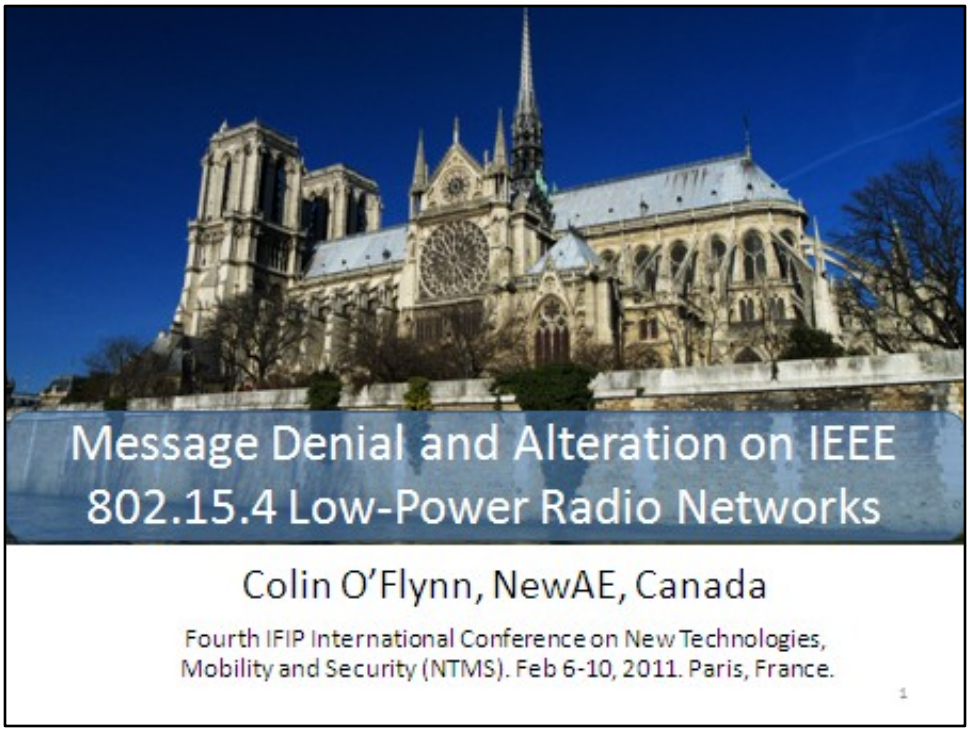


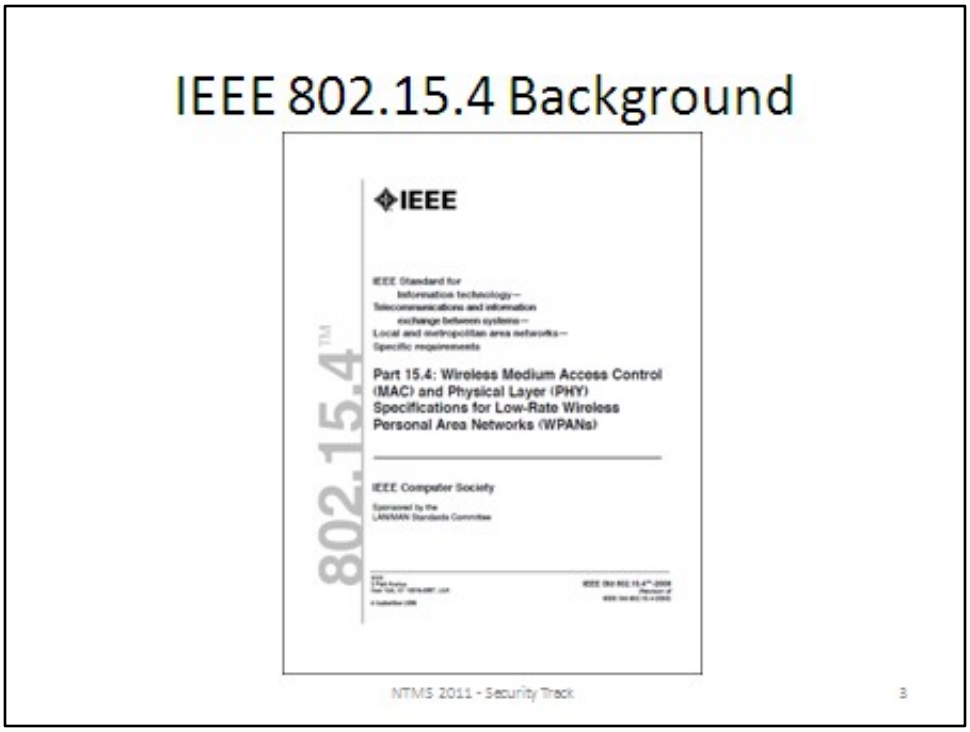
Welcome to my presentation: Message Denial and Alteration on IEEE 802.15.4 Low-Power Radio Networks.

This presentation discusses the susceptibility of IEEE 802.15.4 radio networks to several different attacks. The attacks are based around denial of service, but also branch out to show how to use the attacks as part as a Man-In-The-Middle attack.

The attacks themselves are not unique, and instead designed to demonstrate some basic building blocks. This paper (and presentation) is aimed at security researchers who need to know what attacks are physically capable on these networks. It moves many attacks out of the "interesting academic idea" space, and into the real world.

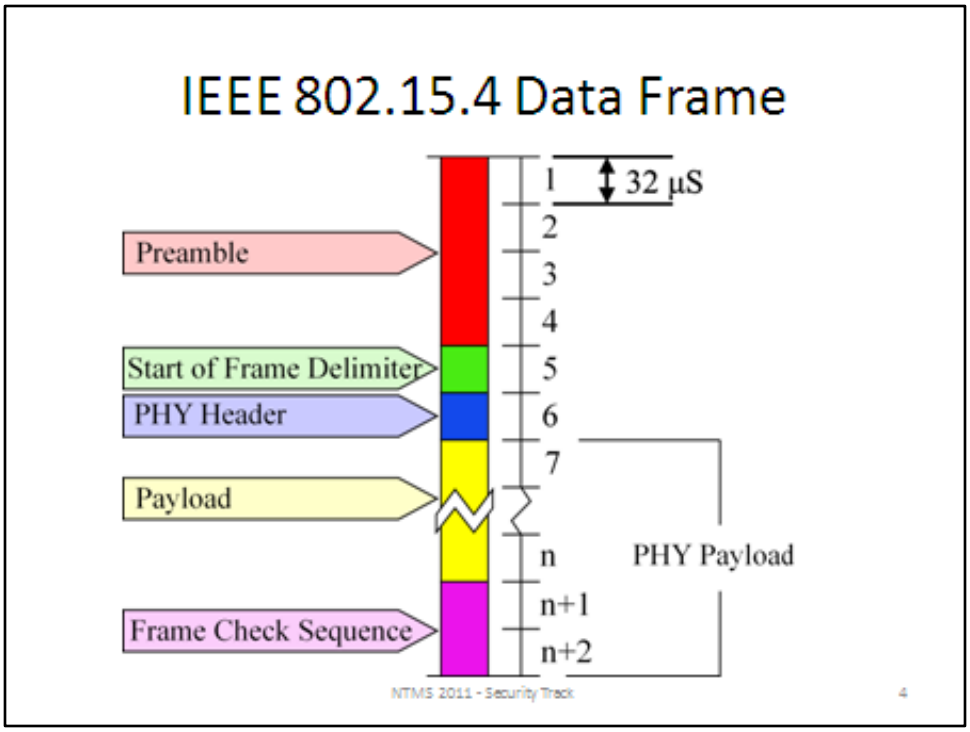
I'll start with a quick background of IEEE 802.15.4 Wireless Networks in case you are not familiar with them. From there I'll move into the attack hardware and it's capabilities, before finally showing the actual attacks and results of those attacks. For brevity this presentation does not cover the countermeasures or transmit power considerations section of the paper.

BACKGROUND



IEEE 802.15.4 is a wireless standard for low-rate wireless personal area networks. It runs in many frequency bands, the most popular is 2.4 GHz which has 250 kbit/second, 10-400 metre range, and uses 16 channels.

