## SSH Key：两个简单步骤实现SSH无密码登录

Public Key认证的基本原理是基于分对称加密方式，分别在服务端对一段数据通过公钥进行加密，如果客户端能够证明其可以使用私钥对这段数据进行解密，则可以说明客户端的身份。因为服务端需要使用客户端生成的密钥对的公钥对数据首先加密，所以需要先将公钥存储到服务端的密钥库（Auhtorized Key）。*还记得Github中使用git协议push代码前需要先添加SSH KEY吗？*

下面详细介绍一个通过Public Key进行客户端认证的过程。

1. 客户端发起一个Public Key的认证请求，并发送RSA Key的模数作为标识符。（如果想深入了解RSA Key详细 --> [维基百科](http://en.wikipedia.org/wiki/RSA_(algorithm))）
2. 服务端检查是否存在请求帐号的公钥（Linux中存储在~/.ssh/authorized\_keys文件中），以及其拥有的访问权限。如果没有则断开连接
3. 服务端使用对应的公钥对一个随机的256位的字符串进行加密，并发送给客户端
4. 客户端使用私钥对字符串进行解密，并将其结合session id生成一个MD5值发送给服务端。 *结合session id的目的是为了避免攻击者采用重放攻击（replay attack）。*
5. 服务端采用同样的方式生成MD5值与客户端返回的MD5值进行比较，完成对客户端的认证。

无密码访问

1. 在需要无需密码登陆远程服务器的机器上生成密钥对：#ssh-keygen –t rsa

执行上面一步，会在~.ssh目录下生成两个文件id\_rsa和id\_rsa.pub，其中id\_rsa是私钥，保存在本机；id\_rsa.pub是公钥，要上传到远程服务器的

1. 上传公钥到需要无密码登陆的远程服务器B上并改名为authorized\_keys:

#chmod 755 .ssh

#scp .ssh/id\_rsa.pub [root@192.168.3.34:/root/.ssh/authorized\_keys](mailto:root@192.168.3.34:/root/.ssh/authorized_keys)

1. 测试

普通用户ssh无密码登陆： ssh root@10.0.0.2