Bluetooth blog

Bluetooth Mesh Security Overview

Posted on September 11, 2017 by Kai Ren and Martin Woolley
Chapter 8 of the Bluetooth Mesh Networking Series

The Criticality of Security

One of the most discussed issues related to the Internet of Things (IoT) is security. From agriculture to hospitals, from residential smart homes to commercial smart buildings, and from power stations to traffic management systems, IoT systems and technologies will touch many parts of the world we live in. Security breaches in IoT systems could have catastrophic consequences.

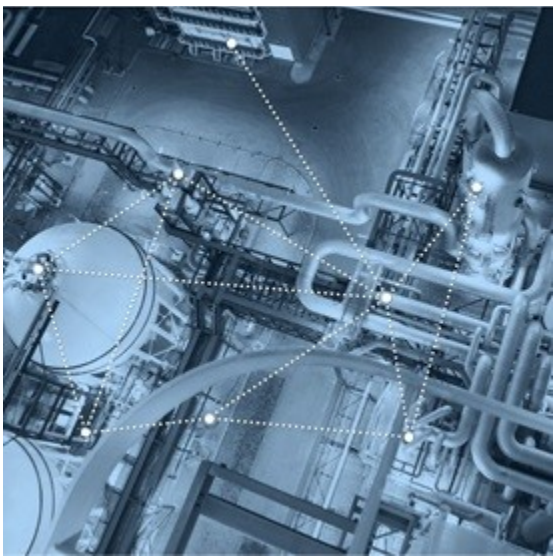

Figure 1 - Security breaches in IoT systems could be catastrophic.

Bluetooth® mesh networking was designed with security as its number one priority and from the ground up. In this article, you'll get an overview of the key security features and the security issues addressed. Further articles in the series will examine aspects of Bluetooth mesh networking security in more detail.

Security in Bluetooth Mesh Networking is Mandatory

Bluetooth Low Energy (LE) GATT devices may implement a range of security measures as defined in the Bluetooth core specification. It's the responsibility of the product designer to decide what security measures are required and it's permissible to decide to adopt none of the available security features at all. In other words, security in Bluetooth Low Energy GATT is optional. This makes sense if we're talking about the security of a single device and its connection with one other device, provided the product designer performs their risk assessment correctly. However, security in Bluetooth mesh networking is concerned with the security of more than individual devices or connections between peer devices; it's concerned with the security of an entire network of devices and of various groupings of devices in the network.

Consequently, **security in Bluetooth mesh networking is mandatory.**

FEATURED RESOURCE

Bluetooth Mesh Networking - An Introduction for Developers

Download this comprehensive technology overview to learn more about the key concepts and terminology, system architecture, and security mechanisms, as well as the unique message publication and delivery technique behind Bluetooth mesh networking.

Bluetooth Mesh Networking Security Fundamentals

The following fundamental security statements apply to all Bluetooth mesh networks:

Encryption and Authentication	All Bluetooth mesh messages are encrypted and authenticated.
Separation of Concerns	Network security, application security, and device security are addressed independently. See *Separation of Concerns* below.
Area Isolation	A Bluetooth mesh network can be divided into subnets, each cryptographically distinct and secure from the others.
Key Refresh	Security keys can be changed during the life of the Bluetooth mesh network via a Key Refresh procedure.

Message Obfuscation	Message obfuscation makes it difficult to track messages sent within the network and, as such, provides a privacy mechanism to make it difficult to track nodes.
Replay Attack Protection	Bluetooth mesh security protects the network against replay attacks.
Trashcan Attack Protection	Nodes can be removed from the network securely, in a way which prevents trashcan attacks.
Secure Device Provisioning	The process by which devices are added to the Bluetooth mesh network to become nodes is a secure process.

Separation of Concerns and Security Keys

At the heart of Bluetooth mesh security are three types of security keys. These keys provide security to different aspects of the Bluetooth mesh network and achieve a critical capability in Bluetooth mesh networking security called a separation of concerns.

Consider a mesh light which acts as a relay. In its capacity as a relay, it may find itself handling messages relating to the building's Bluetooth mesh door and window security system. A light has no business accessing and processing the details of these messages, but it does need to relay them to other nodes.

To deal with this potential conflict of interest, Bluetooth mesh uses different security keys called AppKeys for securing messages at the network layer from those used to secure data relating to specific applications, such as lighting, physical security, heating, etc.

