Нахождение открытого и закрытого ключа для RSA

Аметов Имиль, гр. М07-903

22 мая 2020 г.

Задача:

Найти открытый и закрытый ключ для RSA при $p = 67$ и $q = 31$. Продемонстрировать шифрование и расшифрование. Привести соображения о сложности вычислений с очень большими числами.

Предлагаемое решение:

Найдём открытый и закрытый ключи.

Находим $n = p \cdot q = 67 \cdot 31 = 2077$.

Вычисляем значение функции Эйлера $\varphi(2077) = (67 - 1)(31 - 1) = 1980$.

Выбираем значение $e \in [3, 1980]$ такое, что $\gcd(e, \varphi(n)) = 1$. Я выбрал $e = 727$. Число 727 простое и $1980 = 11 \cdot 5 \cdot 3^2 \cdot 2^2$. Отсюда $\gcd(727, 1980) = 1$.

Теперь находим $d = e^{-1} \mod \varphi(n) = 727^{-1} \mod 1980$. У меня получилось $d = 463$.

Отсюда у меня открытый ключ PK $= (e : 727, n : 2077)$ и закрытый ключ SK $= (d : 463, n : 2077)$.

Шифрование:

Пусть сообщение $m = 117$. Вычисляем шифртекст:

$$c = 117^{727} \mod 2077 = 251.$$

Расшифрование:

Полученный шифртекст $c = 251$. Вычисляем открытый текст:

$$m = 251^{463} \mod 2077 = 117.$$

Соображения о сложности вычислений с очень большими числами

При вычислениях с очень большими числами возникает много проблем. Нужно искать очень большие простые числа, что само по себе непростая задача. Кроме того, нужно подбирать число e, такое, что НОД для чисел e и $\varphi(n)$ был бы равен единице.

Наивное нахождение числа d обратного для e путём перебора также дорогостоящая процедура и в худшем случае сложность может быть $O(\varphi(n))$.

Пример наивной реализации обратного числа для $e = 1979$ на языке Haskell:

```
sn :: Int -> Int
sn x
  | (1979 * x) `mod` 1980 == 1 = x
  | otherwise = sn (x + 1)

> sn 1
1979
```