

目录

介绍 / 作者序	1.1
tinc 简介	1.2
基础使用	1.3
组网实例 (home_vpn)	1.4
1. 核心主机 / Ubuntu 16.04	1.4.1
2. 台式机 / Ubuntu 18.04	1.4.2
3. 笔记本 / Win10	1.4.3
4. 开发机 / MacOS mojave	1.4.4
5. 群晖 NAS / Synology 5.2 5529	1.4.5
路由器 / OpenWrt	1.4.6
进阶	1.5
结束语	1.6

介绍

本教程是一本关于 tinc VPN 的实操指南，记录了作者几年来使用 tinc 的经验、教训和建议，可作为 tinc 入门参考和快速指南。

本教程线上版本：<https://chanix.github.io/TincCookbook>

作者序

谨以此文献给我的家人，感谢一直以来对我的容忍和支持。我不擅长表达，在这里说声谢谢，或许未来他们能够知道。

在工作和生活中，我有将不同地点和网络环境的设备连接在一起，形成虚拟专有网络（VPN）的需要：

希望能进行异地之间数据的交换（远程办公，远程访问家庭NAS，远程备份……）；
希望安全的连接，不用担心隐私和数据的泄露；
希望能简化安全和网络配置，不用考虑各台设备所处环境的区别；
希望尽可能的降低搭建和运维成本，最好是折腾完了就扔一边不用管；
希望尽可能的简单快捷，在下不是IT科班出身，太深奥的东西理解不了；

经过比较和测试，最终选择了 tinc 作为虚拟专有网络的实现工具，搭建的几个 tinc 网络已经稳定运行了超过3年。在实际操作中，遇到了不少问题，相对小众所以资料也不多。因此我将相关的经验和文档整理了一下，形成本教程，作为个人使用 tinc 的总结和备忘。同时也公开分享出来，如果能为您答疑解惑，将不胜荣幸。

本教程针对非专业人士，重点在于实操。作为读者，您不需要是专业IT人员，但需对网络有基本的了解并有一定的动手能力。本教程重点在于如何快捷的使用 tinc，而不是对其原理和源码深入的剖析和讲解。我试图让读者读完本教程后，拥有快速搭建和管理 tinc 网络的能力。换句话说，如果完成一件事有很多种方法，本教程只会介绍一种经过验证和总结的方法（可能最笨但确保实用），不会教你“茴香”的“茴”有九种写法。

本教程的目的在于让一个具有网络使用和电脑使用经验的普通用户在半小时内搭建完成自己的 tinc 网络。如果您遇到了困难和麻烦，那一定是本教程写的还不够清晰明了，请及时反馈，这是我逐步修改和完善的动力和方向。

相关内容

- 本书源码：<https://github.com/Chanix/TincCookbook>
- tinc 官网：<https://www.tinc-vpn.org>
- tinc 源码：<https://github.com/gsliopen/tinc>

转载注意事项

本教程采用 [Creative Commons BY-NC-ND 4.0](#) （自由转载-保持署名-非商用-禁止演绎） 协议发布，可以在非商业的前提下免费转载，但同时必须：

- 保持文章原文，不作修改。
- 明确署名，即至少注明 作者: Chanix 字样以及文章的原始链接，且不得使用 `rel="nofollow"` 标记。
- 商业用途请联系本人（微信同号）。
- 微信公众号转载一律不授权 原创 标志。

tinc 简介

本章节包含一些技术介绍。对于普通使用者而言不需要太过关心。可以跳过本章直接阅读后继章节。

tinc 是一个组建虚拟专用网络 (VPN) 的工具，以 GNU 协议发布，通过隧道及加密技术在互联网上点与点之间创建专有网络。tinc 在网络层工作，因此无需对现有软件进行修改和配置。您可以使用 tinc 搭建专属的低延迟、高带宽、可扩展的 P2P VPN。其数据通讯经过加密和压缩，能避免敏感数据和隐私的泄露。

无论何时何地，只要能连入互联网，就可以安全的访问 VPN。

具体技术细节和功能不多说了，参见官网。对我来说，选择 tinc 有以下几个理由：

- 开源，截止目前还在不断更新完善；
- 分布式网状路由，避免单点高负载和故障；
- 可运行多个实例来接入多个VPN；
- 通过虚拟网卡通讯，无需对现有应用软件进行修改和配置；
- 通讯支持 加密/认证/压缩，并支持参数选择；
- 支持常见的操作系统和网络拓扑，适用场景广泛；

基础使用

对于普通使用者，快速的搭建起一个能用的安全强度适合的 VPN 是最大的需求。因此，本章节主要讲解基本概念和实践操作。配置内核、编译源代码、甚至交叉编译、高级定制等内容请参考本教程的进阶部分。

预备知识

- 每个 tinc VPN 必须有个名称，一个 VPN 可以包括很多主机；
- 每台主机必须有个名称，同时需要运行 tinc，一台主机可以通过运行多个 tinc 实例来加入多个 tinc VPN；
- tinc 启动时接受参数来指定要启动的网络，并定位到对应的网络配置目录读取配置；
- 启动后，读取网络配置目录中的主配置文件 tinc.conf，执行启动脚本（tinc-up），然后 ConnectTo 指定的主机，同时接受其他主机对本机的 ConnetTo；
- tinc 通过读取主机描述文件来获得主机信息，当前主机和 ConnectTo 的目标主机上都必须有双方的主机描述文件；
- ConnectTo 成功（认证通过），则加入 tinc VPN；
- 当 tinc 结束的时候，执行关闭脚本（tinc-down）；

tinc VPN 名称接受 a-z 0-9 _ 中的字符，主机名称也是一样。

组网步骤

搭建 tinc VPN 非常快捷，每台主机的设置仅需几步，所有主机重复这套步骤即可：

```
graph LR A(安装) --> B(配置和交换密钥); B(配置和交换密钥) --> C(运行); C(运行) --> D(检查);
```

其中的安装、运行和检查非常简单，各平台的操作大同小异，有个大致的概念就可以了，后继章节有详细的针对各个平台的说明，请参阅实例部分。

主要需要理解和掌握的是“配置和交换密钥”这部分。

如何配置和交换密钥

- 配置，就是在 tinc 主配置目录里，按照配置规范，建立好相应的子目录和相应的文件。
- 交换密钥，就是在网络配置目录中，包含通讯双方的主机描述文件。

tinc 的配置方式是通过一套目录和其中的文件（以下简称为主配置目录）来完成的，主配置目录中按规范存放一系列目录和文件。主配置目录默认值为 /etc/tinc （Unix-like OS）或 C:\Program Files\tinc （Windows）。

主配置目录下，每个子目录是以该目录名为名称的 VPN 网络的配置目录（以下简称为 网络配置目录）。每个网络配置目录，指定了该网络的相关配置。tinc 实例启动时接受参数来指定要启动的网络，并定位到对应的网络配置目录读取配置。

每个网络配置目录中，有以下内容：

- **tinc.conf**

主配置文件，其中的内容指定了该网络下 tinc 的配置。其中的 Name 说明本主机名称，ConnectTo 指定启动后要自动连接的主机（可以多个）。

- **tinc-up**、**tinc-down**

脚本文件，这两个脚本是在 tinc 启动和关闭该网络时被调用。一般在这个脚本中用 “ifconfig/ipconfig” 等命令来设置虚拟网卡的 IP、路由等网络设置。如果是 Unix-like 系统，需要有运行权限，如果是 Windows 系统，则需要增加 “.bat” 后缀，即 “tinc-up.bat” 和 “tinc-down.bat”。

