6LoWPAN Security: Adding Compromise Resilience to the 802.15.4 Security Sublayer

Eirik Klevstad

January 19, 2016

1 Abstract

Abstract:
- "IPv6 over Low-power Wireless Personal Area Networks"
- Used for integrating 802.15.4 based sensor networks with IPv6 networks.
- Its security depends on the security of the sublayer.
- Uses pairwise keys to mitigate node compromises. Establishment of these pairwise keys are currently unspecified.
- Broadcast keys are shared between multiple nodes.
- Two proposals:
1) Pairwise key establishment scheme
2) Easy-to-implement and compromise-resilient protocol for authenticating broadcast frames.
- Implemented in Contiki, tested on TelosB nodes.

2 Introduction

.
- 6LoWPANs = much like Wireless Sensor Networks (WSN)
- Enables (6LoWPAN) nodes to communicate with each other or remote hosts using IPv6
- Used in smart cities, industrial monitoring, precision agriculture
- Many 6LoWPAN protocols depend on the 802.15.4 security layer to protect them from packet injection or replay attacks
- 802.15.4 sublayer does so by using Message Integrity Codes (MIC) and a frame counter (sequence number?) to each frame.
- However: Key establishment is NOT specified.
- Not smart to preload a shared key on the nodes, as the nodes may be placed in hostile environments where they may be tampered with.
- Creating a tamper-proof node is often expensive and difficult, hence not a solution that we "want".

- With a network key, an attacker is able to inject frames whenever he wishes. Possible to add unauthorized nodes to the network.

- Another approach, where each node is preloaded with the communication key for each of the other nodes. This introduces these problems:

1) Memory problems when a node is to store a 128-bit key for each of say... 30000 nodes in the network. (TelosB nodes has a memory capacity of 1 MB). Also needs to store a key for the yet-not-deployed nodes for supporting adding new nodes to the network after deployment.

2) Needs to store the frame count / sequence number of each source address to avoid replay attacks. Meaning that at some point, counters for disappeared nodes needs to be moved from RAM to Storage. This is energy consuming.

3) Difficult to secure broadcast frames, which have to be authenticated with keys that are shared among all receivers of the frame. Meaning that a compromised node also reveals the broadcast keys of other nodes.

- Public key cryptography is inappropriate for solving these problems, mostly because of it being both time and energy consuming.

- Possible for an attacker to drain the nodes battery by issuing energy consuming PK requests to the node.

- Should have a certificate-based authorization mechanism to verify nodes that joins the network. May be subject to DoS-attacks.

- Two contributions:

1) Adaptable Pairwise Key Establishment Scheme (APKES): Framework for establishing pairwise keys without using public key cryptography.

2) Easy Broadcast Encryption and Authentication Protocol (EBEAP) for authenticating and encryption 802.15.4-frames. Does not use publick key crypto, delays, hash chains or time synchronization.

3 Background

3.1 6LoWPAN and vulnerabilities

3.1.1 6LoWPAN protocol stack

.

Layer 1: 802.15.4 PHYSICAL layer

Layer 2: 802.15.4 MAC layer, 802.15.4 security sublayer, APKES, EBEAP

Layer 2.5: 6LoWPAN adaption layer: Fragments and compresses to IPv6. Compression reduces energy consumption for transmitting/receiving IPv6 packets.

Layer 3: IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) routes IPv6 packets.

Layer 4/7: UDP-based protocol such as Constrained Application Protocol (CoAP)

If the 802.15.4 security sublayer is not secure, an attacker may be able to inject or replay frames. On Layer 2.5, an attacker can launch fragmentation attacks, on Layer 3, attackers may launch path-based DoS attacks.

3.2 The 802.15.4 Security Sublayer

Essential security related fields in 802.15.4 frame is MIC and Frame Counter.
Enables receiver to verify both authenticity and freshness.
MIC generated by Cipher Block Chaining MIC (CCM*). AES 128-bit
CCM* can also encrypt the payload. Useful if it is only one hop from sender to receiver. If the path consists of multiple hops, a secure path using IPsec should be set up.

