Security in ZigBee Wireless Sensor Networks

Presented By:
Anton Kiriwas
Agenda

- General Background
- 802.15.4 Background
- ZigBee Background
  - Network Layers
  - Stack Layer
  - ZigBee Profiles
  - ZigBee Security
  - ZigBee Cluster Library

- Smart Energy Profile (SEP)
- Security, Keys and Commissioning
  - Forward Security
  - Backward Security
  - Certificate Management
Agenda (3)

  - Radio Jamming
  - Link Layer Jamming
  - Backoff Manipulation
  - Same-nonce Attack
  - Replay Protection Attack
  - ACK Attack
  - PANid Conflict Attack
Disclaimer

- You could easily run a series of lectures on each of the network layers of the ZigBee protocol
- I am not an RF expert or a networking expert
- These papers are broad “Survey” type papers – often lead to more questions than answers. I've tried to include references where I can
**General Background**

- ZigBee is a communication specification built upon the PHY and MAC layers of IEEE 802.15.4 Wireless specification.

- Designed for:
  - Low cost
  - Low power
  - Mesh networking
  - Fits in niche between Bluetooth and WiFi

- ZigBee 2004, ZigBee 2006, ZigBee Pro
## General Background (2)

### Rationale for ZigBee and 802.15.4

<table>
<thead>
<tr>
<th>Focus Application</th>
<th>ZigBee and 802.15.4</th>
<th>GSM/GPRS CDMA</th>
<th>802.11</th>
<th>Bluetooth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Battery Life</td>
<td>Years</td>
<td>1 Week</td>
<td>1 Week</td>
<td>1 Week</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>250 Kbps</td>
<td>Up to 2 Mbps</td>
<td>Up to 54 Mbps</td>
<td>720 Kbps</td>
</tr>
<tr>
<td>Typical Range</td>
<td>100+ Meters</td>
<td>Several Kilometers</td>
<td>50-100 Meters</td>
<td>10-100 Meters</td>
</tr>
<tr>
<td>Advantages</td>
<td>Low Power, Cost</td>
<td>Existing Infrastructure</td>
<td>Speed, Ubiquity</td>
<td>Convenience</td>
</tr>
</tbody>
</table>
802.15.4 Background

- **868/900 MHz – Channels 0 to 10**
  - 2 MHz between bands
  - 20 or 40 kbit/s
  - Not used by ZigBee

- **2.4 GHz – Channels 11-26**
  - 5 MHz between bands
  - 100 and 250 kbit/s
  - DSSS (Direct Sequence Spread Spectrum)
  - QPSK (Quadrature Phase Shift Keying)

802.15.4 Background (2)

- **Full Function Devices (FFD)**
  - Capable of being a PAN Coordinator, Coordinator or device
  - Implements entire protocol
  - Can talk to FFDs or RFDs

- **Reduced Function Devices (RFD)**
  - Reduced protocol set
  - Must connect to some established PAN
802.15.4 Background (3)

- **PAN Coordinator (FFD)**
  - Coordinates and acts as control node for entire WPAN

- **Coordinator (FFD)**
  - A device capable of routing and relaying messages between other devices. (Beacon based networks)

- **End Device (FFD or RFD)**
  - Simplest device, not capable of routing
802.15.4 Background (4)

- **Network Device**: Any device that connects to other devices via the 802.15.4 MAC and PHY layer.
- **Coordinator**: A FFD that provides coordination between other FFDs and RFDs.
- **PAN Coordinator**: A coordinator device that is a principal controller of a PAN. Each network has a single PAN coordinator.
802.15.4 Background (5)

- Topologies
  - Star
  - Peer to Peer
  - Combined
Beacon Mode –
- Divides time into periods
  - Beacon
  - Contention Access Period (CSMA/CA)
  - Contention Free Period
    - Guaranteed Time Slots
802.15.4 Background (6)

