INTRODUCTION

- Matt Knight
- Software Engineer and Threat Researcher @ Bastille
- Background in electrical engineering, embedded software, etc.
- I love wireless!
BEFORE WE GET STARTED . . .

- No 0-days or exploits
- We’re going to take apart a cutting edge protocol
- This is a software defined radio talk
- I’ll assume you’re technical, but no SDR experience required
WHY IS THIS RELEVANT

▸ Cisco IBSG: 50 billion devices by 2020
  ▸ Fewer wires every year

▸ Wireshark wasn’t always a thing

▸ Monitor mode wasn’t always a thing

▸ Low-level access to interfaces is paramount for enabling comprehensive security research
AGENDA

1. Survey of IoT-focused wireless
2. Introduce LPWANs
3. Review technical radio concepts
4. Reverse engineer the LoRa PHY with SDR
IOT == EMBEDDED
EMBEDDED/IOT REALITIES

- Low intelligence (basic CPUs)
- Battery powered
- Hard-to-reach places
- Longevity/high endurance
- Must be easy to configure
EMBEDDED/IOT COMMUNICATION — REQUIREMENTS

- Given these constraints, an ideal IoT interface is...
  - Wireless
  - Easy on the battery
  - Capable of being installed anywhere
  - No configuration/easy to configure
  - Inexpensive!
EMBEDDED/IOT COMMUNICATION — REQUIREMENTS

- IoT interfaces do NOT require:
  - High throughput
  - Persistent/always-on connections

- Current IoT devices are grossly overserved
IOT INTERFACES

STATE OF THE ART
EXISTING IOT WIRELESS

- PAN-like technologies
  - 802.15.4 family
    - ZigBee, 6LoWPAN, Thread, etc.
  - Bluetooth
  - Bluetooth Low Energy
- Cellular
  - 2G (incl. GPRS/EDGE)
  - 3G
- Proprietary
  - OOK, ASK, FSK, etc.

EXISTING IOT WIRELESS

- PAN-like technologies
  - 802.15.4 family
    - ZigBee, 6LoWPAN, Thread, etc.
  - Bluetooth
  - Bluetooth Low Energy
- Cellular
  - 2G (incl. GPRS/EDGE)
  - 3G
- Proprietary
  - OOK, ASK, FSK, etc.

THEIR WEAKNESSES

- PAN-like technologies
  - Limited range
  - High-touch provisioning
- Cellular
  - Not inexpensive
  - Not fantastic on battery
- Proprietary
  - Some combination of the above
  - Insecurity by obscurity
WHICH IS BEST
DEPENDS
IF YOU NEED...

1. RANGE
2. COVERAGE
3. EASE OF INSTALLATION
EXISTING IOT WIRELESS

- PAN-like technologies
  - 802.15.4 family
    - ZigBee, 6LoWPAN, Thread, etc.
  - Bluetooth
  - Bluetooth Low Energy
- Cellular
  - 2G (incl. GPRS/EDGE)
  - 3G
- Proprietary
  -OOK, ASK, FSK, etc.

THEIR WEAKNESSES

- PAN-like technologies
- Limited range
- High-touch provisioning
- Cellular
- Not inexpensive
- Not fantastic on battery
- Proprietary
- Some combination of the above
- Insecurity by obscurity
CELLULAR’S NOT GOING ANYWHERE*

*2G, HOWEVER, IS
DEPRECIATION: A DEVELOPER’S CONUNDRUM

- AT&T to sunset 2G on January 1, 2017
- Other major carriers to follow

- 2G advantages: ubiquitous, battery-conscious, somewhat cheap
  - Exactly what IoT devices require
REPLACING 2G

- 3G
  - More expensive
  - Harder power requirements
- LTE-M/NB-LTE Release 13
  - IoT focused cellular protocols
  - Not ready by the sunset date, which means...
VOID IN MARKET
INTRODUCING LPWANS
LPWAN

- **LPWAN**: Low Power Wide Area Network
- Like cellular, but optimized for IoT/M2M
  - Network of basestations worldwide
  - Star network to endpoints, UL/DL traffic
  - Range in miles
EMERGING STANDARDS
REVERSING LORA // BASTILLE NETWORKS

AGGRESSIVE INVESTMENT

- SIGFOX raised 115MM last year
- WSJ: Possible US IPO soon
- Senet and Actility, LoRa backers, raised a combined 51MM
- LoRa alliance membership tripled last year

[Image of SIGFOX logo]
[Image of Senet logo]
[Image of Actility logo]
LPWANS

THE STACKS

OPTIMIZED FOR IOT

- Battery-conscious
  - SIGFOX advertises 10 years on 1 AA battery
- Long range
  - LoRa advertises up to 13.6 miles

Compare this with...