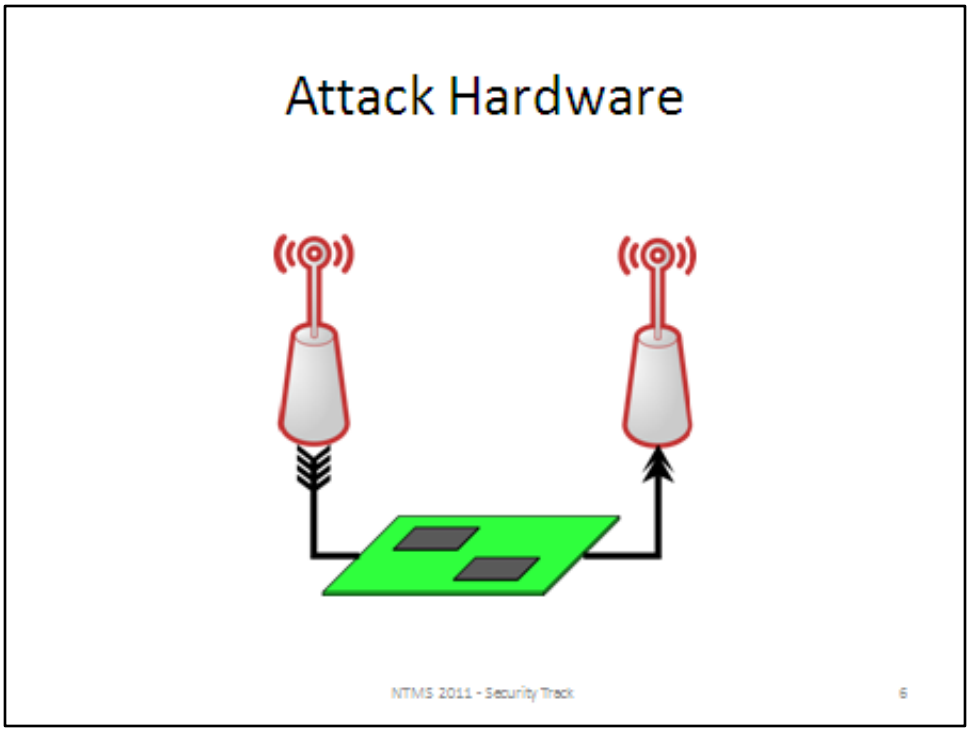
IEEE 802.15.4 is used as a lower layer by several standards. The most prevalent is ZigBee, but other organizations use it as well. 802.15.4 is in home automation, smart energy, security systems, remote controls, and medical devices.



This is the 802.15.4 data frame as transmitted over the air. The numbers indicate the byte – so the start of the frame is at the top of the page. Each byte takes 32 microseconds to transmit (at 250 kbit/sec).

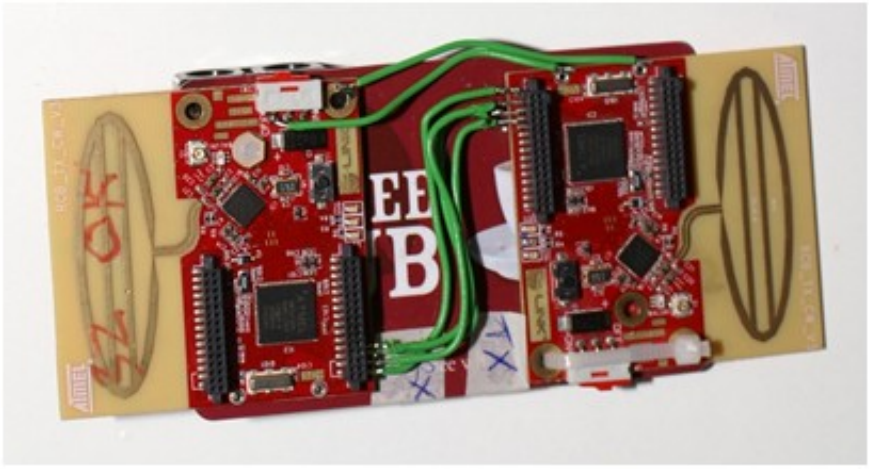
The PHY payload is limited to 127 bytes. The Preamble & SFD are used by the radio to synchronize to an incoming frame. The PHY header is just the number of bytes which will be received.

ATTACK HARDWARE



The attack hardware uses two radios. One is always in receive mode, one is always in 'ready to transmit' mode. It takes time for the radio to switch from receive to transmit, so by using two separate radios the attacker reacts much quicker.

Attack Hardware

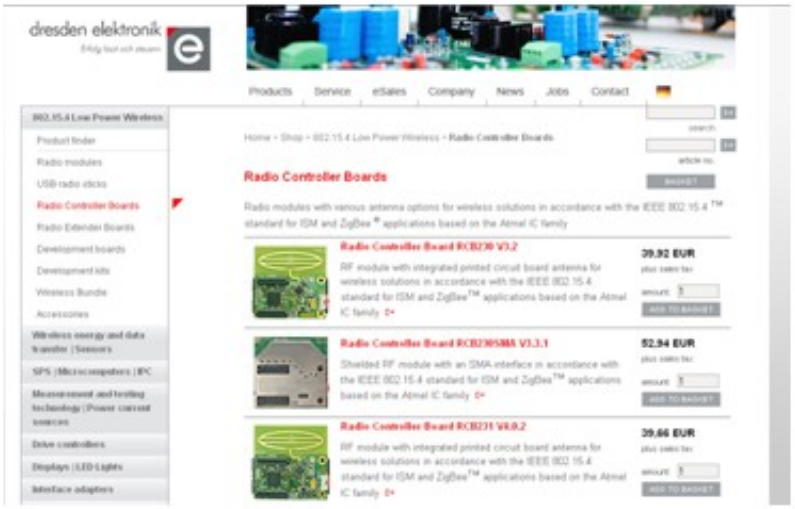


NTMS 2011 - Security Track 7

The actual attack hardware, a commercial development kit. The microcontroller is an 8-bit AVR, the radio an Atmel AT86RF231. Any 802.15.4 board could be used though in a similar way.

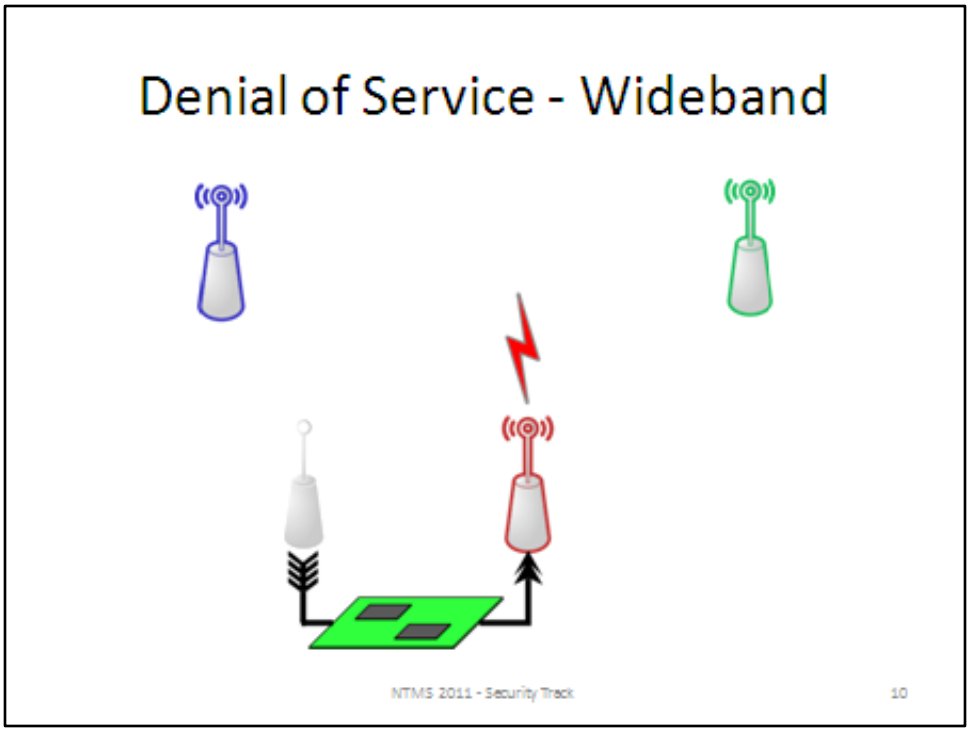
Since originally designing the attack, a number of radio+MCU chips have become available, including Atmel's MegaRF version. Using these would make the attack even better, since access to the radio data occurs quicker compared to having to clock in the radio data over SPI lines.

Attack Hardware



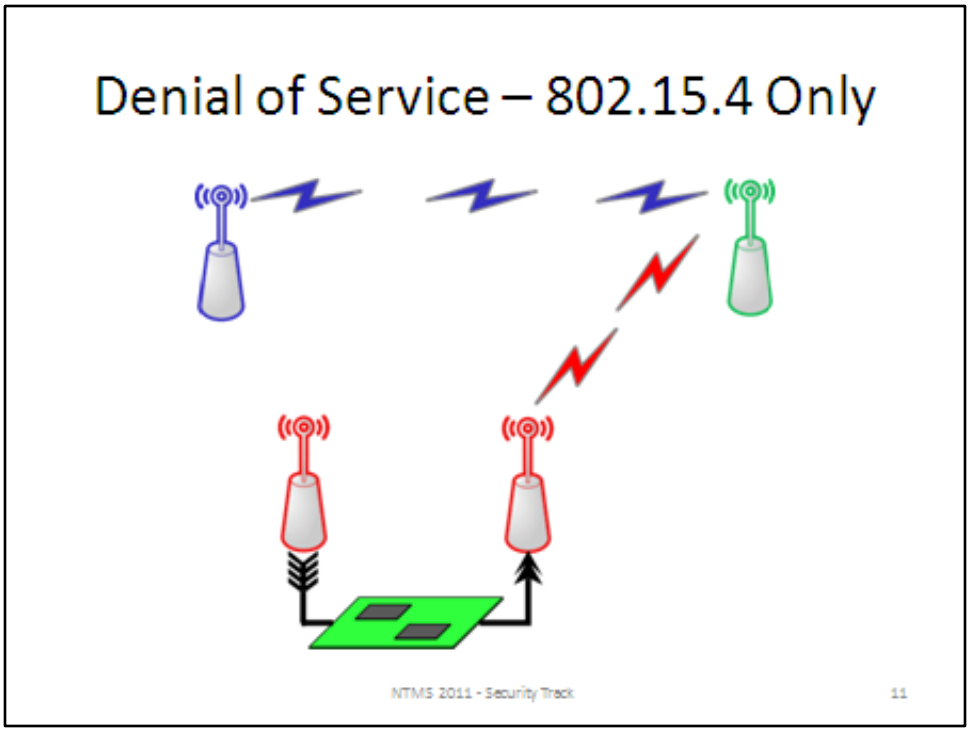
This hardware is available for 80 EUR for two boards.