- Non-beacon Mode –
  - CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance)
  - CCA (Clear Channel Assessment)
    - Energy level threshold
    - Pattern matching of modulation and spreading characteristics
  - Wait until channel is clear before trying to transmit
• ZigBee Stack
  - PHY layer:
  - Adopted the 802.15.4 PHY and MAC layers
• PAN Coordinator → Coordinator
• Router → Coordinator
• End device → End device
ZigBee Background (2)

- ZigBee Stack
  - PHY layer:
    - Preamble provides sync pattern for receiver and decommutator
    - Start of packet is similar
    - Frame length is needed because no footer exists at this level
    - Finally, a payload of up to 127 bytes can be sent with each frame.

<table>
<thead>
<tr>
<th>Preamble</th>
<th>Start of Packet Delimeter</th>
<th>Frame Length</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 Octets</td>
<td>1</td>
<td>1</td>
<td>0-127</td>
</tr>
</tbody>
</table>
ZigBee Background (3)

- ZigBee Stack
  - MAC layer: MAC Frame Format
### ZigBee Background (4)

- **ZigBee Stack**
  - MAC layer: MAC Beacon Frame Format

<table>
<thead>
<tr>
<th>MAC header</th>
<th>MAC payload</th>
<th>MAC footer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame Control</td>
<td>Data Sequence Number</td>
<td>Address</td>
</tr>
<tr>
<td>Frame order</td>
<td>Beacon order</td>
<td>Superframe Spec.</td>
</tr>
<tr>
<td>2 Octet</td>
<td>1</td>
<td>4/20</td>
</tr>
</tbody>
</table>

- **Bits:**
  - 0-3: Beacon order
  - 4-7: Superframe order
  - 8-11: Final CAP slot
  - 12: Batt. Life extension
  - 13: Reserved
  - 14: PAN Coordinator
  - 15: Association permit

- **Payload fields:**
  - Variable
  - Variable
  - Variable
  - 2
**ZigBee Background (5)**

- **ZigBee Stack**
  - MAC layer: MAC Command Frame Format

<table>
<thead>
<tr>
<th>Frame Control</th>
<th>Data Sequence Number</th>
<th>Src Address</th>
<th>Command Type</th>
<th>Beacon Payload</th>
<th>Frame-check Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Octet</td>
<td>1</td>
<td>4/10</td>
<td>1</td>
<td>Variable</td>
<td>2</td>
</tr>
</tbody>
</table>

- Association Request
- Association Response
- Disassociation notification
- Data request
- PAN id conflict notification
- Orphan Notification
- Beacon Request
- Coordinator Realignment
- GTS Request
ZigBee Background (6)

- ZigBee Stack
  - MAC layer: MAC Data/ACK Frame Format

<table>
<thead>
<tr>
<th>MAC header</th>
<th>MAC payload</th>
<th>MAC footer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame Control</td>
<td>Data Sequence Number</td>
<td>Address</td>
</tr>
<tr>
<td>Data Sequence Number</td>
<td>Beacon Payload</td>
<td>Frame-check Sequence</td>
</tr>
</tbody>
</table>

- Frame Control: 2 Octet
- Data Sequence Number: 1
- Address: 4/20
- Beacon Payload: Variable
- Frame-check Sequence: 2

- Frame Control: 2 Octet
- Data Sequence Number: 1
- Frame-check Sequence: 2
Previous stack details have allowed for the transmission of data frames, ZigBee provides an Application Framework on top of this.
ZigBee Clusters and ZCL

- Application Objects and Endpoints
  - Lowest addressable object in the model
    - Addresses 1-240
  - Typically represents a physical device or at least something that models something physical
    - Ex: lightswitch, lamp, temperature probe, filter
  - Lends itself very much to an Object-oriented view of the stack
  - Special Endpoint 0 called ZDO used to communicate with other layers
  - Special Endpoint 255 for broadcasting to ALL endpoints
ZigBee Clusters and ZCL (2)

- **Endpoints consist of a set of clusters**
  - Each cluster is a set of attributes and a set of commands with respect to the attributes

- **Clusters are client/server oriented**
  - **Server**: also known as an input, responsible for storing the attribute values
  - **Client**: also known as an output, manipulates or requests the attribute values
  - Compatible Client and Server clusters on different Endpoints can be “bound”
ZigBee Clusters and ZCL (3)

- **ZigBee Cluster Library** - Standardized set of Cluster definitions organized together into Profiles. Profiles allow for grouping and reuse of Cluster IDs as well as for user-defined Profiles for non-standardized Clusters.