- 2G: typically 1-2 miles, max 22 miles, a few days
- 802.15.4: 10-100 meters, months-years
- WiFi: 30 meters, a few days
COMPROMISES
EMBRACING COMPROMISE

- Conservative *duty-cycling* and listening
- Very *sparse* datagrams
- Highly *rate-limited*
EMBRACING COMPROMISE

Examples

- SIGFOX limits devices to 140 12-byte datagrams per day
- Weightless-N is uplink-only
- LoRa Class A devices can only receive downlink momentarily after uplinking
RADICALLY DIFFERENT
INTRODUCING

LORA
HISTORY

- LPWAN developed by Semtech
- PHY patented in June 2014
- LoRaWAN MAC/NWK stack released in January 2015
- Supported by LoRa Alliance
NOMENCLATURE

- LoRa vs. LoRaWAN
  - LoRa: PHY layer
  - LoRaWAN: MAC, NWK, and APP built on LoRa
LORAWAN NETWORK TOPOLOGY

- Gateway: low-IQ basestation / LoRa concentrator
- Network Server: routes between gateways and application servers
  - Roaming being defined
- Application Servers

LORAWAN SECURITY ARCHITECTURE

- NwkSKey: end device to network server
- AppSKey: end device to application server
- Keys can be preconfigured or sent OTA
LICENSURE, OR LACK THEREOF

- 900 MHz ISM
  - US: 902-928 MHz
  - EU: 868 MHz
- No special license required
LICENSURE, OR LACK THEREOF

- Compare this with cellular
- FCC auctions cellular spectrum licenses for billions
- Restricts building infrastructure to biggest telcos
- Left: opening bid list for FCC TV whitespace reverse auction
LORAWAN NETWORK PROVIDERS

- Senet
  - Commercial network
- The Things Network
  - Crowdsourced
- No licensed spectrum required...!!

http://img.scoop.it/pkne9jh9_fl-PFplrbFgL4XXL4j3HpehxjNOf_P3YmryPKwJ94QGRtDb3Sbc6KY
OBSCENELY SHORT

RADIO CRASH COURSE
PHY LAYER

- Lowest layer in communication stack
- In wired protocols: voltage, timing, and wiring defining 1s and 0s
- In wireless: patterns of energy being sent over RF medium
WHAT IS RF?

- “Radio Frequency”
- Electromagnetic waves
- Energy
MANIPULATING RF

- Done with a radio
- Hardware defined
  - RF and protocol in silicon
- Software defined
  - Flexible silicon handles RF
- Protocol-specific components implemented in software (CPU or FPGA)
PHY COMPONENTS

- Modulation
  - How digital values are mapped to RF energy
- RF parameters that can be modulated:
  - Amplitude
  - Frequency
  - Phase
  - some combination of the above

http://xenon.colorado.edu/spotlight/kb/gps_basics/modulations.001.png
MODULATION

- Modulators can modulate analog or digital information

- Digital modulation
  - **Symbols**: RF energy state representing some quantity of information
COMMON IOT PHYS

- **FSK/GFSK**
  - RF energy **altersates** between two frequencies to signify digital values

- **ASK/OOK**
  - Changes in RF **power** on a certain frequency signify digital values

http://3.bp.blogspot.com/-w66qwKucSJl/UgWKKmPUP2I/AAAAAAAADYA/B9NMGYzqJvkJ/s1600/Screenshot-2013-08-03-04-41-52.png
SYMBOLS ILLUSTRATED

- Time domain
- Top: FSK
- Bottom: OOK/ASK

- Compare with analog modulation
  - Analog = infinite possible symbols
  - Digital = finite number of possible symbols, defined by modulation
MORE COMPLICATED IOT PHYS

- Spread spectrum
  - Data bits are encoded at a higher rate and occupy more spectrum
- Resilient to RF noise

- Examples
  - 802.15.4
  - Bluetooth
  - Bluetooth Low Energy
TOOLS USED IN THIS TALK

- **Transmitter**: Microchip LoRa RN2903 Module
  - Contains hardware-defined Semtech LoRa radio
- **Receiver**: Ettus B210
  - Software-defined radio
  - Python, GNURadio, and Baudline to process
LAST THING . . . FFT

- Fast Fourier Transform

- Decomposes a signal into its component frequencies
  - Any periodic signal can be modeled as the sum of harmonic sine waves
- Allows analysis and visualization of frequency domain
LORA’S PROPRIETARY PHY

- Modulation: Chirp Spread Spectrum (CSS)
- What’s a chirp?
  - A signal of continuously increasing or decreasing frequency
  - i.e. a “swepted tone”
CSS CHIRPS

- **Upchirp** (top)
  - Increasing frequency
- **Downchirp** (bottom)
  - Decreasing frequency
CSS ADVANTAGES

- Great link budget
  - Resilience to interference
  - Performance at low power
- Resistance to multi-path effects
- Resistance to Doppler effect (mobile applications)
- Interesting set of pros... where else are chirps used?
RADAR
CHIRPS IN RADAR

- Various military and marine radars
  - Wideband and pulse compression
- Open source GNU Chirp Sounder
  - Ionospheric radars
  - Space weather
SEEKING LORA

- December 2015: No LoRa sightings in Boston, Atlanta, San Francisco, or New York
- I encountered Senet at a Meetup event in Cambridge, MA
- While watching one of their marketing videos...

