

Security Considerations for 802.15.4 Networks

Naveen Sastry David Wagner
UC Berkeley

October 1, 2004

IEEE 802.15.4 Overview

- Specification for low speed wireless communication
 - PHY and MAC Layers
 - Low power, cheap
 - Includes link-layer cryptography
-
- Intended: personal area networks
 - Game controllers
 - Industrial process control
 - Uses in sensor networks



Security Risks in 802.15.4

802.15.4

- Eavesdropping
 - Confidentiality
- Packet Injection
 - Access control
 - Integrity
- Replay
- Jamming
- Application vulnerabilities



This Talk

- 802.15.4 overview
- Problems:
 - DoS attacks
 - IV reuse
 - Limited keying models
- Overall sentiments

802.15.4 Security Overview

3 Security Options:

1. Encryption: Uses AES-CTR mode (stream cipher).
2. Integrity: AES-CBC-MAC. Includes a (4/8/16) byte integrity code.
3. Replay protection: implemented with a sequence number.
Requires encryption to be enabled.

When using encryption + integrity, AES-CCM mode is used.

Encryption

	On	Off
On	✓ (Replay optional)	✓
Off	✓ (Replay optional)	✓

Security Mechanism

Sender
(Addr = 0xA)

Dest	Key	IV
...
0xB	k ₁	0x2
...
*	k ₂	0x4

Confidentiality &
replay protection

Integrity
protection

Dest Src IV/CTR



Receiver
(Addr = 0xB)

Src	Key	CTR
...
...
0xA	k ₁	0x2
*	k ₂	—

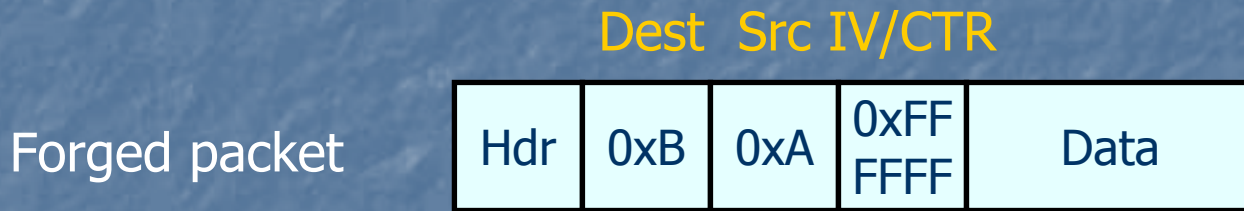
Dest is
sender's
index

Src is receiver's
index

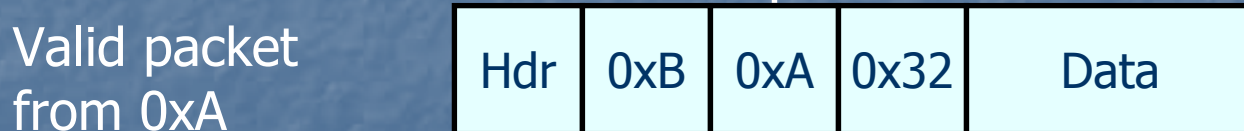
- Index into state table to obtain keying material
- IV: Used for confidentiality
- CTR: Used as a high-water mark for replay
- IV and CTR: two names for the same quantity

Attack I: Insufficient Integrity

Step 1: Attacker sends forged packet
IV=0xFF FFFF FFFF



Step 2: Receiver updates
its replay counter



Receiver
(Addr = 0xB)

Src	Key	CTR
...
...
0xA	k_1	0xFF FFFF
*	k_2	—

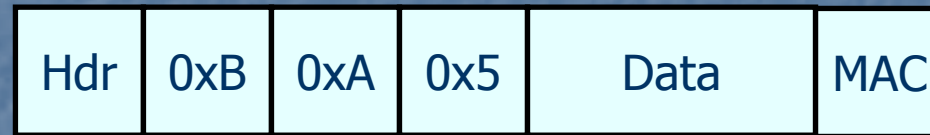
Impact: one packet denial of service.

Attack II: Reusing Keys

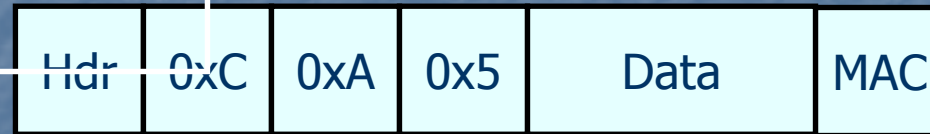
Sender
(Addr = 0xA)