All nodes in a Bluetooth mesh network possess one or more Network Keys (NetKey), each corresponding to a subnet which may be the primary subnet. It's possession of a network key which makes a node a member of the network. Network Encryption Keys and Privacy Keys are derived directly from the NetKey.

Being in possession of a NetKey allows a node to decrypt and authenticate up to the Network Layer so that network functions, such as relaying, can be carried out. It does not allow application data to be decrypted.

Each node also has a unique security key called the Device Key or DevKey. The DevKey is used in the provisioning and configuration of the node.

Area Isolation

Possession of the primary NetKey defines membership of and grants access to the Bluetooth mesh network. But it's also possible to divide the network into distinct subnets, each with its own subnet key. This means that only devices in possession of a given subnet key can communicate with other devices that are members of that subnet. Subnet keys can be created and assigned on an adhoc basis too. A great example is isolating nodes in different hotel rooms from each other.

Node Removal, Key Refresh, and Trashcan Attacks

As described above, nodes contain various Bluetooth mesh security keys. Should a node become faulty and need to be disposed of, or if the owner decides to sell the node to another owner, it's important that the device and the keys it contains cannot be used to mount an attack on the network the node was taken from.

Figure 2 - Bluetooth mesh networking ensures devices can be disposed of securely.

A procedure for removing a node from a network is defined. The Provisioner application is used to add the node to a *black list* and then a Key Refresh Procedure is initiated.

The Key Refresh Procedure issues all nodes in the network, except those which are members of the black list, new network keys, application keys, and all related, derived data. In other words, the entire set of security keys which form the basis for network and application security are replaced.

As such, a node which was removed from the network, and which contains an old NetKey and old set of AppKeys, is no longer a member of the network and poses no threat.

Privacy

A Privacy Key, derived from the NetKey, is used to obfuscate network PDU header values, such as the source address. Obfuscation ensures that casual, passive eavesdropping cannot be used to track nodes and the people using them. It also makes attacks based upon traffic analysis difficult.

Replay Attacks

In network security, a replay attack is a technique whereby an eavesdropper intercepts and captures one or more messages and simply retransmits them later, with the goal of tricking the recipient into carrying out something which the attacking device is not authorized to do. A commonly cited example is a car's keyless entry system being compromised by an attacker who intercepts the authentication sequence between the car's owner and the car, later replaying those messages to gain entry to the car and steal it.

Bluetooth mesh networking protects against replay attacks by using two network PDU fields called the Sequence Number (SEQ) and IV Index. Elements increment the SEQ value every time they publish a message. A node which receives a message from an element containing an SEQ value less than or equal to that of the last valid message will discard it since it likely relates to a replay attack. Similarly, IV Index is a separate field considered alongside SEQ. IV Index values within messages from a given element must always be equal to or greater than the last valid message from that element.

Cryptography Toolbox

Most of the security features of Bluetooth mesh networking rely upon industry-standard cryptographic algorithms and procedures. Reference will be made to them in other security-related articles in this series, but we'll explain the most important ones here.

There are two key security functions used in the Bluetooth mesh stack: AES-CMAC and AES-CCM. These are fundamental encryption and authentication functions, and all other functions used for key generation are based upon them.

AES-CMAC

Cipher-based Message Authentication Code (CMAC) is an algorithm which can generate a fixed length, 128-bit message authentication value for any variable length input. The formula for generating a message authentication code MAC using the AES-CMAC algorithm is written as:

$$MAC = AES\text{-}CMAC_k(m)$$

The inputs to AES-CMAC are:

k - the 128-bit key.

m - the variable length data to be authenticated.

AES-CMAC has excellent error-detection capabilities. Other techniques, involving the verification of a checksum or use of an error-detecting code, may detect only accidental modifications of the data. AES-CMAC is designed to detect intentional, unauthorized modifications of the data, as well as accidental modifications. If you are interested in learning more about this function, please refer to RFC4493 which defines it.

AES-CCM

AES-CCM is a generic, authenticated encryption algorithm, intended for use with cryptographic block ciphers. In the Bluetooth mesh specification, AES-CCM is used as the basic encryption and authentication function in all cases. The formula for its use is as follows:

ciphertext , MIC = $AES\text{-}CCM_k$ (n,m,a)

There are four inputs to AES-CCM:

k - the 128-bit key.

n - a 104-bit nonce.

m - the variable length data to be encrypted and authenticated.

a - the variable length data to be authenticated but not encrypted, also known as *Additional Data*. This input parameter may be zero bytes in length.

