**GitHub - 配置Git免密登陆GitHub**

**1、为什么要配置Git免密登陆**

在使用Git的过程中，每次向GitHub仓库推送代码，或者从GitHub上拉取数据的时候，都要输入Git的账号与密码，进行身份验证才可访问，非常麻烦。那么如何才能避免每次推送代码都要输入账号密码呢？

我们要设置Git免密登录，使Git本地版本库可以免密登录并访问GitHub，也就是配置SSHKey。

即：因为我们是要把Git版本库中的代码推送到GitHub上，GitHub上需要对我们版本库进行验证，所以要输入Git的用户名和密码。

**2、免密登录原理**

**（1）兔密登录机制**

Git主机间的通信采用的是SSH协议，即Sercure Shell协议。

该协议的兔密登录机制，要求主机之间采用SSH-key，即SSH密钥的方式进行身份验证。

SSH密钥包含“公钥与私钥"，所以我们首先要了解什么是“公钥与私钥”，然后还要理解“公钥与私钥”在免密登录中的作用，即免密登录的工作原理。

**（2）公钥与私钥** 对于公钥与私钥，要了解以下三点：

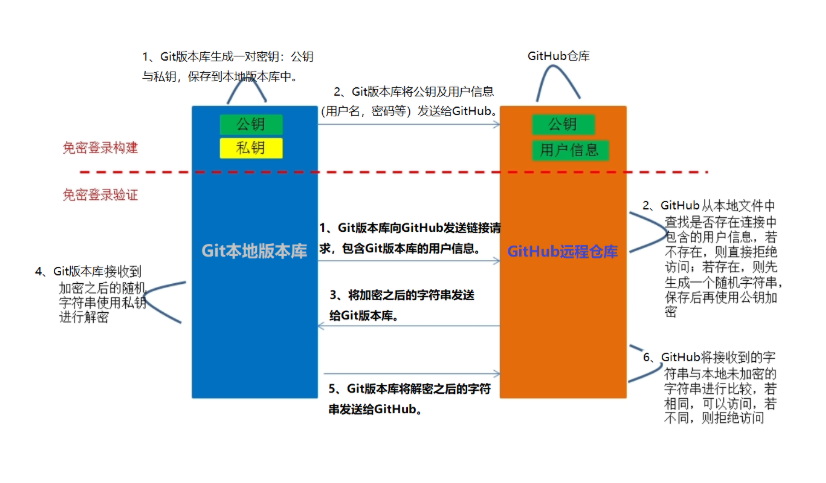
1. “公钥与私钥加密”是一种“不对称的加密方式”，是传统“对称加密方式”功能的增强。
2. 公钥与私钥是成对的，即一个公钥对应一个私钥。使用公钥加密后，只能使用私钥进行解密。
3. 公钥与私钥的关系：就好比“锁与钥匙”的关系。公钥相当于“锁”，锁是可以被他人看到的，是要发送给别人的，所以称为公钥。私钥相当于“钥匙”，它是不能公开的，只能有公钥发出者保存。

**（3）免密码登录的工作原理**

对于免密登录的机制，主要由两部分构成：构建与验证。

* 免密登录构建： Git版本库所在的主机上生成一对密钥：公钥与私钥，保存到本地主机中。 Git版本库将公钥及用户信息（用户名，密码等）保存到GitHub上。
* 免密登录验证： 1、Git版本库向GitHub发送连接请求，包含Git版本库的用户信息。 2、GitHub从本地文件中查找是否存连接中包含的用户信息。若不存在：则拒绝访问，若存在：可以访问。 3、GitHub将加密后的随机字符串，发送给Git版本库。 4、Git版本库将加密后的随机字符串，使用私钥进行解密。 5、Git版本库解密后的字符发送给GitHub。 6、GitHub将接收到的字符串，与本地未加密的字符串进行比较，相同则可以访问，不同则被拒绝。