4 Three Pluggable Schemes and their trade-offs

4.1 The IOWEU Criteria

Inoculation
Opaqueness
Welcomingness
Efficiency
Universality

4.1.1 Inoculation

An attacker can not aid unauthorized nodes to join the sensor network by compromising some other nodes. An attacker reuses the identity of a compromised node an unauthorized node. Can blacklist compromised nodes to prevent this.

4.1.2 Opaqueness

Pairwise key establishment scheme is opaque if in event of a node compromise, only links to and from that particular node are compromised. Useful for detecting malicious nodes.

4.1.3 Welcomingness

Supports addition of new nodes at runtime

4.1.4 Efficiency

Memory and energy efficiency

4.1.5 Universality

Does not rely on one particular network protocol

4.2 Localized Encryption and Authentication Protocol (LEAP)

Main idea: Preload each node with a Master Key K_m, erase K_m after key establishment.

Used as follows:

1) Suppose node u is begin deployed.

2) It first derives its individual key K_u from K_m: $K_u = F(K_m, ID_u)$

3) F is a pseudorandom function family. ID may be u's address.

4) u broadcasts a HELLO message:

$u \leftarrow * : HELLO(ID_u)$

$v \leftarrow u : ACK(ID_v, ID_u)K_v$

The neighbor responds with an ACK that is authenticated with a MIC generated with K_v.

u can generate K_v from K_m and verify the received ACK.

Pairwise key = $K_{u,v} = F(K_v, ID_u)$

Erase K_m in case of being compromised.

4.2.1 Inoculation and opaqueness

LEAP: No way for v to authenticate u. APKES fixes this by adding bidirectional authentication. LEAP + APKES = Inoculated and opaque if K_m is secret.

4.2.2 Welcomingness

Assumes that no jamming attacks occur during neighbor discovering. Eavesdrop on pairwise key establishment messages while deploying nodes.

4.2.3 Efficiency

LEAP is the most energy and memory efficient scheme that we are aware of (2013).

4.2.4 Universality

LEAP does not support mobility.

4.3 Blom's Scheme

Symmetric threshold key exchange protocol. Uses matrixes over finite fields. Uses a trusted party which gives each party a public identity and a secret key so that they can independently create a shared key for communicating. If an attacker compromises x nodes (keys), he can break the scheme and reconstruct all keys used for communication between nodes. Used by HDCP copy protection used in HD DVD and HD TV.

λ: Number of node compromises that are tolerable in Blom's Scheme. n: Number of nodes in network including not-yet deployed nodes. l: Desired key length (bits)

4.3.1 Inoculation and opaqueness

As long as no more than λ nodes are compromised, Blom's Scheme is perfectly inoculated and opaque.

4.3.2 Welcomingness

Welcomingness if n takes future deployments into account.

4.3.3 Efficiency

Computational and memory effort increases linearly with λ.

4.3.4 Universality

Blom's Scheme is universal.

4.4 Random Pairwise Keys Scheme (RPKS)

Only some of the node pairs have pairwise keys.

Example: Network of 10000 nodes, each node only stores 75 pairwise keys to achieve a probability of 0.5 of having a pairwise key with another node.

Similar to the birthday paradox.

If two nodes does not have a pairwise keys, establish a path through common neighbors. Unfortunately, the intermediate node(s) learn the established path key.

4.4.1 Inoculation

The random pairwise keys scheme is inoculated.

4.4.2 Opaqueness

Directly established pairwise keys are opaque, paths through intermediate nodes are revealed.

4.4.3 Welcomingness

Is welcoming

4.4.4 Efficiency

Low memory consumption. Retrieving keys from storage not too energy consuming as keys are short. However, path key establishment is the most energy consuming part.

4.4.5 Universality

Requires a minimum network density.

4.5 The "IOWEU" Dilemma

Limiting factors of LEAP: Universality, Inoculation and Opaqueness.
LEAP does not support 6LoWPAN networks with mobile nodes.
Master key MUST BE secure to be inoculated and opaque.
Limiting factors of Blum's Scheme: Efficiency.
Limiting factors of RPKS: Opaqueness and universality
Nuff said: All schemes have certain trade-offs. RPKS is the best choice given that memory consumption is acceptable.

