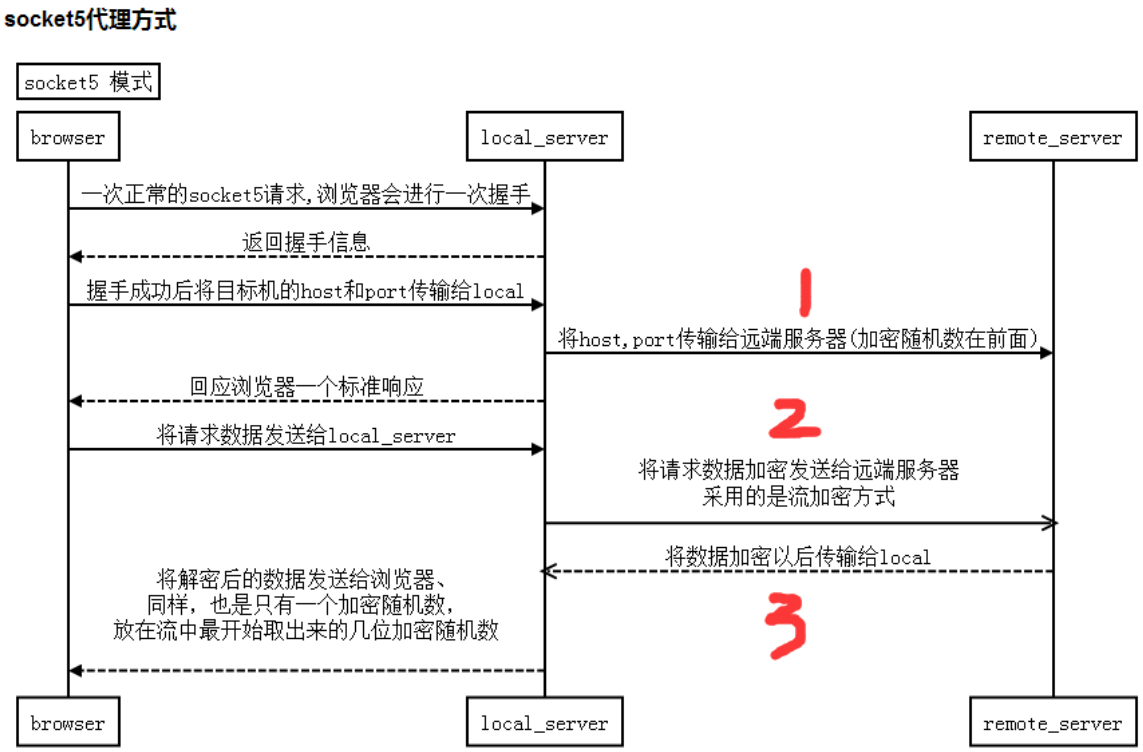
1，原有的shadowsocks翻墙原理实现：

python版本：



在**1**处传输给remote_server的是 16位加密随机数+encrypt(type(1byte) + host_len(1byte) + host(n bytes))
在**2**处传输给remote_server的也是 流加密的数据
在**3**处传输给local_server的是采用的流加密的数据

在这里传输使用的是aes cfb的流加密方式，密码是客户端和服务端使用相同的密码

2，我所实现的翻墙协议改进。

改进后的数据方式是：
在**1**处发送的是：
16位加密随机数+encrypt_1(user_name_len(1byte) + user_name(n bytes))
+
16位加密随机数 + encrypt_2(type(1byte) + host_len(1byte) + host(n bytes))

encryt_1处加密用户名（identity）使用的是公有密码，即 local 和 server使用的是同一个配置的公有密码

encryt_2 处加密使用的是每个用户单独配置的密码，以后再这个tcp连接中使用的都是这个密码进行的加密

3，代码改动细节

server端python代码改动
在tcprelay中新增了一个 handle_pass（ ）方法用于分析encrypt_1中的用户数据

4，快速开始，

server端采用python，local端采用java

1，配置server端的config_server.json

python server.py

2，配置local端的gui-cofig.json

运行 com.stfl.Main 类中的main方法