Portsmouth, NH!
ROAD TRIP
EXAMINING THE LORA PHY FRAME

- Repeated upchirps
  - Preamble/Training Sequence
- Two downchirps
  - Start of frame delimiter (SFD)
- Choppy upchirps of varying length
  - Data!
PHY DATA UNIT STRUCTURE

- Chirp frequency is static
- Chirp “jumps” throughout band
- Instantaneous frequency changes are result of data being modulated onto the chirps
  - Chirp “phase”
DEMODULATING

LORA
BEFORE WE GET STARTED... WHAT DO WE KNOW?

- Technical documentation
  - Semtech European patent application 13154071.8
  - LoRa Alliance LoRaWAN spec (MAC/NWK only, not a PHY spec)
  - Semtech app notes AN1200.18 and AN1200.22
- Prior art
  - Partial implementation in open source rtl-sdrangelove
  - Observations at https://revspace.nl/DecodingLora
SOME DEFINITIONS . . .

- **Bandwidth**: width of spectrum occupied by chirp
- **Spreading factor**: number of bits encoded per symbol (RF state, remember?)
- **Chirp rate**: first derivative of chirp frequency
Some Definitions...  

- **Bandwidth**: width of spectrum occupied by chirp
- **Spreading factor**: number of bits encoded per symbol (RF state, remember?)
- **Chirp rate**: first derivative of chirp frequency

Some Numbers...  

- US: 125kHz, 250kHz, 500kHz
- US: [7-12] bits per symbol
- bandwidth/(2**spreading_factor)
SO WHAT’S A SYMBOL?

- Instantaneous change in frequency
- FM modulated chirps
DEMODULATING THE PHY

1. Identify the beginning of a frame
2. Find the beginning of the PHY data unit
3. Extract data from instantaneous frequency transitions

- How? We need to quantify the frequency transitions
TRANSFORMING THE SIGNAL

- De-chirping the signal makes analysis easier
- Generate local upchirp and downchirp at the appropriate chirp rate
- Multiply each against the signal and something interesting happens...
REMEMBER SYMBOLS

- Symbol: RF state representing some quantity of information
- LoRa spreading factor: number of bits encoded into each symbol
- How many possible symbols are there?
  - $2^{\text{spreading\_factor}}$
**EXTRACTING SYMBOLS**

- Length of FFT == number of component frequencies to be extracted
- Channelize and resample signal to chirp bandwidth
- Make the length of the FFT equal to the number of possible symbols
- Most powerful component in each FFT is the symbol!
DEMODULATION SUMMARY

1. Identify the beginning of a frame
2. Find the beginning of the PHY data unit
3. Extract data from instantaneous frequency transitions
DEMODULATION SUMMARY

1. Identify the beginning of a packet
   - Preamble signified by continuous up-chirp
   - == same symbol being transmitted over and over
   - Look for some number of consecutive FFTs with maximum power in the same bin
DEMODULATION SUMMARY

2. Find the beginning of the PHY data unit
   - Repeat same process looking for SFD down-chirps
   - Down-chirp is complex conjugate of the up-chirp
   - PHY data unit begins 2 symbols after the SFD
BUT WAIT!

- Accurately finding SFD is essential for receiver synchronization
- Bad sync can spread symbol energy between adjacent FFTs
  - == incorrect data!
SFD SYNC SOLUTION

- Increase FFT time-based precision once preamble is found
- Overlapping FFT sample buffers!
OVERLAPPING FFTS

- **Top: non-overlapping FFT**
  - Each sample processed exactly once
  - 
    \[[i*\text{fft\_len}:(i+1)*\text{fft\_len}-1]\]

- **Bottom: overlapping FFT**
  - Samples shifted across multiple FFTs
  - 
    \[[i*(\text{fft\_len}/n\_overlaps):(i+1)*(\text{fft\_len}/n\_overlaps)-1]\]
OVERLAPPING FFTS

- Use overlapping FFTs to synchronize to first sample in the first SFD symbol
- Re-compute with non-overlapping FFTs to get your data!
3. Extract data from chirp phase transitions
   - Use described FFT method
   - **Normalize** data about the value of the preamble (always value 0)
WE'RE DONE, RIGHT?
NO.
DATA ENCODING

- Symbols represent encoded data
  - What?

- Data is *transformed* before it is transmitted
  - Why?

- Encoding increases OTA resiliency
“ARFZ BE LIKE IT IS.”

Marc Newlin
**RF IS A BRUTAL ENVIRONMENT**

- All systems see interference from weather, geomagnetic activity, etc.
- Some systems have protected/reserved frequencies
- LoRa is ISM – TONS OF CHATTER
  - RF contention/collision is guaranteed
- **Encoding** scrambles and replicates data within frame
WHAT KIND OF ENCODING?

