D1155486848684498855111111111111CD5BCCF7CF83999DDD

Reverse Engineering a Mass Transit Ticketing System

Who are we?

- Damon Stacey, Dougall Johnson, Karla Burnett, Theo Julienne
- University students who do security research on the side

Disclaimer

- Research exercise
- Travelling without a valid ticket is illegal
- The views expressed here are entirely our own
- Data and algorithm have been modified

Reverse Engineering

- Figuring out how something was designed
- Hacking stuff that isn't open source

White Box Reverse Engineering

- Can look at the implementation
- Closed source software, malware
- Always possible
- Dynamic Analysis (debuggers)
- Static Analysis (disassemblers, decompilers)
- Tons of cool research
- Not the topic of this talk

Black Box Reverse Engineering

- Can use the implementation
- File formats, network protocols, magnetic stripes
- Not necessarily possible
- System analysis
- Data analysis
- The topic of this talk

Contrived Example

```
$ ./mystery
Not enough arguments.
$ ./mystery 1
Not enough arguments.
$ ./mystery 1 2
Saved to out.txt
$ ./mystery 1 2 3
Too many arguments.
```

```
100000d61:  mov    %rsp,%rbp
100000d64:  sub    $0x40,%rsp
100000d68:  mov    %edi,-0x4(%rbp)
100000d6b:  mov    %rsi,-0x10(%rbp)
100000d6f:  mov    -0x4(%rbp),%eax
100000d72:  cmp    $0x2,%eax
100000d75:  jg     100000d92
100000d77:  lea    0x162(%rip),%rax        # 100000ee0
100000d7e:  mov    %rax,%rdi
100000d81:  callq  100000e8e <_puts$stub>
100000d86:  movl   $0x1,-0x1c(%rbp)
100000d8d:  jmpq   100000e5b
100000d92:  mov    -0x4(%rbp),%eax
100000d95:  cmp    $0x3,%eax
100000d98:  jle    100000db5
100000d9a:  lea    0x155(%rip),%rax        # 100000ef6
100000da1:  mov    %rax,%rdi
100000da4:  callq  100000e8e <_puts$stub>
100000da9:  movl   $0x1,-0x1c(%rbp)
100000db0:  jmpq   100000e5b
...
100000e5b:  mov    -0x1c(%rbp),%eax
100000e5e:  mov    %eax,-0x18(%rbp)
100000e61:  mov    -0x18(%rbp),%eax
100000e64:  mov    %eax,-0x14(%rbp)
100000e67:  mov    -0x14(%rbp),%eax
100000e6a:  add    $0x40,%rsp
100000e6e:  pop    %rbp
100000e6f:  retq

100000ee0:  4e6f7420 656e6f75 67682061 7267756d  Not enough argum
100000ef0:  656e7473 2e00546f 6f206d61 6e792061  ents..Too many a
100000f00:  7267756d 656e7473 2e006f75 742e7478  rguments..out.tx
```