### Basic Cluster
- Versions (ZCL, HW, etc)
- Manufacturer
- Model ID
- Date code
- Power source
- Location description
- Device enable
- Alarm mask
- Reset to Factory

### Server Level Control Cluster
- Current level
- On level
- On/Off transition time
- Remaining Time
- Move to level
- Move
- Step
- Stop
- 'With On/Off' variants
ZigBee Security Model

- IEEE 802.15.4 Security
  - Data confidentiality
  - Data authenticity
  - Replay protection

- ZigBee Security
  - Trust Centers
  - Keys
New devices must be “commissioned”

- Devices can form a new network or join an established one
- Out of band means (buttons, web login, phone, etc.) are used to notify network that the device ID that is authorized to join
- Network enters *permit joining ON* state
- Device attempts to join network and keys are established
Security Keys

• Link Key
  - An end-to-end key that a device may share with another device.
    • Established through Trust Center
  - Devices MUST share a Link Key with the Trust Center (TCLK)
  - TCLK is established during Join procedure
  - Protects App layer and Stack commands:
    • Time, commissioning, price, demand response, load control, simple metering, message smart energy tunneling and pre-payment
  - Infrequently refreshed from Trust Center
Security Keys (2)

- **Network Key**
  - Shared by all devices on a network
  - Protects management and control communications
  - Can be used to protect app layer in cases where Link Key unavailable or unaccepted
  - Protected clusters:
    - Basic, Identify, Alarms, Power configuration and Key establishment
  - Should be periodically refreshed
Security Keys (3)

- Transport Key
  - Shared with the Trust Center and derived from the TCLK
  - Secures the Network Key refresh
    - This refresh is point-to-point with no broadcast mechanism
  - SWITCH_KEY command signals to start using newly established key
JOIN Procedure

- TCLK $L_K_i$ provided to Trust Center by out-of-band means
  - usually while informing the TC that the device wants to join
- Trust Center and device $i$ obtain Transport Key $T_K_i$ from $L_K_i$
- Trust Center now sends device $i$ the Network Key $N_K$ encrypted by $T_K_i$.
- Trust Center updates the Link Key $L_K_i$ of device $i$
Certificate Based Key Establishment

- Every device holds a certificate from a Certificate Authority (CA)
- Can derive the public key from the Certificate
  - Elliptic Curve MQV key agreement scheme
    - Basically provides a safe/secret way for 2 devices to agree on a shared value using a public channel
  - Key Derivative Function
    - Used in conjunction with non-secret parameters to derive one or more keys from a common secret value
LEAVE and REJOIN Procedure

- Secured REJOIN using currently known Network Key
  - May fail if Network Key has been refreshed in the meantime
- Unsecured REJOIN uses the LK that the Trust Center shares with the device to retransmit Network Key
- If Unsecured REJOIN fails device must go back and do a normal join
  - Usually requiring some out-of-band method again
- If a device leaves the network, the Trust Center removes their Link Key $LK_i$
Forward Security
Problems with Smart Energy Profile (SEP)

- Devices SHOULD be prevented from accessing communication on the network after it has left
- SEP specifies revocation of the Trust Center Link Key for a leaving device
- Specifies nothing about Network Key and Link Key
- Device can continue to listen to traffic encoded with Network Key and communicate with devices it shares a Link Key with
- Network and Application layer commands are at risk
  - Highly disruptive routing attacks from a compromised device
Forward Security (2)