- Semtech European patent application clues:
  1. Symbol “gray indexing” — Adds error tolerance
  2. Data whitening — Induces randomness
  3. Interleaving — Scrambles bits within frame
  4. Forward Error Correction — Adds correcting parity bits

- 4 distinct operations to reverse!
FORWARD ERROR CORRECTION

- Parity bits that can repair bit errors
- Hamming(M,N)
  - M: total bits per codeword [5:8]
  - N: number of bits which are data bits [4]

- Hamming error correction rule
  - \((2^{\text{num\_parity\_bits}}) \geq (\text{num\_parity\_bits} + \text{num\_data\_bits} + 1)\)
HAMMING(M,4) CAPABILITIES

- (5,4): Parity
- (6,4): Parity
- (7,4): Single bit error correction
- (8,4): Single bit error correction, double error detection
GRAY INDEXING

- Originally read as gray-coded before transmission
- Actually de-grayed before transmission
WHITENING

- Transmitter XORs frame with a pseudorandom sequence
- Receiver XORs RX’d frame with same sequence
  - Returns original frame, b/c XOR is its own inverse
- Why? Randomizing data helps receiver synchronization
  - Similar to Manchester encoding
  - Manchester reduces bit rate, whitening does not
FINDING THE WHITENING SEQUENCE

- Several whitening algos defined in Semtech AN1200.18
- Other examples in rtl-sdrangelove
- None of them worked
DERIVING THE WHITENING SEQUENCE

- data XOR 0 == data

- Transmitting a frame of all 0s actually transmits the whitening sequence!

- Hamming (8,4) of 0b0000 = 0b00000000

- Non-additive interleaving of 8x 0b00000000 = 8x 0b00000000
FINDING THE INTERLEAVER

- Semtech European patent application defines diagonal interleaver
  - data(byte,bit) = symbol(bit,(bit+byte)%len_word)

- Also doesn’t work!
DERIVING THE INTERLEAVER

- This was hard

- Exploit properties of Hamming FEC to reveal patterns
  - Most (8,4) codewords contain 4 set bits, however...
    - (8,4) 0b0000 = 0b00000000
    - (8,4) 0b1111 = 0b11111111
DERIVING THE INTERLEAVER, ILLUSTRATED

- spreading_factor = 8, code_rate=4/8

<table>
<thead>
<tr>
<th>Payload: 0x0000000F</th>
<th>Payload: 0x000000F0</th>
<th>Payload: 0x00000F00</th>
<th>Payload: 0x00F00000</th>
</tr>
</thead>
<tbody>
<tr>
<td>00100011 11000000 00001001 11010000</td>
<td>00100101 00000011 00000111 00001001</td>
<td>00010010 00010001 00001001 00000000</td>
<td>00000011 00000011 00000011 00000011</td>
</tr>
<tr>
<td>00010011 00100101 00000011 00000011</td>
<td>00010001 00000101 00000101 00000101</td>
<td>00100010 00001001 00001001 00001001</td>
<td>01000011 01000011 01000011 01000011</td>
</tr>
<tr>
<td>00001001 00010011 00000111 00000000</td>
<td>00001101 00001100 00001100 00001100</td>
<td>00000000 00000000 00000000 00000000</td>
<td>00001000 00001000 00001000 00001000</td>
</tr>
<tr>
<td>00001111 00001101 00000111 00000111</td>
<td>00001100 00000000 00000000 00000000</td>
<td>01000010 00100001 00100001 00100001</td>
<td>00010001 00010001 00010001 00010001</td>
</tr>
<tr>
<td>00000000 00001100 00000111 00000111</td>
<td>00000000 00000000 00000000 00000000</td>
<td>00100001 00100001 00100001 00100001</td>
<td>00001001 00001001 00001001 00001001</td>
</tr>
<tr>
<td>00000100 00000000 00000000 00000000</td>
<td>00000000 00000000 00000000 00000000</td>
<td>01000010 00100001 00100001 00100001</td>
<td>00001001 00001001 00001001 00001001</td>
</tr>
<tr>
<td>01000011 00000001 00000000 00000000</td>
<td>00000000 00000000 00000000 00000000</td>
<td>00100001 00100001 00100001 00100001</td>
<td>00001001 00001001 00001001 00001001</td>
</tr>
<tr>
<td>10000101 01000111 00000000 00000000</td>
<td>00000000 00000000 00000000 00000000</td>
<td>00100001 00100001 00100001 00100001</td>
<td>00001001 00001001 00001001 00001001</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Payload: 0x000F0000</th>
<th>Payload: 0x00F00000</th>
<th>Payload: 0xF0000000</th>
<th>Payload: 0xF0000000</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000011 00000011 00100001 10001001</td>
<td>00000011 00000011 00100001 10001001</td>
<td>00000011 00000011 00100001 10001001</td>
<td>00000011 00000011 00100001 10001001</td>
</tr>
<tr>
<td>00000011 00000011 00100001 10001001</td>
<td>00000011 00000011 00100001 10001001</td>
<td>00000011 00000011 00100001 10001001</td>
<td>00000011 00000011 00100001 10001001</td>
</tr>
<tr>
<td>00000011 00000011 00100001 10001001</td>
<td>00000011 00000011 00100001 10001001</td>
<td>00000011 00000011 00100001 10001001</td>
<td>00000011 00000011 00100001 10001001</td>
</tr>
<tr>
<td>00000011 00000011 00100001 10001001</td>
<td>00000011 00000011 00100001 10001001</td>
<td>00000011 00000011 00100001 10001001</td>
<td>00000011 00000011 00100001 10001001</td>
</tr>
<tr>
<td>00000011 00000011 00100001 10001001</td>
<td>00000011 00000011 00100001 10001001</td>
<td>00000011 00000011 00100001 10001001</td>
<td>00000011 00000011 00100001 10001001</td>
</tr>
<tr>
<td>00000011 00000011 00100001 10001001</td>
<td>00000011 00000011 00100001 10001001</td>
<td>00000011 00000011 00100001 10001001</td>
<td>00000011 00000011 00100001 10001001</td>
</tr>
<tr>
<td>00000011 00000011 00100001 10001001</td>
<td>00000011 00000011 00100001 10001001</td>
<td>00000011 00000011 00100001 10001001</td>
<td>00000011 00000011 00100001 10001001</td>
</tr>
<tr>
<td>00000011 00000011 00100001 10001001</td>
<td>00000011 00000011 00100001 10001001</td>
<td>00000011 00000011 00100001 10001001</td>
<td>00000011 00000011 00100001 10001001</td>
</tr>
</tbody>
</table>
DERIVING THE INTERLEAVER, ILLUSTRATED