Dest	Key	IV
...
0xB	k_1	0x5
0xC	k_1	0x5
*	k_2	0x9

Dest Src IV/CTR



\oplus



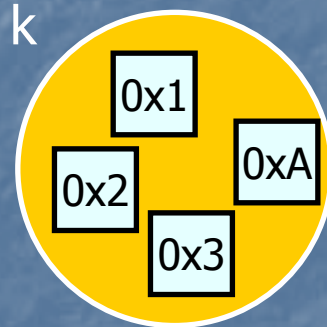
xor of
plaintexts

- If reuse keys: confidentiality broken
- See shortly how this can arise

Network Shared Keys

Sender
(Addr = 0xA)

Dest	Key	IV
...
0x1	k	0x2
0x2	k	0x5
0x3	k	0x1
*	—	—

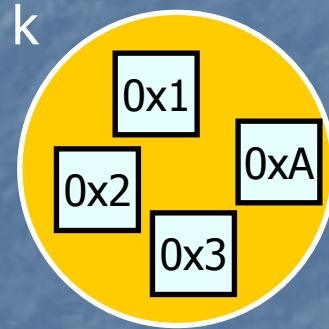


- As we saw: invites IV reuse.
- So can't list members in table and stay secure

Network Shared Keys

Sender
(Addr = 0xA)

Dest	Key	IV
...
...
...
*	k	0x3



Receiver
(Addr = 0xB)

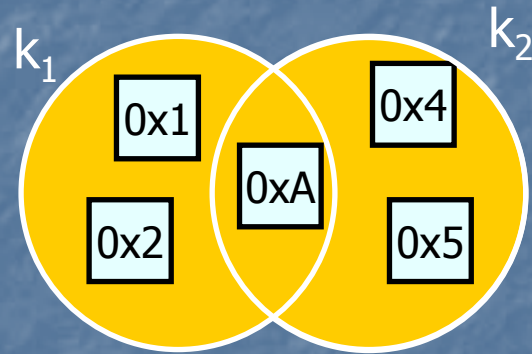
Src	Key	CTR
...
...
...
*	k	—

Solution: Default key entry

Keying Model Problems: Group Keying

Sender
(Addr = 0xA)

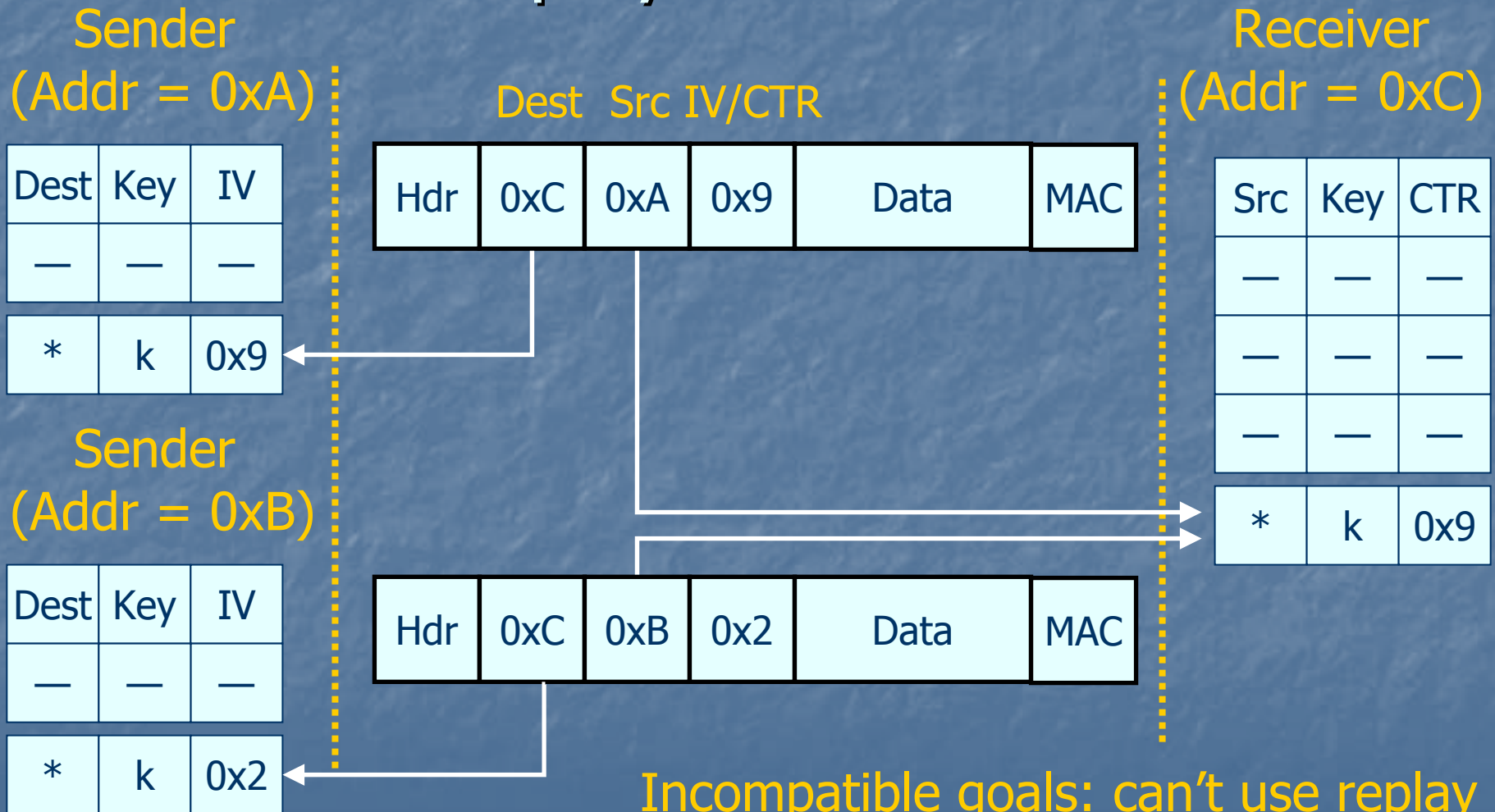
Dest	Key	IV
...
0x4	k_2	0x3
0x5	k_2	0x9
*	k_1	0x4



