

EC2X&AG35-QuecOpen

VPN 工具移植应用指导

LTE Standard/Automotive 模块系列

版本：EC2X&AG35-QuecOpen_VPN 工具移植_应用指导_V1.0

日期：2018-02-22

状态：临时文件

上海移远通信技术股份有限公司始终以为客户提供最及时、最全面的服务为宗旨。如需任何帮助，请随时联系我司上海总部，联系方式如下：

上海移远通信技术股份有限公司

上海市徐汇区虹梅路 1801 号宏业大厦 7 楼 邮编：200233

电话：+86 21 51086236 邮箱：info@quectel.com

或联系我司当地办事处，详情请登录：

<http://quectel.com/cn/support/sales.htm>

如需技术支持或反馈我司技术文档中的问题，可随时登陆如下网址：

<http://quectel.com/cn/support/technical.htm>

或发送邮件至：support@quectel.com

前言

上海移远通信技术股份有限公司提供该文档内容用以支持其客户的产品设计。客户须按照文档中提供的规范、参数来设计其产品。由于客户操作不当而造成的人身伤害或财产损失，本公司不承担任何责任。在未声明前，上海移远通信技术股份有限公司有权对该文档进行更新。

版权申明

本文档版权属于上海移远通信技术股份有限公司，任何人未经我司允许而复制转载该文档将承担法律责任。

版权所有 ©上海移远通信技术股份有限公司 2019，保留一切权利。

Copyright © Quectel Wireless Solutions Co., Ltd. 2019.

文档历史

修订记录

版本	日期	作者	变更表述
1.0	2018-02-22	匡昌盛	初始版本

目录

文档历史	2
目录	3
1 文档目的	4
2 编译准备	5
2.1. 源码下载	5
2.2. Mafile 以及目录结构	5
2.3. 内核选项修改	9
2.4. 编译	10
2.5. 打包到 rootfs	10
3 测试	12
3.1. PPTP	12

Quectel
Confidential

1 文档目的

本文档介绍 LINUX VPN 工具 PPTP, L2TP, IPSEC 在 QuecOpen 中的移植方法, 在 QuecOpen 中移植 VPN 工具和其他交叉编译环境下移植第三方开源软件没有区别, 用户也可以参考本文档在 QuecOpen 中移植其他开源软件。

在阅读文档之前, 请确定已经搭建 QuecOpen 的编译环境, QuecOpen 的使用请参考《[Quectel_EC2x&AG35-QuecOpen_快速入门](#)》

Quectel
Confidential

2 编译准备

2.1. 源码下载

本文档使用的 PPTP 软件为 pptp-linux, L2TP 软件为 xl2tpd, IPSEC 软件为 strongswan。PPTP 和 L2TP 依赖 ppp 相关插件, 在使用 PPTP 和 L2TP 之前, 需要重新编译 PPP。L2SEC 使用 strongswan, strongswan 依赖库 libgmp, 在编译 strongswan 之前, 需要先编译 libgmp。本文档使用的软件版本及下载地址如下所示:

PPTP

软件: pptp-linux

主页: <http://pptpclient.sourceforge.net/>

版本: 1.10.0

下载地址: <https://nchc.dl.sourceforge.net/project/pptpclient/pptp/pptp-1.10.0/pptp-1.10.0.tar.gz>

L2TP:

软件: xl2tpd

版本: 1.3.11

主页: <https://www.xelerance.com/archives/155>

下载地址: <https://codeload.github.com/xelerance/xl2tpd/zip/1.3.11>

PPP:

软件: pppd

版本: 2.4.7

下载地址: <https://download.samba.org/pub/ppp/ppp-2.4.7.tar.gz>

GMP:

软件: gmp

主页: <https://gmplib.org/>

版本: 6.1.2

下载地址: <https://gmplib.org/download/gmp/gmp-6.1.2.tar.bz2>

IPSEC:

软件: strongswan

主页: <https://www.strongswan.org/>

版本: 5.6.2

下载地址: <https://download.strongswan.org/strongswan-5.6.2.tar.bz2>

2.2. Mafile 以及目录结构

第一步：在 QuecOpen sdk 目录下新建 opensrc 目录，将下载好的源码文件放到 opensrc 目录下。

```
mkdir opensrc
```

第二部：创建 opensrc/Makefile 文件并输入以下内容：

```
CURR_DIR := $(shell pwd)
OPENSRC_DIR := $(subst /opensrc, /opensrc, $(CURR_DIR))
OPENSRC_DIR := $(word 1, $(WORKSPACE_DIR))

export PKG_CONFIG_SYSROOT_DIR=$(SDKTARGETSYSROOT)
export PKG_CONFIG_PATH=$(SDKTARGETSYSROOT)/usr/lib/pkgconfig

BUILD_DESTDIR=$(CURR_DIR)/rootfs
BUILD_HOST=arm-oe-linux-gnueabi
BUILD_TARGET=arm-oe-linux-gnueabi

targets_build=pppd_build pptp_build xl2tpd_build libgmp_build strongswan_build
targets_clean=pppd_clean pptp_clean xl2tpd_clean libgmp_clean strongswan_clean

CFLAGS+=-I$(BUILD_DESTDIR)/include -I$(BUILD_DESTDIR)/usr/include
LDFLAGS+=-L$(BUILD_DESTDIR)/lib -L$(BUILD_DESTDIR)/usr/lib

SRC_PPPD:=ppp-2.4.7
SRC_GMP:=gmp-6.1.2
SRC_STRONGSWAN=strongswan-5.6.2
SRC_XL2TPD=xl2tpd-1.3.11
SRC_PPTP=pptp-1.10.0

.PHONY: all
all: $(targets_build)
    rm -rf rootfs_build;
    cp -arf rootfs rootfs_build;
    rm -rf rootfs_build/include rootfs_build/usr/include rootfs_build/share rootfs_build/usr/share;
    find rootfs_build -name "*.a" | xargs rm -f
    @for ff in $(shell find rootfs_build -type f); do \
        $(STRIP) $$ff 2>/dev/null && echo "STRIP FILE : " $$ff;\
    done
    @echo "=====compile $(targets) complete===== "

clean: $(targets_clean)
    rm -rf $(BUILD_DESTDIR)
    @echo "=====clean $(targets) complete===== "
```

pptp_build:

```
if [ ! -d $(SRC_PPTP) ]; then \
    tar xkf $(SRC_PPTP).tar.gz 2>/dev/null; \
fi
cd $(SRC_PPTP) && make DESTDIR=$(BUILD_DESTDIR) CC="$(CC)" IP="/sbin/ip"
```

PPPD="/usr/sbin/pppd" && \

```
fakeroot make install DESTDIR=$(BUILD_DESTDIR)
@echo "compile $(SRC_PPTP) completed"
```

pptp_clean:

```
if [ -d $(SRC_PPTP) ]; then \
    cd $(SRC_PPTP) && make clean; \
fi
```

xl2tpd_build:

```
if [ ! -d $(SRC_XL2TPD) ]; then \
    unzip -n $(SRC_XL2TPD).zip 2>/dev/null; \
fi
cd $(SRC_XL2TPD) && make PREFIX=$(BUILD_DESTDIR) && \
make install PREFIX=$(BUILD_DESTDIR)
@echo "compile $(SRC_XL2TPD) completed"
```

xl2tpd_clean:

```
if [ -d $(SRC_XL2TPD) ]; then \
    cd $(SRC_XL2TPD) && make clean; \
fi
```

