GitHub SSH key使用方法

1. github sshkey是怎么用的？是我的本地主机私钥加密成公钥发送给github吗？

非常好的问题，这个理解对 SSH 的原理抓得很准。我们来一步步把 SSH key 和 GitHub 的过程讲清楚。

🧩 一、SSH key 的本质

SSH（Secure Shell）是一种安全通信协议，它通过**非对称加密**来实现"我是谁"的验证。

当你生成 SSH key 时，会得到两部分：

- **私钥（private key）**：保存在你的电脑上，不能泄露。
- **公钥（public key）**：可以安全地公开，比如上传到 GitHub。

🔐 二、核心机制

你理解得没错，过程本质上是这样的：

> "本地主机生成一对密钥（私钥 + 公钥），
> 然后把公钥发送给 GitHub，
> 之后 GitHub 只认你这台机器的私钥。"

具体过程：

1. 你在本地生成一对密钥

   ```
   bash

   ssh-keygen -t ed25519 -C "your_email@example.com"
   ```

   执行后会生成两个文件（默认位置是 `~/.ssh/`）：

   - `id_ed25519`（私钥）
   - `id_ed25519.pub`（公钥）

2. 你把 **公钥** 内容复制到 GitHub
   路径：
   `GitHub -> Settings -> SSH and GPG keys -> New SSH key`
   然后粘贴 `id_ed25519.pub` 文件的全部内容。

3. 当你执行 `git push` 或 `git pull` 时：

   - GitHub 先发来一个"加密挑战"（只有你的公钥对应的私钥能解密）。
   - 你的本地 SSH 客户端用 **私钥** 解密并返回。
   - GitHub 验证成功后，就知道"哦，这确实是你"。

整个过程没有明文密码，也不会把私钥发出去。

🔄 三、工作原理对比理解

方式	验证手段	安全性	是否需要每次输入密码
HTTPS	账号 + 密码	中等	要输入或用 token
SSH	公钥 + 私钥	高	一次设置长期生效

所以对于频繁使用 GitHub 的开发者，**SSH 是更方便安全的选择**。

🧠 四、测试与使用

设置完 SSH key 后，可以测试是否成功：

```
bash

ssh -T git@github.com
```

如果提示：

```
vbnet

Hi whytreatme! You've successfully authenticated, but GitHub does not provide shell access.
```

说明认证成功。

之后你可以用 SSH 地址 clone：

```
bash

git clone git@github.com:whytreatme/linux_tools.git
```

而不是 HTTPS 的：

```
bash

git clone https://github.com/whytreatme/linux_tools.git
```

✅ 总结一句话

> 是的，SSH key 的机制就是：
> 你本地主机生成密钥对，用私钥加密，
> GitHub 存公钥来验证身份。
> 这样你推送时就不用密码，GitHub 只认你的机器。

要不要我帮你写一份你本机从生成、配置、测试 SSH key 到替换远程仓库的完整命令步骤？这样你照着执行就能一次搞定。