ATTACK HARDWARE CAPABILITIES

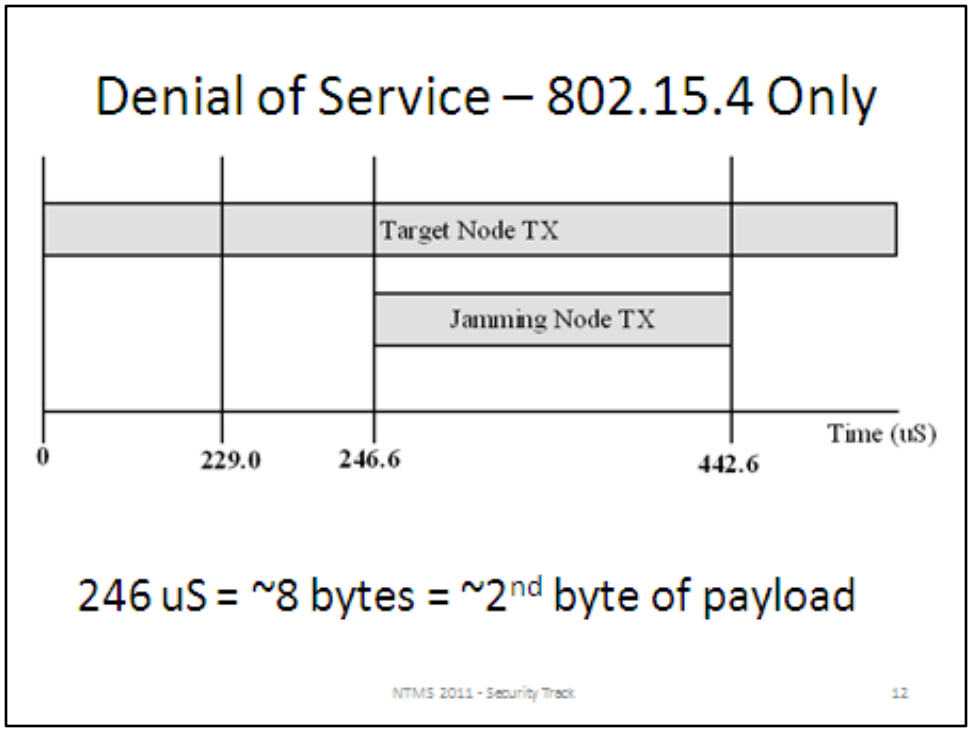


Easiest & dumbest attack is a wideband 'pulse jamming'. Here only the transmit radio is used. It simply transmits pulses of traffic which will disrupt any activity occurring. By hoping channels the entire 802.15.4 spectrum (16 channels) can be disrupted with a single radio for messages greater than 50 octets.

Disadvantage:
- Very easy to detect / track down
- Disrupts other traffic, including WiFi. Makes the attack much more obvious.

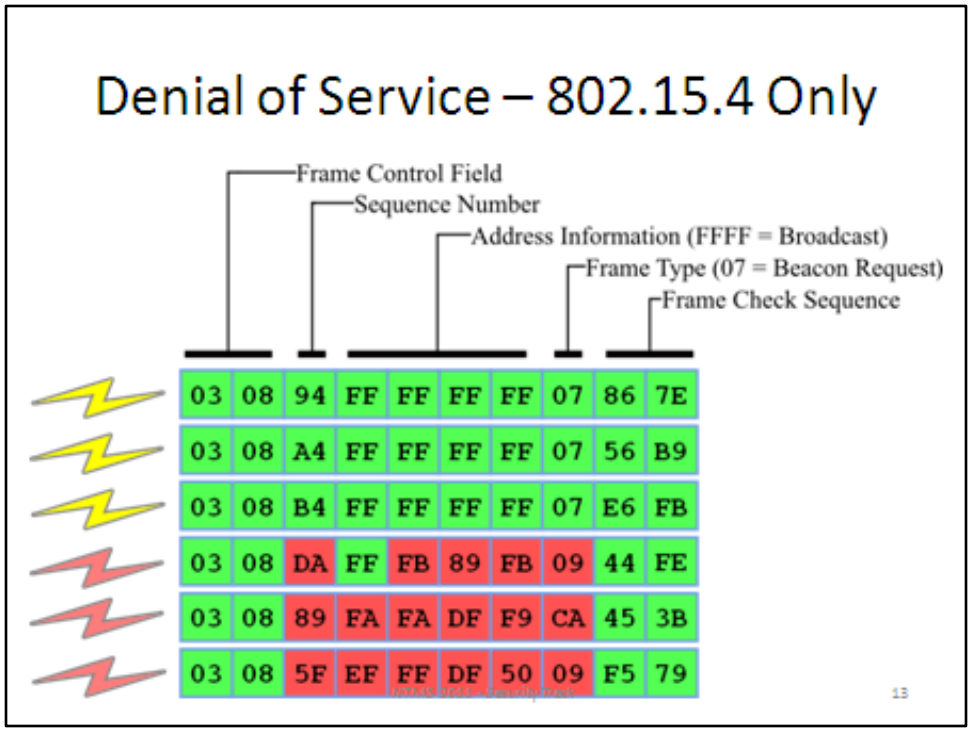


Smarter attack is to wait for 802.15.4 traffic, and then jam. This avoids disrupting other users if you just want to target an 802.15.4 network.



Here is a timing diagram of this operating. The target node starts transmitting at 0 uS. At 229 uS the jamming node has detected the transmission, and it takes a further 17.6 uS to transmit. This means that the collision over the air occurs after about the 2nd payload byte, and should last about 6 bytes.

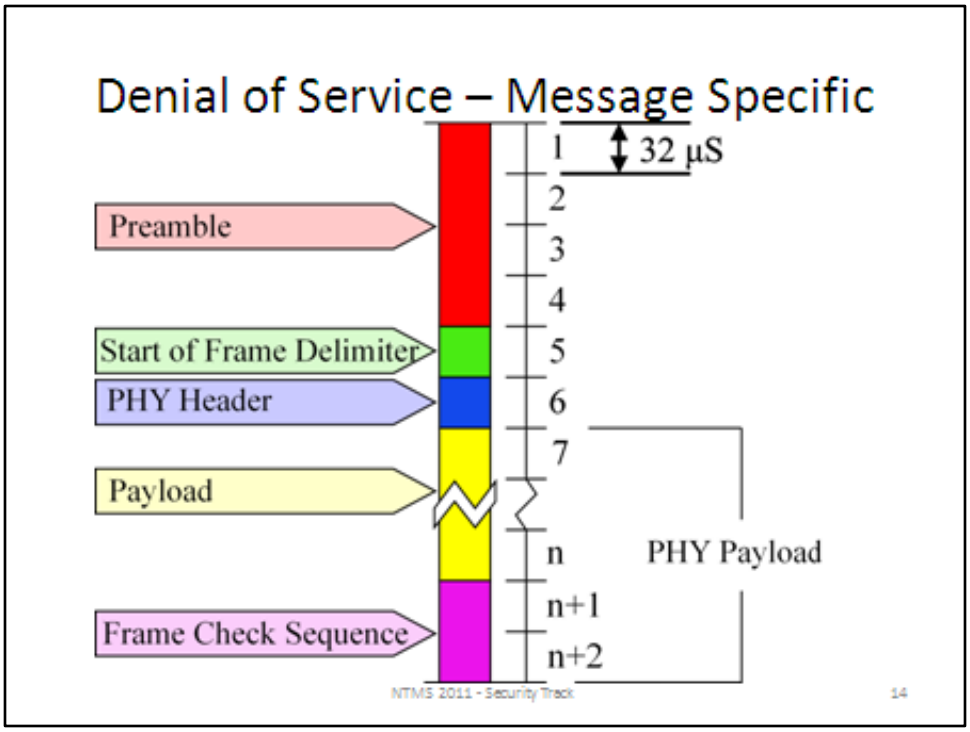
The existence of the jamming transmission occurs entirely within the 'intended' transmission. This eliminates interference with any other uses of the spectrum and should make tracking down the attacker more difficult.



This shows the results of six transmissions of an 802.15.4 beacon request. Time goes from left-to-right in each data frame so "03" is the first byte.