- **rsa_key.priv**

RSA 私钥文件，存放本主机的 RSA 私钥，这个文件的内容注意保密；

- **hosts 子目录**

主机描述文件存放目录。其中的每一个文件描述了一台主机的信息，文件名与主机名保持一致。

这是我笔记本（notebook）上的主配置目录（注：某些系统可能会有点差异，但差不多，万变不离其宗）：

```
/etc/tinc          (主配置目录)
|
└── home_vpn      (第一个 VPN 的网络配置目录，目录名和网络名保持一致。)
    ├── hosts
    │   ├── tinc_ali    (主机 tinc_ali 的描述文件)
    │   └── notebook    (主机 notebook 的描述文件)
    ├── rsa_key.priv
    ├── tinc.conf
    ├── tinc-down     (当关闭 home_vpn 时，执行该脚本)
    └── tinc-up       (当启动 home_vpn 时，执行该脚本)

└── office_vpn     (第二个 VPN 的网络配置目录，目录名和网络名保持一致。)
    ├── hosts
    │   ├── server      (主机 server 的描述文件)
    │   └── notebook    (主机 notebook 的描述文件)
    ├── rsa_key.priv
    ├── tinc.conf
    ├── tinc-down     (当关闭 office_vpn 时，执行该脚本)
    └── tinc-up       (当启动 office_vpn 时，执行该脚本)
```

在这个示例中，主机上共配置了两个 VPN，*home_vpn*（家庭用）和*office_vpn*（公司用）。这两个网络的配置都是在主配置目录下的一个子目录，子目录名称和网络名称一致。

以 *home_vpn* 为例：

tinc.conf

```
# 说明本主机名称
Name = notebook

# 指定启动时自动连接的主机。
# 可以使用多个ConnectTo来自动连接多个主机。
# 也可以没有，等待其他主机发起连接。
ConnectTo = tinc_ali
```

tinc-down

```
#!/bin/sh

# 关闭虚拟网卡
ifconfig $INTERFACE down
```

tinc-up

```
#!/bin/sh

# 启用虚拟网卡，并设置其 IP 为 10.0.0.101，子网掩码为 255.255.255.0
ifconfig $INTERFACE 10.0.0.100 netmask 255.255.255.0
```

hosts/tinc_ali

```
# tinc_ali 是一台公网机器, 公网IP为 111.111.111.111
Address = 111.111.111.111

# tinc_ecs 的 VPN 内部 IP 为 10.0.0.1
# /32 说明其为一台机器而不是子网 (普通用户直接用 /32 就可以了)
Subnet = 10.0.0.1/32

-----BEGIN RSA PUBLIC KEY-----
.....
.....
.....
-----END RSA PUBLIC KEY-----
```

hosts/notebook

```
# notebook 是移动办公的笔记本, 没有公网IP, 所以没有 Address 这一行。
# notebook 的 VPN 内部 IP 为 10.0.0.101, 这里要和 tinc-up 里面设定的一致。
# /32 说明其为一台机器而不是子网 (普通用户直接用 /32 就可以了)
Subnet = 10.0.0.101/32

-----BEGIN RSA PUBLIC KEY-----
.....
.....
.....
-----END RSA PUBLIC KEY-----
```

总结一下, 其实就是本节开头的话:

- 所谓配置, 就是按照上面的例子建立相应的目录, 并在目录中建立相应的文件;
- 所谓交换密钥, 就是确保通讯的双方主机有对方的公钥, 而公钥存放在主机描述文件中。

文笔有限, 可能看到这里您还是有点模糊。不要紧, 有了基本的概念和知识, 跟着我一起规划和搭建一次 tinc VPN。先不求甚解, 依葫芦画瓢, 实用优先, 然后再求其索。

组网实例（home_vpn）

了解了基本知识以后，下面结合实例一起来组建属于自己的 tinc VPN。下面是作者日常家庭使用的 VPN，包括了目前可以看到的大多数网络环境和应用场景，目前已经稳定运行了超过3年。读者可以按照自己的需要，从中截取需要的部分。

所有实例都经过实际操作验证，即使不了解网络，同样的步骤依葫芦画瓢操作一次，也一定能组建起您的专属 tinc VPN。

复习、思路和规划

- 加入 VPN 的每台主机都需运行 tinc，需有所有加入主机的操作权限；
- 主机通过网络配置文件中指定的 ConnectTo，主动连接到指定的主机；
- 为了加密和认证，ConnectTo 的双方皆需有对方主机的描述文件；
- 出于稳定原则，tinc 的版本选择其稳定分支 1.0；
- 通过互联网建立 VPN（工作在 IP 层面），使用默认路由器（router）模式即可；
- 开机即连通 VPN，无需其他操作，降低使用难度；
- 要能方便的进行主机的增加和删除，降低维护难度；

网络示意图

网络名称定为 *home_vpn*，下面是示意图，图中的箭头表示 ConnectTo 方向。基本上覆盖了常见的各种网络环境，可以按照顺序来读，或者直接跳到对应的章节截取需要的部分：

```
graph RL; L[2.台式机] --> Z[1.核心主机]; W[3.笔记本] --> Z; M[4.开发机] --> Z; D[NAS] --> Z; R[路由器 NAT] --> Z;
I[家人设备] --- R; J[临时访问设备] --- R;
```

注：数字编码和后面配置的先后顺序一致，同时也作为机器的编号。

不是分布式吗？为什么要核心主机？

一方面，我们基于互联网建立 VPN，主机必须能直接或间接的进行通讯，这要求整个网络中至少要有一台具有公网 IP 的主机。

另一方面，出于简单实用的原则，采用集中式的管理。主机之间的加密和认证是通过读取主机描述文件中的密钥来实现的。tinc 提供自动的主机发现和信息交换，因此只需在核心主机上保存所有的主机描述文件即可。其他主机连接核心主机成功后，能自动发现其他主机，并通过核心主机获得其描述文件。如果分布式管理，将主机描述文件散落在每一台主机上，更新和维护相对是一件比较麻烦的事情。

核心主机选用什么方案？

核心主机最关键的两个条件是，具有公网 IP，可以运行 tinc。因此云主机（VPS）或有公网 IP 的设备（家里的电脑、路由器等）都可以选择。

出于简单和稳定原则，我建议购买云主机。云主机无需担心断网断电，而且一般都具 90% 以上的可靠性。我购置了一台阿里云主机专门用来运行 tinc。

个人家庭用途的 VPN 规模不会很大，入门级的云主机就够了。带宽流量方面，阿里云有按使用付费和固定带宽两种模式收费方式。我个人喜欢简单的付费模式，选择 1M 固定带宽。这样的配置，目前价格在人民币 70 元/月 左右（长期续费有优惠），未来还可以按需升级配置和带宽。于是我索性买了 10 多年，差不多涵盖了有生之年……

1. 核心主机 / Ubuntu Server 16.04

本节我们将配置核心主机，这是一台购买的阿里云主机。这台核心主机的 tinc 作为服务自动启动，启动后等待其他主机来连接，其他所有的主机都配置为主动连接核心主机。核心主机负责处理其他主机的认证和必要的中继转发。