Proposed Solution

- Network Key
  - Revoke and refresh the Network Key every time a device leaves the network
  - Utilize point-to-point rather than Broadcast based refresh for Network Key
    - Limits scalability
- Link Key
  - Propose defining a Key Revocation Cluster within the application layer
  - Trust Center would use a broadcast to notify all remaining devices that a device has left the network
  - Each device then invalidates $LK_{ij}$
Certificate Management

(Problem)

- Every device holds a certificate issued by Certificate Authority (CA) and the public key of the certificate called the Root Key.
- ZigBee Key Establishment Cluster allows many organizations to issue certificates (manufacturers, device distributors, end-users, etc.).
- This means that a Coordinator must authenticate certificates from various organizations.
  - Every device must store Root Key of every possible certificate source.
  - This is counter to the idea of small, fast, low-power RFDs.
Certificate Management
(Proposed Solution)

- Home Certification Authority ($CA_H$)
  - Root Key Database: Stores root keys of the certificate authorities
  - Verifies the certificates pre-installed in a device and issues a new certificate for that device
Certificate Management (2)
(Proposed Solution)

- Assume Device D, with public key $K_D$ and certificate $<D>_{CA}$ from certificate authority CA
  - Home Certificate Authority $CA_H$ obtains the device's certificate $<D>_{CA}$
  - $CA_H$ retrieves the CA's root key from the Root Key Database
    - Verifies $<D>_{CA}$ by that key
    - May possibly access larger database over internet
  - $CA_H$ issues new home-certificate for D, $<D>_{CA_H}$
    - $<D>_{CA_H}$ is installed in device D
Critiques

- Great outline of the security of IEEE 802.15.4/ZigBee/Smart Energy Profile
- Proposals seem logical given the stated issues
  - No formal verification of these proposals
  - Simulation/Implementation to show effects (or lack of) on a realistic network
Wireless Sensor Networks generally present in unrestricted environments – thus prone to attacks.

Attacker: One who attacks the network with the aim of damaging nodes or gaining selfish benefits from the network itself.

Can occur at any level of the protocol stack.

In general, attacks are unpredictable.
An Investigation on IEEE 802.15.4 MAC Layer Attacks

- Literature Survey of MAC Layer attacks
- 2 attack and solution contributions of their own
Radio Jamming

● Description
  - Simplest of the attacks
  - Act of emitting competing radio signals directed at a specific channel
  - Classified as a misbehavior attack of the 802.15.4 PHY layer

● Evaluation
  - Constant jamming is the easiest
  - Jammer continuously emits signal to corrupt communication
  - Simple to detect using statistical analysis of signal strength and SNR

Radio Jamming (2)

- **Deceptive Jamming**
  - Emits “regular” packets as an interference pattern. More difficult to detect than simple jamming but security schemes can be used to detect bogus node. Assumed constant

- **Random Jamming**
  - Emits “regular” packets or noise at random intervals. Less likely to be detected due to non-constant nature as well as saving power
Radio Jamming (3)

- Reactive Jamming:
  - Based on sensing network activity
  - Stays quiet while network is idle and begins transmitting only upon sensing activity
  - More energy efficient and harder to detect
  - Detection necessitates more advanced methods such as Jammed Area Mapping
Link Layer Jamming

- **Description**
  - Utilizes knowledge of the Link Layer to be as effective as radio jamming on much less power

- **Evaluation**
  - Typical reaction times against this attack are similar to reactive radio attacks

---

Backoff Manipulation

- **Description**
  - IEEE 802.15.4 CSMA uses a back-off period if a node finds the medium to be busy when it wishes to transmit.
  - Backoff period is randomly chosen in a contention window (which increases).
  - A malicious node may consistently choose a shorter back-off period to give unequal medium access.

