Лабораторная работа № 4.

КРИПТОСИСТЕМА RSA

Эта криптосистема использует одностороннюю функцию с потайным ходом, предложенную в 1978 году в статье Ривеста, Шамира и Адлмана. Односторонняя функция с потайным ходом, используемая в RSA, – это дискретное возведение в степень $Y = f_z(X) = E_{(e,n)}(X) = X^e \bmod n$, где открытое сообщение X представляет собой положительное целое, не превосходящее n. Модуль $n = pq$, где p и q - это большие неравные простые числа, e - положительное целое, удовлетворяющее условиям $e \le \phi(n)$, $HOD(e, \phi(n)) = 1$, где $\phi(n) = (p-1)(q-1)$ - функция Эйлера (число положительных целых i, не превосходящих n и взаимно простых с n. Публикация алгоритма шифрования $E_z = E_{(e,n)}$ - это публикация открытого ключа (e,n).

Обратная функция имеет вид $f_z^{-1}(Y) = Y^d (\bmod n)$, где d - это единственное положительное целое меньшее n и удовлетворяющее условию $de = 1(\bmod \phi(n))$. Таким образом, дешифрование также выполняется как возведение в степень, но с другим показателем d, являющимся закрытым ключом

$$Y = D_{(d,n)}(Y) = Y^d (\bmod n).$$

В основе согласованности этих преобразований лежит так называемая ***малая теорема Ферма***, в соответствии с которой для каждого числа X такого, что $X \in \{1, 2, ..., p-1\}$, где $p > 1$ - простое,

$$X^{p-1} \equiv 1 \bmod p.$$

Если обе стороны сравнения домножить X, то получится сравнение

$$X^p = X \bmod p,$$

которое верно и при $X = 0$. Мы знаем, что e и d взаимно обратны по модулю $\phi(n)$, т.е.

$$ed = 1 + k(p-1)(q-1)$$

для некоторого целого k. Тогда шифрование и дешифрование в системе RSA можно описать как

$$\left(X^e \bmod n\right)^d \bmod n = X^{ed} \bmod n = X\left(X^{p-1}\right)^{k(q-1)} \bmod pq.$$

По малой теореме Ферма $X^{ed} = X1^{k(q-1)} = X \bmod p$. Аналогично $X^{ed} = X \bmod q$, отсюда следует $X^{ed} = X \bmod n$.

Надежность системы RSA основывается на трудности решения задачи разложения составного числа на множители. Временная сложность существующих алгоритмов разложения числа n на простые множители пропорциональна $\sqrt{n} = 2^{(\log_2 n)/2}$. Если противник разложит число n на множители p и q, то секретный ключ d он может найти, пользуясь расширенным алгоритмом Евклида, определяющим такие числа d и k, что $ed + k(p-1)(q-1) = 1$. Временная сложность расширенного алгоритма Евклида пропорциональна $(\log_2 n)^2$.

Пример построения системы RSA.

Пусть $p = 17$, $q = 31$, $e = 7$. Тогда $n = 17 \cdot 31 = 527$, $\phi(n) = 16 \cdot 30 = 480$. Найдем секретный ключ d, пользуясь расширенным алгоритмом Евклида. Этот алгоритм находит наибольший общий делитель (НОД) чисел $\phi(n) = 480$ и $e = 7$ и коэффициенты в разложении НОД, т.е. $НОД(\phi(n), e) = ed + k\phi(n)$. Расширенный алгоритм Евклида для нахождения $НОД(a_0, a_1)$, где $a_0 > a_1$ - целые положительные числа, состоит в нахождении последовательности остатков a_i, $i = 2,3,..., j$ от деления a_{i-2} на a_{i-1}:

$$a_i = a_{i-2} - Q_{i-1}a_{i-1},$$

где $Q_{i-1} = \left\lfloor \frac{a_{i-2}}{a_{i-1}} \right\rfloor$. Если $a_j = 0$, т.е. a_{j-1} делит a_{j-2} нацело, то $НОД(a_0, a_1) = a_{j-1}$. Для нахождения коэффициентов разложения $НОД(a_0, a_1) = a_0 x + a_1 y$ выполняем инициализацию $x_0 = 1$, $y_0 = 0$, $x_1 = 0$, $y_1 = 1$ и находим последовательности коэффициентов x_i, y_i, $i = 2,3,..., j-1$ по формулам

$$x_i = x_{i-2} - Q_{i-1}x_{i-1},\ y_i = y_{i-2} - Q_{i-1}y_{i-1}.$$

Коэффициенты x_{j-1}, y_{j-1} являются искомыми коэффициентами разложения. В нашем примере $a_0 = 480$, $a_1 = 7$, получаем

$$a_2 = 480 - 68 \cdot 7 = 4$$
$$a_3 = 7 - 1 \cdot 4 = 3$$
$$a_4 = 4 - 1 \cdot 3 = 1$$
$$a_5 = 3 - 3 \cdot 1 = 0.$$
$$x_2 = 1 - 68 \cdot 0 = 1$$
$$x_3 = 0 - 1 \cdot 1 = -1$$
$$x_4 = 1 - 1 \cdot (-1) = 2.$$
$$y_2 = 0 - 68 \cdot 1 = -68$$
$$y_3 = 1 - 1 \cdot (-68) = 69$$
$$y_4 = -68 - 1 \cdot 69 = -137.$$

Получаем, что $НОД(480,7) = 1 = 480 \cdot 2 - 7 \cdot 137$ и наш секретный ключ равен $d = -137 \bmod 480 = 343$.