- spreading_factor = 8, code_rate=4/8

Payload: 0x0000000F
0100011
0010011
0001001
0000111
0000000
0000100
0000000
0100000
10000101

Payload: 0x000F0000
0000000
1100000
00100101
00010001
00001101
00001100
0000000
00100001
00010000

Payload: 0x00F00000
10000100
0000011
00000011
00000111
00000000
00100001
10000001
01000001
11010000

Payload: 0xF0000000
00000000
10000000
00100000
00001000
00000010
00000000
00010000
00001001
10000000

ALIGNING CODEWORDS

- Mapping diagonals returns bits in each code word
- However bits in each word are scrambled

- Solution: Transmit known words, read out diagonals, and look for recognizable Hamming code words
ALIGNING CODEWORDS, ILLUSTRATED

- spreading_factor = 8, code_rate=4/8

Payload: 0xDEADBEEF

<table>
<thead>
<tr>
<th>bit</th>
<th>76543210</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>00110011</td>
</tr>
<tr>
<td></td>
<td>10111110</td>
</tr>
<tr>
<td></td>
<td>11111010</td>
</tr>
<tr>
<td></td>
<td>11011101</td>
</tr>
<tr>
<td></td>
<td>10000010</td>
</tr>
<tr>
<td></td>
<td>10000000</td>
</tr>
<tr>
<td></td>
<td>10000000</td>
</tr>
<tr>
<td></td>
<td>10000010</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Top</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ALIGNING CODEWORDS, ILLUSTRATED

- spreading_factor = 8, code_rate=4/8

Payload: 0xDEADBEEF

<table>
<thead>
<tr>
<th>bit</th>
<th>76543210</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>00110011</td>
</tr>
<tr>
<td></td>
<td>01111110</td>
</tr>
<tr>
<td></td>
<td>11111010</td>
</tr>
<tr>
<td></td>
<td>11011101</td>
</tr>
<tr>
<td></td>
<td>10000010</td>
</tr>
<tr>
<td></td>
<td>10000111</td>
</tr>
<tr>
<td></td>
<td>11000000</td>
</tr>
<tr>
<td></td>
<td>10000010</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Top</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Bot</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ALIGNING CODEWORDS, ILLUSTRATED

- spreading_factor = 8, code_rate=4/8

Payload: 0xDEADBEEF

<table>
<thead>
<tr>
<th>bit</th>
<th>76543210</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0110011</td>
</tr>
<tr>
<td>1</td>
<td>01111110</td>
</tr>
<tr>
<td>111</td>
<td>111010</td>
</tr>
<tr>
<td>1101</td>
<td>1101</td>
</tr>
<tr>
<td>1000</td>
<td>0101</td>
</tr>
<tr>
<td>1000</td>
<td>0111</td>
</tr>
<tr>
<td>1100</td>
<td>0000</td>
</tr>
<tr>
<td>1000</td>
<td>0101</td>
</tr>
</tbody>
</table>

```
Top

D 1 0 1 0 0 0 0 0 1
E 0 1 1 1 1 0 1 0 0
A
D
B
E
E
F
```
### Aligning Codewords, Illustrated