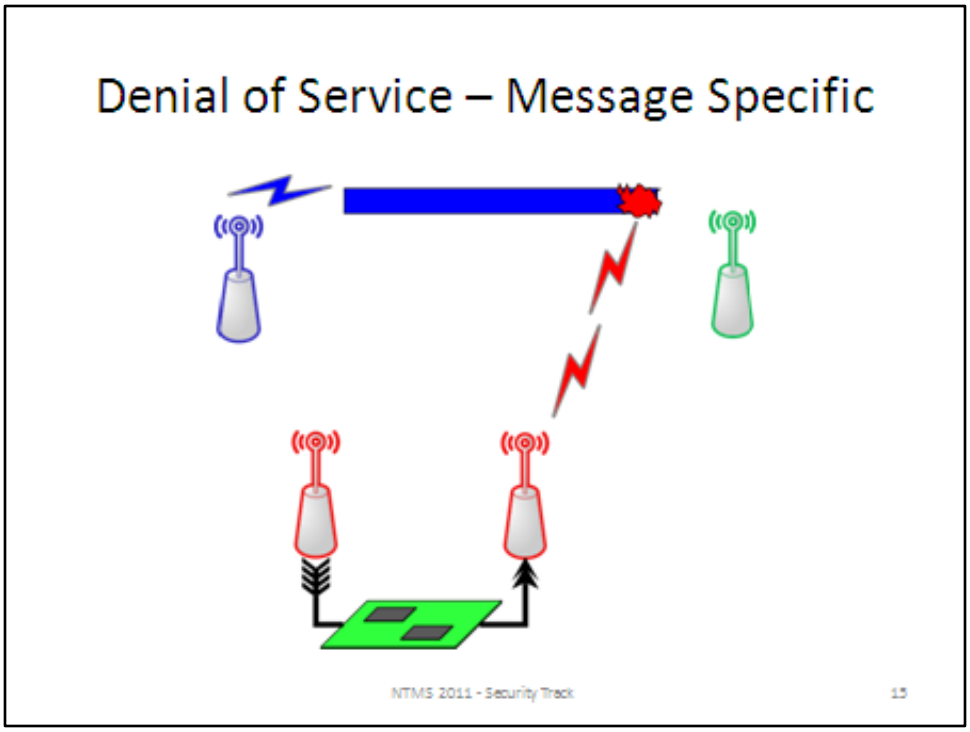
The first three (with yellow lighting bolt) are into clean space with no attacker. The next three (with red bolt) have an attacker present.

Note how six bytes have been corrupted in the middle of the frame. The FCS will no longer be valid so the message will be discarded by the receiver



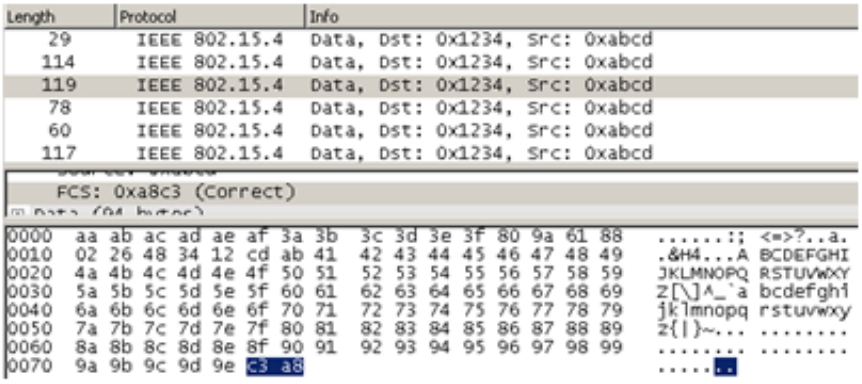
Simply corrupting the frame check sequence (FCS) is enough to stop a message from being received. If an attacker was able to jam JUST the FCS this would mean the attacker still had access to the information that message contained, but denies it to the intended target.

When the message is transmitted the Start of Frame Delimeter (SFD) contains the length of the message, which means you can predict exactly where the FCS will occur before the FCS is actually transmitted. This lets you jam just the FCS.



This is the schematic of the idea. Only transmit to destroy the last byte or two.

Denial of Service – Message Specific



Length	Protocol	Info
29	IEEE 802.15.4	Data, Dst: 0x1234, Src: 0xabcd
114	IEEE 802.15.4	Data, Dst: 0x1234, Src: 0xabcd
119	IEEE 802.15.4	Data, Dst: 0x1234, Src: 0xabcd
78	IEEE 802.15.4	Data, Dst: 0x1234, Src: 0xabcd
60	IEEE 802.15.4	Data, Dst: 0x1234, Src: 0xabcd
117	IEEE 802.15.4	Data, Dst: 0x1234, Src: 0xabcd

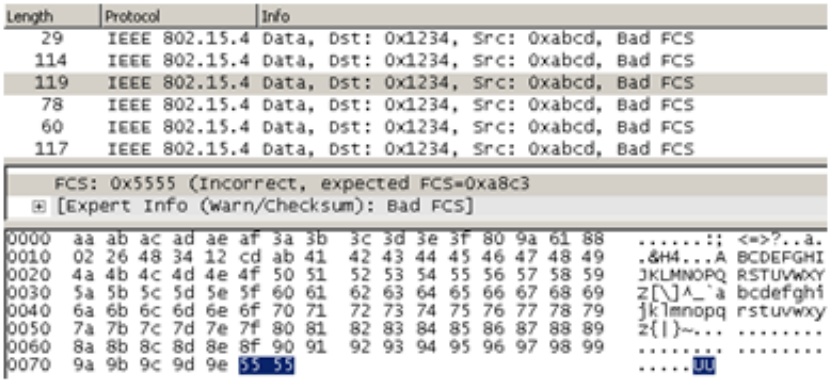
FCS: 0xa8c3 (Correct)

```
0000  aa ab ac ad ae af 3a 3b  3c 3d 3e 3f 80 9a 61 88   ......:; <=>?..a.
0010  02 26 48 34 12 cd ab 41  42 43 44 45 46 47 48 49   .&H4...A BCDEFGHI
0020  4a 4b 4c 4d 4e 4f 50 51  52 53 54 55 56 57 58 59   JKLMNOPQ RSTUVWXY
0030  5a 5b 5c 5d 5e 5f 60 61  62 63 64 65 66 67 68 69   Z[\]^_`a bcdefghi
0040  6a 6b 6c 6d 6e 6f 70 71  72 73 74 75 76 77 78 79   jklmnopq rstuvwxy
0050  7a 7b 7c 7d 7e 7f 80 81  82 83 84 85 86 87 88 89   z{|}~... ........
0060  8a 8b 8c 8d 8e 8f 90 91  92 93 94 95 96 97 98 99   ........ ........
0070  9a 9b 9c 9d 9e c3 a8                               .....
```

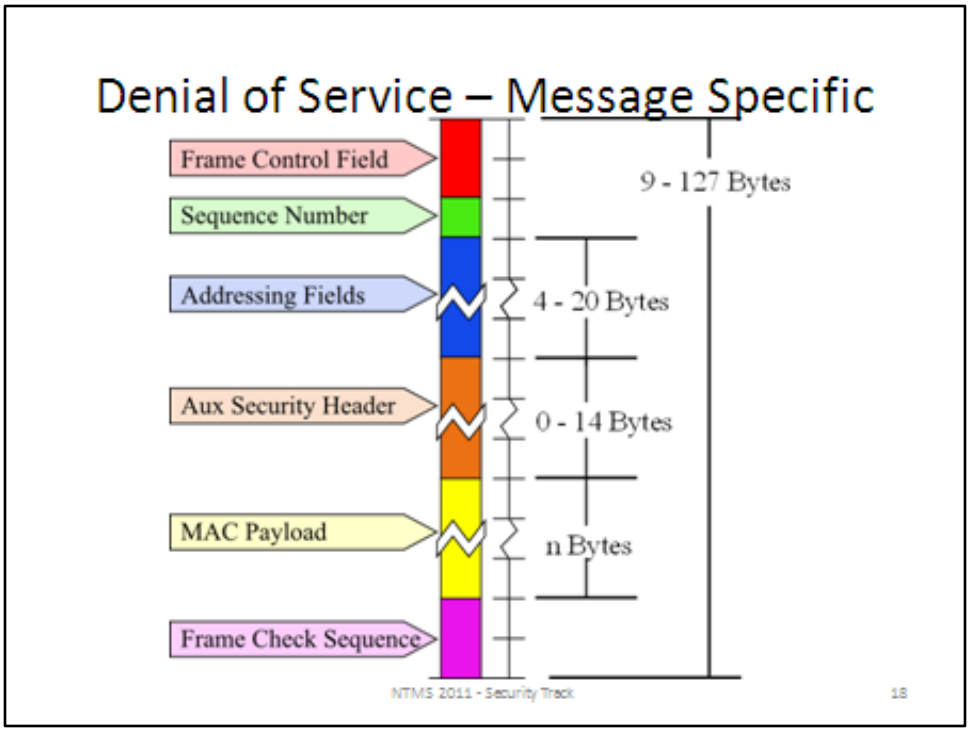

Here are the results. This example network was set up to transmit random lengths of packets from one node to another, note the 'Length' field shows various packets. The data is an incrementing integer starting at hex 41.

Wireshark is being used to decode, and marks the FCS as Correct.

