

主通信协议

| | | | | | | | |
|------|-----|-------|------|-------|-------|-------|--------|
| 16字节 | 8字节 | 8字节 | 32字节 | 1字节 | 4字节 | 1字节 | 1字节 |
| 认证头 | 时间戳 | noise | 校验和 | 主协议版本 | 数据块长度 | 子协议类型 | 随机填充长度 |
| 任意字节 | | | | | | | |
| 加密数据 | | | | | | | |
| 随机字节 | | | | | | | |
| 随机数据 | | | | | | | |

此为主通信协议。SC协议的子协议可以任意定制，主通信协议一般不做变动。

协议认证与解密

认证需要服务器与客户端预定密码PASS，且需要客户端与服务端时间同步
认证部分为 `sha256(用户密码+时间戳+noise)[:16]` 其中时间戳为8byte，网络字节序，noise为8字节随机值。认证头部分不加密。解密算法采用 `AES-128-CFB`
协议解密密钥为 `sha256(用户密码+认证头)[0:16]`
协议解密IV为 `sha256(用户密码+认证头)[16:32]`

在对协议头时间戳和noise进行解密之后，立即进行认证部分校验工作。其中时间戳与服务器时间误差需要在上下30s之内。入认证通过，记录noise的值，在60s的过期之内如果有其他连接使用相同noise，则认定为重放攻击包。立即断开连接。

开发注释

使用此认证主要是为了无特征和防重放。为了无特征，每次连接请求时的认证头应该不同，使用的加密密钥也应该不同。所以使用了noise和时间戳双重认证的办法。
有关noise重复问题。经计算，在每分钟100w连接数量的情况下，noise有重复的概率是亿分之2左右。以每分钟100w次请求不断跑190年左右可能发生一次noise重复。以可以忽略。

协议协商部分

协议协商部分头长度为48byte。

- 认证头
- 时间戳
- noise
- 校验和
 - 除了末尾随机填充之外的所有部分的sha256校验和。检验和字段在运算中用0字节代替

- 主协议版本
 - 当前协议版本，暂定为1
 - 数据块长度
 - 4字节数据块长度，标示包括协议头，末尾随机混淆数据在内的有效数据长度。
 - 子协议类型
 - 标识子协议的类型，即加密数据部分含义。有如下协议
 - 1，转发协议请求
 - 2，转发协议应答
 - 3，数据协议。即数据段内容为需要转发的数据包
 - 4-255 保留
 - 填充长度
 - 表示数据末尾的随机填充长度。末尾填充长度应该为0-256之间随机值。也可永久置0表示关闭此功能
 - 随机填充
 - 头保留字段。填0
-

开发注释

说实话，每一次数据包发送都有最起码 $48+32 = 80$ 字节的冗余稍微感觉有点多，可以考虑只在每一次连接的第一个包加入认证数据，并且压缩数据校验部分长度。

随机填充

在有效数据末尾填充0-255字节直接的随机填充。填充长度在协议头中给出，此部分中不包含任何有效信息，且会被处理程序直接丢弃。可不加密此部分以节约资源