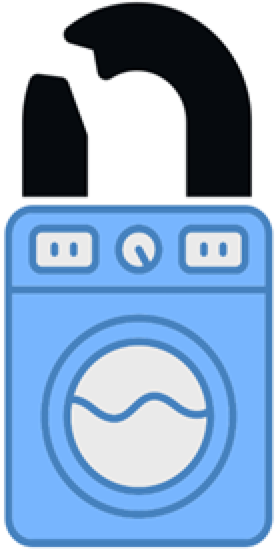
Hung Out to Dry

…airing the dirty laundry of stored-value wash cards



DEF CON 33 – RF Village

Presented by: **Luu** (Aidan N.) & **Equip** (Alexander H.)

Yes, we told them. Disclosure made responsibly.

Outline

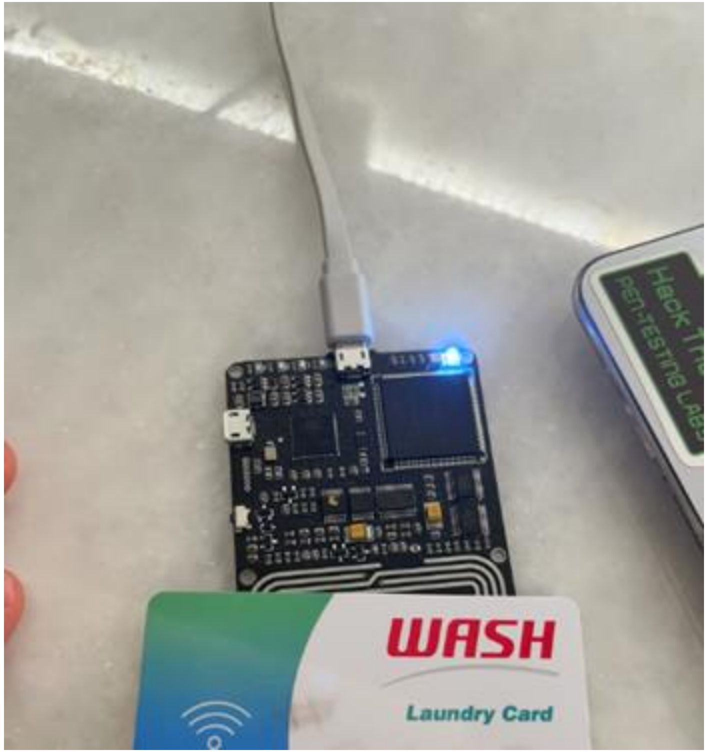
1. Disclaimer
2. The calling..
3. Affected Systems
4. Card Structure
5. Finding the value
6. Labelling the variables and data structure
7. Testing with Value Increments
8. Encoding pattern discovery
9. Function Mirrors
10. Why a valid card cannot be created from nothing
11. Challenges in system upgrades
12. DEMO

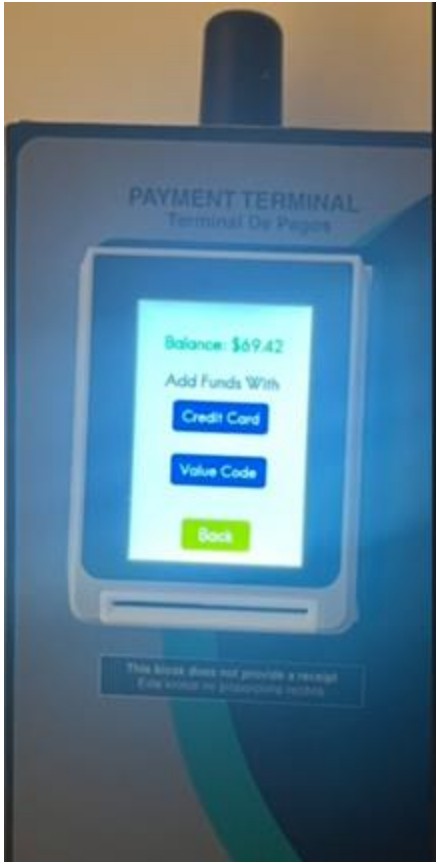
DISCLAIMER

- Educational research only
- Ethical testing, good faith
- $47 legitimately loaded, ($15 used)
- All funds legally contributed

The calling..

PAYMENT TERMINAL
WASH
PLEASE TAP YOUR CARD HERE
Buy Card
CARD DISPENSER


Hack The
PEN-TESTING LABS
WASH
Laundry Card


PAYMENT TERMINAL
Terminal De Pagos
Balance: $69.42
Add Funds With
Credit Card
Value Code
Back
This kiosk does not provide a receipt

Affected Systems

UNIVERSITY OF NORTH CAROLINA CHAPEL HILL LVX LIBERTAS

MASSACHUSETTS COLLEGE NORTH ADAMS OF LIBERAL ARTS EST. 1894

CASE WESTERN RESERVE UNIVERSITY

THE UNIVERSITY OF IOWA ORGANIZED FEB. 25th 1847.