Case Study

- Mass Transit Ticketing System
- Magnetic stripe tickets

case study

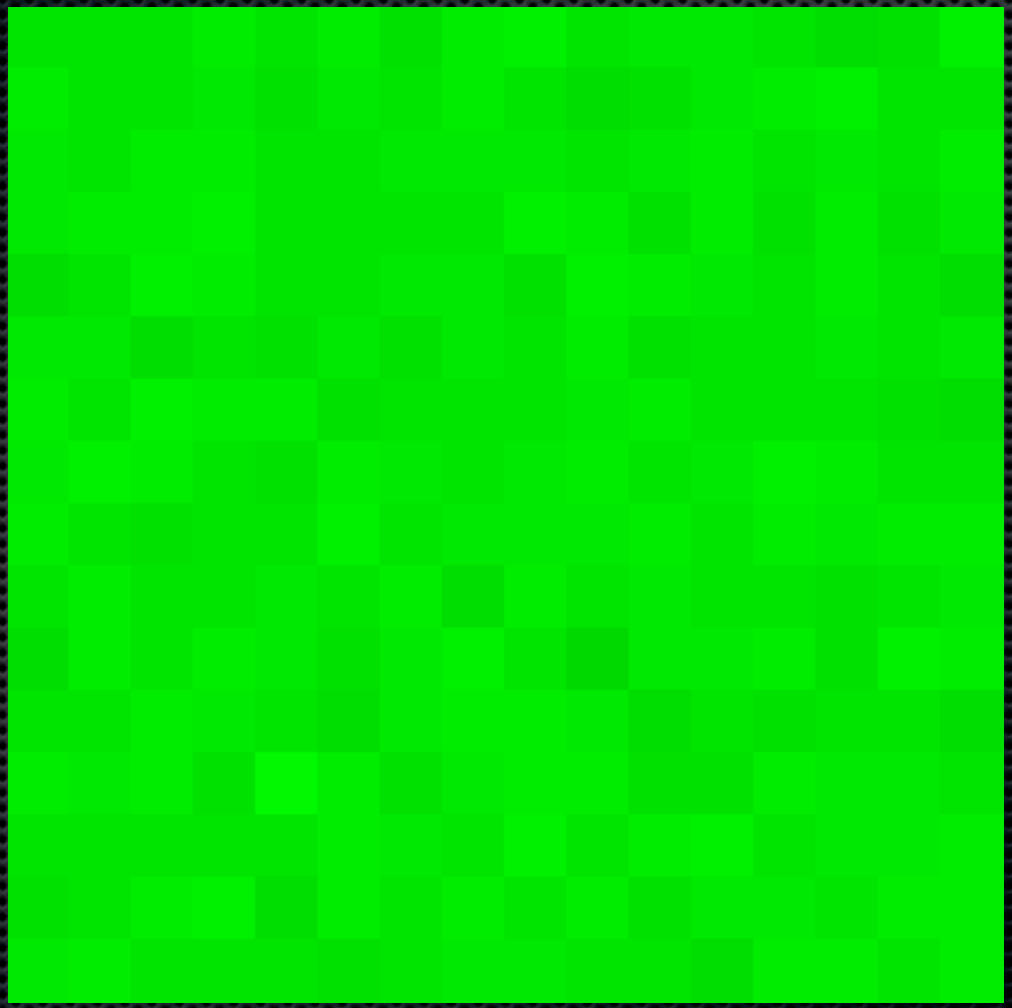
Which Tickets

- Need to figure out how they work
- How much data do we need?
- Which data do we need?
- Large dataset for analysis
- Specially-purchased data to answer specific questions

