**SSH secure shell protocol**

1. **简介**

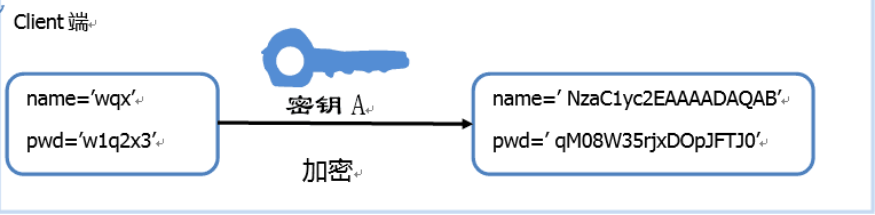
SSH是一种协议标准，其目的是实现安全远程登录以及其它安全网络服务。

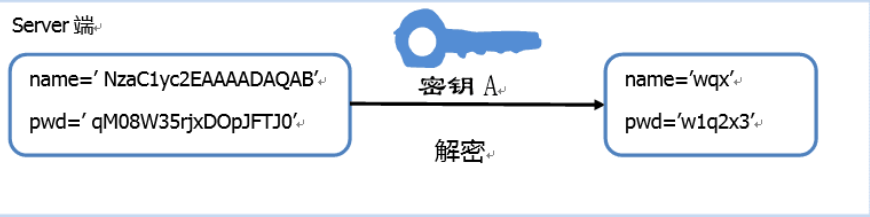
SSH仅仅是一协议标准，其具体的实现有很多，既有开源实现的OpenSSH，也有商业实现方案。使用范围最广泛的当然是开源实现OpenSSH。

1. **原理**

SSH加密的方式主要有两种：

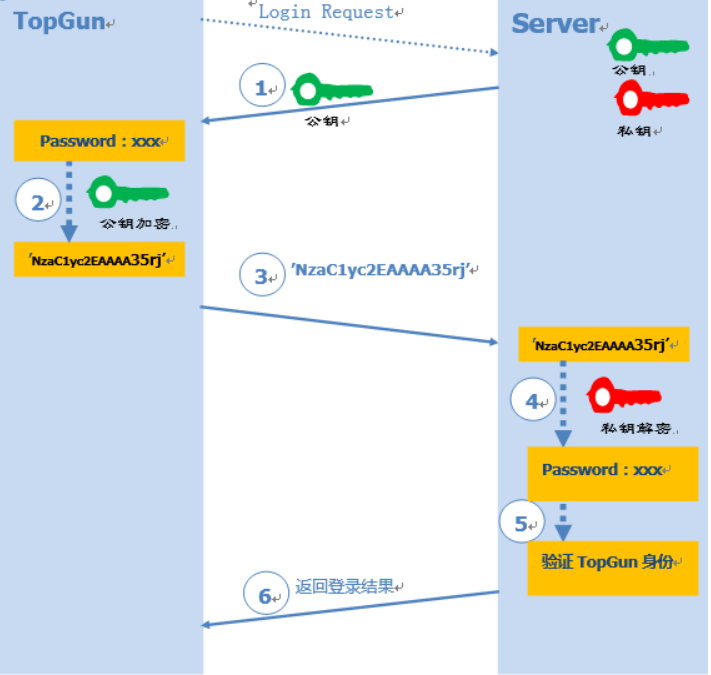
**对称加密**（也称为秘钥加密）加密解密使用同一套秘钥。如下图所示：





对称加密的加密强度高，很难破解。但是在实际应用过程中不得不面临一个棘手的问题：**如何安全的保存密钥呢？**尤其是考虑到数量庞大的Client端，很难保证密钥不被泄露。一旦一个Client端的密钥被窃据，那么整个系统的安全性也就不复存在。为了解决这个问题，**非对称加密**应运而生。非对称加密有两个密钥：**“公钥”**和**“私钥”**。