CSC SERVICEWORKS

WASH

MONDO

FMB LAUNDRY

KIOSOFT

automatic laundry
superior service solutions

washbar

Hercules

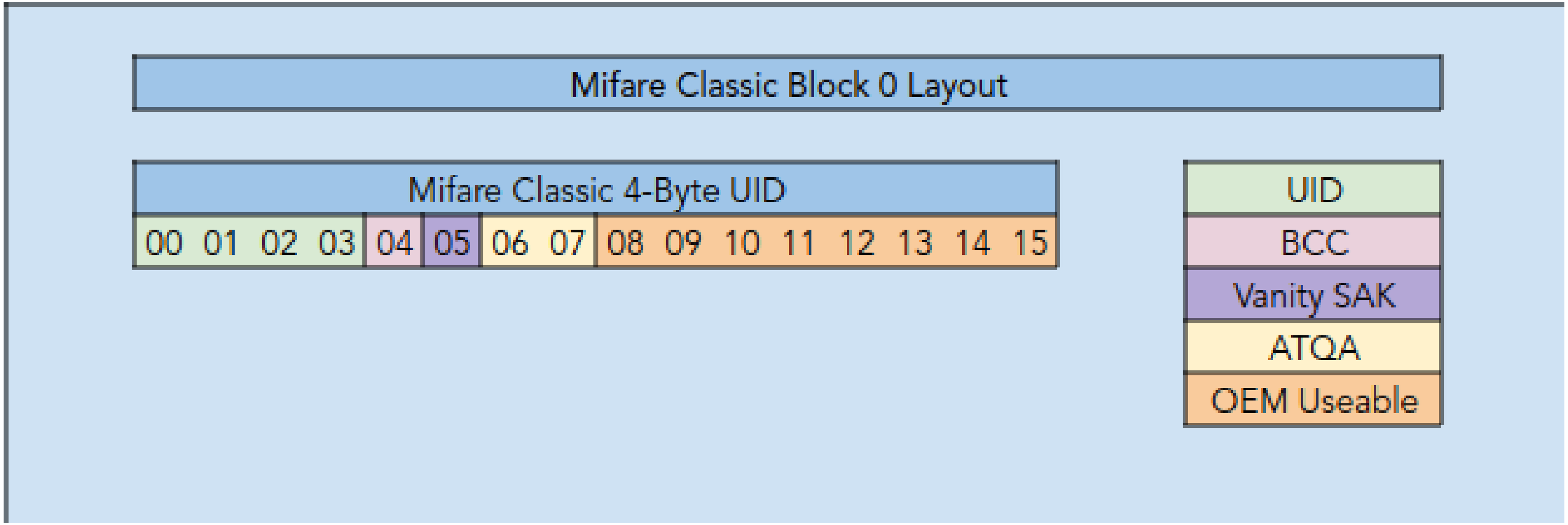
Coinamatic®
Laundry & Beyond. Simply Clean.

Commercial Laundries, Inc.

DEMO

Make sure you stay until the end for a quick video demo.

Card Structure (Mifare Classic)



Mifare Classic Block 0 Layout

Mifare Classic 4-Byte UID

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15

UID

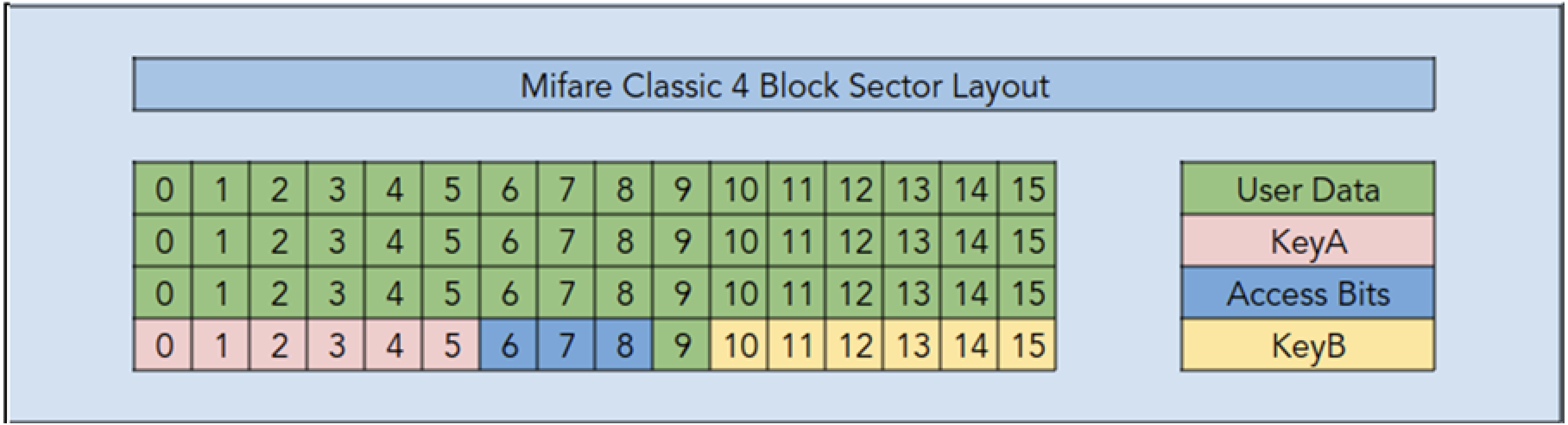
BCC

Vanity SAK

ATQA

OEM Useable

Card Structure (Mifare Classic)



