First Year Review

Information Leakage in Sensor Network Traffic

Yan Yan

Review

- May 2014 – Decmber 2014
  - Background of sensor network
    - Contiki
    - 6lowPAN, CoAP, DTLS
    - tinyDTLS
  - Existing attacks:
    - Content length fingerprinting
      - Generally applicable
      - Application specific
      - Low accuracy in some circumstances
      - Affected by dynamic contents like ads
    - Mutual Information analysis
      - Good coverage
      - Requires high computational power

Review

- More existing attacks:
  - Compression ratio attack: BREACH, CRIME
    - Efficient
    - Prevented by disabling compression
  - Padding oracle and Lucky 13
    - Efficient
    - Countermeasures implemented
    - Requires specific setup (Lucky 13)
    - Latency sensitive (Lucky 13)

Review(cont)

- Reflection
  - Applications are mostly experimental
  - Not much security took into concern
- Plan
  - Start with some simple applications
  - Demonstrate the potential of similar attacks

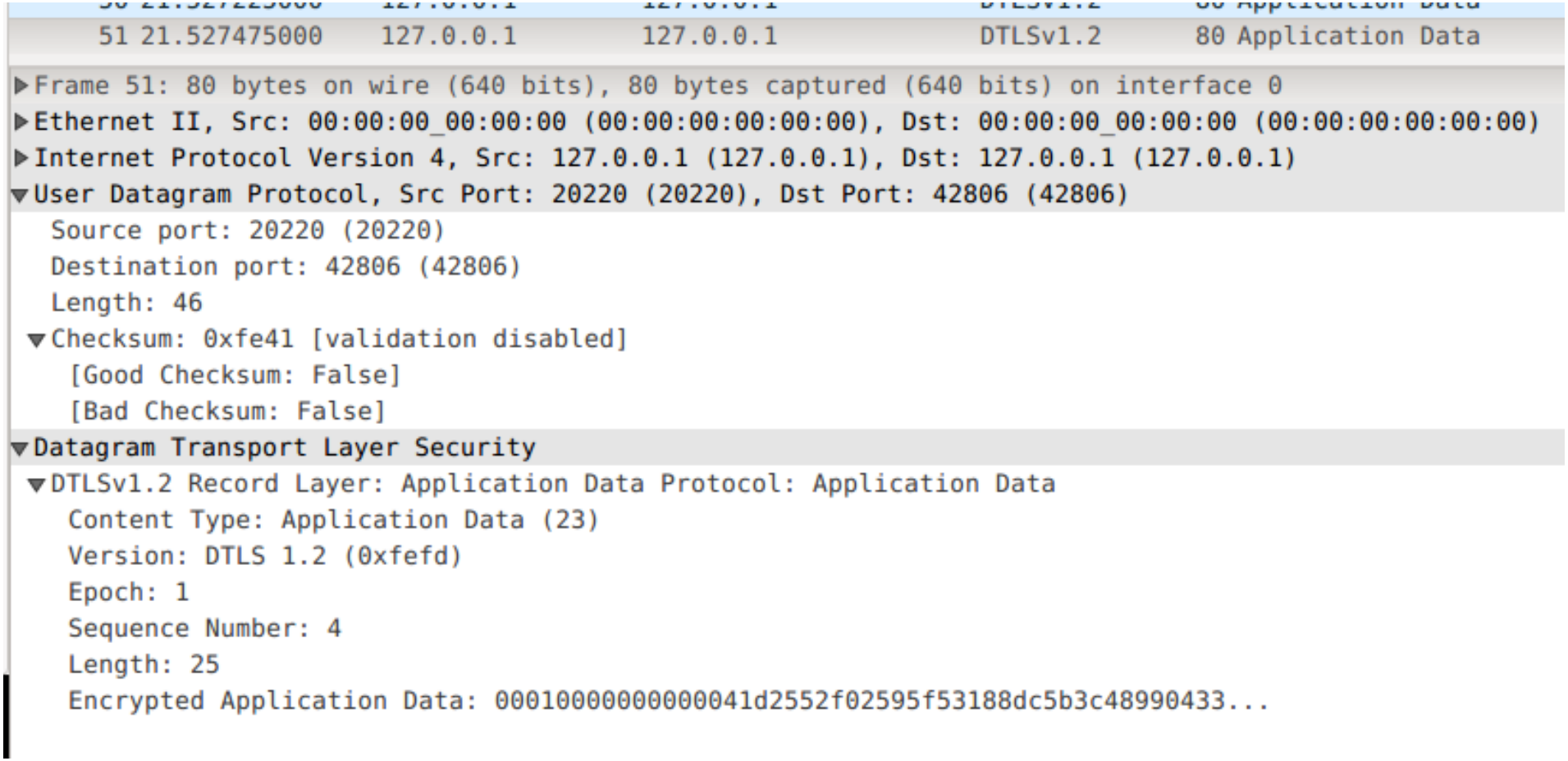
Recent Work

- January 2015 -
  - Developed two toy applications:
    - Odd or Even
    - Leaky Coffee
- Traffic analysis
  - Timestamp*
  - Fields
  - Length

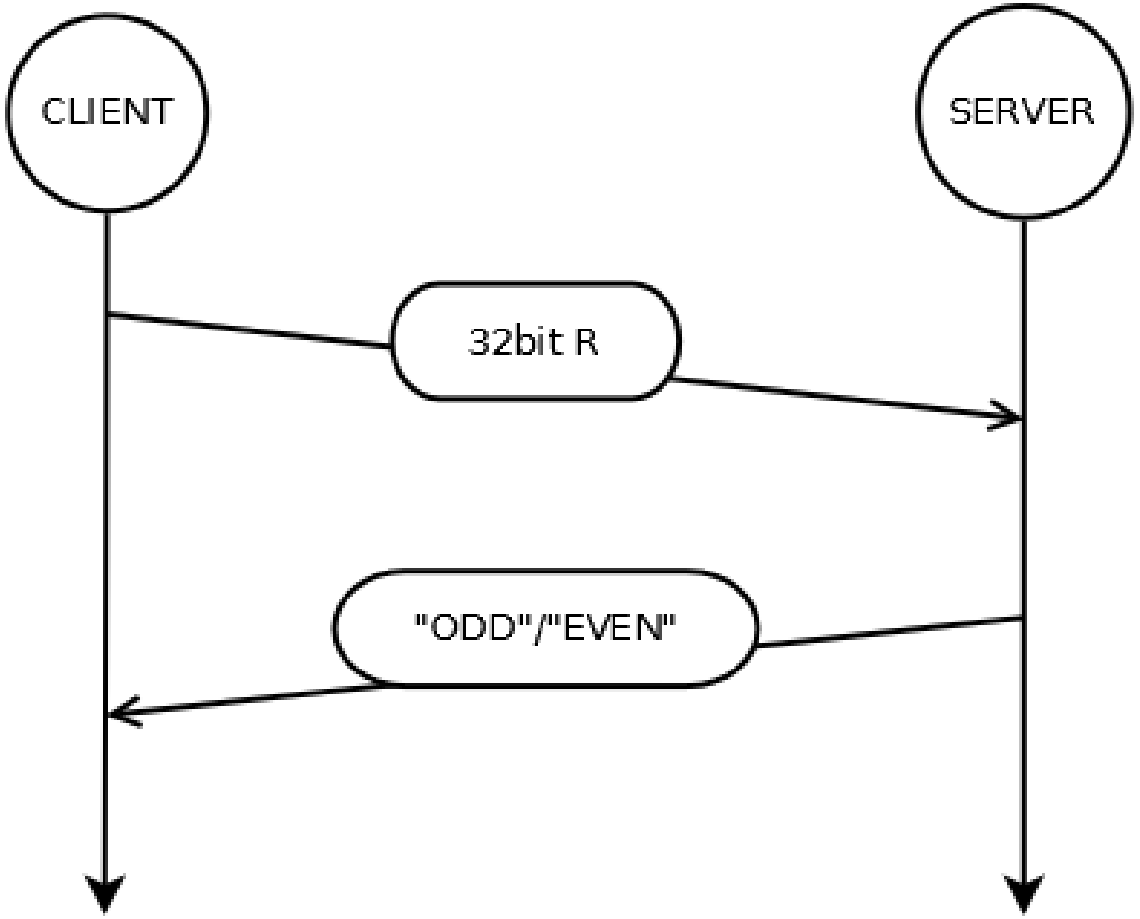
Environment

- Ubuntu 14.04
- Locallink
- Tinydtls-0.8.1
  - Pre-shared key
  - TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8
    - There is no padding!
    - Plaintext Length = DTLS length field - 17