免密登录原理示意图：



简单的理解：

SSH Key也可以简单的理解为你的身份标识，放在GitHub上面标明你是这个项目的一个开发人员，但是别人可以截获，但是你本机上的私钥无法截获，SSH Key也就保证了每次传输是安全的。

**3、设置本地库对GitHub的免密登录**

此时我们要设置Git本地库对GitHub的免密登录，密钥对应由本地版本库主机生成，而GitHub中只需要保存公钥即可。

**（1）在本地版本库主机中生成SSHkey密钥**

**1）生成密钥**

进入Git bash客户端，执行命令ssh-keygen -t rsa -C "邮箱地址"，来生成SSHkey密钥。

ssh-keygen命令常用参数：

* -t：指定生成密钥的类型，默认使用RSA类型密钥。
* -f：指定生成密钥的文件名，默认id\_rsa（私钥id\_rsa，公钥id\_rsa.pub）
* -P：提供旧密码，空表示不需要密码（-P ' '）。
* -N：提供新密码，空表示不需要密码(-N ' ')。
* -C：提供一个新注释，比如邮箱。

|  |
| --- |
| 注意：   * 这里的 xxxxx@xxxxx.com 只是生成的 sshkey 的名称，并不约束或要求具体命名为某个邮箱。现网的大部分教程均讲解的使用邮箱生成，其一开始的初衷仅仅是为了便于辨识，所以使用了邮箱。 * 直接执行ssh-keygen命令也可以生成SSHkey密钥。。 * 命令在任何路径下都可以执行。 |

生成SSHkey密钥示例：

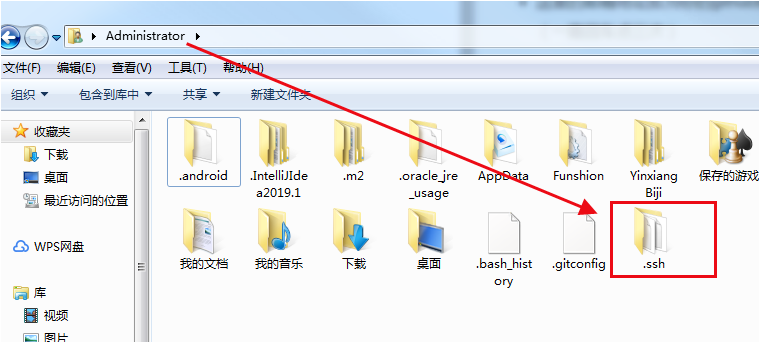
|  |
| --- |
| Bash # 1.执行ssh-keygen命令，生成SSHkey密钥 L@DESKTOP-T2AI2SU MINGW64 /j $ ssh-keygen -t rsa -C "309877988@qq.com"  # 生成公共/私有rsa密钥对。 Generating public/private rsa key pair.  # 生成公共/私有rsa密钥对。 # 输入要在/c/Users/L/.ssh/目录中，保存密钥的文件名称 # 回车代表默认，即id\_rsa，例如：/c/Users/L/.ssh/id\_rsa # 如果输入指定的文件名。则为/c/Users/L/.ssh/指定的文件名。 # 我们回车，执行默认选项即可。 Enter file in which to save the key (/c/Users/L/.ssh/id\_rsa):  # 创建目录“ /c/Users/L/.ssh”。 Created directory '/c/Users/L/.ssh'.  # 提示你：不输入密码，则密码为空。这里我们是可以设定密码的，例如123456。 # 我们回车即可。 Enter passphrase (empty for no passphrase):  # 再次输入相同的密码：（同上也回车） Enter same passphrase again:  # 经过上面三次回车，下面SSHKey密钥就自动生成了 # 您的私钥已保存在/c/Users/L/.ssh/id\_rsa中。 Your identification has been saved in /c/Users/L/.ssh/id\_rsa. # 您的公钥已保存在/c/Users/L/.ssh/id\_rsa.pub中。 Your public key has been saved in /c/Users/L/.ssh/id\_rsa.pub. # 关键指纹是：（如果没有指定-C 参数内容，默认应该是用系统用户@主机名来代替） The key fingerprint is: SHA256:hcDGM+lT+gMRTqxOZnGtxFwiqZ/dMRRIpCXlWJLG3GY 309877988@qq.com # 密钥的随机图像为： The key's randomart image is: +---[RSA 3072]----+ | o+&@\*o. | | B%Eo+. | | ooXoO. . | | . = \* o. | | \* o =So | | + . + | | . | | | | | +----[SHA256]-----+ |