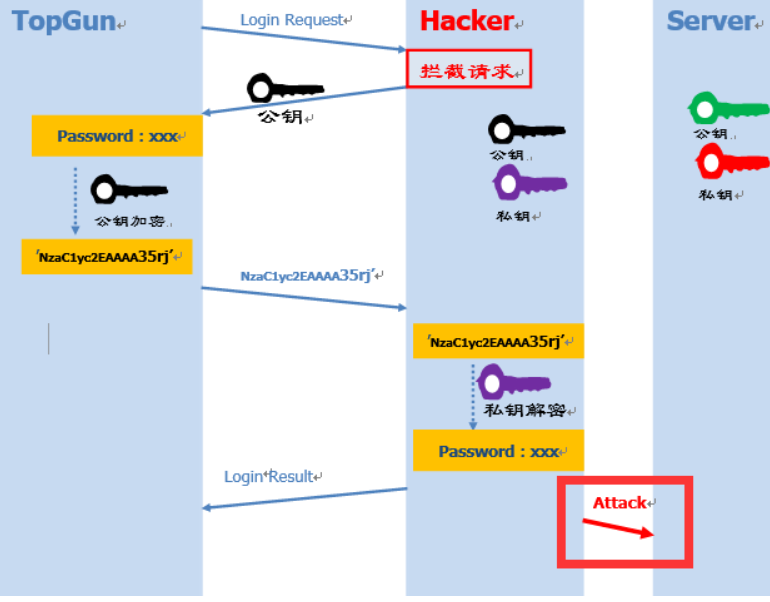
**非对称加密**（也称公钥加密），



远程Server收到Client端用户TopGun的登录请求，Server把自己的公钥发给用户。Client使用这个公钥，将密码进行加密。Client将加密的密码发送给Server端。远程Server用自己的私钥，解密登录密码，然后验证其合法性。若验证结果，给Client相应的响应。

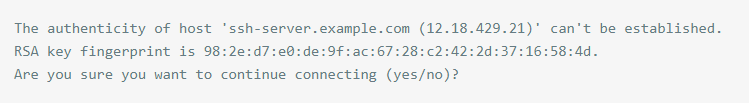
私钥是Server端独有，这就保证了Client的登录信息即使在网络传输过程中被窃据，也没有私钥进行解密，保证了数据的安全性，这充分利用了非对称加密的特性。

但是存在中间人攻击。



1. **基于口令的认证**

从上面的描述可以看出，问题就在于如何对Server的公钥进行认证？在https中可以通过CA来进行公证，可是SSH的publish key和private key都是自己生成的，没法公证。只能通过Client端自己对公钥进行确认。通常在第一次登录的时候，系统会出现下面提示信息：



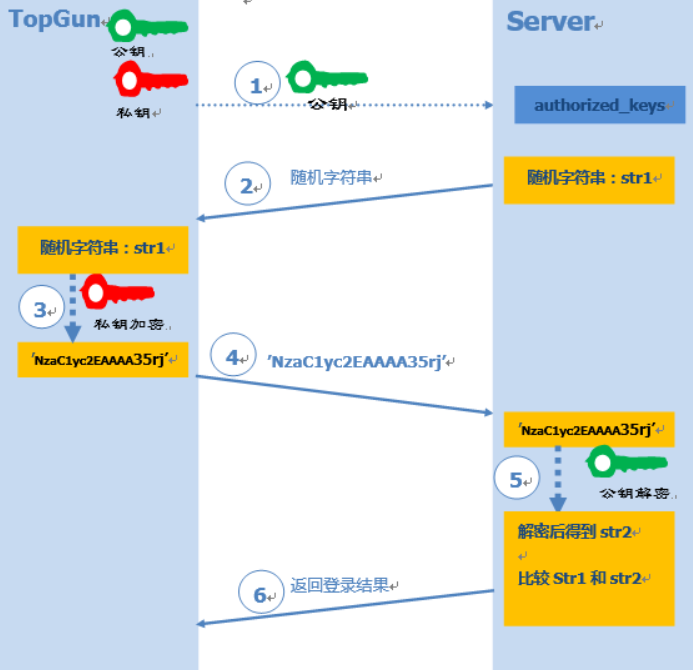
如果输入yes后，会出现下面信息：



该host已被确认，并被追加到文件known\_hosts中，然后就需要输入密码，之后的流程就按照上面的进行。

1. **基于公钥认证**

在上面介绍的登录流程中可以发现，每次登录都需要输入密码，很麻烦。SSH提供了另外一种可以免去输入密码过程的登录方式：公钥登录。流程如下：



Client端用户TopGun将自己的公钥存放在Server上，追加在文authorized\_keys中。Server收到登录请求后，随机生成一个字符串str1，并发送给Client。Client用自己的私钥对字符串str1进行加密。将加密后字符串发送给Server。Server用之前存储的公钥进行解密，比较解密后的str2和str1。根据比较结果，返回客户端登陆结果。

1. 启动服务

默认启动

/etc/init.d/sshd start

chconfig sshd on

1. **默认设置优化**

vi /etc/ssh/sshd\_config 查看配置文件

UserDNS no

GSSAP IAuthentication no

1. **Linux实践**

基于口令：

ssh -p22 [root@10.0.0.3](mailto:root@10.0.0.3) [OP]可以加命令

选择yes后会提示输入密码

1. **生成密钥对，然后分发**

ssh-keygen -t rsa

ssh-copy-id [-i [identity\_file]] [user@]machine

ssh-copy-id -i .ssh/id\_rsa.pub "-p 1212 [root@10.0.0.5](mailto:root@10.0.0.5)"

ssh-copy-id -i .ssh/id\_rsa.pub [root@10.0.0.5](mailto:root@10.0.0.5) 默认端口22