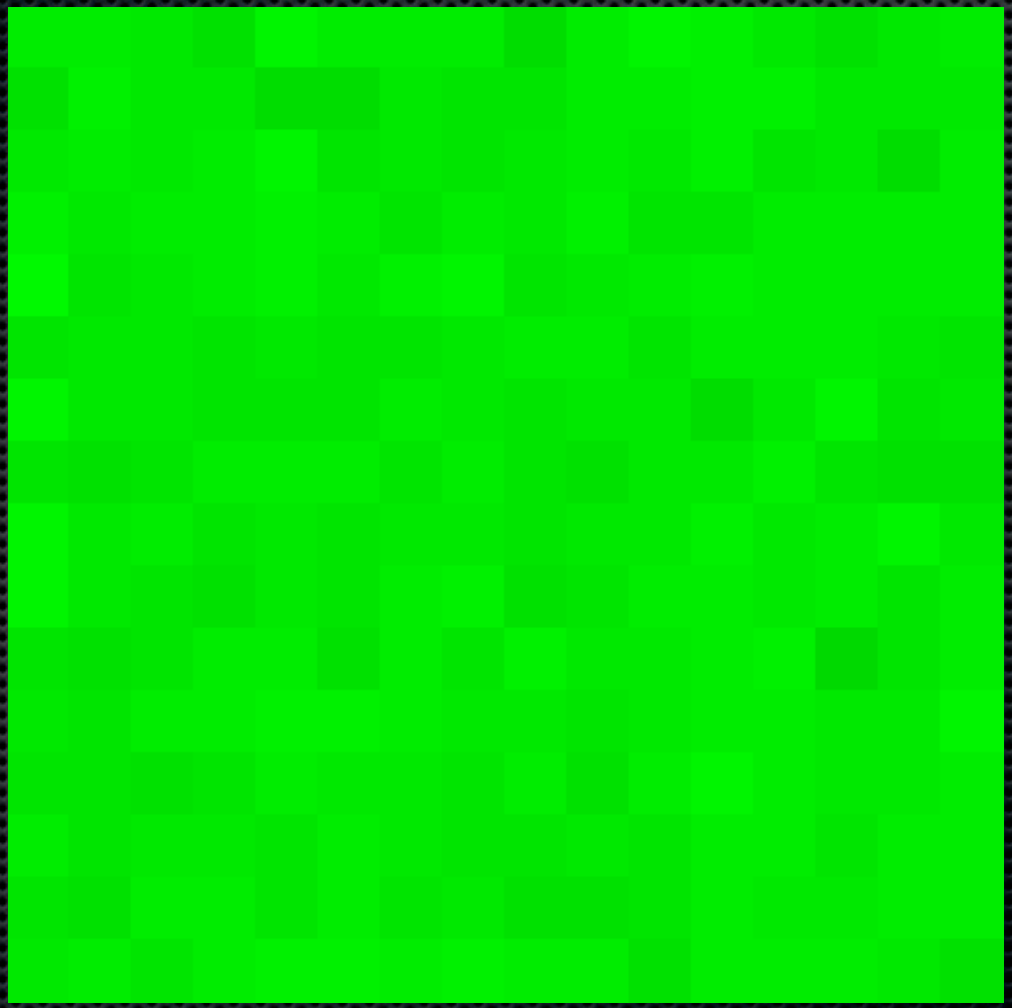
Data Analysis

- What do you know about the data?
- Look for correlations
- Look at common stuff first
- How would you encode the data?