pppd_build:

```
if [ ! -e $(SRC_PPPD)/Makefile ]; then \
    tar xkf $(SRC_PPPD).tar.gz 2>/dev/null; \
    cd $(SRC_PPPD); \
    ./configure \
    INSTROOT="$(BUILD_DESTDIR)" \
    DESTDIR="$(BUILD_DESTDIR)" \
    BINDIR=$(BUILD_DESTDIR)/usr/sbin; \
fi
cd $(SRC_PPPD) && make && make INSTROOT="$(BUILD_DESTDIR)" \
INSTALL="install --strip-program=$(STRIP)" \
DESTDIR="$(BUILD_DESTDIR)" \
BINDIR=$(BUILD_DESTDIR)/usr/sbin install
@echo "compile $(SRC_PPPD) completed"
```

pppd_clean:

```
if [ -e $(SRC_PPPD)/Makefile ]; then \
```



```

        cd $(SRC_PPPD) && make clean;\
    fi

libgmp_build:
    if [ ! -e $(SRC_GMP)/Makefile ]; then \
        tar xkf $(SRC_GMP).tar.bz2 2>/dev/null; \
        cd $(SRC_GMP); \
        ./configure \
        --host=$(BUILD_HOST) \
        --target=$(BUILD_TARGET) \
        --prefix=$(BUILD_DESTDIR) \
        --disable-silent-rules \
        --disable-dependency-tracking \
        --enable-cxx=detect \
        --with-readline=no; \
    fi
    cd $(SRC_GMP) && make && make install
    @echo "compile $(SRC_GMP) completed"

```

```

libgmp_clean:
    if [ -e $(SRC_GMP)/Makefile ]; then \
        cd $(SRC_GMP) && make clean; \
    fi

```

```

strongswan_build:
    if [ ! -e $(SRC_STRONGWAN)/Makefile ]; then \
        tar xkf $(SRC_STRONGWAN).tar.bz2 2>/dev/null; \
        cd $(SRC_STRONGWAN); \
        ./configure \
        --host=$(BUILD_HOST) \
        --target=$(BUILD_TARGET) \
        --prefix=/ \
        --disable-silent-rules \
        --disable-dependency-tracking \
        --without-lib-prefix \
        --without-systemdsystemunitdir \
        --disable-aesni \
        --enable-charon \
        --enable-curl \
        --enable-gmp \
        --enable-eap-md5 \
        --disable-ldap \
        --disable-mysql \

```

```
--enable-openssl \
--disable-scepclient \
--disable-soup \
--enable-sqlite \
--enable-stroke \
--disable-swanctl \
--disable-systemd \
CFLAGS="$(CFLAGS)" \
LDFLAGS="$(LDFLAGS)"; \
fi
cd $(SRC_STRONGWAN) && make && make install DESTDIR=$(BUILD_DESTDIR)
@echo "compile $(SRC_STRONGWAN) completed"

strongswan_clean:
if [ -e $(SRC_STRONGWAN)/Makefile ]; then \
cd $(SRC_STRONGWAN) && make clean; \
fi
```

最后 QuecOpen 的目录结构如下：

ql-ol-sdk

```
├── Makefile
├── opensrc
│   ├── gmp-6.1.2.tar.bz2
│   ├── Makefile
│   ├── ppp-2.4.7.tar.gz
│   ├── pptp-1.10.0.tar.gz
│   ├── strongswan-5.6.2.tar.bz2
│   └── xl2tpd-1.3.11.zip
├── ql-ol-bootloader
├── ql-ol-crosstool
├── ql-ol-extsdk
├── ql-ol-kernel
├── ql-ol-rootfs
├── ql-ol-usrdata
├── ql-ol-usrfs
└── target
```

2.3. 内核选项修改

pptp, ipsec 需要修改内核编译选项。

内核编译选项文件位置：ql-ol-kernel/msm-3.18/arch/arm/configs/ mdm9607-perf_defconfig

修改如下:

PPTP:

```
CONFIG_PPP_MPPE=y
```

IPSEC:

```
CONFIG_INET_AH=m
```

```
CONFIG_INET_ESP=m
```

```
CONFIG_INET_IPCOMP=m
```

```
CONFIG_INET_XFRM_TUNNEL=m
```

```
CONFIG_INET_TUNNEL=m
```

```
CONFIG_XFRM_USER=m
```

