# 开放式社交网络通信协议

Open Social Network Communication Protocol

## 版本

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 作者 | 版本 | 备注 |
| 2019-09-01 | 加农炮.雷 | v1.0 | 开放式社交网络通信协议初稿 |
|  |  |  |  |

## 地址（Address）

地址是作为网络上身份的唯一标识。

本协议中的地址是ECDSA公钥的散列组合，ECDSA采用Prime256V1曲线。

他的定义如下

|  |  |  |  |
| --- | --- | --- | --- |
| 字段尺寸 | 描述 | 数据类型 | 说明 |
| 2byte | Version | uchar | 地址使用版本号，初始版本为0x1000。 |
| 1byte | PUBLIC flag | uchar | ECDSA公钥采用压缩模式和非压缩模式两种  04表示非压缩模式，公钥长度为64byte  03 02 表示压缩模式，公钥长度为32byte |
| 64/32byte | Publickey | uchar[] | 非压缩公钥长度为64byte，压缩公钥只保留X，长度为32byte。 |
| 32byte | Shadow hash | uchar[] | 一个地址的生成需要2对椭圆加密密钥对，后一对在地址中仅保留公钥的hash |

该地址进行base58以后加上标志头。

地址标志头：OSN

Base58编码与bitcoin的base58编码一致。

明私钥：public key对应的私钥。

暗私钥：shadow hash对应的私钥。

## 网络地址(Network address)

需要网络地址时会用到这个结构。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段尺寸 | 描述 | 数据类型 | 说明 |
| 16byte | IPv6/4 | Uchar[16] | Ipv6地址，以网络字节顺序存储。IPv4读取最后4个字节。IPv4地址以16字节的IPv4映射位址格式写入结构。(12字节 00 00 00 00 00 00 00 00 00 00 FF FF, 后跟4 字节IPv4地址) |
| 2byte | port | Uchar[2] | 端口 |
|  |  |  |  |

## 变长字符串（Variable length string）

|  |  |  |  |
| --- | --- | --- | --- |
| 字段尺寸 | 描述 | 数据类型 | 说明 |
| 4byte | length | Uchar[4] | 字符串长度 |
|  | string | Uchar[] | 字符串内容 |
|  |  |  |  |

## 签名

本协议使用椭圆曲线数字签名算法(ECDSA)对交易进行签名 ECDSA 使用了prime256v1曲线

公钥以 04 <x> <y>的形式给出，支持压缩公钥，压缩公钥以03<x>或者02<x>的形式输出。x和y是表示曲线上点的坐标的32字节字符串。

签名格式（signdata）：签名使用DER 编码 将 r 和 s 写入一个字节流中(因为这是OpenSSL的默认输出)。70字节或者71字节长度。

## Base58

Base58编码与bitcoin的base58编码一致。

## 广播找人

节点p1有Bob的消息，但是p1并不知道Bob在哪儿，就会广播消息寻找Bob。

节点p1会告诉所有人，我是节点p1，我在哪儿，寻找Bob，当前时间是多少，我的地址是多少。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段尺寸 | 描述 | 数据类型 | 说明 |
| 4byte | magic | uint32 | OSNP |
| 2byte | version | uchar[2] | 第一版版本为10 00 |
| 4byte | length | uint32 | 长度 |
| 2byte | command | uchar[2] | 01 00（find） |
| 18byte | network | uchar[18] | 节点p1的IPv6/4地址 |
| 4byte | timestamp | uint32 | 时间戳 |
|  | who | vls | 找寻的人（Bob）的地址 |
|  | address | vls | 节点p1的地址 |
|  | message | vls | 附带的消息。 |
| 70/71byte | signature | signdata | 使用p1地址的明私钥签名 |

Message：message里的内容为随机数。Message的长度不超过64字节。

## 获取描述

可以通过获取描述指令获取某个节点的描述。描述中可能会包含该节点上用户消息的获取方式。

## 获取消息来源列表

该命令可以获取节点上有关于某人的消息来源列表。

返回消息格式：json

{“command”:”getlist”,

“version”:”1000”,

”who”:”OSNXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX”}

**返回**

**{“command”:”getlist”,**

**“version”:”1000”,**

**“to”:”OSNXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX”,**

**“from”:[“user1”,”user2”,”user3”]**

**}**

## 获取指定来源的消息

该命令可以获取你需要的需要的消息。

{“command”:”getmsg”,

“version”:”1000”,

“to”:”OSNxxxxxxxxxxxxxxxxxxxxxxxxxx”,

“from”:”OSNxxxxxxxxxxxxxxxxxxxxxxxxxxx”}

返回：

{“command”:”getmsg”,

“version”:”1000”,

“to”:”OSNxxxxxxxxxxxxxxxxxxxxxxxx”,

“message”:

{“hash1”:{message},

“hash2”: {message},

“hash3”: {message}}

}

## 发送收到回执

该命令在收到消息发送给服务器的回执。

{“command”:”complete”,

“version”:”1000”,

“from”:”OSNXXXXXXXXXXXXXXXXXXXX”,

“to”:”OSNXXXXXXXXXXXXXXXXXXXX”,

“sign”:”xxxxxxxxxxxxxxxxxxxxxxxxxxxxx”,

“msghash”:[“hash1”,”hash2”,”hash3”]}

Sign:签名是将所有的hash链接成字符串以后签名。

## 消息层协议

消息层协议是在用户发送消息时使用，见getmsg中的hash字段的message。

{“command”:”message”,

“version”,”1000”

“from”:”OSNXXXXXXXXXXXXXXXXXXXXXXXXXXXX”,

“to”:” OSNXXXXXXXXXXXXXXXXXXXXXXXXXXXX”,

“crpyto”:”yes/no”,

“content”:”base58 string”,

“description”: ”base58 string”,

“hash”:”hash”,

“timestamp”:””

“sign”:”xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx”}

content的生成方式：

content使用base58编码，采用asn1格式，并分成key和cipher两部分。

key中包含了随机密码，其生成方式为

1.由客户端生成随机SHA256字符串。

2.使用to的公钥对随机SHA256进行加密。

cipher的生成方式为

1.使用随机SHA256作为key对需要发送的明文进行加密。

2.加密方式采用AES128cbc，hash前128bit作为key，后128bit作为vi。

key和cipher组成asn1格式以后使用base58编码，如果crypto为no，则解码以后无需解密。

解密需要from的私钥先解码出key，然后用key解码cipher。

description生成方式同content。

sign的生成方式：

使用通讯录中的key对hash进行签名。

### 文件传输

Description解密以后的内容为

{

“command”:” file”,

“type”:”audio/video/file”,

“name”:”file.txt”,

“total”:number,

“count”:number,

“size”:number

}

### 更改账号

content解密以后的内容为

{

“command”:” changeac”,

“to”:” OSNXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX”

“newaccount”:” OSNXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX”,

“publickey”:” 04xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx”,

“sign”:”暗私钥的签名，对command+to+newaccount进行签名”

}

### 回复更改账号

content解密以后的内容为

{

“command”:” rechangeac”,

“newaccount”:” OSNXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX”,

}

### 请求添加好友

content解密以后的内容为

{

“command”:” add”,

“info”:” xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx”,

}

### 回复请求添加好友

content解密以后的内容为

{

“command”:” add”,

“result”:”approval/refuse”,

“info”:” xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx”,

}

### 请求更新公共信息

content解密以后的内容为

{

“command”:” update”,

“hash”:””

}

### 向你的朋友更新你的公共信息

content解密以后的内容为

{

“command”:”update”,

“timestamp”:””,

“type”:”person”,

“nickname”:””,

“face”:””,

“personalized”,””

}

### 介绍你的朋友给另外一个朋友

Bob希望介绍Alice和Chris成为朋友。

{

“command”:”introduce”,

”account”:”OSNXXXXXXXXXXXXXXXX”,

“nickname”:”Chris”,

“description”:””

}

### 收到介绍以后的回复

拒绝：

Content解密后的内容：

{

“command”:”reintroduce”,

“action”:”refuse”,

“description”:””

}

同意：

Content解密后的内容

{

“command”:”reintroduce”,

“action”:”approval”,

}

Description解密后的内容

{

“command”:”create”

“from”:”osnxxxxxxxxxxxxxxxxxx” //Alice

“to”:”osnxxxxxxxxxxxxxxxxx” //Chris

“newaccount”:”osnxxxxxxxxxxxxxxxxxxxxx” //Alice2

“sign”:”” // by Alice

}

Description由Chris解密。Content由Bob解密。

### 中转消息

Bob收到消息reintroduce以后，会发送一个中转消息。

{

“command”:”accept”

“content”:””

}

Content为discription中的密文。