There are two outputs from AES-CCM:

ciphertext - the variable length data after it has been encrypted.

MIC - the Message Integrity Check value of m and a.

Figure 3 shows a plain-text payload, which may be from the Bluetooth mesh network layer or upper transport layer, being processed by AES-CCM with an input encryption key, nonce, and plaintext payload. An encrypted payload and MIC is output.

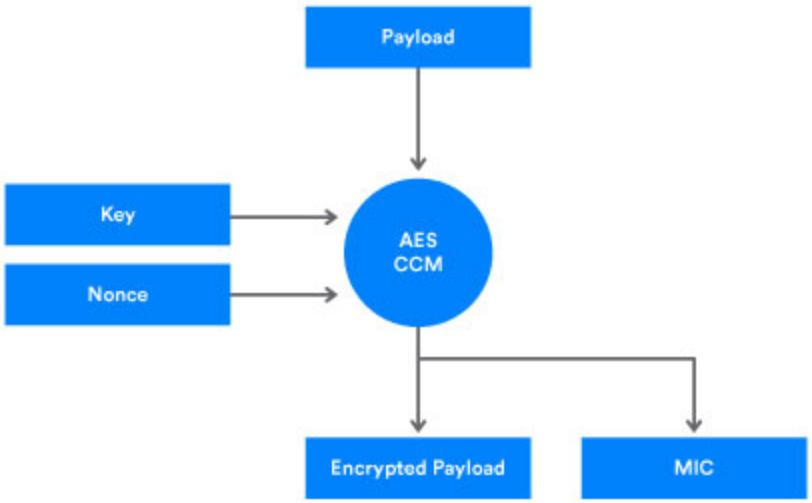

Figure 3 - AES-CCM used for packet payload encryption and authentication.

SALT Generation

Bluetooth mesh security defines a SALT generation function known as s1, which uses the AES-CMAC function. As explained above, AES-CMAC has two input parameters: k and m. When used for SALT generation though, only the input parameter m varies. k is always set to the 128-bit value: *0x0000 0000 0000 0000 0000 0000 0000 0000,* which is referred to as ZERO in the Bluetooth mesh specification.

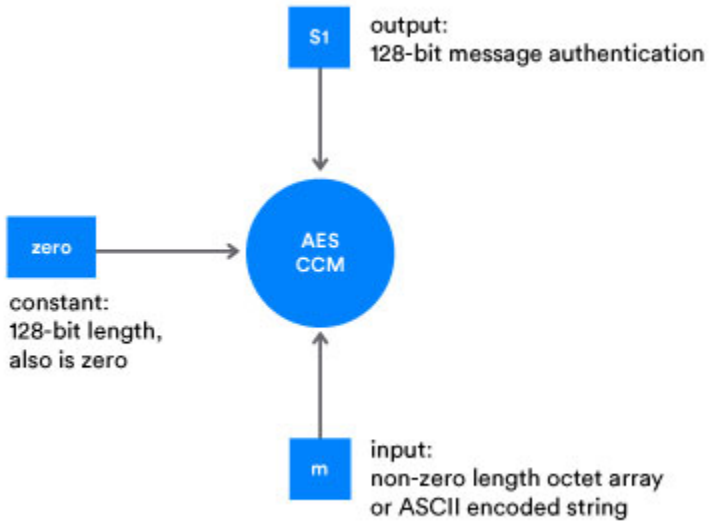

Figure 4 - The SALT generation function.

The input to the SALT generation function is:

m - a non-zero length octet array or ASCII encoded string.

The output is a 128-bit MAC value and the s1 formula is written as:

$$s1(m) = \text{AES-CMAC}_{ZERO}(m)$$

Other security functions

In the Bluetooth mesh networking specification section 3.8.2, *Security Toolbox*, you will find other security functions defined, such as various key derivation functions. All of them are based on AES-CMAC and the SALT generation function, s1 (the SALT generation function is also based on AES-CMAC).

Onwards!

Security is an important issue for Bluetooth, and the topic will come up repeatedly in our series on Bluetooth mesh networking. After reading this article, you should have a good understanding of the primary Bluetooth mesh networking security features and some of the underlying cryptography techniques involved. You're now ready to take a deeper dive into Bluetooth mesh network security when we cover the subject in future articles.