Backoff Manipulation (2)

• Evaluation
  - Very difficult to discern between legitimate node and misbehaving node
  - Indiscernible with low congestion
  - Sequential Probability Ratio Test (SPRT)

\[ S_k = \sum_{j=1}^{k} \ln \frac{f_1(x_j)}{f_0(x_j)}, \]

with \( f_i(\cdot) \) the probability density function of hypothesis \( \mathbf{H}_i, i = 0, 1 \). The decision is taken based on the criteria:

\[ S_k \geq a \Rightarrow \text{accept } \mathbf{H}_1, \]

\[ S_k < b \Rightarrow \text{accept } \mathbf{H}_0, \]

\[ b \leq S_k < a \Rightarrow \text{take another observation.} \]
Same-nonce Attack

- Description
  - 802.15.4 Secured Mode uses Access Control Lists (ACL) made of destination address, secured mode options, keys and nonce info.
  - In the case where a sender has 2 ACL entries with the same keys and nonce
  - When encoded message $C_1$ and $C_2$ use the same nonce data and keys
    - $C_1 = [D_1 \text{ XOR } E_{\text{key}}(\text{nonce})]$
    - $C_2 = [D_2 \text{ XOR } E_{\text{key}}(\text{nonce})]$
    - $[C_1 \text{ XOR } C_2] = [D_1 \text{ XOR } D_2]$

Nonce - arbitrary number used only once to sign a cryptographic communication.

Another great paper

Same-nonce Attack (2)

- **Example**
  - AES-CCM-64 Security Suite, recipients $r_1$ and $r_2$ use the same key $k$.
  - Frame and key counters initied to 0x0 (common)
  - Sender transmits $D_1=0xAA00$ to $r_1$ as $C_1$
  - Sender transmits $D_2=0x00BB$ to $r_2$ as $C_2$
  - Using previous formula, an adversary can obtain
    - $[D_1 \text{ XOR } D_2] = [C_1 \text{ XOR } C_2] = 0xAABB$

- **Evaluation**
  - Likely occurrence if nonce data is not managed
  - "The general principle to prevent nonce reuse is that the nonce state should never be separated from the key"
Replay Protection Attack

- **Description**
  - Replay protection in 802.15.4 checks a counter of a message with the previously obtained counter. Rejects message if counter is less than or equal to previous one (can't go backwards).
  - Attack repeatedly sends high value counters.
  - Legitimate frame has lower counter value and is rejected.

Replay Protection Attack (2)

- Evaluation
  - Proposed solution of utilizing a timestamp as the frame counter field
  - Timestamp updated using Beacon frame from coordinator and updates all ACL entries with new time stamp
  - Attacker cannot send future times and so Replay protection attack fails
  - Requires larger field to support timestamp
**ACK Attack**

- Frame sequence numbers are unencrypted
- Attacker uses MAC or PHY layer interference to prevent transmission after Frame sequence number is seen
- Then sends fake ACK frame to sender, preventing retry
- Proposed solution: Use Message Integrity Bits (MIC) which encode even ACK frames

---

PANid Conflict Attack

(Original work)

- PANid conflict occurs when two Coordinators contain the same PANid.
- Can be detected via Beacons or a member node can inform its coordinator.
- Induced Conflict Resolution Procedure:
  - Initiate scan, choose new PANid, broadcast to PANid member nodes.
  - Nodes resync'ed at next Beacon Frame.
- Attacker can perform DoS by generating fake PANid conflict notifications (PCN) causing entire PAN to reset and resync.
PANid Conflict Attack (2)  
(original work)

- Simple formula for detection time:
  - \( p_1 \): Max number of PCN from an attacker
  - \( p_2 \): Max number of PCN in fixed duration
  - \( p_3 \): Fixed duration length
  - \( \varepsilon \): Misbehavior detection algorithm running time

\[
f(p_1) = \text{successful}\,(p_1 + 1)^{th}\,\text{attack time} + \varepsilon_2 \quad (1)
\]
\[
|f(p_2, p_3) = \text{successful}\,(p_2 + 1)^{th}\,\text{attack time in the last } p_3 \text{ duration} + \varepsilon_2 | \quad (2)
\]
\[
detection\,\text{time} = \min(f(p_1), f(p_2, p_3)) \quad (3)
\]
Methodology:

- Simulated on ns2.31 IEEE 802.15.4 simulator
- Star topology with 5 and 10 nodes
- Attackers send fake PCN at random times
- Set initial first attack at 15 second to allow initial network establishment
- Observed realignment time upon conflict at approximately 3 seconds
  - Attackers do not attack within this range of (3-\(e\), 3+\(e\)) where \(e\) is the realignment message. Doing some could leave them orphaned
**PANid Conflict Attack (4) (original work)**

### Attack times vs number of attackers

<table>
<thead>
<tr>
<th>Attacker Type</th>
<th>Attack Times (s)</th>
<th>Misbehavior Type</th>
<th>Attack Solution Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single</td>
<td>15 19 27 31 35 100</td>
<td>2</td>
<td>27 + ε₂</td>
</tr>
<tr>
<td>Attacker</td>
<td>16 23 39 44 63 71</td>
<td>1</td>
<td>63 + ε₂</td>
</tr>
<tr>
<td>Double</td>
<td>15 19 27 31 35 100</td>
<td>2</td>
<td>27 + ε₂</td>
</tr>
<tr>
<td>Attacker</td>
<td>16 23 39 44 63 71</td>
<td>1</td>
<td>71 + ε₂</td>
</tr>
<tr>
<td>Triple</td>
<td>15 19 27 31 35 100</td>
<td>2</td>
<td>27 + ε₂</td>
</tr>
<tr>
<td>Attacker</td>
<td>16 23 39 44 63 71</td>
<td>1</td>
<td>71 + ε₂</td>
</tr>
<tr>
<td></td>
<td>17 21 33 40 50 56</td>
<td></td>
<td>67 + ε₂</td>
</tr>
</tbody>
</table>

- **Type 1**: Max per node
- **Type 2**: Max per time period
- **Green**: Successful attack
- **Red**: Detected attack
- **Uncolored**: Redundant attack (ignored at MAC layer)

### Attack times vs coordinator detection parameters

<table>
<thead>
<tr>
<th>Attack Times (s)</th>
<th>Misbehavior Type</th>
<th>Attack Solution Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>15 19 27 31 35 100</td>
<td>3</td>
<td>27 + ε₂</td>
</tr>
<tr>
<td>16 23 39 44 63 71</td>
<td>1</td>
<td>44 + ε₂</td>
</tr>
<tr>
<td>15 19 27 31 35 100</td>
<td>2</td>
<td>27 + ε₂</td>
</tr>
<tr>
<td>16 23 39 44 63 71</td>
<td>1</td>
<td>63 + ε₂</td>
</tr>
</tbody>
</table>

- **Double Attacker**
- **P₁=2**
- **P₂=2**
- **P₃=20**
GTS Attack
(original work)

- A more targeted version of the MAC layer interference DoS attack
- Use the Beacon frame which contains the GTS descriptions to target a specific adversary node
- Interfere specifically during the GTS slot of the adversary
- Jamming can be noise-based or legitimate message-based
- Fine-grained nature makes attack difficult to detect
- If detected, difficult to pinpoint source node

GTS Attack (2)  
(original work)

- Simulated using ns2.31 IEEE 802.15.4 simulator modified to support full MAC layer standard
- Simulated targeted and random attackers
  - Targeted attackers chose same slot
  - Random attackers randomly chose slot
- Attackers jammed using legitimate messages
GTS Attack (3)
(original work)

- Results
  - (with more useful rotation)
Critiques

- Rehashes the same kinds of attacks that almost every paper does
- Doesn't do as good a job reviewing those attacks as other papers do (didn't find this out until later)
- Paper proposes 2 original attacks
  - PANid work is good, but analysis and simulation data is weak
  - GTS attack was not even properly simulated until a later paper which I had to find separately