SMP

**1 安全性**

通过调用函数bt\_conn\_security()设置安全性。

* 1. 安全性改变

BLE安全性改变的发起有两种方式，可以由MASTER或SLAVE发起，但配对只能由MASTER发起，因此两种安全性改变方式有不同的过程。若MASTER主动发起安全性改变，则直接判断LTK是否存在，存在则开始加密连接，否则发起配对。若SLAVE主动发起安全性改变，则先发送安全性改变请求，再有MASTER进行后续处理。

绑定：若使能了绑定，则在配对时会保存生成的LTK等密钥，下次再改变安全性时，首先根据地址查找LTK，若找到则直接开始加密，而无需再次进行配对。

注：密钥的存储由协议栈完成，存储方法的实现在Host/storage.c文件中，利用了zephyr的文件系统，若用户想要自己实现，只需修改storage\_init函数，调用Host/Hci\_core.c中的bt\_storage\_register()函数为bt\_storage结构体注册读写等函数。

设置连接安全性

bt\_conn\_security()

发送安全性改变请求

bt\_smp\_send\_security\_req()

MASTER

SLAVE

处理安全性改变请求

smp\_security\_request()

查找LTK是否存在

bt\_keys\_find(BT\_KEYS\_LTK)

开始加密连接

bt\_keys\_find(BT\_KEYS\_LTK)

发送配对请求

bt\_smp\_send\_pairing\_req()

否

是

* 1. 配对过程

注意：决定协议栈采用LE遗留配对还是LE安全连接配对由bool变量sc\_supported决定，sc\_supported的值在bt\_smp\_init()函数中进行初始化（通过读取蓝牙设备支持的命令来进行判断，要支持SC配对，控制器必须支持LE Read Local P256 Public Key和LE Generate DH Key命令）。

函数作用说明：

**bt\_smp\_send\_pairing\_req()**：通过conn获取smp通道，检查是否超时，配对是否正在进行，安全等级是否可实现，初始化SMP（smp\_init），设置配对请求参数，发送配对请求。

**smp\_init()**：将smp（除了L2CAP通道部分）置0，生成本地随机数（smp->prnd）。

**smp\_pairing\_req()**：初始化SMP，存储配对请求（smp->preq），设置配对响应参数，发送配对响应。

**legacy\_pairing\_req()**：获取配对方法（legacy\_get\_pair\_method），若配对方法为仅工作且认证回调(bt\_auth)的配对确认函数（pairing\_confirm）不为空，则调用配对确认函数。发送配对响应，申请STK（legacy\_request\_tk）。

**legacy\_get\_pair\_method()**：若配对请求和配对响应都未要求中间人认证（AUTH\_MITM），则配对方法为仅工作。否则根据本机和远程设备的IO能力获取配对方法，当双方都有输入显示（KeyboardDisplay）能力时，主机使用密码显示方法，从机使用密码输入方法。

**legacy\_request\_tk()**：根据连接的目标地址查找对应的密钥，若配对方法为仅工作且密钥是经过身份认证的，则返回错误码。若配对方法为显示密码（PASSKEY\_DISPLAY），则生成6位整数的密码，并调用认证回调的密码显示函数（passkey\_display），将密码复制到STK（smp->tk）中。若配对方法为输入密码（PASSKEY\_INPUT），则设置smp的标志为用户，调用认证回调的密码输入函数（passkey\_entry）。若配对方法为仅工作，则直接返回。

