



ZigBee™ Alliance

Wireless Control That Simply Works

ZigBee Security

Joseph Soma Reddy
Chair, Security WG, ZigBee Alliance
Sr. Systems Architect, Texas Instruments



Topics

- ZigBee system constraints lead to design goals
- Need to answer
 - ▶ What to secure ?
 - ▶ Who to secure against ?
 - ▶ How much security ?
- Security scope
 - ▶ Packet security
 - ▶ Key setup and maintenance
- ZigBee 2006 scenario
- What is not in the specification !



Design objectives

ZigBee devices are

- based on tiny microcontrollers
- have low memory (code and data)
- deployed in home/industrial scenarios
- easy to use

So we need

- encryption primitive must be simple to implement and execute
- low overhead for key storage / maintenance
- flexible enough to support home/industrial
- easy to use



**Describes Key setup
and maintenance**

Defines Key Types
(Master, Link, Network)

CCM*
(Unified/Simpler
mode of operation)

802.15.4 Security
AES encryption
CCM security modes

ZigBee Security

Uses 128-bit AES algorithm
Strong, NIST approved
security

ZigBee uses the basic security
elements in 802.15.4



What is secured ?

Infrastructure security

- Network access control
- Integrity of packet routing
- Prevent unauthorized use of packet transport



What is secured ?

Application data security

■ Message integrity

- ▶ protects message from being modified in transit

■ Authentication

- ▶ provides assurance on the originator of message

■ Freshness

- ▶ prevents replay attacks

■ Privacy

- ▶ prevents an eavesdropper from listening messages



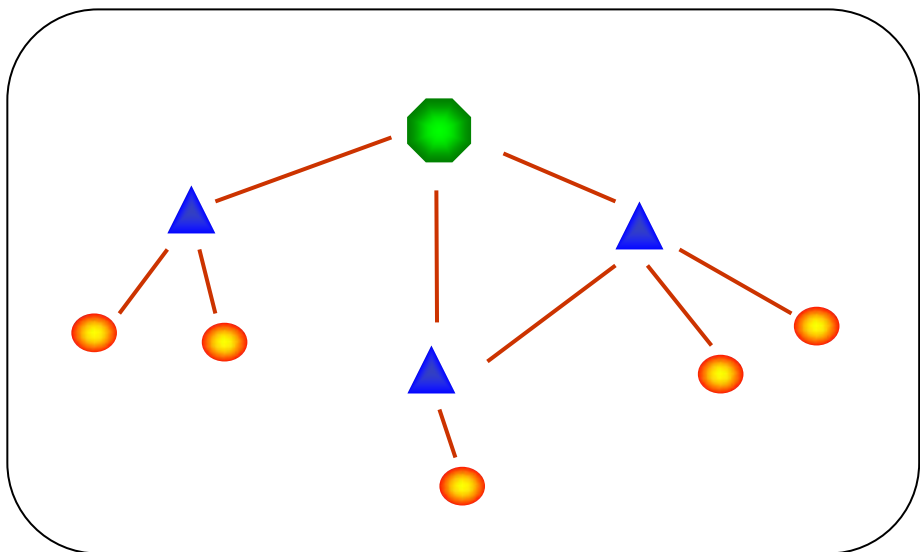
Who is it secured against ?

- Only outsider attacks
 - ▶ Trust all devices in the network
- Insider attacks also
 - ▶ Protect data from other devices on network
 - ▶ Suitable for situations when devices need to use some other network for transport



How much security ?

- Flexible
- Actual cost Vs security tradeoff is configurable by application
- Memory constraints
 - ▶ how many keys and associated info can we store
- Overhead
 - ▶ extra bytes added to each packet
 - ▶ security-specific signaling packets



Coordinator



Router



End Device

ZigBee Network

- When device joins, it must be “admitted” into network
- All devices share a common “*network key*”
- Any two devices may share a “*link key*”
- Key management (key-transport, key-update) is performed by the “*Trust Center*” device



Key types

Network Key *shared key; basis of network-wide security; protects infrastructure and application data from outsider attacks*

Master Key

Derived using SKKE

Link Key

Basis for long-term security between two devices

Basis of security between two devices (insider protection)

- *Keys can be factory-installed or setup over the air or using out-of-band mechanisms (eavesdropping should be prevented when this is setup)*
- *Link and Network keys can be updated periodically*



Trust Center

- Trust Manager
 - ▶ Authenticate devices that request to join network
- Network Manager
 - ▶ Maintains and distributes network keys
- Configuration Manager
 - ▶ Enables end-to-end security between devices by assisting in setup of link keys
- Could be on Coordinator or a dedicated commissioning tool



Use case: ZigBee 2006

- What is secured ?
 - ▶ Infrastructure security ✓
 - ▶ Application data security ✓
- Who to secure against ?
 - ▶ Only outsider attacks ✓
 - ▶ Insider attacks also ✗
- How far to go in providing security ?
 - ▶ Memory, packet overhead etc. >>low



Use case: ZigBee 2006..

- Single network key (no link keys) in use at any time
 - ▶ low memory storage
 - ▶ no protection against insider attacks
- Encryption plus 32-bit MIC
- Trust center is on coordinator
 - ▶ network access policy: app-specific (e.g. button press allows devices for 10 seconds)
 - ▶ network key update policy: app-specific (e.g. update keys every week or manual update)



Application-specific

- Out of band methods for key setup
- Cost/Security tradeoff for number of link keys needed
- Policy for expiration and update of keys
- Policy for accepting new devices



Implementation differences

- Random number generator
 - ▶ If not truly random, encryption primitive is not secure
- Hardware support
 - ▶ CCM and/or AES engine
 - ▶ Random bits
- Handling error conditions
 - ▶ loss of key sync etc.
- Check input (packet) sizes before processing
 - ▶ Buffer over/under flow attacks

Most security attacks are due to implementation flaws!!