Denial of Service – Message Specific

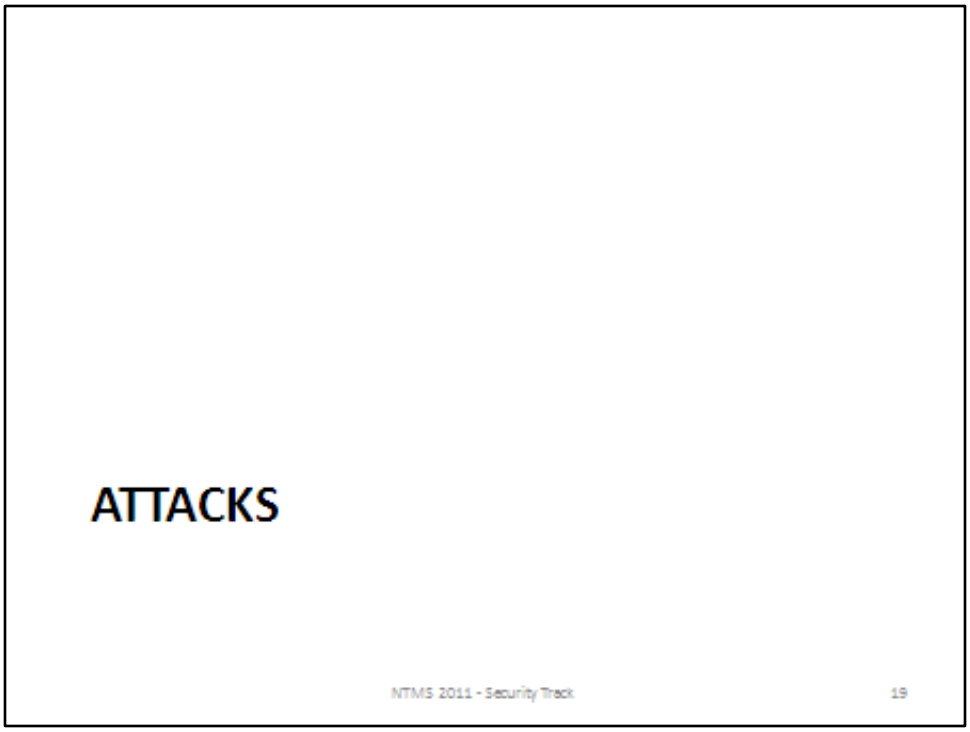


Repeat the same network as before, except now activate the FCS jammer. Note that each packet has been marked as "bad FCS". Looking at the data note the data is not corrupt and contains the expected value, only the FCS has been jammed.

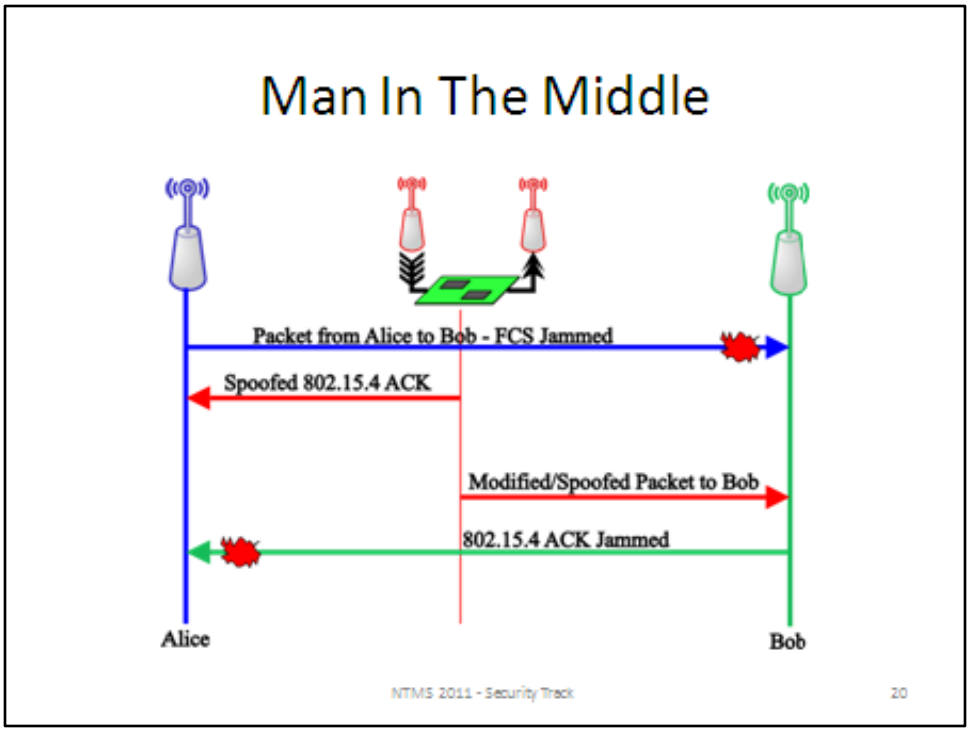


It gets even better. In the 802.15.4 PHY payload all the addressing and frame type information is present. This shows what is included in the PHY payload, starting with the first byte at the top of the page.

This means an attacker is able to block messages to/from a specific target, or specific types of messages. An attacker could even look into packets if they are not encrypted to decide if it should be blocked or not.



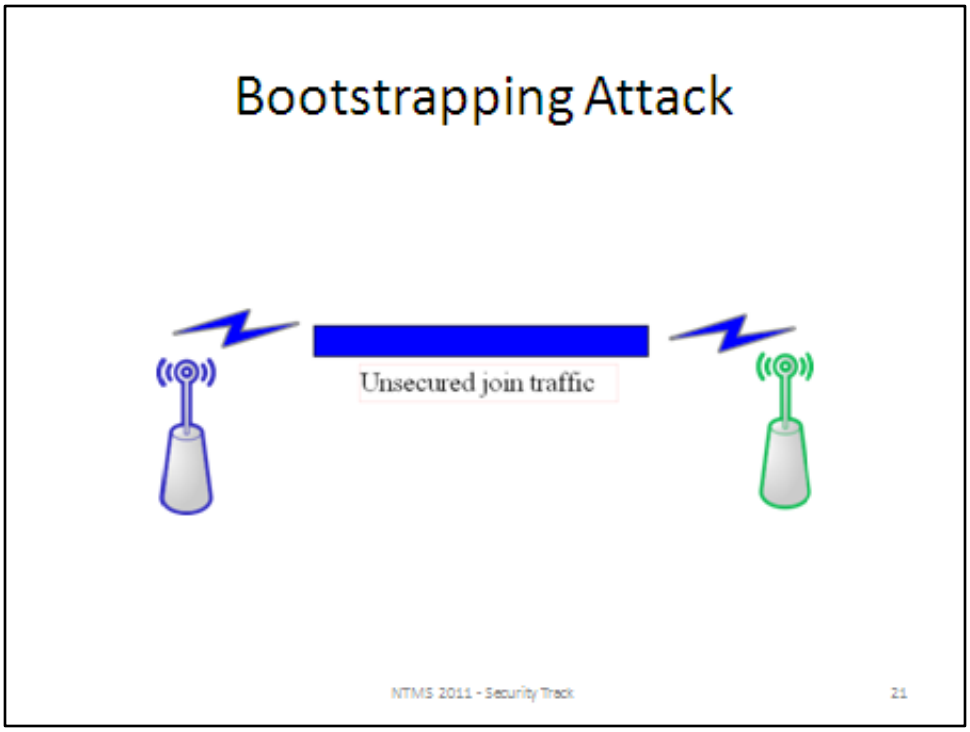
The previous building blocks shows what is capable with the systems.



Here is an example of a simple MITM attack:

The packet from Alice to Bob as the FCS jammed, so Bob never receives it. Alice might be expecting an IEEE 802.15.4-level acknowledgement (ACK) packet from Bob. So the attacker needs to send a fake IEEE 802.15.4 ACK to make it seem like Bob received the message OK.

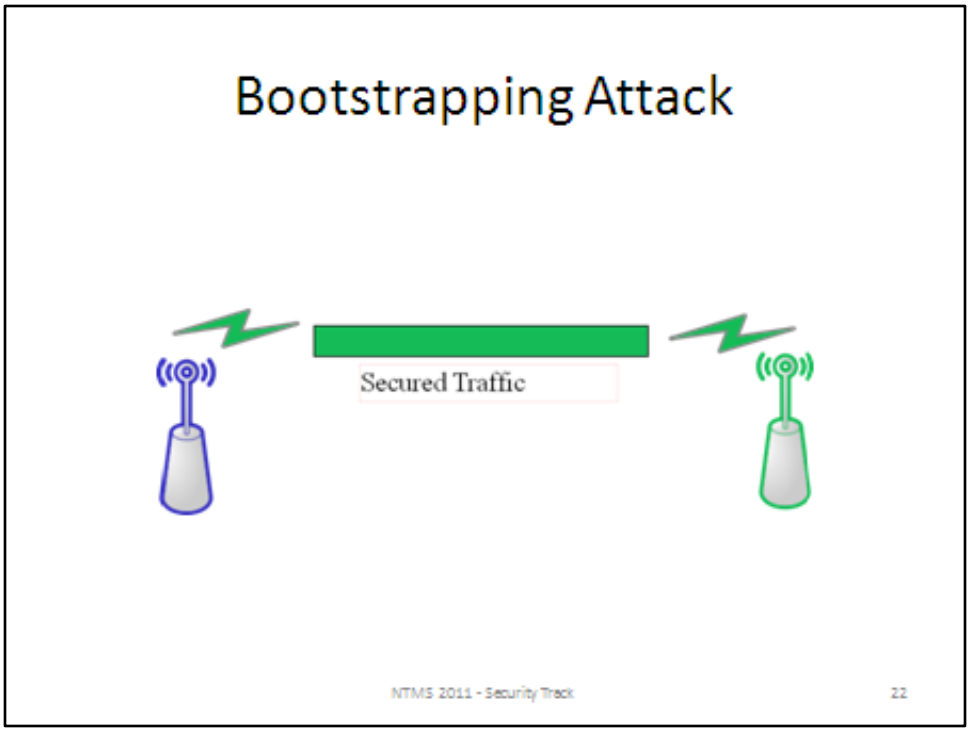
The attacker then sends the modified message onward to Bob. This message can either have the IEEE 802.15.4 ACK request disabled, or jam the resulting ACK packet to stop Alice from noticing it.