**passkey\_entry()**：此函数由用户注册，用户输入密码后应将密码传递给协议栈（bt\_conn\_auth\_passkey\_entry）。

**bt\_conn\_auth\_passkey\_entry()**：将密码传递给SMP层处理（bt\_smp\_auth\_passkey\_entry）。

**bt\_smp\_auth\_passkey\_entry()**：根据连接获取smp，进行遗留密码输入（legacy\_passkey\_entry）。

**legacy\_passkey\_entry()**：将密码复制到STK（smp->tk）中，进行遗留配对确认（legacy\_pairing\_confirm）。

**legacy\_pairing\_confirm()**：发送遗留配对确认（legacy\_send\_pairing\_confirm）。

**legacy\_send\_pairing\_confirm()**：创建配对确认数据包，生成确认值（smp\_c1）保存在req->val中，发送配对确认命令。

**smp\_pairing\_confirm()**：将确认值复制到smp->pcnf中，若为主机，则发送配对随机值（smp\_send\_pairing\_random）。否则进行遗留配对确认。

**smp\_send\_pairing\_random()**：创建配对随机值数据包，复制本地随机值到数据包中，发送配对随机值命令。

**smp\_pairing\_random()**：复制收到的配对随机值到远程随机数中（smp->rrnd），进行遗留配对随机数处理（legacy\_pairing\_random）。

**legacy\_pairing\_random()**：根据收到的随机数生成确认值（smp\_c1），并与之前收到的确认值（smp->pcnf）进行比较，不相同的话说明配对失败。若为主机，则生成STK（smp\_s1）并开始加密（bt\_conn\_le\_start\_encryption）。若为从机，则生成STK并将其保存到smp->tk中，然后发送配对随机值。

**bt\_conn\_le\_start\_encryption()**：创建开始加密数据包，设置参数，发送开始加密HCI命令。

**hci\_encrypt\_change()**：调用用户注册的安全性改变回调函数。

**bt\_smp\_encrypt\_change()**：

**bt\_smp\_distribute\_keys()**：遗留分发密钥，使用IDENT\_INFO命令发送IRK（bt\_dev.irk），使用IDENT\_ADDR\_INFO命令发送设备公共地址（bt\_dev.id\_addr），生成CSRK，使用SIGNING\_INFO命令发送CSRK。若使能了绑定，则复制CSRK到conn->le.keys->local\_csrk中。

**legacy\_distribute\_keys()**：生成LTK、Rand以及EDIV，使用ENCRYPT\_INFO命令发送LTK，使用MASTER\_IDENT命令发送Rand和EDIV。如果使能了绑定，则将上述密钥复制到conn->le.keys->slave\_ltk中。

**smp\_encrypt\_info()**：若使能了绑定，则由连接的目标地址获取对应密钥，然后将收到的LTK复制到密钥中。

**smp\_master\_ident()**：若使能了绑定，则由连接的目标地址获取对应密钥，然后将收到的EDIV和Rand复制到密钥中。若为主机，则分发密钥并完成配对。

**smp\_ident\_info()**：若使能了绑定，则由连接的目标地址获取对应密钥，然后将收到的IRK复制到密钥中。

**smp\_ident\_addr\_info()**：先判断接收到的地址是否为身份地址（公共地址或静态地址），不是则返回错误。若使能了绑定，则由连接的目标地址获取对应密钥，然后将收到的复制到密钥中。调用用户注册的身份地址解析函数（identity\_resolved）。

**smp\_signing\_info()**：若使能了绑定，则根据连接的目标地址获取对应密钥（bt\_keys），并将收到的CSRK复制到密钥中。若为主设备且还有密钥未分发，则继续分发密钥。若所有密钥都已分发完成，则完成配对。

**smp\_pairing\_complete()**：重置SMP（smp\_reset）。

**smp\_reset()**：将配对方法（smp->method）设置为JUST\_WORKS，清除标志位（smp->allowed\_cmds和smp->flags），若为主设备，则允许接收安全请求，若位从设备，则允许接收配对请求。

发送配对请求

bt\_smp\_send\_pairing\_req()

MASTER

SLAVE

处理配对请求

smp\_security\_request()

执行 LE 遗留配对

legacy\_pairing\_req()

发送配对响应

send\_pairing\_rsp()

处理配对响应

smp\_pairing\_rsp()

执行 LE 遗留配对

legacy\_pairing\_rsp()

用户输入配对密钥

bt\_auth->passkey\_entry()

调用此函数传入密钥

bt\_conn\_auth\_passkey\_entry ()

LE 遗留配对密钥输入

legacy\_passkey\_entry()

LE 遗留配对确认

legacy\_pairing\_confirm()

LE 遗留配对发送确认请求

legacy\_send\_pairing\_confirm()

处理配对确认请求

smp\_pairing\_confirm()

处理配对确认请求

legacy\_pairing\_confirm()

向用户显示密钥并保存

bt\_auth->passkey\_display()