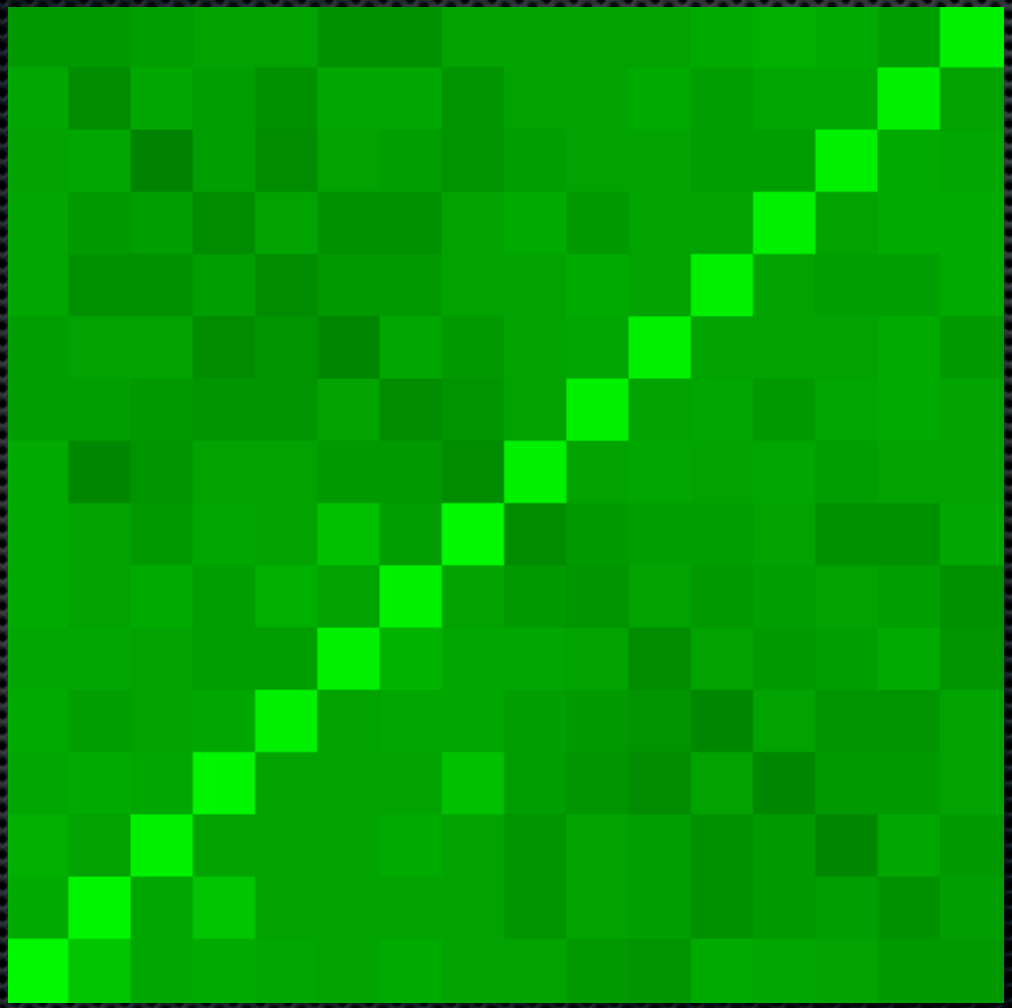
Entropy - Random



Entropy - AES



Entropy - Case Study



case study

Encryption

- Modern cryptography looks like random data
- Patterns indicate weaker cryptography
- Frequency analysis
- Entropy and compressibility

case study

General Observations

- Must encode validity dates, origin, destination, etc.
- Physical ticket ID encodes station and machine ID

Finding Patterns

- Specially purchased, sequential tickets are significantly different

```
D11554868486844988551111111111CD5BCCF7CF83999DDD - 17:57:56
D667730B030B033400776666666666157C11DF10D39998AD - 17:57:59
DBBAAED6DED6DEE9DDAABBBBBBBBBBC8A1CC02C94A0001ED - 17:58:02
```

Finding Patterns

- Clearly not random

```
D11554868486844988551111111111CD5BCCF7CF83999DDD - 17:57:56
D667730B030B033400776666666666157C11DF10D39998AD - 17:57:59
DBBAAED6DED6DEE9DDAABBBBBBBBBBC8A1CC02C94A0001ED - 17:58:02
```

Finding Patterns

- XOR each nibble with ‘1’

D1155486848684498855**11111111111**CD5BCCF7CF83999DDD - 17:57:56

C0044597959795589944**00000000000**DC4ADDE6DE92888CCC

Finding Patterns

case study

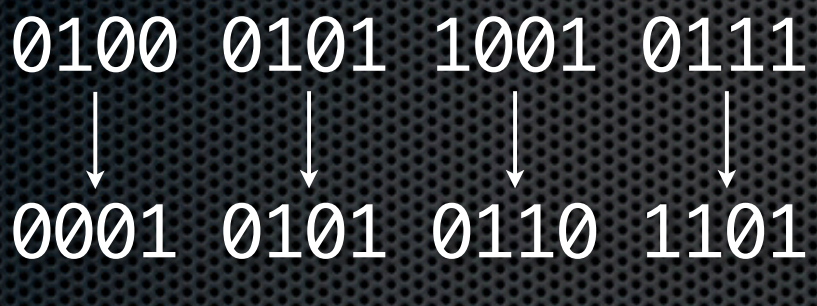
- Data after XOR

```
D1155486848684498855111111111111CD5BCCF7CF83999DDD - 17:57:56
D667730B030B03340077666666666666157C11DF10D39998AD - 17:57:59
DBBAAED6DED6DEE9DDAABBBBBBBBBBBC8A1CC02C94A0001ED - 17:58:02
```

```
C00445979597955899440000000000DC4ADDE6DE92888CCC
B001156D656D6552661100000000000731A77B976B5FFFECB
6001156D656D6552661100000000000731A77B972F1BBBA56
```

Finding Patterns

case study



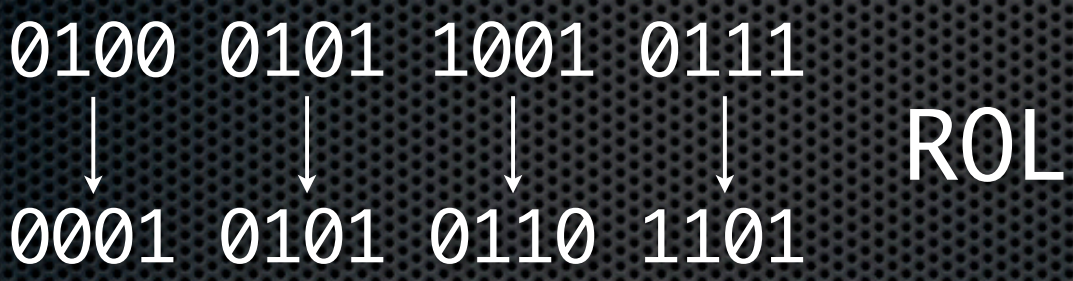
C004459795979558994400000000000DC4ADDE6DE92888CCC

