4.4. Алгоритм RSA

Алгоритм, разработанный Ривестом, Шамиром и Адлеманом, использует выражения с экспонентами. Данные шифруются блоками, каждый блок рассматривается как число, меньшее некоторого числа n. Шифрование и дешифрование имеют следующий вид для некоторого незашифрованного блока M и зашифрованного блока C.

$C = M^e \pmod n$

$M = C^d \pmod n = (M^e)^d \pmod n = M^{ed} \pmod n$

Как отправитель, так и получатель должны знать значение n. Отправитель знает значение e, получатель знает значение d. Таким образом, *открытый ключ* есть KU = {e, n} и *закрытый ключ* есть KR = {d, n}. При этом должны выполняться следующие условия:

1. Возможность найти *e, d* и *n* такие, что $M^{ed} = M \bmod n$ для всех $M < n$.
2. Относительная легкость вычисления M^e и C^d для всех значений $M < n$.
3. Невозможность определить d, зная e и n.

Рассмотрим некоторые математические понятия, свойства и теоремы, которые позволят нам определить e, d и n.

1. Если $(a \cdot b) \equiv (a \cdot c) \bmod n$, то $b \equiv c \bmod n$, если a и n взаимнопростые, т.е gcd (a, n) = 1.
2. Обозначим Z_p - все числа, взаимнопростые с p и меньшие p. Если p - простое, то Z_p - это все остатки. Обозначим w^{-1} такое число, что $w \cdot w^{-1} \equiv 1 \bmod p$.

   Тогда $\forall w \in Z_p \; \exists z: w \cdot z \equiv 1 \bmod p$
3. Определим функцию Эйлера следующим образом: Φ(n) - число положительных чисел, меньших n и взаимнопростых с n. Если p - простое, то $\Phi(p) = p-1$.

   Если p и q - простые, то $\Phi(p \cdot q) = (p-1) \cdot (q-1)$.
4. *Теорема Ферма*.

   $a^{n-1} \equiv 1 \bmod n$, если n - простое.
5. *Теорема Эйлера*.

   $a^{\Phi(n)} \equiv 1 \bmod n$ для всех взаимнопростых a и n.

   Теперь рассмотрим все элементы *алгоритма RSA*.

p, q - два простых целых числа	-открыто, вычисляемо.
$n = p \cdot q$	- закрыто, вычисляемо.
d, gcd (Φ(n), d) = 1;	- открыто, выбираемо.
$1 < d < \Phi(n)$	
$e \equiv d^{-1} \bmod \Phi(n)$	- закрыты, выбираемы.

Закрытый ключ состоит из {d, n}, *открытый ключ* состоит из {e, n}. Предположим, что пользователь A опубликовал свой *открытый ключ*, и что пользователь B хочет послать пользователю A сообщение M. Тогда B вычисляет $C = M^e \pmod n$ и передает C. При получении этого

зашифрованного текста пользователь A дешифрует вычислением $M = C^d \pmod n$.

Суммируем *алгоритм RSA*:

Создание ключей

Выбрать простые p и q

Вычислить $n = p \cdot q$

Выбрать d $\gcd(\Phi(n), d) = 1;\ 1 < d < \Phi(n)$

Вычислить e $e = d^{-1} \bmod \Phi(n)$

Открытый ключ $KU = \{e, n\}$

Закрытый ключ $KR = \{d, n\}$

Шифрование

Незашифрованный текст: $M < n$

Зашифрованный текст: $C = M^e \pmod n$

Дешифрование

Зашифрованный текст: C

Незашифрованный текст: $M = C^d \pmod n$

Обсуждение криптоанализа

Можно определить четыре возможных подхода для криптоанализа *алгоритма RSA*:

1. Лобовая атака: перебрать все возможные *закрытые ключи*.
2. Разложить n на два простых сомножителя. Это даст возможность вычислить $\Phi(n) = (p-1) \cdot (q-1)$ и $d = e^{-1} \pmod{\Phi(n)}$.
3. Определить $\Phi(n)$ непосредственно, без начального определения p и q. Это также даст возможность определить $d = e^{-1} \pmod{\Phi(n)}$.
4. Определить d непосредственно, без начального определения $\Phi(n)$.

Защита от лобовой атаки для *RSA* и ему подобных алгоритмов приводится далее.