配对过程，这里假设MASTER具备输入能力，SLAVE具备显示能力

发送配对确认值

legacy\_send\_pairing\_confirm()

处理配对确认值

smp\_pairing\_confirm()

发送配对随机值

smp\_send\_pairing\_random()

处理配对随机值：检查确认值是否匹配，不匹配则返回错误

smp\_pairing\_random()

LE处理配对随机值

legacy\_pairing\_random()

发送配对随机值

smp\_send\_pairing\_random()

处理配对随机值

smp\_pairing\_random()

LE处理配对随机值

legacy\_pairing\_random()

LE 开始连接加密（发送命令）

bt\_conn\_le\_start\_encryption()

处理加密改变事件(收到事件)

hci\_encrypt\_change()

1.3 开始加密

开始加密过程中首先由MASTER发送HCI\_LE\_START\_ENCRYPTION命令，注意这个HCI命令由控制器处理时会发送LL\_ENC\_REQ数据包（Spec Vol6 PartB 2.4.2.4）给SLAVE。

LL\_ENC\_REQ的CtrData由4个字段组成：

Rand：8字节，

EDIV：2字节，

SKDm：8字节，

IVm：4字节，

发送此命令后，若连接还未加密，MASTER和SLAVE的控制器都会生成Encryption Change事件，否则生成Encryption Key Refresh Complete事件。

事件的处理见Hci\_core.c中的hci\_event函数。

在加密改变事件的处理中，先由SLAVE将密钥分发给MASTER，包括以下密钥：

LTK：使用Encryption Information命令。

EDIV和Rand：使用Master Identification命令。

IRK：使用Identity Information命令。

公共设备地址或静态随机地址：使用Identity Address Information命令。

CSRK：使用Signing Information命令。

之后再由MASTER将密钥分发给SLAVE，分发的密钥同上。

身份解析密钥（Identify Resolving Key，IRK）用于可解析私有地址的构造。主机接收到从机的IRK后可以解析从机的随机设备地址。从机接收到主机的IRK后可以解析主机的随机设备地址。IRK可以指定，也可以在设备制作过程中随机生成，或者使用其他方法。

连接签名解析密钥（Connection Signature Resolving Key，CSRK）用于对连接中的数据进行签名。接收到CSRK的设备可以对分发设备生成的签名进行验证。CSRK可以由设备在制造过程中分配或随机生成，也可以使用其他方法。

加密多元化（Encrypted Diversifier，EDIV）用于标识LE遗留配对期间分发的LTK，每次分发一个新的LTK时，都会生成一个新的EDIV。

随机数（Random Number，Rand）用于失败LE遗留配对中分发的LTK，每次分发一个新的LTK时，都会生成一个新的Rand。

注：EDIV和Rand用于生成以前共享的LTK，以便于以前配对的主设备启动加密连接。

开始加密过程

LE 开始连接加密（发送命令）

bt\_conn\_le\_start\_encryption()

MASTER

SLAVE

处理LTK Request事件

Hci\_core.c: le\_ltk\_request()

处理加密改变事件(收到事件)

hci\_encrypt\_change()

发送加密命令

Hci.c: le\_start\_encryption()

LL\_ENC\_REQ

处理加密改变事件

bt\_l2cap\_encrypt\_change()

调用安全性改变回调函数

bt\_conn\_security\_change()

ATT加密改变处理

Att.c: bt\_att\_encrypt\_change()

SMP加密改变处理

Smp.c: bt\_smp\_encrypt\_change

发送LTK\_REQ\_REPLY命令

LL\_START\_ENC\_REQ

处理加密改变事件(收到事件)

hci\_encrypt\_change()

SMP加密改变处理

bt\_smp\_encrypt\_change()

ATT加密改变处理

bt\_att\_encrypt\_change()

SMP分发密钥

bt\_smp\_distribute\_keys()

遗留分发密钥

Smp.c: legacy\_distribute\_keys()

遗留分发密钥

ENCRYPT\_INFO

MASTER\_IDENT

IDENT\_INFO

IDENT\_ADDR\_INFO

SIGNING\_INFO

处理分发密钥事件

smp\_encrypt\_info()

smp\_master\_ident()

smp\_ident\_info()

smp\_ident\_addr\_info()

smp\_signing\_info()