B001156D656D655266110000000000731A77B976B5FFFECB

Finding Patterns

case study

- Left rotation of each nibble by 2
- ((nibble << 2) | (nibble >> 2)) & 0xF



C00445979597957589944000000000000DC4ADDE6DE92888CCC

B001156D656D655266110000000000000731A77B976B5FFFECB

Finding Patterns

case study

- First ticket with bits ROLed

```
C00445979597955899440000000000DC4ADDE6DE92888CCC
```

```
3001156D656D6552661100000000000731A77B97B68222333
```

case study

Finding Patterns

- Data after XOR (with first ticket ROLed)

```
D1155486848684498855111111111111CD5BCCF7CF83999DDD - 17:57:56
D667730B030B03340077666666666666157C11DF10D39998AD - 17:57:59
DBBAAED6DED6DEE9DDAABBBBBBBBBBBC8A1CC02C94A0001ED - 17:58:02
```

```
3001156D656D6552661100000000000731A77B97B68222333
B001156D656D6552661100000000000731A77B976B5FFFECB
6001156D656D6552661100000000000731A77B972F1BBBA56
```

Finding More Patterns

- Worked on those 3 tickets
- Failed on all other tickets
- Try other nibbles for XOR: 4, 8, 15, 23 then 42

Small vs Large Data Sets

- Small known data has little variation
  - Same values, more correlations
  - Great for making data look the same
- Large dataset will have much more variation
  - More values, less correlations
- Need to move to a larger data set

case study

Data Gathering

- Magnetic stripe tickets
- Ticket vending machines
- Cost a lot of money to get a good sample
- Once used, they're basically free

Ticket Database

case study

- About a thousand tickets
- Efficient data digitisation
- Need magnetic stripe data and printed data
- Took an afternoon

LEGO DEMO

case study

Automation

- Don’t go through massive datasets by hand
- Automated search for correlations
- Automated search for possible encodings of known data

Search Scripts

case study

- Group full data set into known field values
  - Origin station from physical ticket
- Easy with decrypted data
  - Our data only partially decoded
- Weak encryption
  - Brute force

case study

Finding the Origin

- Find all nibbles that are the same between all tickets with same origin
- Iterate through all nibbles as the XOR key
- Output in a visual way

Analyse Results

```
11 PALL MALL      ________________________________________________
11 OXFORD ST      _____0__________________________________________
11 KINGS CROSS    ________________________________________________
11 PICCADILLY     B_______________________________________________
11 FLEET ST       ________________________________________________
11 BOND ST        B_________44F2246AA8A___________________________
...
12 PALL MALL      __________0E6___________________________________
12 OXFORD ST      _____0____0744__________________________________
12 KINGS CROSS    __________061___________________________________
12 PICCADILLY     B_________0755__________________________________
12 FLEET ST       __________19A___________________________________
12 BOND ST        B_________0B602EECE_____________________________
...
13 PALL MALL      ___________E6___________________________________
13 OXFORD ST      _____0_____744__________________________________
13 KINGS CROSS    ___________61___________________________________
13 PICCADILLY     B__________755__________________________________
13 FLEET ST       ___________8B___________________________________
13 BOND ST        B__________B602EECE_____________________________
```

case study

Finding More Fields

- Can now decode the ticket origin and destination stations
  - Origin and destination codes different
    - ROLing some nibbles corrects this
  - Data is still not decrypted completely
- Next want to find date and time

case study

Date Field Location

- Origin Station vs Date
  - Downside: Less tickets with same date values
- Analyse data from any date with > 2 samples
- Find common nibbles with 95% accuracy

