解密蓝牙mesh系列 | 第十篇

2017-11-10 任凯 蓝牙技术联盟

蓝牙mesh网络
启动配置Part2

蓝牙技术联盟亚太区技术项目经理
任凯

在本文的Part1中，我介绍了启动配置承载层(provisioning bearer layer)和蓝牙mesh启动配置流程的前三个阶段：**发送Beacon信号**、**邀请**和**交换公共密钥**。

启动配置流程包括五个阶段：

1. **发送Beacon信号**
2. **邀请**
3. **交换公共密钥**
4. **认证**
5. **启动配置数据分发**

1. **发送Beacon信号**：如果未经启动配置的设备支持PB-ADV承载层，则其作为未经启动配置设备Beacon进行广播；如果使用的是PB-GATT承载层，则发送可连接的广播数据包。这就向启动配置设备（Provisioner）表明未经启动配置的设备已做好准备，可进入启动配置流程。

2. **邀请**：启动配置设备（Provisioner）邀请未经启动配置的设备发送自身启动配置功能信息。

3. **交换公共密钥**：在此阶段，根据未经启动配置设备的功能，启动配置设备(Provisioner)选择合适的验证方法，并通知未经启动配置设备将要采取的方式。之后，启动配置设备和未经启动配置设备会创建一个椭圆曲线公私密钥对并交换公钥。然后，每台设备使用自己的私钥和对等设备的公钥来计算对称密钥，即ECDHSecret。该密钥用于验证对端设备的身份。

本文将介绍启动配置流程的后两个阶段：**认证**和**启动配置数据分发**。

④ 认证

在此步骤中，启动配置设备使用所选的验证方法，对未经启动配置设备进行验证。有三种可用的验证方法(OOB, Out-Of-Band)：**输出OOB（Output OOB）**、**输入OOB（Input OOB）**、以及**静态OOB（Static OOB）**或**无OOB（No OOB）**。

输出带外（Output OOB）

若选择的是输出带外（Output OOB）验证方法，**则未经启动配置设备会选择一个随机数，并通过与其功能兼容的方式输出该数字**。例如，如果未经启动配置设备是一个灯泡，则它能够闪烁指定的次数。如果设备具有LCD屏幕，则可以将随机数显示为多位数值。启动配置设备（Provisioner）的用户需要输入观察到的数字，来验证未经启动配置的设备。**输出带外验证方法的工作流程如图1所示**。

图1 – 通过输出OOB进行验证

输入随机数后，启动配置设备（Provisioner）生成并检查确认值。**无论采用哪种验证方式，整个验证步骤中的检查确认值（check confirmation value）计算方式都是相同的**，请继续往下看。

输入带外（Input OOB）

输入带外（Input OOB）验证方法与输出带外（Output OOB）方法类似，**但设备的角色相反**。启动配置设备（Provisioner）生成并显示随机数，然后提示用户采取适当的操作，将随机数输入未经启动配置的设备。**以照明开关为例，用户可以在一定时间内数次按下按钮，以这种形式输入随机数**。

与输出带外验证（Output OOB）相比，**输入带外（Input OOB）方法需要发送一个附加的启动配置协议PDU**。在完成认证操作之后，未经启动配置的设备向启动配置设备发送一个启动配置输入完成PDU（Provisioning Input Complete PDU），通知其随机数已输入完成。随后进入到执行检查确认值操作的步骤。

图 2 – 通过输入OOB进行验证

静态带外(Static OOB) 或无带外（No OOB）

在输入带外或输出带外都不可用的情况下，启动配置设备（Provisioner）和未经启动配置的设备可采用静态带外（Static OOB）验证或无带外（No OOB）验证：**采用静态OOB信息；或静态OOB信息不可用，直接以数值0代替**。在此情况下，启动配置设备和未经启动配置的设备各自生成一个随机数，然后进行检查确认值操作。

检查确认值(Check Confirmation Value)

无论采用何种验证方法，都会进行确认值生成和检查。根据蓝牙mesh规格，启动配置设备(Provisioner) 和未经启动配置设备应分别计算确认值。这两个值被称为**ConfirmationProvisioner**和**ConfirmationDevice**。这两个值的计算都使用一系列相同的函数，不同之处仅在于所使用的随机数输入。

蓝牙mesh规格中有两页关于计算过程的内容说明。**确认值生成函数需要8个参数，参数值来自启动配置（Provisioning）过程中的后续步骤**。可参考规格的5.4.2.4认证和5.4.1启动配置PDU部分，了解更多详细信息。（详见：https://www.bluetooth.com/specifications/mesh-specifications ）

表1列出了用于确认值生成及其来源的参数。

参数	来源
ProvisioningInvitePDUValue ProvisioningCapabilitiesPDUValue	第2步
ProvisioningStartPDUValue PublicKeyProvisioner PublicKeyDevice	第3步
ECDHSecret	第3步
AuthValue	第4步：如果采用了输出带外或输入带外的方法，则AuthValue是用户输入的值。
RandomProvisioner	第4步：来自启动配置设备（Provisioner）的随机数。
RandomDevice	第4步：来自未经启动配置设备的随机数。

表1 - 用于确认值计算的参数

让我们来看看用于确认值生成的算法。**图3是一个流程图，其中包括了几轮AES-CMAC和SALT生成**[1]。该流程图对于ConfirmationProvisioner和ConfirmationDevice值均适用。

- **如果由启动配置设备(Provisioner)执行输入**，则从一个名为RandomProvisioner的值开始（下图中以蓝色表示），并生成ConfirmationProvisioner值。