2.4. 编译

strongswan 在编译配置过程中需要使用 m4 工具, ubuntu 可以运行如下命令安装:

```
sudo apt install m4
```

进入 QuecOpen 主目录, 运行如下命令进行编译

```
source ql-ol-croostool/ql-ol-croostool-env-init
```

```
make kernel
```

```
make kernel_module
```

```
cd opensrc
```

```
make
```

编译完成后, 目标文件位于 **opensrc/rootfs_build**。

2.5. 打包到 rootfs

编辑 QuecOpen 主目录中的 makefile 文件, 找到编译 rootfs 的位置(使用 mkfs.ubifs 工具打包 ql-ol-rootfs 的地方), 以下以本文档使用的 SDK 为例, 说明如何修改。

```
$(rootfs):
ifneq ($(filter $(QUECTEL_PROJECT_NAME), AG35C AG35CE), )
    cd $(TOPDIR) ; chmod +x ./ql-ol-extsdk/tools/quectel_ubi/* ; ./ql-ol-extsdk/tools/quectel_ubi/mkfs.ubi
fs -r ql-ol-$(rootfs) -o machine-image-mdm9610.ubifs -m 2048 -e 126976 -c 4292 -F ; \
    ./ql-ol-extsdk/tools/quectel_ubi/mkfs.ubifs -r ql-ol-usrfs -o mdm9607-usrfs.ubifs -m 2048 -e 126976 -c
4292 -F ; \
```

修改为:

```
cd $(TOPDIR) ; chmod +x ./ql-ol-extsdk/tools/quectel_ubi/* ; \
cp -arf ql-ol-rootfs temprootfs; \
cp -arf opensrc/rootfs_build/* temprootfs/ ; \
./ql-ol-extsdk/tools/quectel_ubi/mkfs.ubifs -r temprootfs -o machine-image-mdm9610.ubifs -m 2048 -e 12
6976 -c 4292 -F ; \
rm -rf temprootfs; \
./ql-ol-extsdk/tools/quectel_ubi/mkfs.ubifs -r ql-ol-usrfs -o mdm9607-usrfs.ubifs -m 2048 -e 126976 -c
4292 -F ; \
```

Quectel
Confidential

3 测试

以下举例说明 QuecOpen 下 VPN 工具简单的测试，此测试的目的是确认移植是否成功。

3.1. PPTP

以下举例说明使用用户名/密码 test/11111111 连接 PPTP 服务器 192.168.20.49 的配置。

step 1: 编辑文件/etc/ppp/chap-secrets, 添加认证用户名密码:

```
test * 11111111 *
```

step 2: 编辑文件/etc/ppp/peers/pptpvpn

```
pty "pptp 192.168.20.49 --nolaunchpppd"
```

```
lock
```

```
noauth
```

```
nobsdcomp
```

```
nodeflate
```

```
name test
```

```
remotename pptpvpn
```

```
ipparam pptpvpn
```

```
require-mppe-128
```

step 3: 开始拨号:

```
pppd call pptpvpn updetach
```

如果拨号成功, 可以看到 ppp 开头的网络设备。

```
/etc/ppp/peers # ifconfig ppp0
ppp0      Link encap:Point-to-Point Protocol
          inet addr:192.168.20.26  P-t-P:192.168.20.230  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1396  Metric:1
          RX packets:30 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:2509 (2.4 KiB)  TX bytes:90 (90.0 B)
```