- spreading_factor = 8, code_rate=4/8

**Payload:** 0xDEADBEEF

<table>
<thead>
<tr>
<th>bit</th>
<th>76543210</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>00110011</td>
</tr>
<tr>
<td></td>
<td>10111101</td>
</tr>
<tr>
<td></td>
<td>11111010</td>
</tr>
<tr>
<td></td>
<td>10111101</td>
</tr>
<tr>
<td></td>
<td>10000010</td>
</tr>
<tr>
<td></td>
<td>10000111</td>
</tr>
<tr>
<td></td>
<td>11000000</td>
</tr>
<tr>
<td></td>
<td>10000010</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Top</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**ALIGNING CODEWORDS, ILLUSTRATED**

- spreading_factor = 8, code_rate=4/8

**Payload: **0xDEADBEEF

<table>
<thead>
<tr>
<th>bit</th>
<th>76543210</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>00110011</td>
</tr>
<tr>
<td></td>
<td>10111110</td>
</tr>
<tr>
<td></td>
<td>11110110</td>
</tr>
<tr>
<td></td>
<td>11011101</td>
</tr>
<tr>
<td></td>
<td>10000010</td>
</tr>
<tr>
<td></td>
<td>10000111</td>
</tr>
<tr>
<td></td>
<td>11000000</td>
</tr>
<tr>
<td></td>
<td>10000010</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Top</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Bot</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>10100000000001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>01110101000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>01011100000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>10110000000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ALIGNING CODEWORDS, ILLUSTRATED

- spreading_factor = 8, code_rate=4/8

Payload: 0xDEADBEF

<table>
<thead>
<tr>
<th>bit</th>
<th>76543210</th>
</tr>
</thead>
<tbody>
<tr>
<td>0011</td>
<td>10011</td>
</tr>
<tr>
<td>1011</td>
<td>1110</td>
</tr>
<tr>
<td>1111</td>
<td>010</td>
</tr>
<tr>
<td>1101</td>
<td>1101</td>
</tr>
<tr>
<td>1000</td>
<td>010</td>
</tr>
<tr>
<td>1010</td>
<td>11111</td>
</tr>
<tr>
<td>1100</td>
<td>00000</td>
</tr>
<tr>
<td>1000</td>
<td>0010</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Top</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td></td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**ALIGNING CODEWORDS, ILLUSTRATED**

- spreading\_factor = 8, code\_rate=4/8

Payload: 0xDEADBEEF

<table>
<thead>
<tr>
<th>bit 76543210</th>
<th>00110011</th>
<th>10111110</th>
<th>11111010</th>
<th>11011101</th>
<th>00000101</th>
<th>10000111</th>
<th>11000000</th>
<th>10000010</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Top</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ALIGNING CODEWORDS, ILLUSTRATED

- spreading_factor = 8, code_rate=4/8

Payload: 0xDEADBEEF

<table>
<thead>
<tr>
<th>bit</th>
<th>76543210</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>01100111</td>
</tr>
<tr>
<td>1</td>
<td>11111110</td>
</tr>
<tr>
<td>1</td>
<td>11011010</td>
</tr>
<tr>
<td>0</td>
<td>10000101</td>
</tr>
<tr>
<td>0</td>
<td>10000111</td>
</tr>
<tr>
<td>0</td>
<td>11000000</td>
</tr>
<tr>
<td>1</td>
<td>00000100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Top</th>
<th></th>
<th></th>
<th></th>
<th>Bot</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>1 0 1 0 0 0 0 0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>0 1 1 1 0 1 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0 1 0 1 1 1 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>1 0 1 1 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>1 1 0 0 0 0 1 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>0 1 1 1 0 1 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>0 1 1 1 0 1 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ALIGNING CODEWORDS, ILLUSTRATED

- spreading_factor = 8, code_rate=4/8

Payload: 0xDEADBEEF

<table>
<thead>
<tr>
<th>bit</th>
<th>76543210</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>10011</td>
</tr>
<tr>
<td>101</td>
<td>11110</td>
</tr>
<tr>
<td>111</td>
<td>1010</td>
</tr>
<tr>
<td>110</td>
<td>1101</td>
</tr>
<tr>
<td>000</td>
<td>010</td>
</tr>
<tr>
<td>100</td>
<td>001</td>
</tr>
<tr>
<td>110</td>
<td>000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Top</th>
<th></th>
<th>Bot</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>1 0 1 0 0 0 0 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>0 1 1 1 0 1 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0 1 0 1 1 0 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>1 0 1 1 0 0 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>1 1 0 0 0 0 1 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>0 1 1 1 0 1 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>0 1 1 1 0 1 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>1 1 1 1 1 1 1 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**ALIGNING CODEWORDS, ILLUSTRATED**