Пусть $X = 2$ тогда шифртекст равен

$$Y = X^e \bmod n = 2^7 \bmod 527 = 128.$$

Для дешифрации используем секретный ключ, т.е. вычислим

$$Y^d \bmod n = 128^{343} \bmod 527 =$$
$$= 128^{256}128^{64}128^{16}128^{4}128^{2}128 \bmod 527 = 35 \cdot 256 \cdot 35 \cdot 101 \cdot 47 \cdot 128 \bmod 527 = 2 \bmod 527.$$

Криптоанализ системы RSA показал, что для стойкого шифрования необходимо соблюдение следующих условий:

1. Числа p и q должны быть достаточно большими, не слишком сильно различаться, но и быть не слишком близкими.

2. $НОД(p-1, q-1)$ должен быть небольшим
3. Числа p и q должны быть ***сильно простыми*** (простое число r называется сильно простым, если $r+1$ имеет большой простой делитель, а $r-1$ имеет большой простой делитель s такой, что $s-1$ также имеет большой простой делитель.

Для эффективной реализации системы RSA необходимо эффективно вычислять x^n по данным x и n. Запишем n в двоичной системе счисления. Затем заменим каждую 1 – парой символов SX, а 0 – символом S и вычеркнем крайнюю слева пару SX. Результат представляет собой правило вычисления x^n, где S трактуется как возведение в квадрат, а X как умножение на x. Например, пусть нужно вычислить x^{25}, тогда получаем 25 в двоичной системе счисления имеет вид 11001, т.е. имеем последовательность SXSSSX или

$$x^{25} = ((((x^2)x)^2)^2)^2 x.$$

Так как простые числа, используемые в системе RSA, имеют разрядность порядка 100 бит, то такие известные методы поиска простого числа, как, например, "Решето Эратосфена" обычно не используются. Простые числа находят с помощью тестов на простоту. Наиболее широко используемый в криптографии тест – тест Миллера-Рабина. Этот тест можно описать с помощью следующего алгоритма:

```
Miller-Rabin(p,s)
for j = 1:s   do
  a ← Random(1, p-1)
   if Witness(a, p)
   then return Composite
 return Prime
```

где

```
Witness(a,p)
Пусть (b_k, b_{k-1}, ..., b_0) - двоичная запись числа p-1
d ← 1
        for i = k:0:-1 do
                x ← d
                d ← (d · d) mod p
                        if d = 1 и x ≠ 1 и x ≠ p-1
                        then return TRUE
                if b_i = 1 then d ← (d · a) mod p
if d ≠ 1 then return TRUE
return FALSE
```

Этот тест проверяет равенство $a^{p-1} = 1 \bmod p$ для s чисел от 1 до p-1. При этом процедура Witness выполняет возведение в степень по описанному выше алгоритму и на каждом шаге проверяет, не получили ли мы для некоторой четной степени a меньшей, чем $p-1$, равенство 1 (при этом $a \neq 1$ и $a \neq p-1$). Так как в этом случае число заведомо составное, то продолжать возведение в степень не имеет смысла. Тест основан не следующей теореме, которая приведена здесь без доказательства.

Теорема.

При простом p *уравнение* $x^2 = 1 \bmod p$ *имеет ровно два решения* $x = 1$ *и* $x = -1 = p - 1 \bmod p$.

Покажем, что для случая составного p корни могут быть другими. Пусть $p = 12$ тогда кроме $1^2 = 11^2 = 1 \bmod 12$ имеем $5^2 \bmod 12 = 1$, и $7^2 = 1 \bmod 12$.

Заметим, что тест не дает гарантии, что найденное число простое и его необходимо проверить, например, делением на простые множители вплоть до $\sqrt{p}$.

ЗАДАНИЕ

1. Построить поле Галуа $GF(p^m)$ по модулю заданного примитивного полинома.
2. С помощью теста Миллера-Рабина выбрать 2 простых числа и вычислить модуль n как произведение этих чисел.
3. С помощью расширенного алгоритма Евклида выбрать пару: открытый, e, и закрытый, d, ключ.
4. Зашифровать заданный файл по алгоритму RSA с использованием пары (n, e).
5. Дешифровать полученный шифртекст по алгоритму RSA с использованием пары (n, d).

ИСХОДНЫЕ ДАННЫЕ: Примитивный полином и простое число p. Исходный файл.

СОДЕРЖАНИЕ ОТЧЕТА: Отчет по лабораторной работе должен содержать

1. Построенное поле $GF(p^m)$.
2. Описание алгоритма шифрования
3. Описание алгоритма дешифрования
4. Выводы по работе.