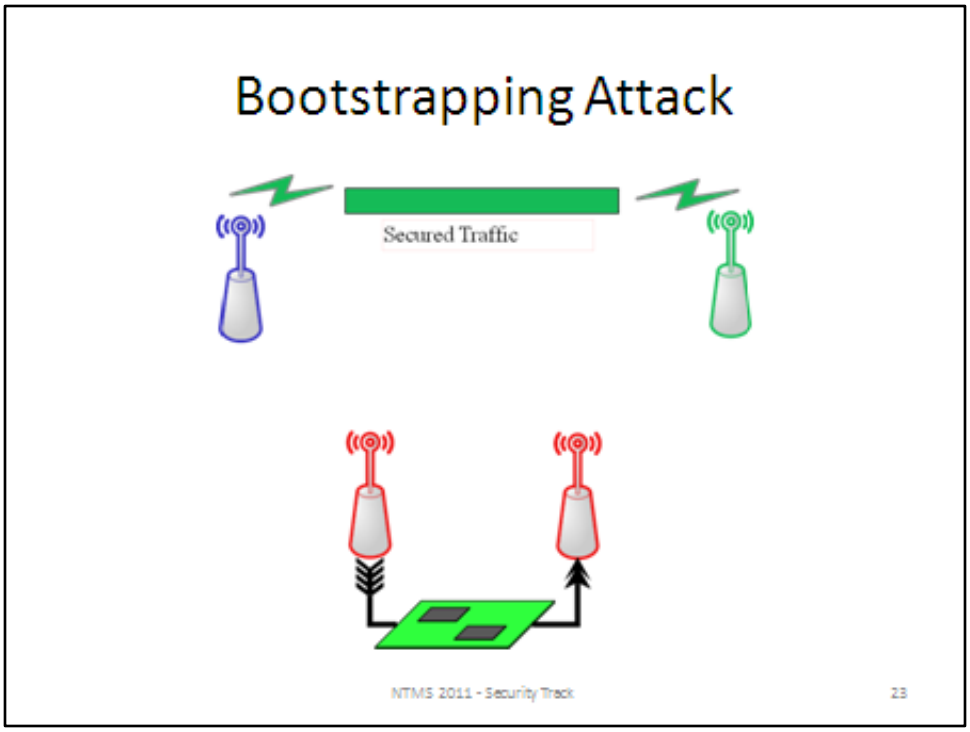
Bootstrapping is what happens when a node joins a network. For example if using a wireless remote control, at some point you need to tell your DVD player it should listen to 'this' remote. Depending on the standard this might be called pairing, joining or other names.

Sometimes an unsecure join method might be used; the assumption being it is very unlikely an attacker will be listening at the exact instant you perform this bootstrapping.

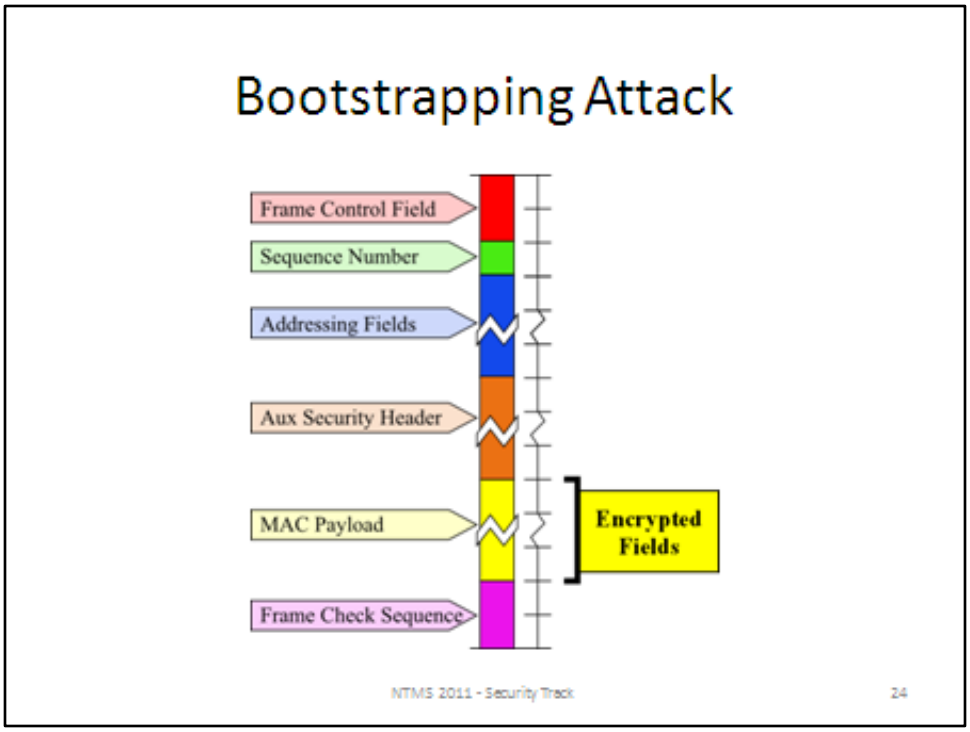
Let's look at how this works with the attacks presented here. So the network starts up, and no attacker is present.



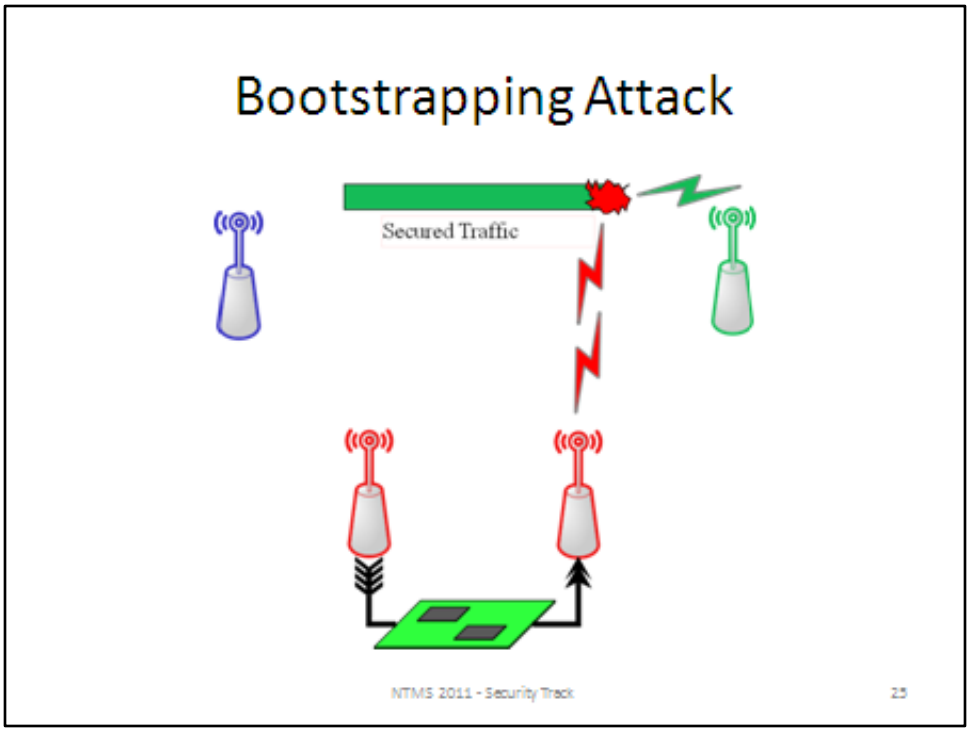
The devices start communicating with encrypted traffic.



An attacker enters the area, but cannot snoop since it cannot decrypt the information.