- spreading_factor = 8, code_rate=4/8

Payload: **0xDEADBEEF**

<table>
<thead>
<tr>
<th>bit</th>
<th>76543210</th>
</tr>
</thead>
<tbody>
<tr>
<td>00110011</td>
<td></td>
</tr>
<tr>
<td>10111110</td>
<td></td>
</tr>
<tr>
<td>11111010</td>
<td></td>
</tr>
<tr>
<td>11011101</td>
<td></td>
</tr>
<tr>
<td>10000010</td>
<td></td>
</tr>
<tr>
<td>10000111</td>
<td></td>
</tr>
<tr>
<td>11000000</td>
<td></td>
</tr>
<tr>
<td>10000010</td>
<td></td>
</tr>
</tbody>
</table>

|   | Top |  |  |  |  |  |  |  | Bot |
|---|-----|---|---|---|---|---|---|-----|
| D | 1   | 0 | 1 | 0 | 0 | 0 | 0 | 1   |
| E | 0   | 1 | 1 | 1 | 0 | 1 | 0 | 0   |
| A | 0   | 1 | 0 | 1 | 1 | 0 | 0 | 0   |
| D | 1   | 0 | 1 | 1 | 0 | 0 | 0 | 0   |
| B | 1   | 1 | 0 | 0 | 0 | 0 | 1 | 0   |
| E | 0   | 1 | 1 | 1 | 0 | 1 | 0 | 0   |
| E | 0   | 1 | 1 | 1 | 0 | 1 | 0 | 0   |
| F | 1   | 1 | 1 | 1 | 1 | 1 | 1 | 1   |
ALIGNING CODEWORDS, ILLUSTRATED

- spreading_factor = 8, code_rate=4/8

**Payload: 0xDEADBEEF**

<table>
<thead>
<tr>
<th>Bin</th>
<th>(8,4) Parity</th>
<th>Top</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Bot</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xD</td>
<td>1101</td>
<td>D</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0xE</td>
<td>1110</td>
<td>E</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0xA</td>
<td>1010</td>
<td>A</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0xD</td>
<td>1101</td>
<td>D</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0xB</td>
<td>1011</td>
<td>B</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0xE</td>
<td>1110</td>
<td>E</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0xE</td>
<td>1110</td>
<td>E</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0xF</td>
<td>1111</td>
<td>F</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
**ALIGNING CODEWORDS, ILLUSTRATED**

- spreading\_factor = 8, code\_rate=4/8

**Payload:** 0xDEADBEEF

<table>
<thead>
<tr>
<th></th>
<th>Bin</th>
<th>(8,4) Parity</th>
<th></th>
<th>Bot</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Top</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xD</td>
<td>1101</td>
<td>1000</td>
<td></td>
<td>D</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0xE</td>
<td>1110</td>
<td>0001</td>
<td></td>
<td>E</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0xA</td>
<td>1010</td>
<td>1010</td>
<td></td>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0xD</td>
<td>1101</td>
<td>1000</td>
<td></td>
<td>D</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0xB</td>
<td>1011</td>
<td>0100</td>
<td></td>
<td>B</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0xE</td>
<td>1110</td>
<td>0001</td>
<td></td>
<td>E</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0xE</td>
<td>1110</td>
<td>0001</td>
<td></td>
<td>E</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0xF</td>
<td>1111</td>
<td>1111</td>
<td></td>
<td>F</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
ALIGNING CODEWORDS, ILLUSTRATED

- spreading_factor = 8, code_rate=4/8

Payload: 0xDEADBEEF

<table>
<thead>
<tr>
<th>Bin</th>
<th>(8,4) Parity</th>
<th>Bot</th>
<th>Top</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xD</td>
<td>1101</td>
<td>D</td>
<td>1</td>
</tr>
<tr>
<td>0xE</td>
<td>1110</td>
<td>E</td>
<td>0</td>
</tr>
<tr>
<td>0xA</td>
<td>1010</td>
<td>A</td>
<td>0</td>
</tr>
<tr>
<td>0xD</td>
<td>1101</td>
<td>D</td>
<td>1</td>
</tr>
<tr>
<td>0xB</td>
<td>1011</td>
<td>B</td>
<td>1</td>
</tr>
<tr>
<td>0xE</td>
<td>1110</td>
<td>E</td>
<td>1</td>
</tr>
<tr>
<td>0xE</td>
<td>1110</td>
<td>E</td>
<td>1</td>
</tr>
<tr>
<td>0xF</td>
<td>1111</td>
<td>F</td>
<td>1</td>
</tr>
</tbody>
</table>
ALIGNING CODEWORDS, ILLUSTRATED

- spreading_factor = 8, code_rate=4/8

Payload: 0xDEADBEEF

<table>
<thead>
<tr>
<th>Hex</th>
<th>Bin</th>
<th>(8,4) Parity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xD</td>
<td>1101</td>
<td>1000</td>
</tr>
<tr>
<td>0xE</td>
<td>1110</td>
<td>0001</td>
</tr>
<tr>
<td>0xA</td>
<td>1010</td>
<td>1010</td>
</tr>
<tr>
<td>0xD</td>
<td>1101</td>
<td>1000</td>
</tr>
<tr>
<td>0xB</td>
<td>1011</td>
<td>0100</td>
</tr>
<tr>
<td>0xE</td>
<td>1110</td>
<td>0001</td>
</tr>
<tr>
<td>0xE</td>
<td>1110</td>
<td>0001</td>
</tr>
<tr>
<td>0xF</td>
<td>1111</td>
<td>1111</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>p1</th>
<th>p2</th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>d4</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bot</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Top</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
### Aligning Codewords, Illustrated