**说明：**

生成的SSHkey密钥存放在，主机的当前用户主目录下的，隐藏目录.ssh下面。

.ssh目录下面有两个文件，默认分别是id\_rsa文件与id\_rsa.pub文件，其中id\_rsa文件中存放的是私钥，id\_rsa.pub文件中存放的是公钥。

执行完成上边命令，在用户根目录自动生成.ssh目录。（windows系统中c:/users/administrator/.ssh可以找到）



进入.ssh目录中可以看到两个文件，id\_rsa文件是私钥，id\_rsa.pub是是公钥，这两个文件是成对出现的，说明生成SSHkey密钥成功。

**2）补充：生成SSHkey密钥到指定的目录中**

执行命令：$ ssh-keygen -t rsa -C 'xxxxx@qq.com' -f ~/.ssh/github\_id\_rsa

这样可以直接指定生成密钥文件的位置和名称。

可以到~/.ssh目录下进行查看，也是生成了一对密钥，github\_id\_rsa和github\_id\_rsa.pub。

这样的操作，也是非常方便的。

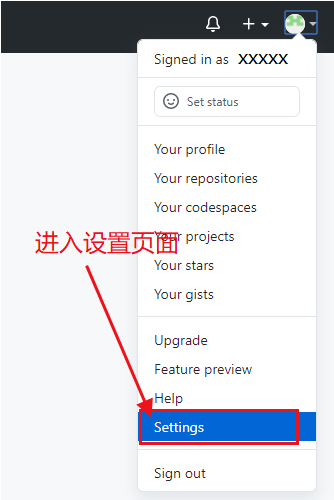
**（2）把公钥的内容配置到GitHub中**

**1）复制公钥文件内容**

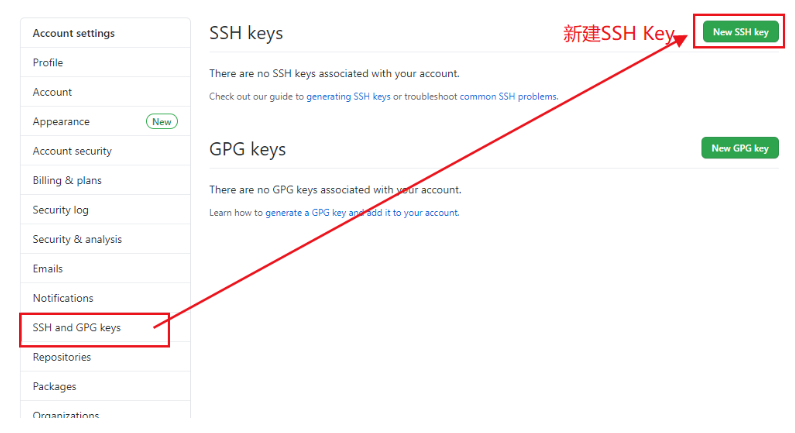
|  |
| --- |
| Bash L@DESKTOP-T2AI2SU MINGW64 /j $ cat /c/Users/L/.ssh/id\_rsa.pub ssh-rsa  309877988@qq.com |