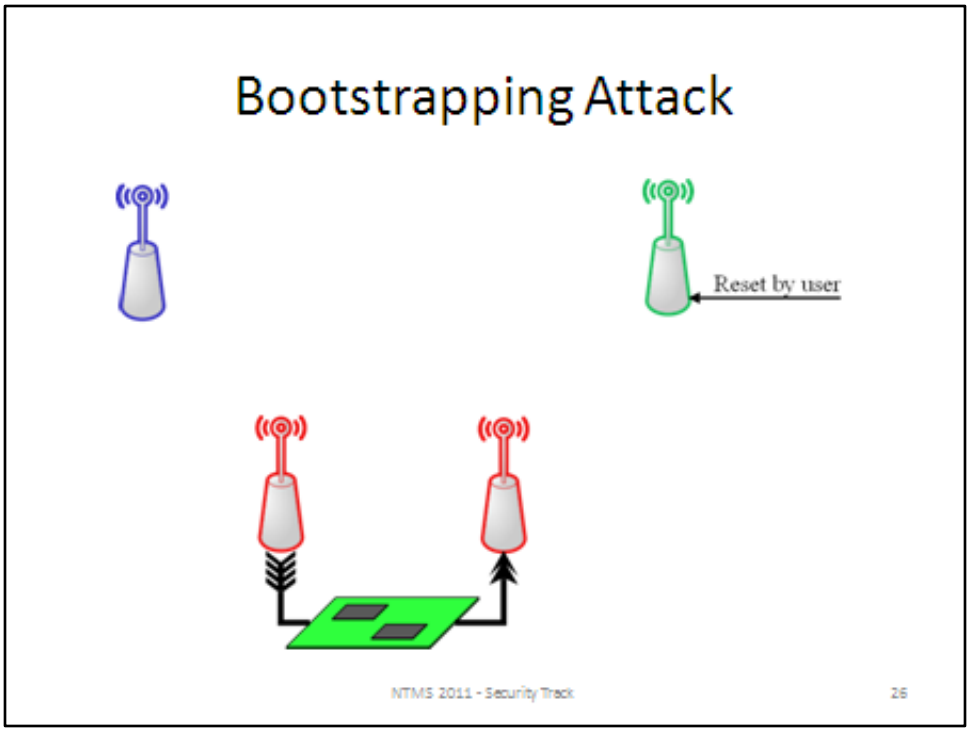


Remember that 802.15.4 security only covers the MAC payload. The addressing information is not encrypted, which is understandable since it would be unrealistic to require every node to decrypt every single packet to see if it is destined to us or not.



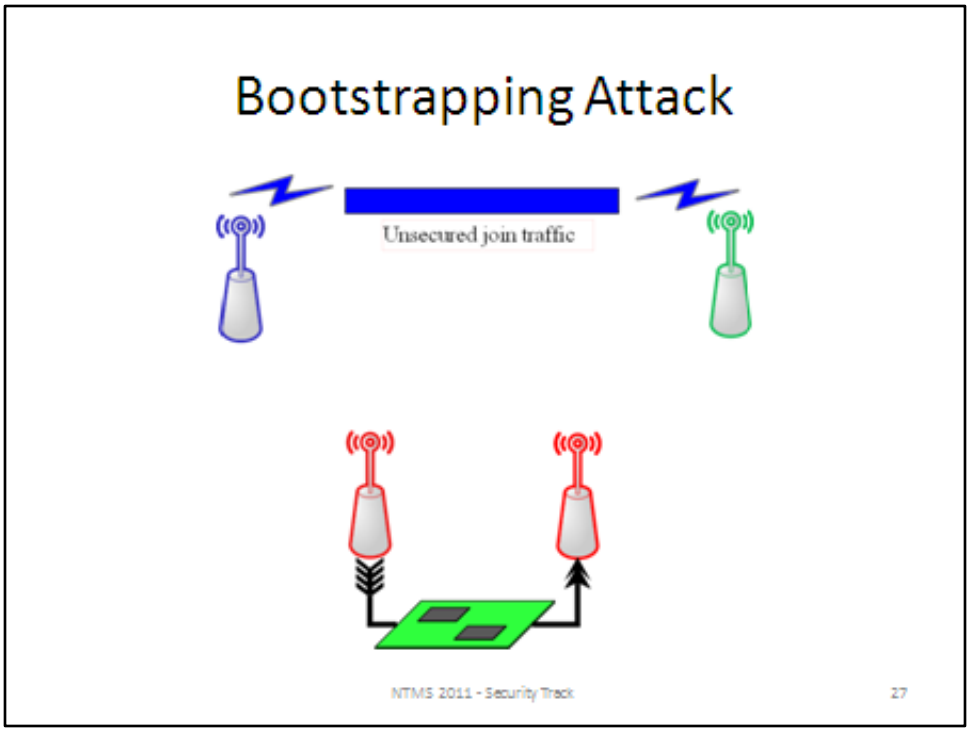
This means the attacking node can perform a selective Denial of Service against a chosen node. Any packet which has a source or destination of the node's address is simply jammed. The attacker can also only block encrypted traffic, since the encryption mode used is transmitted in the 'auxiliary security header'.

At some point the user of the network will consider that this node is broken or misbehaving and requires service.



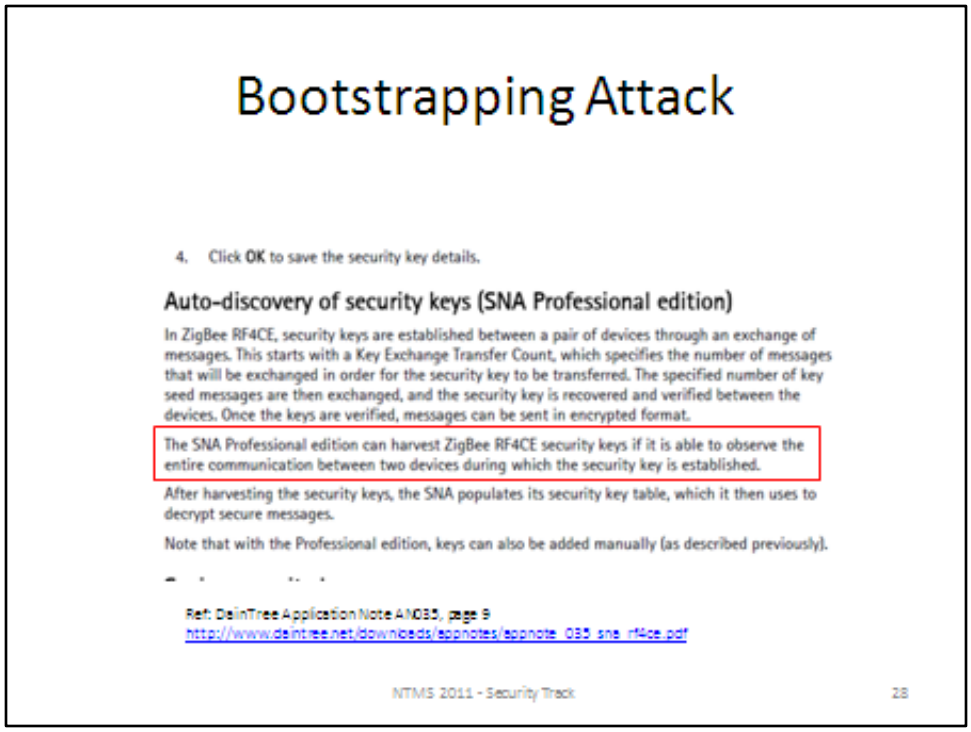
The user will either replace the node, or more likely reset the node in an attempt to fix the problem.

Note the similarity to WiFi deauthentication attack.



When the attacking node detects unencrypted traffic to/from this node, it does NOT block it. The attacker is now present, and records or intercepts the bootstrapping traffic.

The previously blocked node appears to work, so the user is satisfied they have fixed the problem. In reality they have let the unknown attacker learn the required network information.



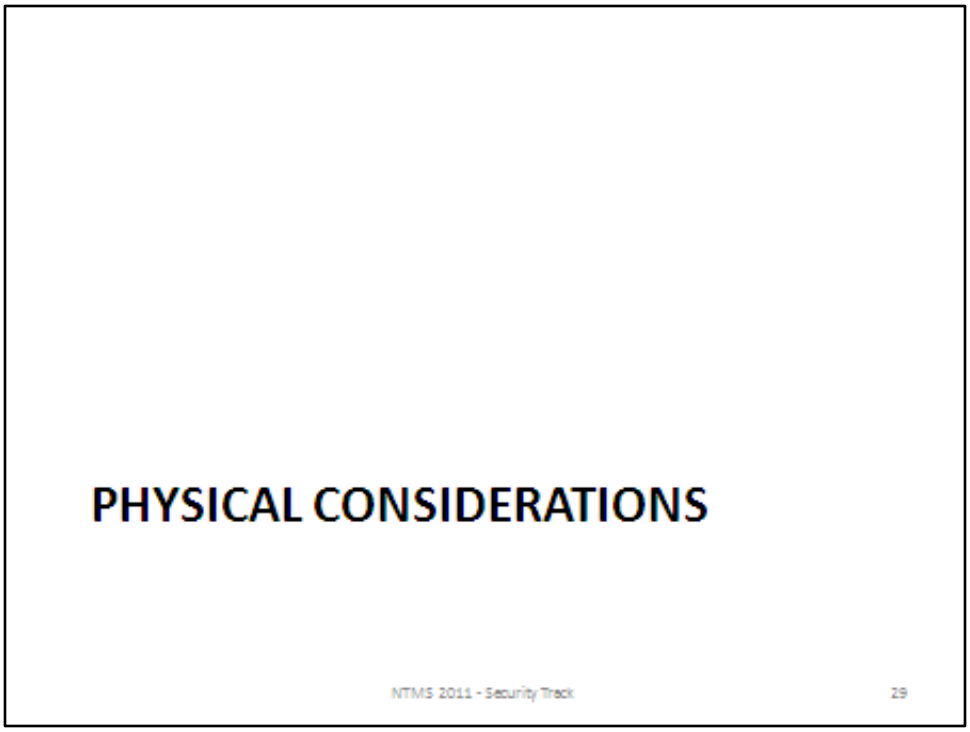
Surely no "real" protocol would send unencrypted data that could be snooped that easily? But they do!