Packet


51 21.527475000 127.0.0.1 127.0.0.1 DTLSv1.2 80 Application Data
Frame 51: 80 bytes on wire (640 bits), 80 bytes captured (640 bits) on interface 0
Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
User Datagram Protocol, Src Port: 20220 (20220), Dst Port: 42806 (42806)
Source port: 20220 (20220)
Destination port: 42806 (42806)
Length: 46
Checksum: 0xfe41 [validation disabled]
[Good Checksum: False]
[Bad Checksum: False]
Datagram Transport Layer Security
DTLSv1.2 Record Layer: Application Data Protocol: Application Data
Content Type: Application Data (23)
Version: DTLS 1.2 (0xfefd)
Epoch: 1
Sequence Number: 4
Length: 25
Encrypted Application Data: 00010000000000041d2552f02595f53188dc5b3c48990433...

Odd or Even

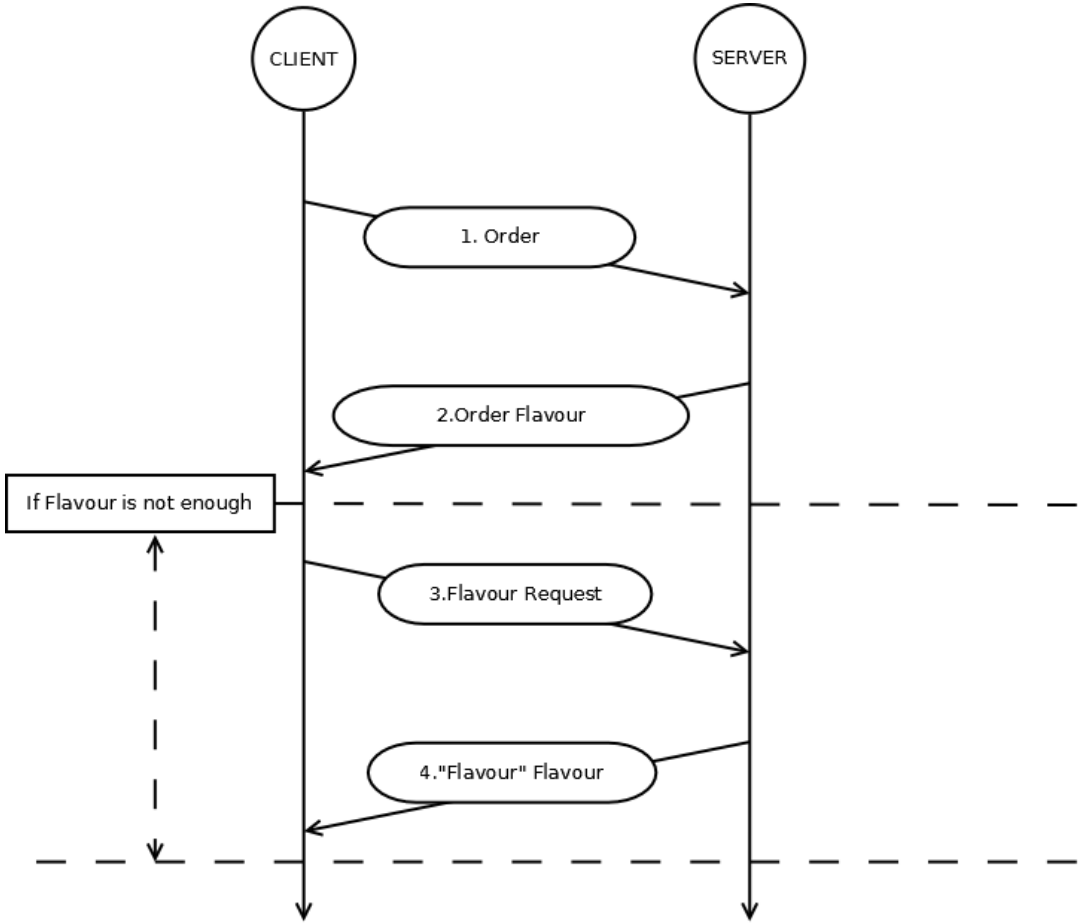


Odd or Even

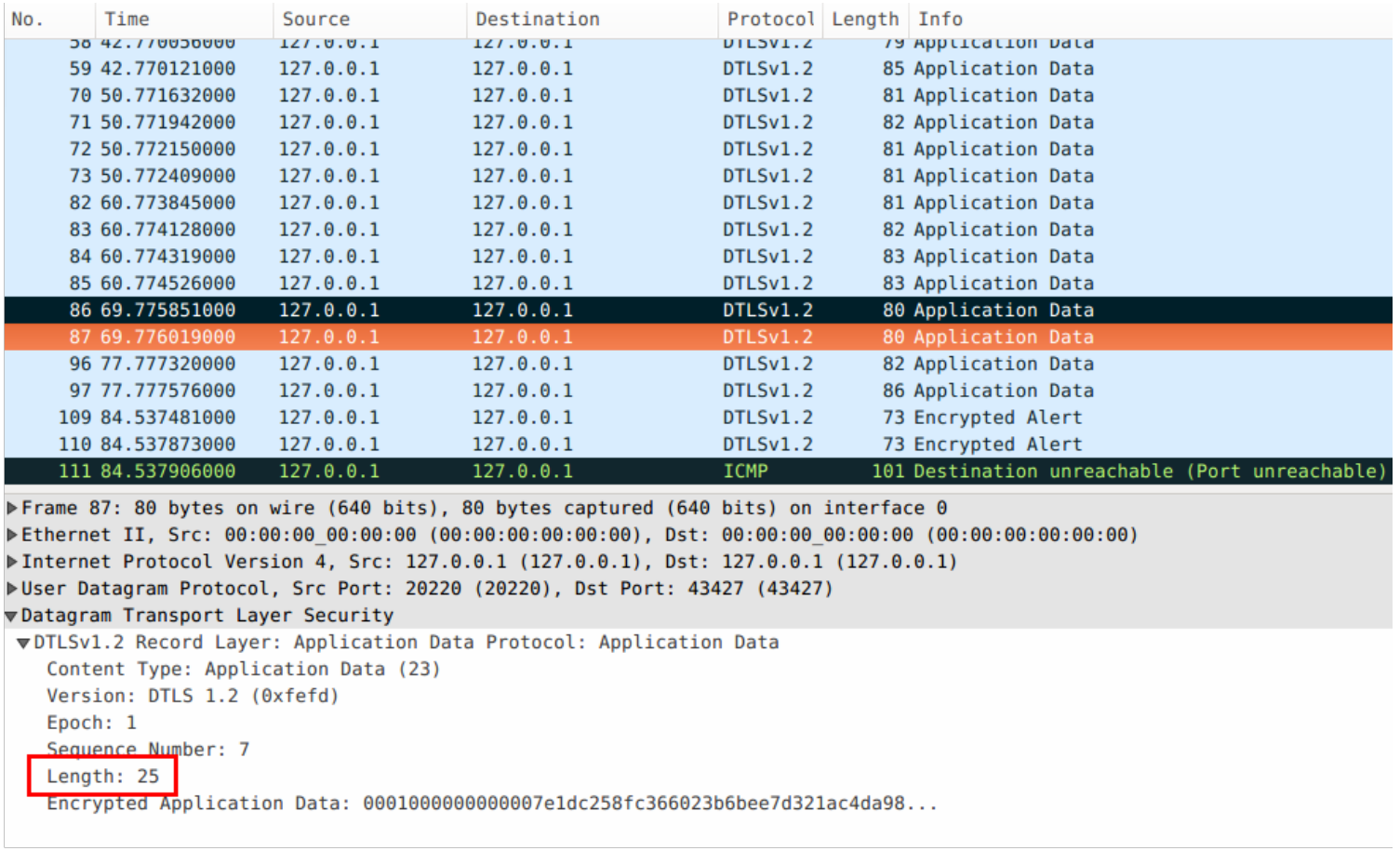
- No padding,
- “EVEN” is 1 byte longer than “ODD”
- Plaintext revealed by length!

