вычислить $s = k^{-1}(m - dr) \pmod q$.

Выход: (r, s) – подпись.

Функция проверки подписи (Verify)

Вход: (p, q, g) – параметры ЭЦП; e – открытый ключ; M – подписываемое сообщение (в виде строки текста произвольной длины; (r, s) – подпись.

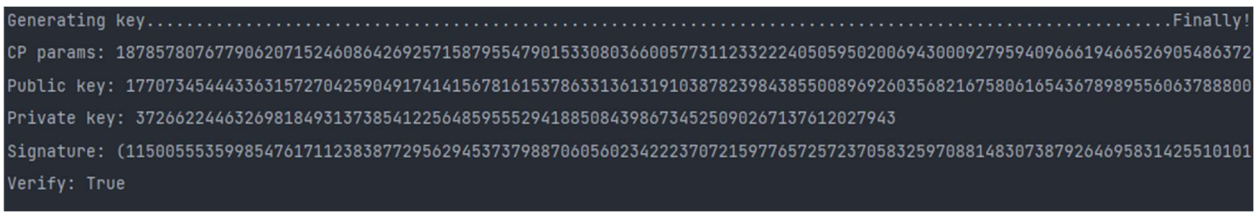
Шаги:

1. Если r не лежит в $Z_p / \{0\}$ или s не лежит в Z_q, то вернуть *FALSE*.
2. Вычислить хэш-значение m от сообщения M: $m = h(M)$ ($h()$ – хэш-функция).
3. Если $e^r r^s = g^m \pmod p$, то вернуть *TRUE*, иначе вернуть *FALSE*.

Выход: TRUE, если подпись корректна; *FALSE,* если подпись некорректна.

Также я написал хеш-функцию SHA-256. Алгоритм написал по псевдокоду из википедии: https://en.wikipedia.org/wiki/SHA-2

Окно вывода программы:



В скриншот не влезают все числа

Пример полученных данных:

P=18785780767790620715246086426925715879554790153308036600577311233222405059502006943000927959409666194665269054863726037331851415166360239264431330321839 61

Q=706328544280902456686127198375891542636741400450633261471218641282896406727 21

G=13266592046015454755200377778289390942692387145799544382166219294150645184175952051656050438600035480762249701203258389416115787705282521452375124884824 60

PublicKey=17707345444336315727042590491741415678161537863313613191038782398438550089692603568216758061654367898955606378880034536974404268740124129882781376031621 72

PrivateKey=37266224463269818493137385412256485955529418850843986734525090267137612027943

Signature:

R=1150055535998547617112383877295629453737988706056023422237072159776572572370583259708814830738792646958314255101011498696337504323043346880440017981922923

S=5470517945704242159332790445914314561827321369477200996543340773 1936416880002

Как видно по выводу, такие данные прошли верификацию, хотя наглядными их нельзя назвать…

Для возведения в степень я использовал алгоритм бинарного возведения в степень из предыдущей лабораторной, как и расширенный алгоритм Евклида для поиска обратного элемента.