4.5. Алгоритм обмена ключа Диффи-Хеллмана

Цель алгоритма состоит в том, чтобы два участника могли безопасно обменяться ключом, который в дальнейшем может использоваться в каком-либо алгоритме симметричного шифрования. Сам *алгоритм Диффи-Хеллмана* может применяться только для обмена ключами.

Алгоритм основан на трудности вычислений *дискретных логарифмов*. *Дискретный логарифм* определяется следующим образом. Вводится понятие *примитивного корня простого числа* Q как числа, чьи степени создают все целые от 1 до Q - 1. Это означает, что если A является *примитивным корнем простого числа* Q, тогда числа

$A \bmod Q, A^2 \bmod Q, \ldots, A^{Q-1} \bmod Q$

являются различными и состоят из целых от 1 до Q - 1 с некоторыми перестановками. В этом случае для любого целого $Y < Q$ и *примитивного корня* A простого числа Q можно найти единственную экспоненту X, такую, что

$Y = A^X \bmod Q$, где $0 \leq X \leq (Q - 1)$

Экспонента X называется *дискретным логарифмом*, или индексом Y, по основанию A mod Q. Это обозначается как

$\text{ind}_{A, Q}(Y)$.

Теперь опишем алгоритм обмена ключей *Диффи-Хеллмана*.

Общеизвестные элементы. *Q*: простое число; *A*: $A < Q$ и *A* является примитивным корнем Q.

Создание пары ключей пользователем I. Выбор случайного числа X_i (*закрытый ключ*), $X_i < Q$. Вычисление числа Y_i (открытый ключ) $Y_i = A\ X_i \bmod Q$.

Создание *открытого ключа* пользователем J. Выбор случайного числа Xj (закрытый ключ) $Xj < Q$. Вычисление случайного числа Yj (открытый ключ) $Yj = A\ Xj \bmod Q$.

Создание общего секретного ключа пользователем I. $K = (Y_j)^{Xi} \bmod Q$.

Создание общего секретного ключа пользователем J. $K = (Y_i)^{Xj} \bmod Q$.

Предполагается, что существуют два известных всем числа: простое число Q и целое A, которое является *примитивным корнем* Q. Теперь предположим, что пользователи I и J хотят обменяться ключом для алгоритма симметричного шифрования. Пользователь I выбирает случайное число $X_i < Q$ и вычисляет $Y_i = A^{Xi} \bmod Q$. Аналогично пользователь J независимо выбирает случайное целое число $X_j < Q$ и вычисляет $Y_j = A^{Xj} \bmod Q$. Каждая сторона держит значение X в секрете и делает значение Y доступным для другой стороны. Теперь пользователь I вычисляет ключ как $K = (Y_j)^{Xi} \bmod Q$, и пользователь J вычисляет ключ как $K = (Y_i)^{Xj} \bmod Q$. В результате оба получат одно и то же значение:

$K = (Y_j)^{Xi} \bmod Q = (A^{Xj} \bmod Q)^{Xi} \bmod Q = (A^{Xj})^{Xi} \bmod Q = A^{Xj\ Xi} \bmod Q$

$= (A^{Xj})Xj \bmod Q = (A^{Xi} \bmod Q)^{Xj} \bmod Q = (Y_i)^{Xj} \bmod Q$

Таким образом, две стороны обменялись секретным ключом. Так как X_i и X_j являются закрытыми, противник может получить только следующие значения: Q, A, Y_i и Y_j. Для вычисления ключа атакующий должен взломать *дискретный логарифм*, т.е. вычислить

$X_j = ind_{a,\ q}(Y_j)$

Следует заметить, что данный алгоритм уязвим для атак типа "man-in-the-middle".