Again: duplicating keys leads to IV reuse.

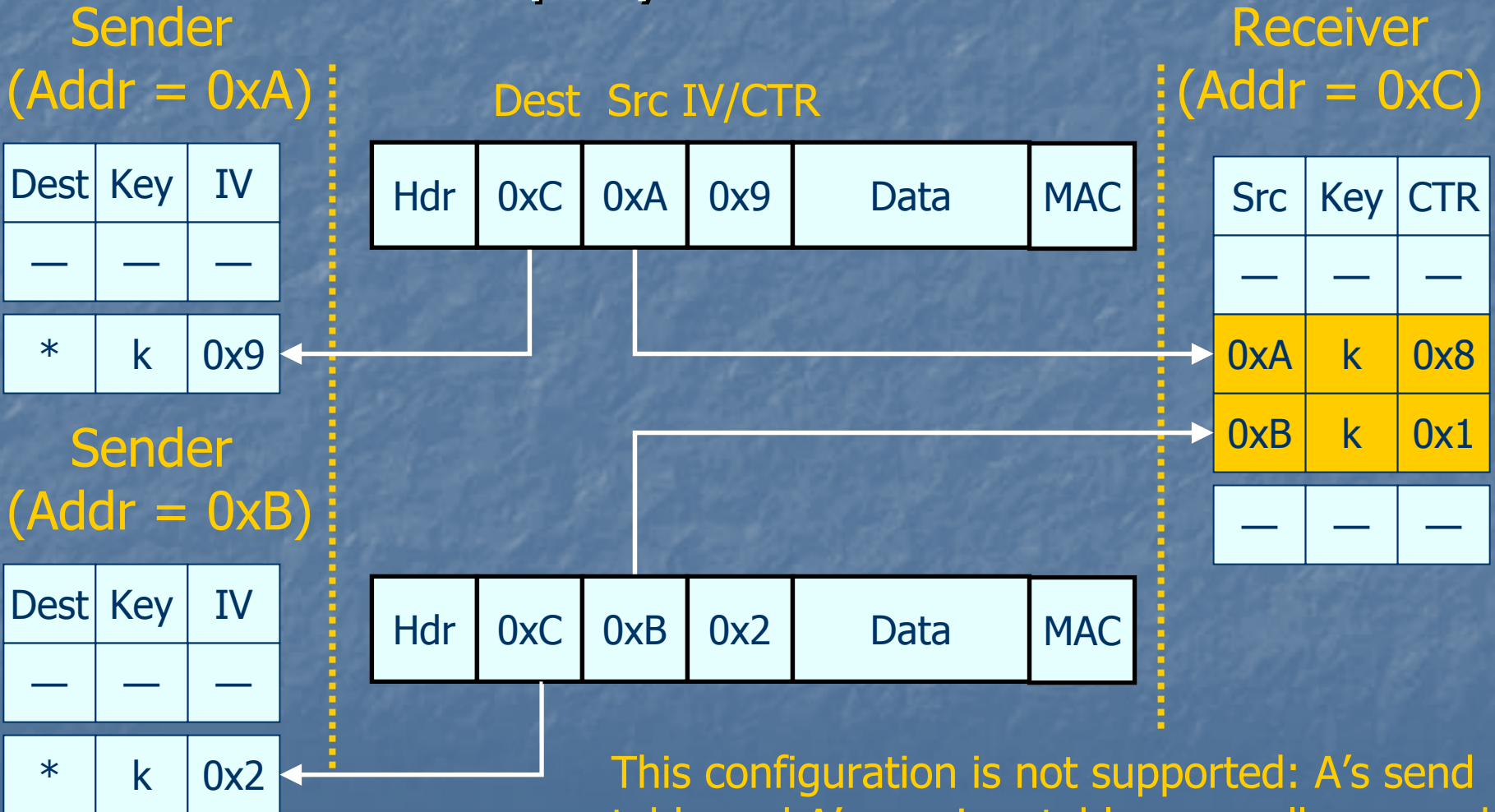
Bottom line: group keying not feasible

Network Shared Key & Replay Protection



Incompatible goals: can't use replay protection and network shared keys

Network Shared Key & Replay Protection



This configuration is not supported: A's send table and A's receiver table are really merged.

