МАТЕРИАЛЫ К ЗАДАНИЮ №37 – RSA

Постановка задачи

Пусть $M = (m_0, m_1, \ldots, m_{t-1})$ – сообщение длины $t \in \mathbb{N}$, причём для каждого $i \in \overline{0, t-1}$ символ m_i имеет шестнадцатеричный ASCII-код, который будем обозначать $\mathrm{Ord}(m_i)$. В итоге сообщению M ставится в соответствие число

$$\mathrm{Ord}(m_{t-1}) \parallel \ldots \parallel \mathrm{Ord}(m_1) \parallel \mathrm{Ord}(m_0),$$

где $\parallel$ – символ конкатенации, то есть склеивания битовых последовательностей. К примеру, в предложении «abcd» символ «a» имеет ASCII-код 97 (или `0x61`), символ «b» – 98 (или `0x62`), символ «c» – 99 (или `0x63`), символ «d» – 100 (или `0x64`). Значит, этому предложению будет соответствовать шестнадцатеричное число `0x64636261`, которое в десятичной системе счисления записывается в виде 1684234849.

Каждое сообщение, зашифрованное с помощью криптосистемы RSA, в числовом представлении не должно превосходить значение ключевого параметра n. В противном случае сообщение разбивается на блоки подходящего размера, каждый из которых зашифровывается отдельно.

В распоряжении криптоаналитика имеются:

- пара чисел (e, n), соответствующих открытому ключу криптосистемы RSA;
- перехваченные блоки закрытого текста y.

$$n = 91125205046729970955254 11, \qquad e = 6910762382649221.$$

y: `0x41105349274c9bb84e96f` `0x742f2ac0de06cb03db07f` `0x4cfb689df9d8fcdffea5`
`0x30d0ee377534705bd357e` `0x5ceac4ce76c95ab6e863f` `0x1f1341dcd3c6296bcb4e`
`0x233aa0fd40b1071a2f24c` `0x233e8de3a8464177fefab` `0x1bfc370eca17fccbd0cb2`
`0x678dc186134b4aac67d9a` `0x2b260c08d5d08cc45f7b5` `0x60e06614fdcef55576392`
`0x437222c7c2e4709097b13` `0x725a08e173e9d680be17b` `0x4789c6ebc006979788f0`
`0x5603de9c33f3ab68b278c` `0x6deb93975b248e876aea1` `0x67716a9738c613b2b203f`
`0x6e9c9bd07761585cd03c9` `0x28c6e7a5001d13c05525b`

Необходимо выполнить следующее:

- изучить и описать схему работы криптосистемы RSA;
- написать программную реализацию RSA на языке программирования C или Python, способную работать с числами произвольной разрядности;
- изучить и описать метод Ферма факторизации целых чисел;
- написать программную реализацию метода Ферма на языке программирования C или Python, способную работать с числами произвольной разрядности;
- по известному открытому ключу (e, n) криптосистемы RSA определить закрытый ключ (d, n) и с его помощью восстановить открытый текст x, из которого был получен закрытый текст y.

Полезные источники

1. *Глухов М.М., Круглов И.А., Пичкур А.Б., Черемушкин А.В.* Введение в теоретико-числовые методы криптографии: Учебное пособие. – СПб.: Издательство «Лань», 2011.
   Здесь, помимо очевидно необходимого описания метода Ферма, излагается алгоритм извлечения квадратного корня, который понадобится реализовать на определённом этапе работы.
2. *Лось А.Б, Нестеренко А.Ю., Рожков М.И.* Криптографические методы защиты информации: учебник для академического бакалавриата. – 2-е изд., испр. – М.: Издательство Юрайт, 2019.
3. *Черемушкин А.В.* Лекции по арифметическим алгоритмам в криптографии. – М.: МЦНМО, 2002.