1)Алгоритм RSA

Генерация ключей

|  |  |  |
| --- | --- | --- |
| № п/п | Операция | Полученные значения |
| 1 | Выбираются два простых числа **p, q** | p=3 q=7 |
| 2 | Вычисляется **n**=pq | n = 21 |
| 3 | Вычисляется функция Эйлера **ф**=(p-1)(q-1) | ф = 12 |
| 4 | Выбирается **e** как число в интервале (0,n), взаимно простое с ф | е=5 |
| 5 | Вычисляется **d** как как обратное число к е по модулю ф  (de) mod ф = 1 | (5d) mod 12 = 1  d=5 |
| 6 | Сохраняются открытый и закрытый ключи | Открытый ключ (e,n)=(5,21)  Закрытый ключ d=5 |

Аутентификация на сервере

|  |  |  |
| --- | --- | --- |
| № п/п | Операция | Полученные значения |
| 1 | Сервер выбирает случайное число **k**, вычисляет **r**=(k^e)mod n и отсылает его клиенту | k=12  r=(12^5)mod21=3 |
| 2 | Клиент вычисляет **k’**=(r^d)mod n и отсылает на сервер | k’=12 |
| 3 | Если k=k’, то аутентификация пройдена | 12=12 |

2)Схема Шнорра

Генерация ключей

|  |  |  |
| --- | --- | --- |
| № п/п | Операция | Полученные значения |
| 1 | Выбираются два простых числа **p** и **q** такие, что (p - 1) mod q = 0 | p = 23, q = 11 |
| 2 | Выбирается секретный ключ **x** в интервале (1, q-1) | x = 6 |
| 3 | Выбирается **g** такое, что g^q mod p = 1 | g=12 |
| 4 | Вычисляется открытый ключ **y** такой, что (g^x\*y) mod p = 1 | y = 18 |
| 5 | Сохраняются открытый и закрытый ключи | Открытый ключ y=18  Закрытый ключ x=6 |

Аутентификация на сервере

|  |  |  |
| --- | --- | --- |
| № п/п | Операция | Полученные значения |
| 1 | На клиенте выбирается случайное число **k** в интервале (1,q-1), вычисляется **r** = gk mod p и посылается на сервер | k = 21 r = 2 |
| 2 | Сервер выбирает случайное число **e** в итервале (0,2t-1), где t - некоторый параметр, и посылает e  клиенту | e = 10 |
| 3 | На клиенте вычисляется **s** = (k + x \* e) mod q и посылается на сервер | s = 4 |
| 4 | Сервер проверяет соотношение **t** = (gs \* ye) mod p и, если оно выполняется, то аутентификация пройдена | t=2=r |

3)Схема Фейге-Фиата-Шамира

Генерация ключей

|  |  |  |
| --- | --- | --- |
| № п/п | Операция | Полученные значения |
| 1 | Выбирается модуль **n**, равный произведению двух простых чисел **p,q** | p = 5, q = 7, n = 35 |
| 2 | Выбирается число **v** (открытый ключ), являющееся квадратичным вычетом по модулю **n** и имеется обратное значение **v-1** по модулю **n** | v = 11 v-1 = 16 |
| 3 | Определяется закрытый ключ **s**, как наименьшее значение, удовлетворяющее следующему выражению s2 mod n = v-1 | s = 4 |
| 4 | Сохраняются открытый и закрытый ключи | Открытый ключ (v,n)=(11,35)  Закрытый ключ s=4 |

Аутентификация на сервере

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Операция | Полученные значения | |
| 1 | На клиенте выбирается случайное число **r** в интервале (1, …, n-1), вычисляется **z** = r2 mod n и посылается на сервер | r = 9 z = 11 | |
| 2 | Сервер посылает клиенту случайный бит **b**. | b = 0 | b = 1 |
| 3 | Если **b**=0, то клиент посылает на сервер **r**, иначе - **y** = (r \* s) mod n. | r = 9 | у = 1 |
| 4 | Если **b**=0, то сервер проверяет, что **z** = r2 mod n, иначе - **z** = (y2 \* v) mod n. | 11=11 | 11=11 |