Real Picture of State Tables

Sender
(Addr = 0xC)

Dest	Key	IV
—	—	—
—	—	—
—	—	—
*	k	0x2

+

Receiver
(Addr = 0xC)

Src	Key	CTR
0xA	k	0x8
0xB	k	0x1
—	—	—
*	k	—

=

Node
(Addr = 0xC)

Src	Key	IV	CTR
0xA	k	0x2	0x8
0xB	k	0x2	0x1
—	—	—	—
*	k	0x2	—

↑
Outbound
only

↑
Inbound
only

But now the send-table has a duplicate entry for the same key: an insecure configuration.

Summary of Keying Problems

- Group keying not supported
- Network keying doesn't support replay protection
- No guarantee on number of table entries on chip
- Would like to view tables as separate for send and receive purposes
- Doesn't fully support common keying models

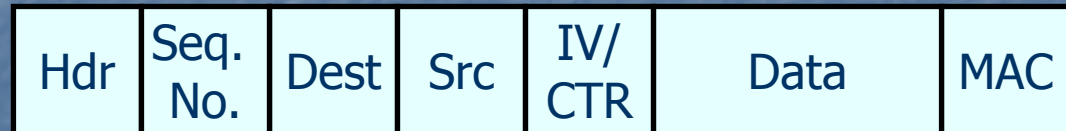
Conclusions

- IEEE 802.15.4 specification gets many things right:
 - Proven security primitives
 - Proper use of many primitives
- Secure keying models:
 - Pairwise
 - Network shared (no replay protection)
- A few pitfalls lead to silent security vulnerabilities
- To avoid traps, see paper. We have suggestions for:
 - Specification writers
 - Chip designers
 - Application writers

Integrity II: Unauthenticated ACKS

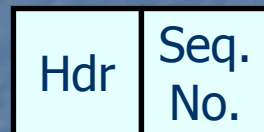
- Have been talking about data packets.
- Sender requests ACK by including 1 byte value in header

Data Packet



↓ Recipient copies

Ack Packet



- But no integrity protection for ACKS.
- Can't trust any ACK in hostile environment: attacker can easily forge ACK packet

Merged View of State Tables

Sender
(Addr = 0xA)

Dest	Key	IV
...
0xB	k_1	0x2
...
*	k_2	0x4

Receiver
(Addr = 0xB)

Src	Key	CTR
...
...
0xA	k_1	0x2
*	k_2	—

Only used when
sending packets

Only used when
receiving packets

Addr	Key	IV	CTR
...
0xB	k_1	0x2	0x2
...
*	k_2	0x4	—

< 256
entries

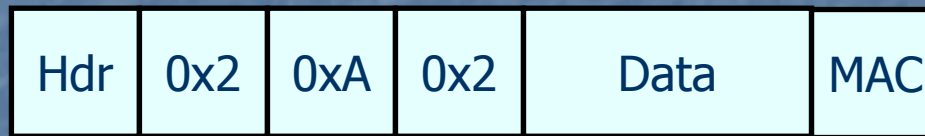
Keying Model Problems: Group Keying

Sender
(Addr = 0xA)

Dest	Key	IV
...
0x2	k ₁	0x2
0x5	k ₂	0x5
*	k ₃	0x4

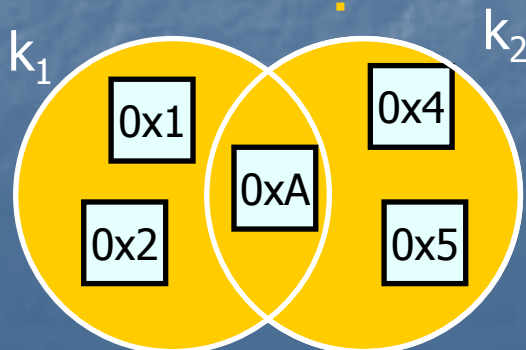
What if we switched destination field in the table before each send?

Dest Src IV/CTR



Receiver
(Addr = 0x2)

Src	Key	CTR
...
...
0x1	k ₁	0x2
*	k ₂	—



But on the receiver,
need to expect a packet
before it arrives!

Difficult to coordinate.