详细配置与 VPN 设置：

项目	数据
硬件配置	1核 1GB 20G云盘
带宽流量	固定带宽 1m/s
操作系统	Ubuntu 16.04
公网 IP	111.111.111.111
内网 IP	222.222.222.222
VPN 网络名称	home_vpn
VPN 主机名称	tinc_ali
VPN IP	10.0.0.254
VPN 子网掩码	255.255.255.0
VPN CIDR	10.0.0.254/24
tinc 端口	655(默认)

注意：云主机一般都配有防火墙（默认可能没有开放 655 端口）。您需要打开 TCP/UDP 协议的 655 端口以供 tinc 使用。具体请参阅阿里云或者您提供商的文档。

安装 tinc

登录服务器，进入终端。阿里云的安装镜像不是最新的，建议先将系统升级到最新状态。下面的这行命令将更新软件列表，升级到最新版本并自动删除不再使用的软件包：

```
sudo apt-get update && sudo apt-get upgrade -y && sudo apt-get dist-upgrade -y && sudo apt-get autoremove -y
```

然后安装 tinc：

```
sudo apt-get install tinc -y
```

好了，安装完成。tinc 的默认主配置目录为 /etc/tinc，现在这个目录里面只有一个文件 nets.boot。

```
>which tincd
/usr/sbin/tincd

>tincd --version
tinc version 1.0.26 (built Jul  5 2015 23:17:56, protocol 17)
Copyright (C) 1998-2015 Ivo Timmermans, Guus Sliepen and others.
See the AUTHORS file for a complete list.

tinc comes with ABSOLUTELY NO WARRANTY.  This is free software,
```

```
and you are welcome to redistribute it under certain conditions;
see the file COPYING for details.

>ls /etc/tinc
nets.boot

>cat /etc/tinc/nets.boot
## This file contains all names of the networks to be started on system startup.
```

创建配置文件

1. 建立网络配置目录（网络名称为 *home vpn*）：

```
sudo mkdir -p /etc/tinc/home_vpn/hosts
```

2. 建立配置文件 *tinc.conf*

```
sudo vi /etc/tinc/home_vpn/tinc.conf
```

编辑 *tinc.conf* 内容如下：

```
Name = tinc_ali
```

由于本主机为核心主机，只负责等待和认证其他主机的连接。因此，本主机没有配置 ConnectTo。

3. 建立启动和关闭脚本 创建启动脚本 *tinc-up*

```
sudo vi /etc/tinc/home_vpn/tinc-up
```

编辑 *tinc-up* 内容如下：

```
#!/bin/sh

ifconfig $INTERFACE 10.0.0.254 netmask 255.255.0.0
```

创建启动脚本 *tinc-down*

```
sudo vi /etc/tinc/home_vpn/tinc-down
```

编辑 *tinc-down* 内容如下：

```
#!/bin/sh

ifconfig $INTERFACE down
```

赋予脚本可执行权限：

```
sudo chmod +x /etc/tinc/home_vpn/tinc-up
sudo chmod +x /etc/tinc/home_vpn/tinc-down
```

4. 创建本主机描述文件（网络名称为 *tinc_ali*）

```
sudo vi /etc/tinc/home_vpn/hosts/tinc_ali
```

编辑 tinc_ali 内容如下：

```
Address = 111.111.111.111
Subnet = 10.0.0.254/32
```

其中 Address 指明公网地址，告诉其他主机怎么连接核心主机。Subnet 中“10.0.0.254”是本主机的 VPN IP，“/32”说明是本主机的一台普通类型的主机。不了解没关系，先这么写，以后可以参阅进阶进一步学习。

生成密钥

执行 tincd 生成脚本，-n 指定网络名称，-K 指明生成密钥，可以在 -K 后以数字指定密钥长度，普通用途用默认值（2048）即可。命令执行过程中，需要指定文件名，不用管直接两次回车用默认值即可。

```
sudo tincd -n home_vpn -K
```

运行完成以后，会生成私钥文件 /etc/tinc/home_vpn/rsa_key.priv，并在本主机的描述文件中增加公钥。

查看私钥文件：

```
cat /etc/tinc/home_vpn/rsa_key.priv
```

/etc/tinc/home_vpn/rsa_key.priv 内容类似下面这样，“----BEGIN RSA PRIVATE KEY----”和“----END RSA PRIVATE KEY----”之间是本机私钥，这个文件的内容注意保密不要泄露。

```
-----BEGIN RSA PRIVATE KEY-----
...
...
...
-----END RSA PRIVATE KEY-----
```

查看本主机描述文件内容：

```
cat /etc/tinc/home_vpn/hosts/tinc_ali
```

可以看到这个文件的内容发生了变化。在原来编辑的两行后增加了“----BEGIN RSA PUBLIC KEY----”和“----END RSA PUBLIC KEY----”之间的内容，这段内容是本主机的公钥。

```
Address = 111.111.111.111
Subnet = 10.0.0.254/32

-----BEGIN RSA PUBLIC KEY-----
...
...
...
-----END RSA PUBLIC KEY-----
```

注：如果您正在设置您的主机，请记得将上面 111.111.111.111 修改为您主机的公网 IP。

交换密钥

由于我们这里配置的是 VPN 里第一台主机，还没有其他主机连接进来。所以交换密钥先略过了，等后面加入其他主机的时候再进行。

设为自启

Ubuntu 中，安装了 tinc 软件包即安装了 tinc 服务。系统启动后会自动运行这个服务，其读取 /etc/tinc/nets.boot 的内容来确定启动哪些 VPN。也就是说，如果想自动启动某个 VPN，只需将编辑该文件，加入 VPN 的网络名称即可。这样每次机器重启后会自动启动 home_vpn。也可以 sudo service tinc start、sudo service tinc stop 等命令来手工控制服务。

编辑/etc/tinc/nets.boot：

```
sudo vi /etc/tinc/nets.boot
```

在文件末尾加上一行：

```
home_vpn
```

重启系统：

```
sudo reboot
```

重启完成后，可以用下列命令来查看进程：

```
ps -ef | grep tinc
```

测试

由于我们这里配置的是 VPN 里第一台主机，还没有其他主机连接进来。所以测试先略过了，等后面加入其他主机的时候再进行。

现在，我们可以先 ping 自己，看看虚拟网卡是否已经启动并绑定了指定的地址：

```
ping -c 5 10.0.0.254
```

完成

核心主机的安装配置过程大概就是这样。由于是第一台主机，所以交换密钥和测试都略过了。截止目前，我们拥有了第一台 tinc VPN 主机。这台主机暂时孤独的运行在互联网上，等待其他 VPN 主机的连接。

2. 台式机主机 / Ubuntu Server 18.04

本节我们配置台式机，并将其加入到 VPN 中，完成后，整个 VPN 内将有两台机器。这台机器位于杂物房，通过路由器连入互联网。

VPN 设置：

项目	数据
VPN 网络名称	home_vpn
VPN 主机名称	desktop
VPN IP	10.0.0.100
VPN 子网掩码	255.255.255.0
VPN CIDR	10.0.0.100/24
tinc 端口	655(默认)

安装 tinc

Ubuntu 系列的安装都差不多，均通过包管理器 apt 进行，不再赘述。其实其他的 Linux 发行版也差不多，只是包管理器的差异，tinc 在流行的发行版中一般都有预编译版本。

登录服务器，进入终端：

```
sudo apt-get update && sudo apt-get upgrade -y && sudo apt-get dist-upgrade -y && sudo apt-get autoremove -y
sudo apt-get install tinc -y
```