L2TP

以下举例说明使用用户名/密码 test/11111111 连接 L2TP 服务器 192.168.20.49 的配置。

step 1: 编辑 xl2tpd 的配置文件/etc/xl2tpd/xl2tpd.conf:

```
[global]
```

```
port = 1701
```

```
debug state = yes
```

```
debug tunnel = yes
```

```
[lac testvpn]
```

```
lns = 192.168.20.49
```

```
require chap = yes
```

```
refuse pap = yes
```

```
require authentication = yes
```

```
name = test
```

```
ppp debug = yes
```

```
pppoptfile = /etc/ppp/peers/testvpn.l2tpd
```

```
length bit = yes
```

step 2: 创建/etc/xl2tpd/xl2tpd.conf 文件中指定的 l2tp 连接配置文件/etc/ppp/peers/testvpn.l2tpd:

```
user test
```

```
password 11111111
```

```
noauth
```

```
lock
```

```
lcp-echo-interval 3
```

```
lcp-echo-failure 30
```

```
asyncmap 0
```

step 3: 开始拨号:

```
mkdir /var/run/xl2tpd
```

```
xl2tpd -D &
```

```
echo "c testvpn" > /var/run/xl2tpd/l2tp-control
```

如果拨号成功，可以查看到 ppp 开头的网络设备。

```
ppp0      Link encap:Point-to-Point Protocol
          inet addr:192.168.20.124  P-t-P:192.168.20.230  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1400  Metric:1
          RX packets:9 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:226 (226.0 B)  TX bytes:64 (64.0 B)
```

IPSEC

strongswan 的测试配置可以参考 <https://www.strongswan.org/testresults.html>，以下使用方式 ikev1/net2net-psk 测试。服务器(192.168.10.154)为 WAN 侧一台运行 strongswan 的主机，客户端(192.168.22.17)为我们的模块。

服务器配置(192.168.10.154):

step 1: 编辑文件/etc/ipsec.conf

```
config setup
```

```
conn %default
```

```
ikelifetime=60m
keylife=20m
rekeymargin=3m
keyingtries=1
keyexchange=ikev2
authby=secret
```

```
conn net-net
left=192.168.10.154
leftid=@moon.strongswan.org
right=192.168.22.17
rightid=@sun.strongswan.org
auto=add
```

step 2: 编辑文件/etc/ipsec.secrets:

```
@moon.strongswan.org @sun.strongswan.org : PSK 0sv+NkxY9LLZvwj4qCC2o/gGrWDF2d21jL
```

step 3: 编辑文件/etc/strongswan.conf:

```
charon {
load = random nonce aes sha1 sha2 curve25519 hmac stroke kernel-netlink socket-default
```

updown

step 4: 启动 strongswan 服务:

```
sudo ipsec start --nofork --debug-all
```

模块配置（192.168.22.17）:

step 1: 编辑文件/etc/ipsec.conf

```
config setup

conn %default
ikelifetime=60m
keylife=20m
rekeymargin=3m
keyingtries=1
keyexchange=ikev2
authby=secret

conn net-net
left=192.168.22.17
leftid=@sun.strongswan.org
leftfirewall=yes
right=192.168.10.154
```

```
rightid=@moon.strongswan.org
```

```
auto=add
```

step 2: 编辑文件/etc/ipsec.secrets:

```
@moon.strongswan.org @sun.strongswan.org : PSK
0sv+NkxY9LLZvwj4qCC2o/gGrWDF2d21jL
```

step 3: 编辑文件/etc/strongswan.conf

```
charon {
```

```
load = random nonce aes sha1 sha2 curve25519 hmac stroke kernel-netlink socket-default updown
```

```
}
```