5 APKES: Adaptable Pairwise Key Establishment Scheme

Principle: 6LoWPAN network developer picks an appropriate pairwise key establishment scheme and plugs it into the APKES implementation of each node.

5.1 Protocol Overview

5.1.1 Optional Preloading of Short Addresses

Needs IDs for generating shared secrets.
APKES reuses 802.15.4 addresses as IDs.
802.15.4 address consist of two parts:
1) 2-byte PAN ID (Subnetwork or PAN of a 6LoWPAN)
2) 2-byte short (unique within PAN) or an 8-byte extended address (globally unique)
Each 802.15.4 transceiver has an unique EUI-64 assigned to it.
To obtain shared secret with a node: APKES sends PAN-ID of node, extended address of node (and short address if present).
Automatic configuration of short addresses with 6LoWPAN is incompatible with APKES because keys have to be established before IPv6 packets can be sent.

5.1.2 Preloading of Cryptographic Material

Crypto material must be preloaded into the node before it is deployed. A seed for generating random numbers. Preloaded with crypto for the plugged-in scheme, and the master key K_m if using LEAP.

5.1.3 Pairwise key establishment

APKES employes a three-way message exchange for establishing pairwise keys between two neighboring nodes that are to ble established.
802.15.4 commands: HELLO, HELLOACK, ACK

Special frames, not passed through upper-layer protocol, processed on Layer 2.

Message sequence:

1. u generates random number R_u and broadcasts a HELLO command.

2. Upon receiving HELLO with R_u, v generates a random number R_v and stores $R_u||R_v$. Waits a time T_w to prevent u from being overwhelmed with HELLOACKs (There might be a lot of nodes that received the HELLO)

3. After waiting time is over, v sends HELLOACK which contains both R_u and R_v and is authenticated with $K_{v,u}$ as CCM* key which is a shared secret between u and v.

4. The actual pairwise key $K'_{v,u}$ is derived from $K_{v,u}$ as $AES(K_{v,u}, R_u||R_v)$

5. Upon receiving HELLOACK, u receives the attached CCM*-MIC by obtaining the shared secret $K_{u,v}$. Also checks if R_u is unchanged from its initial value.

6. If all checks come through, u derives $K'_{v,u}$ too and sends an ACK to v.

5.2 Protocol Specification

To prevent DoS attacks by floating or injecting HELLO / HELLOACK, each node stores a table with its neighbors.

5.2.1 Tentative Neighbors

Created upon receiving a HELLO command if sender is not already stored as neighbor and the number of tentative neighbors is smaller than M_t. A tentative neighbor expires after $T_e = T_c + T_w + T_a$. T_c = time when created, T_w waiting time before sending HELLOACK, T_a waiting time before receiving ACK.

5.2.2 Permanent Neighbors

Potentially created upon receiving valid HELLOACK, ACK. Three cases:

1) If not already neighbor: Store as neighbor.

2) If already neighbor: Send ACK if frame counter is smaller than the one in the ACK received. Update all information.

3) If tentative neighbor: Turn into permanent neighbor and send ACK. Avoids deadlock.

APKES discards frames from non-permanent neighbors. Only those that are from permanent neighbors are authenticated and decrypted before passed to the 6LoWPAN adaption layer.

5.3 Security Analysis

APKES mitigates HELLO flooding attacks by limiting the number of tentative neighbors.

Maximum number of tentative neighbors should be low. Waiting time T_a should be long.

APKES prevents the replay of HELLOACKs and ACKs. HELLOACKs by using frame counters. The replay of ACKs is senseless as a tentative neighbor only becomes permanent once.

Different if permanent neighbor is deleted. Attacker can try to become permanent by spoofing a HELLO command and a subsequent ACK to the HELLOACK.

APKES uses AES for deriving the actual pairwise keys. Two advantages:

1) Disappeared permanent neighbors can be deleted along with the frame counters because of R_u and R_v.

2)Improves the opaqueness of some pluggable schemes. An attacker needs both master key M_k, R_u and R_v in order to crack the secret key used for communication between the two parties.