创建配置文件

1. 建立网络配置目录（网络名称为 *home_vpn*）：

```
sudo mkdir -p /etc/tinc/home_vpn/hosts
```

2. 建立配置文件 *tinc.conf*

```
sudo vi /etc/tinc/home_vpn/tinc.conf
```

编辑 *tinc.conf* 内容如下：

```
Name = desktop
ConnectTo = tinc_ali
```

指明本主机的主机名为 *desktop*。注意这里多了一行 *ConnectTo*，这行 *ConnectTo* 指定启动时，自动连接上一节我们配置好的核心主机 *tinc_ali*。

3. 建立启动和关闭脚本 创建启动脚本 *tinc-up*

```
sudo vi /etc/tinc/home_vpn/tinc-up
```

编辑 tinc-up 内容如下：

```
#!/bin/sh  
  
ifconfig $INTERFACE 10.0.0.100 netmask 255.255.0.0
```

创建启动脚本 tinc-down

```
sudo vi /etc/tinc/home_vpn/tinc-down
```

编辑 tinc-down 内容如下：

```
#!/bin/sh  
  
ifconfig $INTERFACE down
```

赋予脚本可执行权限：

```
sudo chmod +x /etc/tinc/home_vpn/tinc-up  
sudo chmod +x /etc/tinc/home_vpn/tinc-down
```

4. 创建本主机描述文件（主机名称为 *desktop*）

```
sudo vi /etc/tinc/home_vpn/hosts/desktop
```

编辑 desktop 内容如下：

```
Subnet = 10.0.0.100/32
```

与核心主机比较，*desktop* 没有公网IP，所以没有 Address 这一行。Subnet 中“10.0.0.100”是本主机的 VPN IP，“/32”说明是本主机的一台普通类型的主机。不了解没关系，先这么写，以后可以参阅进阶进一步学习。

生成密钥

执行 *tincd* 生成脚本，-n 指定网络名称，-K 指明生成密钥，可以在 -K 后以数字指定密钥长度，普通用途用默认值（2048）即可。命令执行过程中，需要指定文件名，不用管直接两次回车用默认值即可。

```
sudo tincd -n home_vpn -K
```

运行完成以后，会生成私钥文件 */etc/tinc/home_vpn/rsa_key.priv*，并更新本主机的描述文件 */etc/tinc/home_vpn/hosts/desktop*。

交换密钥

之前有提到，*tinc* 的加密和认证需要公钥文件，需要通讯的双方主机都有对方的公钥。所以我们要确保核心主机和本主机都有对方的主机描述文件。

将本主机的 */etc/tinc/home_vpn/hosts/desktop* 复制到核心主机的同样位置。

复制核心主机的 */etc/tinc/home_vpn/hosts/tinc_ali* 到本主机的同样位置。

设为自启

Ubuntu 18.04 不再使用 initd 管理系统，改用 systemd，因此和 16.04 之前的版本设置方法不同。需使用 systemctl 来进行服务的管理：

```
sudo systemctl enable tinc@home_vpn
```

重启系统：

```
sudo reboot
```

测试

重启完成后，通过 ping 来验证网络是否互通。

在 desktop 上：

```
ping -c 10.0.0.254
```

在 tinc_ali 上：

```
ping -c 10.0.0.100
```

如果您是严格按照教程做，无意外的话已经能相互 ping 通了。如果 ping 不通，请检查双方，尤其是 tinc_ali 的防火墙设置是否正确。

完成

现在 VPN 共有两台机器：tinc_ali 和 desktop。这两台机器现在开机即可相互通讯，有兴趣的话还可以再试试 SSH 等网络应用。

3. 笔记本 / Windows 10

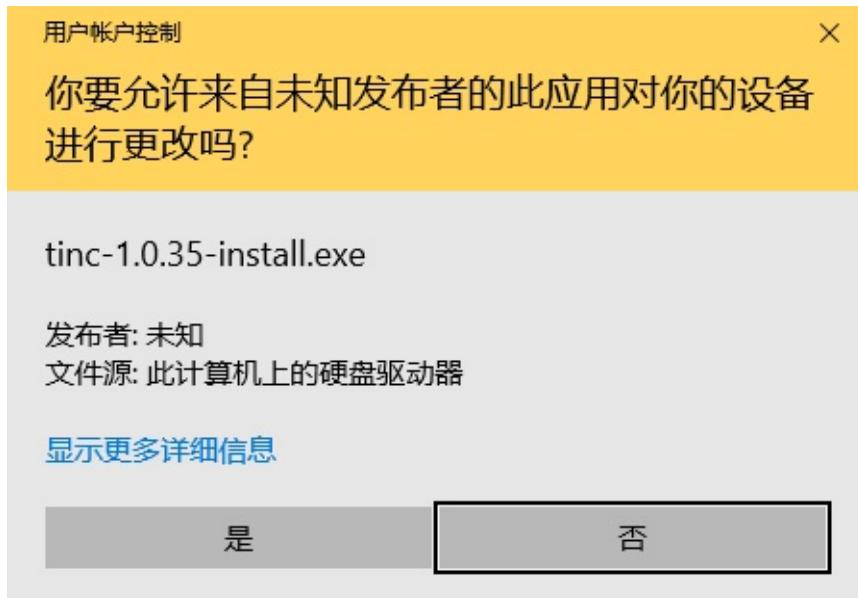
本节将日常使用的笔记本连入 VPN，操作系统为 Windows 10。虽然以 Win10 为例，但 Windows 系列操作基本一致，需要注意的是权限问题以及默认防火墙策略的设置。