- But all other fields seemingly leaks nothing...

Leaky Coffee



Leaky Coffee



No.	Time	Source	Destination	Protocol	Length	Info
58	42.770056000	127.0.0.1	127.0.0.1	DTLSv1.2	79	Application Data
59	42.770121000	127.0.0.1	127.0.0.1	DTLSv1.2	85	Application Data
70	50.771632000	127.0.0.1	127.0.0.1	DTLSv1.2	81	Application Data
71	50.771942000	127.0.0.1	127.0.0.1	DTLSv1.2	82	Application Data
72	50.772150000	127.0.0.1	127.0.0.1	DTLSv1.2	81	Application Data
73	50.772409000	127.0.0.1	127.0.0.1	DTLSv1.2	81	Application Data
82	60.773845000	127.0.0.1	127.0.0.1	DTLSv1.2	81	Application Data
83	60.774128000	127.0.0.1	127.0.0.1	DTLSv1.2	82	Application Data
84	60.774319000	127.0.0.1	127.0.0.1	DTLSv1.2	83	Application Data
85	60.774526000	127.0.0.1	127.0.0.1	DTLSv1.2	83	Application Data
86	69.775851000	127.0.0.1	127.0.0.1	DTLSv1.2	80	Application Data
87	69.776019000	127.0.0.1	127.0.0.1	DTLSv1.2	80	Application Data
96	77.777320000	127.0.0.1	127.0.0.1	DTLSv1.2	82	Application Data
97	77.777576000	127.0.0.1	127.0.0.1	DTLSv1.2	86	Application Data
109	84.537481000	127.0.0.1	127.0.0.1	DTLSv1.2	73	Encrypted Alert
110	84.537873000	127.0.0.1	127.0.0.1	DTLSv1.2	73	Encrypted Alert
111	84.537906000	127.0.0.1	127.0.0.1	ICMP	101	Destination unreachable (Port unreachable)

```
▶ Frame 87: 80 bytes on wire (640 bits), 80 bytes captured (640 bits) on interface 0
▶ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▶ Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
▶ User Datagram Protocol, Src Port: 20220 (20220), Dst Port: 43427 (43427)
▼ Datagram Transport Layer Security
  ▼ DTLSv1.2 Record Layer: Application Data Protocol: Application Data
      Content Type: Application Data (23)
      Version: DTLS 1.2 (0xfefd)
      Epoch: 1
      Sequence Number: 7
      Length: 25
      Encrypted Application Data: 0001000000000007e1dc258fc366023b6bee7d321ac4da98...
```

Leaky Coffee

- Existence of a packet:
  - A session taking place
- Timestamp:
  - Segmenting packets by session
- Length:
  - Constructing a channel to “decode” plaintext

Leaky Coffee

W(Length|*Order*)	5 bytes	8 bytes	9 bytes	Prob(*Order*)
“AMERICANO”			1	1/4
“CAPPUCINO”			1	1/4
“MOCHA”	1			1/4
“ESPRESSO”		1		1/4