- **如果由未经启动配置设备执行输入**，则从一个名为RandomDevice的值开始（下图中以绿色表示），并生成ConfirmationDevice值。

图 3 – 确认值生成图

确认值检查(Confirmation Value Check)

当确认值生成之后，两台设备就会进行交换，并且都会检查接收值的完整性。图4表示确认值检查的过程。

确认过程的开始就是启动配置设备（Provisioner）将其随机数RandomProvisioner发送到未经启动配置的设备。未经启动配置设备使用它来重新计算确认值，并与之前接收的确认值进行比较，进行验证。

图 4 –确认值检查

- **如果由未经启动配置设备计算所得的确认值与接收到的**

ConfirmationProvisioner不匹配，则启动配置（Provisioning）过程将被中止。

- **如果由未经启动配置设备计算所得的确认值与接收到的ConfirmationProvisioner匹配**，则未经启动配置设备将其RandomDevice值发送给启动配置设备。

然后，启动配置设备(Provisioner) 使用相同的过程来重新计算确认值，并通过比较计算所得值与先前接收值来进行验证。

- **如果由启动配置设备(Provisioner) 计算所得的确认值与接收到的ConfirmationDevice不匹配，**则启动配置（Provisioning）流程将被中止。

- **如果由启动配置设备(Provisioner) 计算所得的确认值与接收到的ConfirmationDevice匹配**，则表示验证成功。后续只要启动配置设备（Provisioner）和未经启动配置设备完成启动配置流程的第五步：启动配置数据分发，则未经启动配置设备就能成为蓝牙mesh网络中的节点（node）。

⑤ 启动配置数据分发

认证步骤完成之后，就可以确保在启动配置设备（Provisioner）和未经启动配置设备之间建立的承载层的安全，然后就进入启动配置(Provisioning)过程中最重要的一步：**导出并分发启动配置数据（provisioning data）**。启动配置设备(Provisioner) 负责生成启动配置数据，启动配置数据由多个数据项组成，包括一个称为网络密钥 (NetKey) 的安全密钥。**表2列出了启动配置数据字段**。

字段	大小（字节）	注释
网络密钥	16	简称NetKey。NetKey确保网络层（network layer）通信的安全，并在网络中所有节点(node)之间共享。是否拥有给定的NetKey定义了给定蓝牙mesh网络或子网的成员资格。为设备赋予网络的NetKey是启动配置（Provisioning）流程的主要结果之一。 启动配置设备(Provisioner) 在对要添加到网络中的首台设备进行启动配置时创建NetKey。
设备密钥	16	简称DevKey，只有启动配置设备（Provisioner）和被启动配置的设备拥有的唯一安全密钥。
密钥索引	2	由于NetKey太长，无法在单段消息中传输。为使消息传递尽可能高效，会向密钥分配一个全球唯一的12位索引值，称为密钥索引，用作密钥的短标识符。消息中包括密钥索引值，它可能以配置客户端（Configuration Clients）维护的密钥列表为参考。
标志	1	标志位掩码 - 指示关联密钥的状态。
IV 索引	4	IV（初始化向量）索引是一个32位的值，被网络中的所有节点 (node)共享。其目的是在计算消息随机值时提供熵（随机性）。
单播地址	2	新节点中主要元素的单播地址(Unicast Address)。

表 2 – 启动配置数据(provisioning data)

为安全地进行启动配置数据分发，启动配置设备（Provisioner）采用AES-CCM [2]，**借助共享的会话密钥（SessionKey）对启动配置数据(provisioning data)进行加密，启动配置设备和未经启动配置设备都会进行计算**。 AES-CCM需要三个输入参数：**会话密钥**、**会话随机数**和**纯文本**。纯文本参数值包含需要被加密的启动配置数据。设备密钥（DevKey）、会话密钥（SessionKey）和会话随机数（SessionNonce）的值通过图5所示的过程导出。

图5 - DevKey / SessionKey / SessionNonce生成过程

从图5可以看出：

- 如果将prsk（绿色）的输入值代入k1函数中，则会生成SessionKey。这是用于启动配置数据加密的关键。
- 如果将prsn（黄色）的输入值代入k1函数中，将生成SessionNonce。这是用于启动配置数据加密的随机值。
- 如果将prdk（蓝色）的输入值代入k1函数，则将生成DevKey。

启动配置设备（Provisioner）和未经启动配置设备都需要生成会话密钥（SessionKey）和会话随机数（SessionNonce）。 当SessionKey和SessionNonce值准备就绪时，启动配置设备将对包含导出启动配置数据的启动配置数据PDU (Provisioning Date PDU) 进行加密，并将其发送至未经启动配置的设备。此处，相同的SessionKey和SessionNonce值也可用来对接收到的数据进行解密。

至此，启动配置流程完成。两台对等设备都已知晓新的设备密钥（DevKey）和全网的网络密钥（NetKey），这就**意味着我们的新设备已成为蓝牙mesh网络中的节点（node）和成员**。

开发mesh

启动配置过程是蓝牙mesh安全的基础，让网络设备之间能够可靠安全地进行通信。如果您想了解有关蓝牙mesh的更多信息，请下载蓝牙mesh规格，蓝牙mesh创新产品开发所需的一切都将尽在掌握。

References:

[1] 有关AES-CMAC和SALT的更多信息，请参阅《蓝牙mesh - 安全概述》。
[2] 有关这些术语的更多信息，请参阅《蓝牙mesh - 安全概述》。

精选留言

写留言

刘琦
感谢！困扰多日的问题茅塞顿开
2017年11月19日

以上留言由公众号筛选后显示

了解留言功能详情