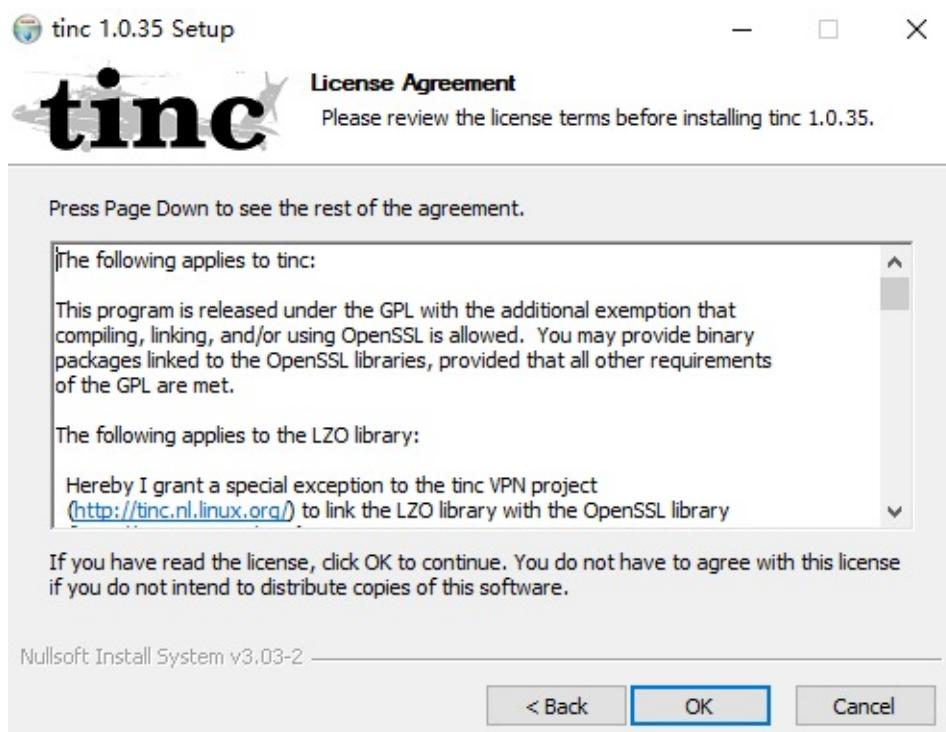
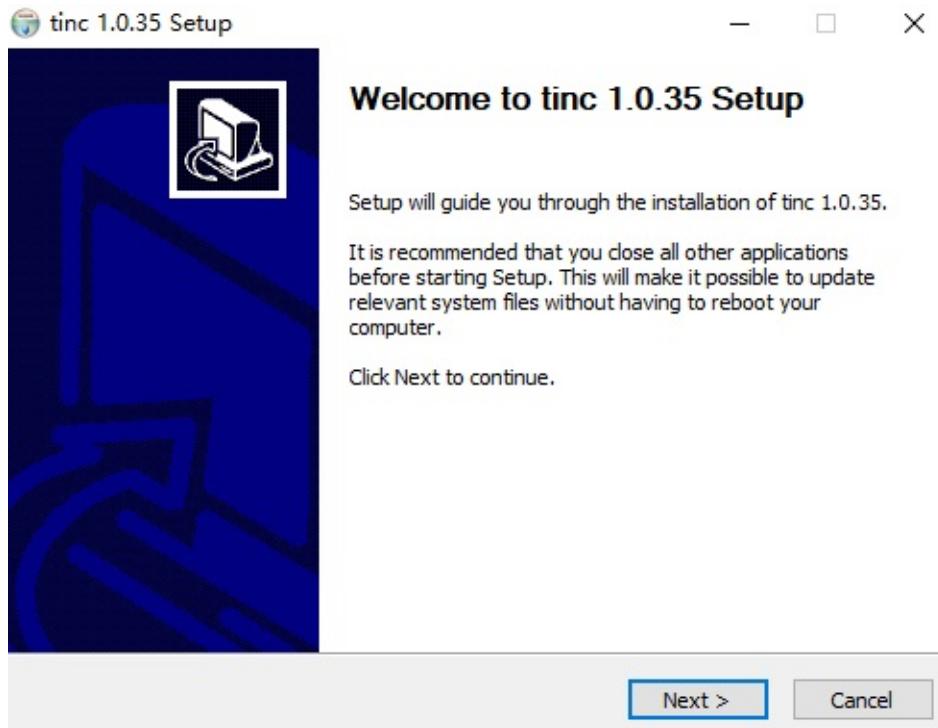
VPN 设置：

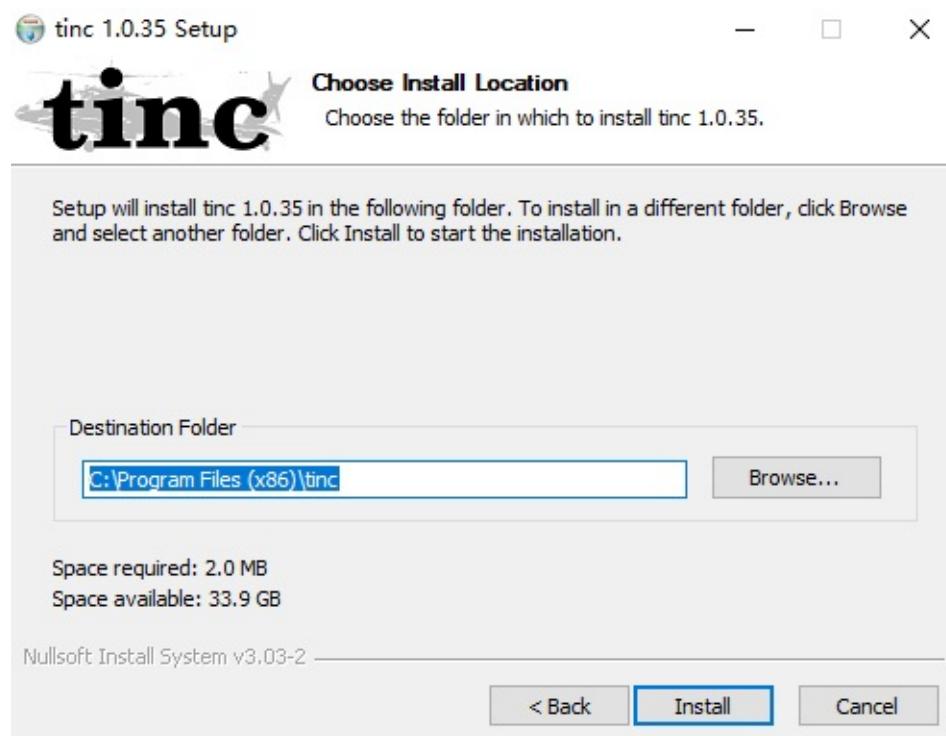
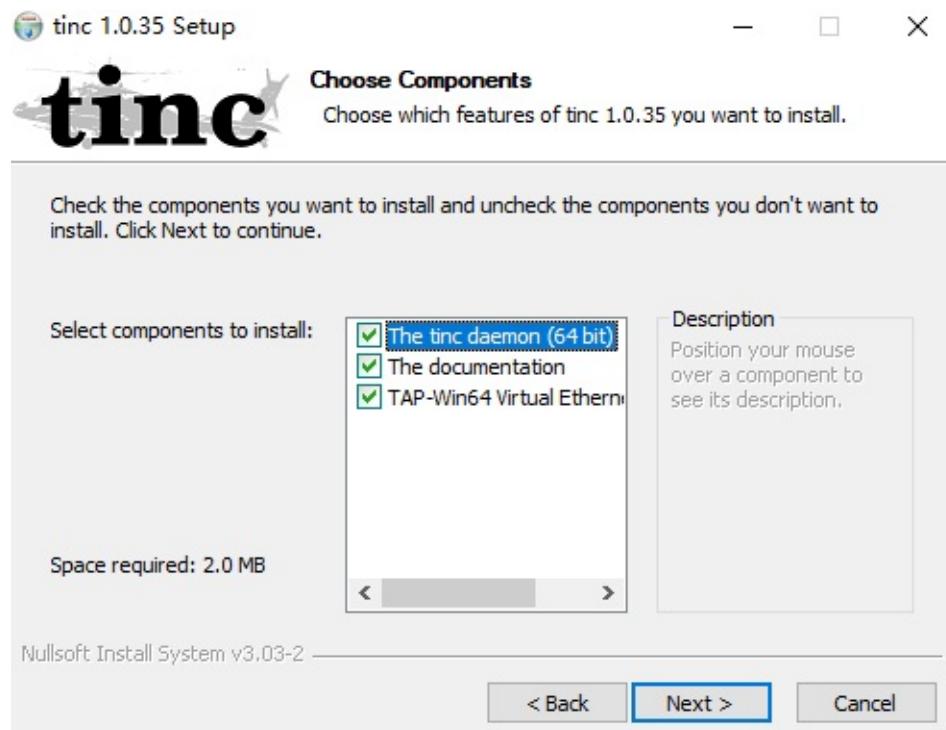
项目	数据
VPN 网络名称	home_vpn
VPN 主机名称	notebook
VPN IP	10.0.0.101
VPN 子网掩码	255.255.255.0
VPN CIDR	10.0.0.101/24
tinc 端口	655(默认)