- spreading_factor = 8, code_rate=4/8

Payload: **0xDEADBEEF**

<table>
<thead>
<tr>
<th>Bin</th>
<th>(8,4) Parity</th>
<th>p1</th>
<th>p2</th>
<th>p4</th>
<th>p3</th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>d4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xD</td>
<td>1101</td>
<td>1000</td>
<td>D</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0xE</td>
<td>1110</td>
<td>0001</td>
<td>E</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0xA</td>
<td>1010</td>
<td>1010</td>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0xD</td>
<td>1101</td>
<td>1000</td>
<td>D</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0xB</td>
<td>1011</td>
<td>0100</td>
<td>B</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0xE</td>
<td>1110</td>
<td>0001</td>
<td>E</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0xE</td>
<td>1110</td>
<td>0001</td>
<td>E</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0xF</td>
<td>1111</td>
<td>1111</td>
<td>F</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
ALIGNING CODEWORDS, ILLUSTRATED

- spreading_factor = 8, code_rate=4/8

Payload: **0xDEADBEEF**

<table>
<thead>
<tr>
<th></th>
<th>Bin</th>
<th>(8,4) Parity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xD</td>
<td>1101</td>
<td>1000</td>
</tr>
<tr>
<td>0xE</td>
<td>1110</td>
<td>0001</td>
</tr>
<tr>
<td>0xA</td>
<td>1010</td>
<td>1010</td>
</tr>
<tr>
<td>0xD</td>
<td>1101</td>
<td>1000</td>
</tr>
<tr>
<td>0xB</td>
<td>1011</td>
<td>0100</td>
</tr>
<tr>
<td>0xE</td>
<td>1110</td>
<td>0001</td>
</tr>
<tr>
<td>0xF</td>
<td>1111</td>
<td>1111</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>p1</th>
<th>p2</th>
<th>p4</th>
<th>p3</th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>d4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bot</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0xD</td>
<td>D</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0xE</td>
<td>E</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0xA</td>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0xD</td>
<td>D</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0xB</td>
<td>B</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0xE</td>
<td>E</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0xE</td>
<td>E</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0xF</td>
<td>F</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
ALMOST THERE
**APPLY FORWARD ERROR CORRECTION**

- spreading\_factor = 8, code\_rate=4/8

Payload: 0xDEADBEEF

<table>
<thead>
<tr>
<th></th>
<th>Bin</th>
<th>(8,4) Parity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xD</td>
<td>1101</td>
<td>1000</td>
</tr>
<tr>
<td>0xE</td>
<td>1110</td>
<td>0001</td>
</tr>
<tr>
<td>0xA</td>
<td>1010</td>
<td>1010</td>
</tr>
<tr>
<td>0xD</td>
<td>1101</td>
<td>1000</td>
</tr>
<tr>
<td>0xB</td>
<td>1011</td>
<td>0100</td>
</tr>
<tr>
<td>0xE</td>
<td>1110</td>
<td>0001</td>
</tr>
<tr>
<td>0xE</td>
<td>1110</td>
<td>0001</td>
</tr>
<tr>
<td>0xF</td>
<td>1111</td>
<td>1111</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>p1</th>
<th>p2</th>
<th>p4</th>
<th>p3</th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>d4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bot</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
APPLY FORWARD ERROR CORRECTION

- spreading_factor = 8, code_rate=4/8

Payload: 0xDEADBEEF

<table>
<thead>
<tr>
<th>Bin</th>
<th>(8,4) Parity</th>
<th>p1</th>
<th>p2</th>
<th>p4</th>
<th>p3</th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>d4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xD</td>
<td>1101</td>
<td>1000</td>
<td>D</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0xE</td>
<td>1110</td>
<td>0001</td>
<td>E</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0xA</td>
<td>1010</td>
<td>1010</td>
<td>A</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0xD</td>
<td>1101</td>
<td>1000</td>
<td>D</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0xB</td>
<td>1011</td>
<td>0100</td>
<td>B</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0xE</td>
<td>1110</td>
<td>0001</td>
<td>E</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0xE</td>
<td>1110</td>
<td>0001</td>
<td>E</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0xF</td>
<td>1111</td>
<td>1111</td>
<td>F</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
APPLY FORWARD ERROR CORRECTION

- spreading_factor = 8, code_rate = 4/8

Payload: 0xDEADBEEF
TO CONCLUDE

- LPWANs have momentum and are proliferating
- RF stacks are becoming more diverse
  - Wireless is not just WiFi anymore
- Shown how to go from obscure RF → bits
ACKNOWLEDGEMENTS

- Balint Seeber, Bastille Threat Research Team
- hexameron and Bertrik Sikken, open source contributors
- Johan Stokking, The Things Network
- Jailbreak for hosting!
THANKS

matt@Bastille.net
@embeddedsec
QUESTIONS?

matt@Bastille.net
@embeddedsec