**2）登陆GitHub官网进入设置页面**

在GitHub网站的右上角，点击下图所示图标，在下拉列表中选择Settings，打开Settings设置页面。



**3）在左边导航栏中，选择SSH and GPG keys，在右边点击new SSH key。**

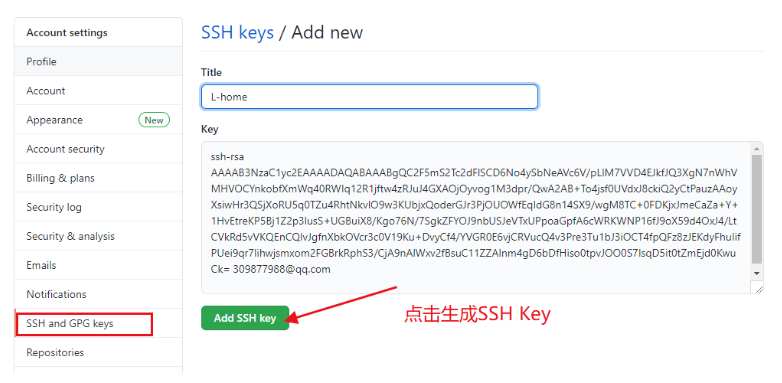


**4）编辑SSH Key信息，并创建SSH Key**

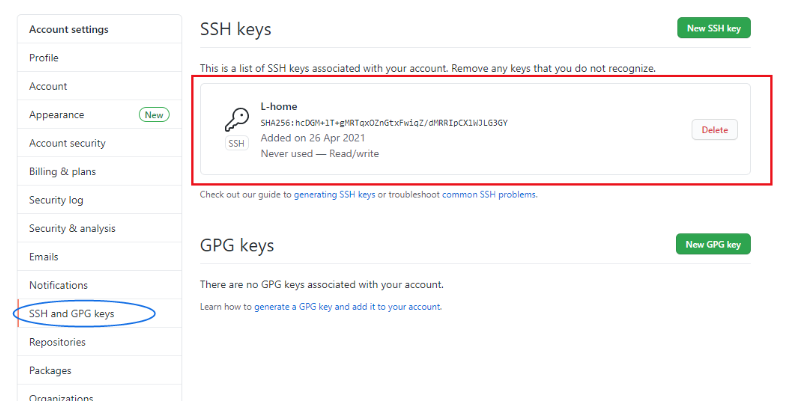
填写页面中信息：

* 需要编辑一个Title来说明此Key认证的是哪个用户。可以随意定义，例如，这是张三在家里主机上的Git要连接该GitHub，所以可以命名为zhangsanHome。
* 把公钥（id\_rsa.pub）文件中的内容填写的第二个栏中。

如下图：



点击Add SSH key后，即可看到刚刚在GitHub中配置的公钥信息。



**（3）验证SSH链接认证是否配置成功**

也就是测试SSHkey密钥是否配置成功。

使用ssh -T命令进行校验，执行ssh -T git@github.com命令。

|  |
| --- |
| Bash # 执行校验命令 L@DESKTOP-T2AI2SU MINGW64 /j $ ssh -T git@github.com  # 无法确定主机“ github.com（192.30.255.113）”的真实性。 # RSA密钥指纹为SHA256：nThbg6kXUpJWGl7E1IGOCspRomTxdCARLviKw6E5SY8。 # 您确定要继续连接（是/否/ [指纹]）吗？ The authenticity of host 'github.com (192.30.255.113)' can't be established. RSA key fingerprint is SHA256:nThbg6kXUpJWGl7E1IGOCspRomTxdCARLviKw6E5SY8. Are you sure you want to continue connecting (yes/no/[fingerprint])? # 我们这里输入yes，需要进行校验。  # 如果设置了密码，就会第二个停顿，就需要输入上边我们设置的密码123456。  # 最后如下内容，证明验证成功！ Hi Lxxxxx! You've successfully authenticated, but GitHub does not provide shell access. |

此时，已经证明我们配置好了，Git账户SSHKey的认证方式。

|  |
| --- |
| 提示：其他远程仓库的SSHKey配置步骤都类似，例如：**Gitee，GitHub、Gitlab等**。 |

作者：繁华似锦Fighting 链接：https://www.jianshu.com/p/a2e04b048955 来源：简书 著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。