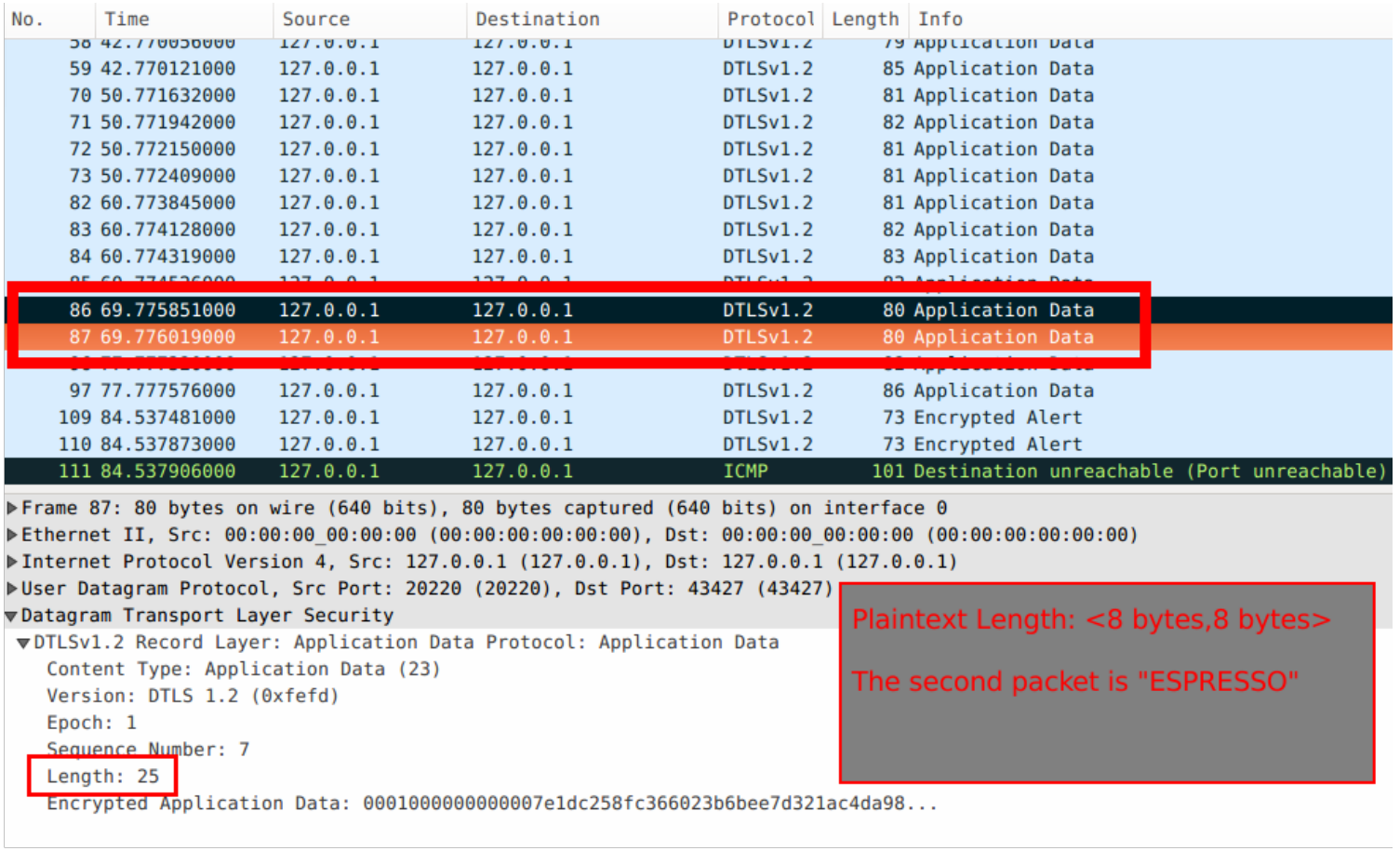
- Plaintext-Length channel for *Order*
  - Prob(*Order*) is known from the implementation
  - Revert it to construct our “decoding” channel!

Leaky Coffee

W(*Order*|Length)	“AMERICANO”	“CAPPUCINO”	“ESPRESSO”	“MOCHA”
5 bytes				1
8 bytes			1	
9 bytes	1/2	1/2		

- This channel decodes length to *Order*
- The attack can be further improved by analysing packets jointly

Leaky Coffee



Reflection

- Similar to content length fingerprinting
  - DTLS instead of TLS
  - No noise
- Constrains
  - Application too simple (low entropy plaintext)
  - Some other cipher suites have padding
  - Timestamp may be affected by underlying protocols
  - Requires pre-knowledge of plaintext distribution

Future Plan

- Wrap up the “toys”
  - Different pre-knowledge?
  - Other types of attack?
  - Countermeasure?
- Other DTLS implementation
  - e.g. PolarSSL (https://tls.mbed.org/)
- Apply the attacks on some real world traffic
  - (If we can get in touch with any...)

Other Activities

- Real World Crypto 2015
  - 06/01/2015 ~ 09/01/2015