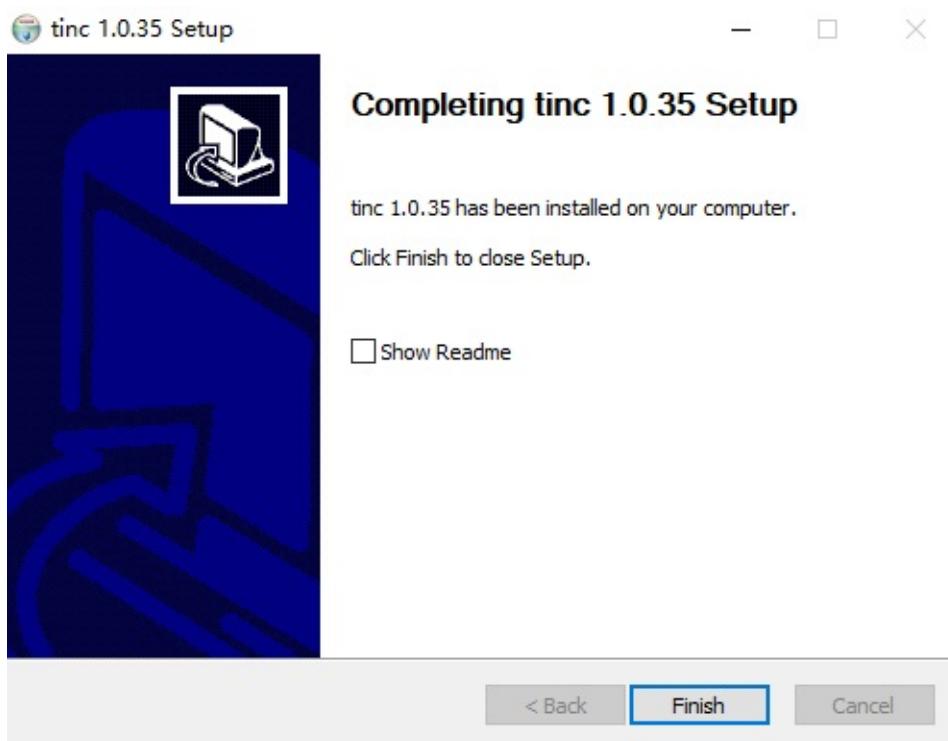
安装 tinc

首先到官网下载安装包 <https://www.tinc-vpn.org/download/>。目前能下载到的最新稳定版是 1.0.35。下载完成后运行安装程序，直接 Next, Next 到 Finish 即可，整个过程非常的简单。









安装完成后，需要先安装虚拟网卡设备。在任务栏 Win 按钮处，按鼠标右键并选择 **Windows PowerShell (管理员) (A)**，后面的操作我们都将使用打开的命令行窗口（方便获得管理员权限也方便读者粘帖命令）：



执行下面的命令增加一块虚拟网卡，安装过程中如果出现安全提示，如下图，请选择 安装：

```
& 'C:\Program Files (x86)\tinc\tap-win64\addtap.bat'
```



在任务栏 Win 按钮处，按鼠标右键并选择 网络连接，然后右侧选择 网络和共享中心，接着左侧选择 更改适配器设置，这时右侧就列出了当前系统中所有的网卡设备，如下图。可以看到系统新增加了一块网卡 以太网 2，请记住这个名称，后面配置文件中需要用到。目前，该设备的状态是个小红叉，处于“网络电缆被拔出”状态。

备注：由于配置环境的差异，您的可能与我不同（例如 Win7 中为 本地连接 2）。



创建配置文件

下面的命令我们仍然在上述打开的 **管理员：Windows PowerShell** 窗口中进行。

1. 建立网络配置目录（网络名称为 `home_vpn`）：

```
mkdir -p 'C:\Program Files (x86)\tinc\home_vpn\hosts'
```

2. 建立配置文件 `tinc.conf`

```
notepad 'C:\Program Files (x86)\tinc\home_vpn\tinc.conf'
```

编辑 `tinc.conf` 内容如下：

```
Name = notebook
ConnectTo tinc_ali
Interface = 以太网 2
```

指明本主机的主机名为 `notebook`，连接核心主机 `tinc_ali`。与之前不同的是，增加了一行 `Interface`。`Interface` 指定了 `tinc` 使用的虚拟网卡名称，该上面安装步骤增加的虚拟网卡设备的名称。注意：不建议将虚拟网卡改名，原因看这里。

3. 建立启动和关闭批处理文件

`Windows` 可以通过图形界面来预先设定虚拟网卡的地址，不需要通过批处理来设置，但建议依然使用批处理。

注意，在 `Windows` 需要增加后缀名 `.bat` 以表明文件类型为 批处理。

创建启动脚本 `tinc-up.bat`：

```
echo '' > 'C:\Program Files (x86)\tinc\home_vpn\tinc-up.bat'
notepad 'C:\Program Files (x86)\tinc\home_vpn\tinc-up.bat'
```

编辑 `tinc-up.bat` 内容如下：

```
netsh interface ip set address "以太网 2" static 10.0.0.101 255.255.255.0
```

创建关闭脚本 `tinc-down.bat`

```
echo '' > 'C:\Program Files (x86)\tinc\home_vpn\tinc-down.bat'  
notepad 'C:\Program Files (x86)\tinc\home_vpn\tinc-down.bat'
```

编辑 tinc-down.bat 内容如下：

```
netsh interface ip set address "以太网 2" source=dhcp
```

4. 创建本主机描述文件（网络名称为 *notebook*）

```
echo '' > 'C:\Program Files (x86)\tinc\home_vpn\hosts\notebook'  
notepad 'C:\Program Files (x86)\tinc\home_vpn\hosts\notebook'
```

编辑 notebook 内容如下：

```
Subnet = 10.0.0.101/32
```

生成密钥

执行 tincd 生成脚本， -n 指定网络名称， -K 指明生成密钥，可以在 -K 后以数字指定密钥长度，普通用途用默认值（2048）即可。命令执行过程中，需要指定文件名，不用管直接两次回车用默认值即可。

```
& 'C:\Program Files (x86)\tinc\tincd.exe' -n home_vpn -K
```

运行完成以后，会生成私钥文件 /etc/tinc/home_vpn/rsa_key.priv，并更新本主机的描述文件 /etc/tinc/home_vpn/hosts/notebook。

交换密钥

将本主机的 C:\Program Files (x86)\tinc\home_vpn\hosts\notebook 复制到核心主机的 /etc/tinc/home_vpn/hosts/notebook。

复制核心主机的 /etc/tinc/home_vpn/hosts/tinc_ali 到本主机的 C:\Program Files (x86)\tinc\home_vpn\hosts\tinc_ali。

设为自启

执行下列命令，手工启动服务，程序将自动注册为系统服务，并出于自动状态，即开机启动，无需登录：

```
& 'C:\Program Files (x86)\tinc\tincd.exe' -n home_vpn
```

系统提示：

```
tinc.home_vpn service installed  
tinc.home_vpn service started
```

可以打开系统的服务面板，来查看详细的信息或进行更进一步的设置。普通用户不用管，默认就可以了。

测试

重启完成后，通过 ping 来验证网络是否互通。

在 notebook 上：

```
ping 10.0.0.254
```

在 tinc_ali 上：

```
ping -c 10.0.0.101
```

如果您是严格按照教程做，无意外的话已经能相互 ping 通了。注意，Windows 系统默认的防火墙设置关闭了 PING，如果 ping 不通，请检查双方，尤其是 Windows 的防火墙设置是否正确。参见[这里](#)。