step 4: 开始拨号:

```
/etc # ipsec start
Starting strongswan 5.6.2 IPsec [starter]...
!! Your strongswan.conf contains manual plugin load options for charon.
!! This is recommended for experts only, see
!! http://wiki.strongswan.org/projects/strongswan/wiki/PluginLoad
/etc # ipsec up net-net
initiating IKE_SA net-net[1] to 192.168.10.154
generating IKE_SA_INIT request 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) N(FRAG_SUP) N(HASH_ALG) N(REDIR_SUP) ]
sending packet: from 192.168.22.17[500] to 192.168.10.154[500] (312 bytes)
received packet: from 192.168.10.154[500] to 192.168.22.17[500] (240 bytes)
parsed IKE_SA_INIT response 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) N(FRAG_SUP) N(HASH_ALG) N(MULT_AUTH) ]
!! local host is behind NAT, sending keep alives
authentication of 'sun.strongswan.org' (myself) with pre-shared key
establishing CHILD_SA net-net[1]
generating IKE_AUTH request 1 [ Idi N(INIT_CONTACT) IDr AUTH SA TSi TSr N(MOBIKE_SUP) N(ADD_4_ADDR) N(ADD_4_ADDR) ]
sending packet: from 192.168.22.17[4500] to 192.168.10.154[4500] (384 bytes)
received packet: from 192.168.10.154[4500] to 192.168.22.17[4500] (256 bytes)
parsed IKE_AUTH response 1 [ IDr AUTH SA TSi TSr N(AUTH_LFT) N(MOBIKE_SUP) N(NO_ADD_ADDR) ]
authentication of 'moon.strongswan.org' with pre-shared key successful
IKE_SA net-net[1] established between 192.168.22.17[sun.strongswan.org]...192.168.10.154[moon.strongswan.org]
scheduling reauthentication in 3353s
maximum IKE_SA lifetime 3533s
CHILD_SA net-net[1] established with SPIs c0e244a8_i c6b9e7cf_o and TS 192.168.22.17/32 === 192.168.10.154/32
connection 'net-net' established successfully
```

step 5: 查看拨号状态:

```
/etc # ipsec status
Security Associations (1 up, 0 connecting):
net-net[1]: ESTABLISHED 15 seconds ago, 192.168.22.17[sun.strongswan.org]...192.168.10.154[moon.strongswan.org]
net-net{1}: INSTALLED, TUNNEL, reqid 1, ESP in UDP SPIs: c3e576f1_i cd50a876_o
net-net{1}: 192.168.22.17/32 === 192.168.10.154/32

/etc # ip -s xfrm state
src 192.168.22.17 dst 192.168.10.154
proto esp spi 0xc6b9e7cf(33340727271) reqid 1(0x00000001) mode tunnel
replay-window 0 seq 0x00000000 flag af-unspec (0x00100000)
auth-trunc hmac(sha256) 0xab1ca4f01aacb4d4b9674950bea3397c82748659a6586591926b9427eea13a0c (256 bits) 121
enc cbc(aes) 0x768ccdaf03ea112f160007c82985fe87 (128 bits)
encap type espinudp sport 4500 dport 4500 addr 0.0.0.0
lifetime config:
limit: soft (INF)(bytes), hard (INF)(bytes)
limit: soft (INF)(packets), hard (INF)(packets)
expire add: soft 944(sec), hard 1200(sec)
expire use: soft 0(sec), hard 0(sec)
lifetime current:
0(bytes), 0(packets)
add 2018-03-04 08:54:15 use -
stats:
replay-window 0 replay 0 failed 0
src 192.168.10.154 dst 192.168.22.17
proto esp spi 0xc0e244a8(3236054184) reqid 1(0x00000001) mode tunnel
replay-window 32 seq 0x00000000 flag af-unspec (0x00100000)
auth-trunc hmac(sha256) 0x5c7e11f1aac4427f472996b9f5cbac6ddde3b4319b7103d921835db11879952a (256 bits) 121
enc cbc(aes) 0xea80a18be09f76c8cc097fbfc1647ca7 (128 bits)
encap type espinudp sport 4500 dport 4500 addr 0.0.0.0
lifetime config:
limit: soft (INF)(bytes), hard (INF)(bytes)
limit: soft (INF)(packets), hard (INF)(packets)
expire add: soft 954(sec), hard 1200(sec)
expire use: soft 0(sec), hard 0(sec)
lifetime current:
0(bytes), 0(packets)
add 2018-03-04 08:54:15 use -
stats:
replay-window 0 replay 0 failed 0
```