Date Field Location

case study

```
...
8 2011-06-16   _________322F_____________________________
8 2011-06-28   _________326A5____________________________
8 2011-06-29   B________3269_____________________________
8 2011-06-30   _________3268_____________________________
8 2011-07-01   _________326F2____________________________
8 2011-07-02   _________326E_____________________________
8 2011-07-03   _________326D738AC8_______________________
...
```

case study

Date Field Encoding

- Origin Station vs Date
  - Upside: Better guess at encoding
  - Probably field incrementing each day
  - Pick a start date, SQL server uses 1900-01-01
- Use all samples this time
- Correlate and visualise

Date Field Encoding

case study

```
Date, Days since 1900, field values
...
2011-06-16 40708 {'322f': 2}
2011-06-17 40709 {'322c': 1}
2011-06-18 40710
2011-06-19 40711
2011-06-20 40712
2011-06-21 40713
2011-06-22 40714
2011-06-23 40715 {'3226': 1}
2011-06-24 40716
2011-06-25 40717 {'3224': 1}
2011-06-26 40718
2011-06-27 40719
2011-06-28 40720 {'326a': 2}
2011-06-29 40721 {'3269': 3}
2011-06-30 40722 {'3268': 2}
2011-07-01 40723 {'326f': 2}
2011-07-02 40724 {'326e': 2}
2011-07-03 40725 {'326d': 2}
2011-07-04 40726 {'326c': 1}
...
```

Date Field

- Found the date field:
  - Last nibble changes once per day
  - Second and third nibbles change every 256 and 16 days
  - First nibble is always 0x3

case study

Decryption

- Date field has large set of values
  - We know the likely encoding
  - Can use to work out more about encryption
- Have to ROL some values to make sense
- Number of days since 1/1/1970

Occam's Razor

- Look for the simplest solution
- If it seems too complex, it probably is

Generalised Algorithm

- Generalise algorithm over whole ticket:
  - XOR each nibble with the previous one
  - ROL nibbles (1, 2), (5, 6)... if bit set else (3, 4), (7, 8)...

Deciphering Fields

case study

- Try to find fields that should be there
  - Guess how they would be stored
  - Dates are days since 1970, seconds?
- Group data by known values, see what they have in common
- Try changing values (not applicable here)
- Look for checksums, version numbers, padding, redundancy

Checksum

- Looked for complex checksums
- XORing all encoded nibbles together gives 0xF

case study

Initialisation Vector
Ticket type (single, return, weekly)
Ticket scan status
Valid from date
Valid to date
Origin station
Destination station
Transit type code (bit packed)
Last scanned date
Last scanned time
Last scanned station
Issuing machine ID
Ticket serial number
Concession type code
Slogan Number
Ticket Price
Checksum

```
X 01 01 3BEC 3BEC 071 070 4 0000 00000 071 8BD 032E3 7E A 0010 X
```

Special Tickets

case study

- Look different

```
800000001A22AA90D090D089310AA222222222239AB799EF9F07333BBA
```

- Many days trying to work out the “special encryption”
- Spent far too long on this:

```
( 1A22AA9... >> 1 ) = D11554...
```

Information Leaks

- Get creative
  - Websites
  - Significant ordering
  - Do some research
- Offsets, outliers

case study

Cool Things

- Physical serial numbers match fields, include station
- Rail enthusiasts detail *everything*
- Station IDs found on website
  - Constant offset

Custom Cryptography

- Has anything good ever come of this?
- Takes cryptographers years to do this right
- Strong cryptography is free and easy
- Learn about it before you use it

Responsible Disclosure

- Difficult
- Takes a while
- Changing large systems is hard
- Know the relevant laws
- Work with the organisation

case study

Message from transport spokesperson

“We acknowledge the group for their interest and research in this important area. We continue to expand our monitoring and fraud protection mechanisms accordingly and are implementing stronger measures in our new technologies.

It should be recognised that fare evasion and product tampering is a criminal offence and such activities are investigated accordingly. It is also an offence and unethical to conduct tests on live systems without proper authorisation.

It is an offence to travel without a valid ticket. A ticket is not valid if it is defaced, mutilated or altered.”

Questions?