完成

tinc VPN 增加主机非常的方便，现在已经有3台机器了。下一节，将继续接入我的开发机，该机器为一台 Mac Mini，运行的操作系统为 MacOS mojave。

4. 开发机 / MacOS Mojave

我有一台 Mac mini 作为日常 MacOS 环境的开发机，已经升级为目前最新的版本 Mojave。本节，将这台主机加入到 VPN 中。有点洁癖，所以这次是先删除数据，彻底重新安装后操作并记录的。但是也适用于 High Seia 等版本。

详细配置与 VPN 设置：

项目	数据
VPN 网络名称	home_vpn
VPN 主机名称	macmini
VPN IP	10.0.0.102
VPN 子网掩码	255.255.255.0
VPN CIDR	10.0.0.102/24
tinc 端口	655(默认)

安装 HomeBrew

官方手册提供的是通过 MacPorts + XCode 安装的方式，不适合普通用户，我推荐使用常用的包管理器 HomeBrew 来安装预编译版，需安装 HomeBrew，如果已经安装则跳过次步。关于 HomeBrew 的介绍和使用，超出了本教程的范围，有需要的读者可以到其官网 <https://brew.sh/> 获得更详细的信息，本节只说明操作步骤。

打开终端，执行命令：

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

安装 tinc

MacOS 自带 tun/tap 设备，出于兼容性的考虑，使用开源的 tun/tap 包作为 tinc 虚拟网卡设备。在终端中执行下列命令：

出于安全的需要，系统会要求输入鉴定密码或者打开安全设置，请按照提示操作。

```
brew cask install tun/tap
brew install -y tinc
```

这里，第一行安装开源 tun/tap 包，第二行安装 tinc 软件。

注意，由于安全性的需要，安装时如果出现鉴定或者安全性警告，请输入管理员密码通过，或者是打开 系统偏好设置 中的安全性与隐私，通过操作。如下列图：



创建配置文件

1. 建立网络配置目录（网络名称为 `home_vpn`）：

通过 HomeBrew 安装的所有包都在 `/usr/local` 下，tinc 的默认主目录为 `/usr/local/etc/tinc`。刚安装完是没有的。

```
sudo mkdir -p /usr/local/etc/tinc/home_vpn/hosts
```

2. 建立配置文件 tinc.conf

```
sudo vi /usr/local/etc/tinc/home_vpn/tinc.conf
```

编辑 tinc.conf 内容如下：

```
Name = macmini
ConnectTo tinc_ali
Device = tap0
```

指明本主机的主机名为 macmini。注意这里通过 Device 指定了设备。

3. 建立启动和关闭脚本 创建启动脚本 tinc-up

```
sudo vi /usr/local/etc/tinc/home_vpn/tinc-up
```

编辑 tinc-up 内容如下：

```
#!/bin/sh

ifconfig $INTERFACE 10.0.0.102 netmask 255.255.0.0
```

创建启动脚本 tinc-down

```
sudo vi /usr/local/etc/tinc/home_vpn/tinc-down
```

编辑 tinc-down 内容如下：

```
#!/bin/sh

ifconfig $INTERFACE down
```

赋予脚本可执行权限：

```
sudo chmod +x /etc/tinc/home_vpn/tinc-up
sudo chmod +x /etc/tinc/home_vpn/tinc-down
```

4. 创建本主机描述文件（网络名称为 *macmini*）

```
sudo vi /usr/local/etc/tinc/home_vpn/hosts/macmini
```

编辑 tinc_ali 内容如下：

```
Subnet = 10.0.0.102/32
```

生成密钥

执行 tincd 生成脚本， -n 指定网络名称， -K 指明生成密钥，可以在 -K 后以数字指定密钥长度，普通用途用默认值（2048）即可。命令执行过程中，需要指定文件名，不用管直接两次回车用默认值即可。

```
sudo tincd -n home_vpn -K
```

运行完成以后，会生成私钥文件 /etc/tinc/home_vpn/rsa_key.priv，并更新本主机的描述文件 /etc/tinc/home_vpn/hosts/macmini。

交换密钥

将本主机的 /usr/local/etc/tinc/home_vpn/hosts/macmini 复制到核心主机 /etc/tinc/home_vpn/hosts/macmini。

复制核心主机的 /etc/tinc/home_vpn/hosts/tinc_ali 到本主机 /usr/local/etc/tinc/home_vpn/hosts/tinc_ali。

设为自启

MacOS 通过 Lauchd 管理系统服务和自启项。需要设置 .plist 文件来设为自启，编辑下列文件：

以root权限向/Library/LaunchDaemons/tincd.home_vpn.plist 写入：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>KeepAlive</key>
    <true/>
    <key>Label</key>
    <string>tinc.home_vpn</string>
    <key>ProgramArguments</key>
    <array>
        <string>/usr/local/sbin/tincd</string>
        <string>-n</string>
        <string>home_vpn</string>
        <string>-D</string>
        <string>--pidfile=/usr/local/var/run/tinc.home_vpn.pid</string>
    </array>
    <key>StandardErrorPath</key>
    <string>/tmp/tinc.home_vpn.err</string>
    <key>StandardOutPath</key>
    <string>/tmp/tinc.home_vpn.out</string>
</dict>
</plist>
```

启动服务：

```
sudo launchctl load tinc.home_vpn
```

重启系统：

```
sudo reboot
```

测试

重启完成后，通过 ping 来验证网络是否互通。

在 desktop 上：

```
ping -c 10.0.0.254
```

在 tinc_ali 上：

```
ping -c 10.0.0.102
```

如果您是严格按照教程做，无意外的话已经能相互 ping 通了。如果 ping 不通，请检查双方，尤其是 tinc_ali 的防火墙设置是否正确。

完成

5. 群晖 / Synology 5.2 5529

我有一台群晖放在房间的角落里，用来集中存储家庭照片和视频。为了在外面也能方便的访问到家里 NAS 上的数据，将群晖也通过 tinc 连入 home_vpn。我的群晖是 x86 配置的，运行的操作系统为群晖 DSM 5.2 5529。可以理解为一台运行群晖 DSM 的定制 PC 机。DSM 从本质上说就是个 Linux 的定制包，因此可以类比 Linux 去理解和操作。

VPN 设置：

项目	数据
VPN 网络名称	home_vpn
VPN 主机名称	notebook
VPN IP	10.0.0.101
VPN 子网掩码	255.255.255.0
VPN CIDR	10.0.0.101/24
tinc 端口	655(默认)

安装 tinc

截止本章截稿，群晖官方商店中没有找到 tinc，需自行安装。最正统和彻底的做法是下载 tinc 的源代码，然后在群晖上进行编译。但这么做的话，群晖默认没有所需的工具链，需要先下载和配置一堆东西（内核、头文件、编译器、编译工具……），这显然超出了本教程的范围。所以我用了一个取巧的办法：复制一个能用的版本。

用 SSH 登录到群晖上（登录名为“root”，密码就是安装群晖时输入的管理员帐号的密码），查看一下相关信息，确认该版本的群晖 DSM 用 gcc 4.7.3 编译的，核心版本是 3.10.35，x86_64位版本，支持 SMP（多CPU）。

从 Ubuntu Server 16.04 系统中复制 /usr/sbin/tincd，复制到群晖上，按群晖 DSM 的存放规范，放到 /usr/sbin/tincd，然后记得 chmod +x /usr/sbin/tincd，赋予其可执行权限。

从 Ubuntu Server 16.04 系统中复制 /lib/x86_64-linux-gnu/libcrypto.so.1.0.0，按群晖 DSM 的的存放规范，放到 /lib64。

这个是取巧的办法，思路是：既然群晖运行在 x86_64 上，本质上是个 Linux，所以复制一个对应环境已编译好的 Linux 版本，应能正确运行。

我将需要复制的文件打了个压缩包，可以 [点这里下载](#)。

创建配置文件

1.建立网络配置目录（网络名称为 *home_vpn*）：

