**Цель:**  
Организовать взаимодействие клиент – сервер, выполняющего регистрацию, авторизацию и аутентификацию пользователя(клиента) по алгоритму SRP.  
  
**Пункты выполнения:**  
1. Запуск сервера и клиента;

2. Подключение клиента к серверу;

3. Регистрация пользователя:

* Ввод I(логин);
* Ввод p(пароль);
* Вычисление своей случайной строки s(соль), для каждого регистрируемого пользователя;
* Для каждого пользователя вычисляются свои g(генератор по модулю N) и N(безопасное простое число): N = 2\*q + 1 (q – простое число), g^x % N = X

(0 < X< N);

* x(закрытый ключ) вычисляется через функцию хеширования Hash, принимающую s(соль) и p(пароль): x = Hash(s,p);
* Вычисление пользователем своего v(верификатора пароля): v = g^x % N;
* Отправка серверу следующих параметров: I(логин), g (генератор по модулю N),

N (безопасное простое число), s(соль), v(верификатор пароля);

* Сервер принимает параметры от клиента и сохраняет их для дальнейшей аутентификации пользователя;

4. Авторизация пользователя:

* Ввод I(логин);
* Ввод p(пароль);
* Клиент по логину определяет необходимые значения: g(генератор по модулю N) и N(безопасное простое число), вычисленные раннее;
* Вычисление клиентом а (случайное секретное значение);
* Вычисление клиентом A (эфемерный ключ): A = g^a % N;
* Аутентификация пользователя:
* Отправка серверу следующих параметров: I(логин), A (эфемерный ключ);
* Сервер, по полученному логину, определяет необходимые значения:

g (генератор по модулю N), N (безопасное простое число), s(соль), v(верификатор пароля) клиента. Проверят A!=0;

* Вычисление сервером b (случайное секретное значение);
* Вычисление сервером B (эфемерный ключ): B = (k\*v + g^b % N) % N, где

k – параметр множителя и равен 3;

* Отправка клиенту B (эфемерный ключ), проверка клиентом B != 0;
* Сервер и клиент вычисляют u(маскированное значение): u = Hash(A,B);
* Вычисление клиентом x(закрытый ключ): x = Hash(s,p) (p – введенный пароль);
* Вычисление клиентом K(общий ключ сессии):

S = ((B - k\*(g^x % N)) ^ (a + u\*x)) % N

K = Hash(S)

* Вычисление сервером K(общий ключ сессии):

S = ((A\*(v^u % N)) ^ b) % N

K = Hash(S)

* Клиент генерирует подтверждение:

Mс = Hash( Hash(N) XOR Hash(g), Hash(I), s, A, B, K)

* Отправка клиента на сервер значения Mc;
* Сервер генерирует свою Ms:

Ms = Hash( Hash(N) XOR Hash(g), Hash(I), s, A, B, K)

* Если Ms != Mc, то сервер обрывает соединение без отправки Rs;
* Если Ms == Mc, то сервер вычисляет и отправляет клиенту Rs:

Rs = Hash( A, Ms, K );

* Клиент генерирует свою Rc:

Rc = Hash( A, Mc, K );

* Если Rc != Rs, то сервер обрывает соединение, если Rc == Rs, то успех -аутентификация прошла успешно, и клиент получил доступ к серверу.

**Вывод:**  
Пользователь регистрируется посредством ввода логина и пароля, пароль не передается в открытом виде серверу, это происходит с помощью верификатора пароля v, который сохраняется сервером. При авторизации также вводятся логин и пароль, сервер по логину ищет все необходимые данные по данному пользователю и происходит аутентификация. При вводе слишком больших паролей при регистрации, сервер при авторизации принимает неверные пароли похожие на истинный пароль.