There are a number of proprietary standards which probably do this, but as they are not published it is a lot of work to figure out their join method, since you need to buy two devices and start sniffing.

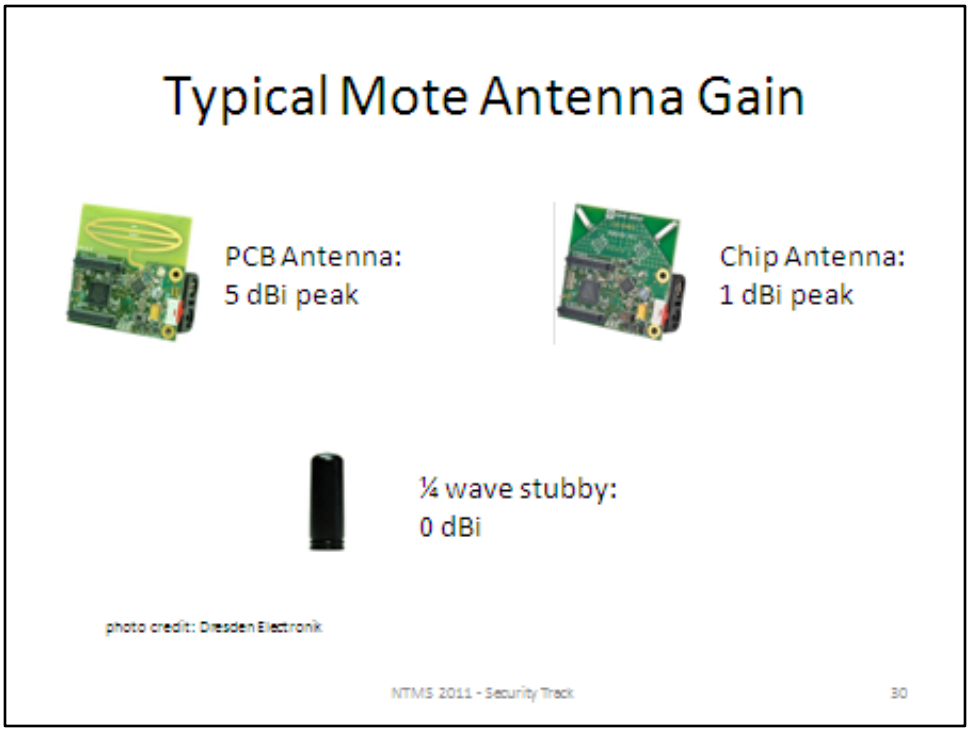
So here is one example from a published commercial standard: ZigBee RF4CE. Zigbee RF4CE is aimed at low-power consumer electronics; for example a remote control & DVD player could communicate with ZigBee RF4CE. The join process (called pairing) is described in the standard, which is available at http://www.zigbee.org/Specifications/ZigBeeRF4CE/download.aspx . This pairing does not send the key completely in clear text, but it can still be easily calculated if the entire join traffic is observed.

Here is proof I'm not just misreading the specification: a snippet from an application note from Daintree, a sniffer manufacture, says that their sniffer can acquire the encryption keys by passively observing the join traffic. Ref: http://www.daintree.net/downloads/appnotes/appnote_035_sna_rf4ce.pdf

This is a very complicated problem, as it's not that the RF4CE standard contains insufficient security. Adding more complicated security that is harder to break would push the computational complexity up, and probably make it unsuitable for the small low-cost devices RF4CE is targeting. If someone is able to control your DVD player, maybe it's a little annoying, but it's hardly a serious problem.

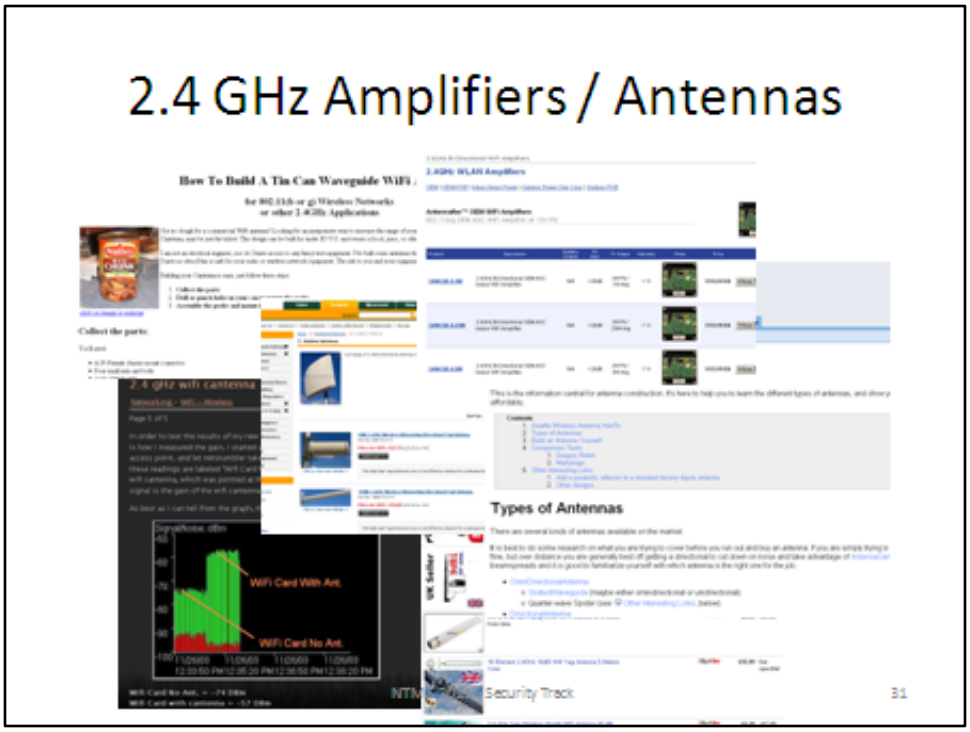


A quick detour into physical capabilities of attacks is required.



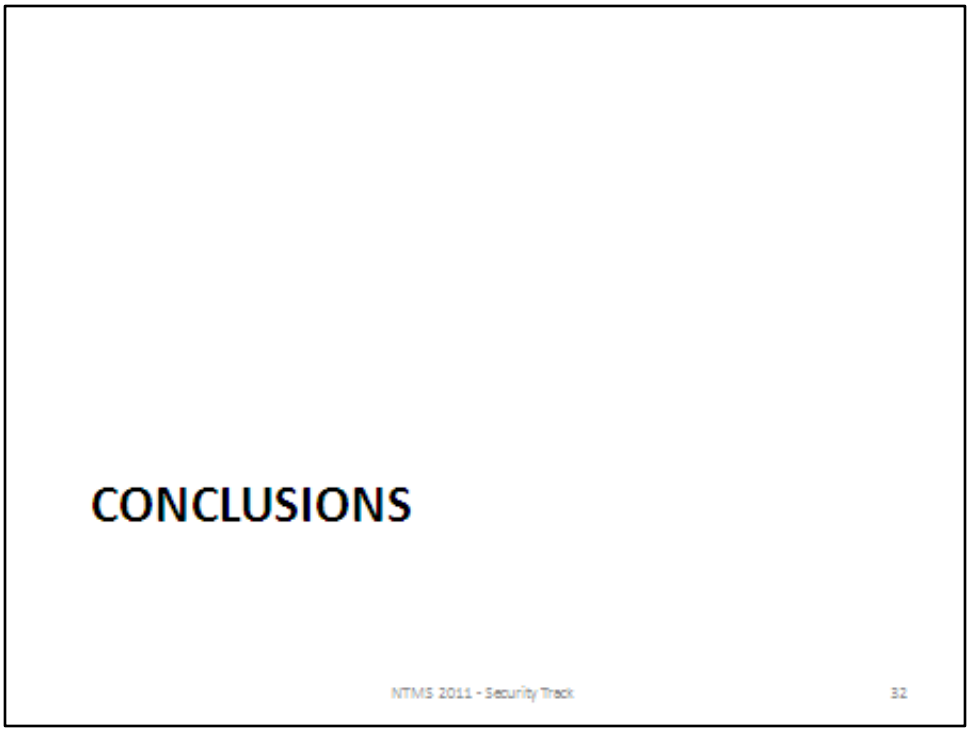
A typical mote/node would use simple and small antennas. Most devices won't use a directional antenna when possible since the RF environment is likely to be always changing.

Output power of the chip itself varies, but the AT86RF231 for example has at most +3 dBm output power.



The 2.4 GHz band of 802.15.4 overlaps almost exactly with WiFi. This means that a wide variety of directional antennas and amplifiers designed for WiFi will work with 802.15.4 networks.

An attacker may be a distance from the network, but can overcome this with antenna and amplifier gain to attack a specific node.



Performing Denial of Service attacks on IEEE 802.15.4 networks can be accomplished easily with minimum hardware. These attacks could be dumb and block access to certain areas, or sophisticated and block certain messages as part of a larger attack; e.g. Man In The Middle. The wide availability of IEEE 802.15.4 development kits and hardware means performing these attacks is trivial. Designing a network with inadequate or missing security is unacceptable, as it leaves nodes open to misuse.

If you have any questions feel free to contact me at coflynn@newae.com .