```
sudo mkdir -p /etc/tinc/home_vpn/hosts
```

2.建立配置文件 *tinc.conf*

```
sudo vi /etc/tinc/home_vpn/tinc.conf
```

编辑 *tinc.conf* 内容如下：

```
Name = nas
ConnectTo = tinc_ali
```

指明本主机的主机名为 *nas*。ConnectTo 指定启动时，自动连接核心主机 *tinc_ali*。

3. 建立启动和关闭脚本 创建启动脚本 *tinc-up*

```
sudo vi /etc/tinc/home_vpn/tinc-up
```

编辑 *tinc-up* 内容如下：

```
#!/bin/sh

ifconfig $INTERFACE 10.0.0.103 netmask 255.255.0.0
```

创建启动脚本 *tinc-down*

```
sudo vi /etc/tinc/home_vpn/tinc-down
```

编辑 *tinc-down* 内容如下：

```
#!/bin/sh

ifconfig $INTERFACE down
```

赋予脚本可执行权限：

```
sudo chmod +x /etc/tinc/home_vpn/tinc-up
sudo chmod +x /etc/tinc/home_vpn/tinc-down
```

4. 创建本主机描述文件（主机名称为 *nas*）

```
sudo vi /etc/tinc/home_vpn/hosts/nas
```

编辑 *nas* 内容如下：

```
Subnet = 10.0.0.103/32
```

生成密钥

执行 *tinced* 生成脚本，*-n* 指定网络名称，*-K* 指明生成密钥，可以在 *-K* 后以数字指定密钥长度，普通用途用默认值（2048）即可。命令执行过程中，需要指定文件名，不用管直接两次回车用默认值即可。

```
sudo tinced -n home_vpn -K
```

运行完成以后，会生成私钥文件 */etc/tinc/home_vpn/rsa_key.priv*，并更新本主机的描述文件 */etc/tinc/home_vpn/hosts/nas*。

交换密钥

将本主机的 /etc/tinc/home_vpn/hosts/nas 复制到核心主机的同样位置。

复制核心主机的 /etc/tinc/home_vpn/hosts/tinc_ali 到本主机的同样位置。

设为自启

在 /usr/syno/etc/rc.d/ 下新建一个文件 S99tinc.sh, 文件内容为 “/usr/sbin/tinced -n home_vpn”, 记得不要忘记 “chmod +x /usr/syno/etc/rc.d/S99tinc.sh”。

重启系统:

```
sudo reboot
```

测试

重启完成后, 通过 ping 来验证网络是否互通。

在 nas 上:

```
ping -c 10.0.0.254
```

在 tinc_ali 上:

```
ping -c 10.0.0.103
```

如果您是严格按照教程做, 无意外的话已经能相互 ping 通了。如果ping不通, 请检查双方, 尤其是 tinc_ali 的防火墙设置是否正确。

完成

群辉加入后, 无论身在何处, 都可以用 <http://10.0.0.103:5000/webman/index.cgi> 来访问, 非常的方便。如何使用和配置群辉不在本教程范围内, 其自带的帮助中心可以解决大部分的问题。

在前面的折腾过程中，都是在设备上安装 tinc，使其成为VPN的一个节点。理论上说，只要这些设备能连上其他节点或者能被其他节点连接上，就加入了VPN并能提供VPN内的通讯转发服务。但这种折腾方式，并不适合所有的场景。

比如我工作中就有个很典型的场景：办公室里的设备需要访问VPN上的机器，只需访问即可，并不需要这些机器能被VPN上的机器访问。首先如果逐台机器去配置，工作量比较大；其次在VPN上暴露了这些机器，有隐患；再次设备并不单指跑Win的PC机啊，MAC呢？MACBOOK呢？苏菲呢？iOs呢？Andriod呢？要针对所有的这些设备安装tinc，累不累啊（苹果只有Cydia的源，需要越狱才能安装。而安卓呢，有对应的软件可以安装，但是使用时需要root）；再后，我司IT管理很有问题，各自维护各自使用的设备，某些同事还特别喜欢折腾，今天重装下机器，明天恢复成出厂设置……如果每台设备分别安装，每天除了帮他们配节点就不用干别的了……最后TINC的密钥文件到处都是，对于安全和管理来说也是个头痛的事情。

所以在上面的场景里，我采用上篇帖子里网络示意图里场景D的模式：在路由器上安装 tinc 并进行设置，需要访问VPN的机器连上这个路由器，不需要进行任何配置就都可以访问VPN里的机器。VPN中的机器也只能看到路由为止（路由屏蔽了下层网络）。另外，这也从物理上限制了访问VPN的区域（必须到公司才能连上VPN），从一定程度上也加强了安全性。

嗯，有人问我只能在办公室访问，那移动办公怎么办。其实这个根本不是问题……既然都装在路由上了，给需要的人发个路由器不完了，啥地方需要连入VPN那就放啥地方不完了，路由器又不大，买个小点的就完事了。

在路由器上安装TINC并不困难，tinc支持的操作系统和硬件非常多。本着耐++实用的原则（随便断电随便摔，不怕发热启动快），最终采用如下方式：软件采用 OpenWRT，硬件采用 NetGear WDNR3800。

NetGear WDNR3800 就是下面这货，淘宝买的二手洋垃圾，饱经风霜的外观透露着沧桑和历史的厚重感……

MT7601方案，128+128，80MHZ的CPU，。NetGear 大厂出品，质量稳定。家里原来也一直用它，好多年了，除了停电的时候重启一下，平时没关心过。小型网络几十台机器这种配置的硬件足够了，未来不够了可以换成更高档的路由器，或者索性用PC DIY一个软路由。其实最主要的是……成本低……买这个才花了100不到，现在应该更便宜了。

买回来，刷好 Uboot，刷 OpenWRT，我刷的是17.01.5这个版本。然后开始配置：

嗯，有人问我只能在办公室访问，那移动办公怎么办。其实这个根本不是问题……既然都装在路由上了，给需要的人发个路由器不完了，啥地方需要连入VPN那就放啥地方不完了，路由器又不大。

advanced

还没写呢！

进阶

进一步的资料

HOST MAIN SCRIPT

tinc网络实际使用来做什么

远程控制， ssh 远程桌面 VNC

代理转发

个人有时候做些项目，需要。例如开发微信，给别人看产品样品，

通过代理访问受限资源

个人/小型公司的代码管理等等 连接多个以太网段形成1个网络

安全、管理和运维

秘钥长度的选择

压缩的选择

主机名用域名还是写死IP地址？

通过工具，例如 ansible 简化客户端管理

通过NAT，简化对客户端的干预

结束语

写完了，没啥说的。有啥意见和建议放issue里吧。

祝您好运。