6 EBEAP: Easy Broadcast Encryption and Authentication Protocol

EBEAP enables nodes to authenticate and encrypt broadcast frames. (APKES only unicast frames)

Broadcast frames: used in RPL and 6LoWPAN-ND to discover neighbor nodes and inform about network changes.

Solely uses CCM* and reuses the established pairwise keys from APKES.

6.1 Protocol Overview

$M(k, f)$: CCM*-MIC over a frame f with key k. Suppose u wants to broadcast a frame to its permanent neighbors.

First adds Security Control and a Frame Counter field to the frame f.

First broadcast: an ANNOUNCE command. Contains one CCM*-MIC over the frame f for each permanent neighbor.

When a permanent neighbor receives an ANNOUNCE command, extracts and stores CCM*-MIC. To be able to do this: Index $i = I_{v,u}$ must be in sending nodes neighbor list.

Such indices are piggybacked to APKES commands.

Second broadcast: Actual broadcast message. when receiving, receiving party v_i generates $M(K'_{v_i,u}, f')$. Accept message if f' is fresh and the resulting CCM*-MIC is stored at node.

EBEAP also supports encryption: Each node u generates a broadcast key $K_u, *$ and passes it securely to neighbors.

6.2 Protocol Specification

EBEAP stores additional data on each node u:

- its index $I_{u,v} \in \{0, 1, ...\}$ is stored in each of its permanent neighbor's neighbour lists.
- Received CCM*-MICs

- Broadcast keys are stored if encryption is turned on. (Length = $L_{bk} < 16$ bytes)

Encryption on: HELLOACK/ACK piggybacks indices and broadcasts keys.

If using broadcast encryption: HELLOACK/ACK encrypted with CCM*.

Key identifier field for command frame identifier, short address

Over time: Gaps in neighbor list. To keep ANNOUNCE commands as short as possible: Reordering of neighbor list by UPDATE command. Same as HELLOACK but without random numbers.

Secured with $K'_{u,v}$

6.3 Security Analysis

Assume a random broadcast frame gets accepted by one of the neighbors and discarded by the others.

Accept frame probability = $\frac{m}{2^l}$

Compensation: Possible to configure length of CCM*-MICs within ANNOUNCE commands and unicast frames independently.

ANNOUNCE messages are sent unauthenticated. Possible to spoof them. Attacker can overwrite the announced CCM*-MICs. Success = the victims neighbors discards the frame received from victim.

Attacker needs to send multiple ANNOUNCE commands because of overwriting of oldest CCM*-MICs.

Choose m very high, or authenticate ANNOUNCE commands with broadcast keys. Attack may be difficult anyway.

If multiple ANNOUNCE commands have to be sent, this reduces the scalability of EBEAP.

Compromise node: Broadcast keys of the node's permanent neighbors will leak.

Attacker can decrypt frames, but the attacker can not get broadcast frames accepted that pretend to originate from one of the neighbors because of the opaqueness of the pairwise keys.

7 Implementation

APKES and EBEAP implemented in WSN operating system Contiki.

Added layer between MAC and 6LoWPAN adaption layer for implementing link layer security.

coresec_driver implements 802.15.4 security sublayer + APKES and EBEAP.

Three simplifications:

1) Did not implement UPDATE commands

2) Restrict the number of allowed neighbors: One ANNOUNCE command is always sufficient.

3) Only LEAP is available as a pluggable scheme.

8 Evaluation

Telos B motes: 48 KB of program memory, 10 KB RAM.

Example allows 15 neighbors with 12-byte pairwise and 8-byte broadcast keys.

Energy consumption not measured.

9 Related Work

Alternative to 802.15.4 security sublayer: Hummen: Mitigation mechanisms against known fragmentation attacks. Introduce complexity. Can protect against PDoS attacks.

Swapping of frame counters avoided by use of Bloom filters. However, false positives makes legitimate frames sometimes discarded.

10 Conclusion and Future Work

Severe threat to 6LoWPANs: Node compromises

Inoculated and opaque pairwise key establishment schemes contains effects of node compromise. APKES and EBEAP solves these issues.

Attackers can inject frames if the compromise a node and obtain its pairwise keys.

If the plugged-in scheme is inoculated and opaque: Sender of malicious authentic frame can be considered compromised.