Mifare Classic 4 Block Sector Layout

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

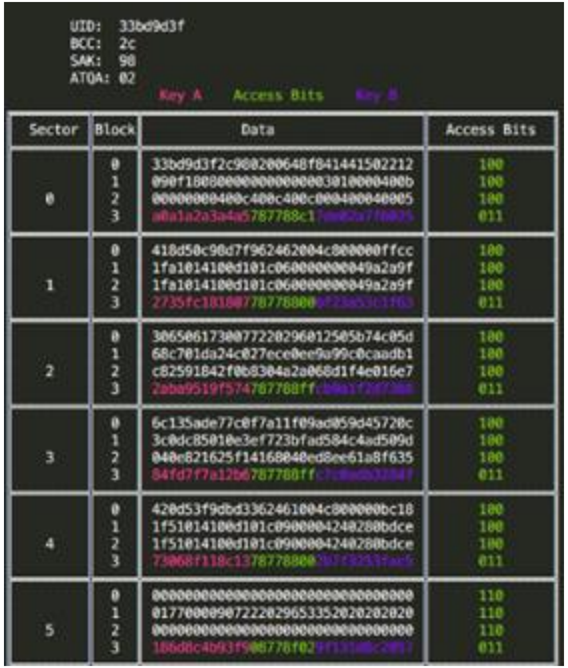
User Data

KeyA

Access Bits

KeyB

Card Structure (Mifare Classic)



```
          UID:  33bd9d3f
          BCC:  2c
          SAK:  98
          ATQA: 02
                       Key A     Access Bits      Key B
 ┌────────┬──────┬──────────────────────────────────┬──────────────┐
 │ Sector │Block │               Data               │ Access Bits  │
 ├────────┼──────┼──────────────────────────────────┼──────────────┤
 │        │  0   │ 33bd9d3f2c980200648f841441502212 │     100      │
 │   0    │  1   │ 090f180800000000000003010000400b │     100      │
 │        │  2   │ 00000000400c400c400c000400040005 │     100      │
 │        │  3   │ a0a1a2a3a4a5787788c17de02a7f6025 │     011      │
 ├────────┼──────┼──────────────────────────────────┼──────────────┤
 │        │  0   │ 418d50c98d7f962462004c800000ffcc │     100      │
 │   1    │  1   │ 1fa1014100d101c060000000049a2a9f │     100      │
 │        │  2   │ 1fa1014100d101c060000000049a2a9f │     100      │
 │        │  3   │ 2735fc18180778778800bf23a53c1f63 │     011      │
 ├────────┼──────┼──────────────────────────────────┼──────────────┤
 │        │  0   │ 3065061730077220296012505b74c05d │     100      │
 │   2    │  1   │ 68c701da24c027ece0ee9a99c0caadb1 │     100      │
 │        │  2   │ c82591842f0b8304a2a068d1f4e016e7 │     100      │
 │        │  3   │ 2aba9519f574787788ffcb9a1f2d7368 │     011      │
 ├────────┼──────┼──────────────────────────────────┼──────────────┤
 │        │  0   │ 6c135ade77c0f7a11f09ad059d45720c │     100      │
 │   3    │  1   │ 3c0dc85010e3ef723bfad584c4ad509d │     100      │
 │        │  2   │ 040e821625f14168040ed8ee61a8f635 │     100      │
 │        │  3   │ 84fd7f7a12b6787788ffc7c0adb3284f │     011      │
 ├────────┼──────┼──────────────────────────────────┼──────────────┤
 │        │  0   │ 420d53f9dbd3362461004c800000bc18 │     100      │
 │   4    │  1   │ 1f51014100d101c0900004240280bdce │     100      │
 │        │  2   │ 1f51014100d101c0900004240280bdce │     100      │
 │        │  3   │ 73068f118c13787788002b7f3253fac5 │     011      │
 ├────────┼──────┼──────────────────────────────────┼──────────────┤
 │        │  0   │ 00000000000000000000000000000000 │     110      │
 │   5    │  1   │ 01770000907222029653352020202020 │     110      │
 │        │  2   │ 00000000000000000000000000000000 │     110      │
 │        │  3   │ 186d8c4b93f908778f029f131d8c2057 │     011      │
 └────────┴──────┴──────────────────────────────────┴──────────────┘
```

Finding the Value

Known balance on card: $18.75

Finding the Value

$18.75 → 1875 cents

Finding the Value

1875 → 0x0753

Finding the Value

Dump of Blocks 1 to 9

```
Block 1: 30 30 00 01 00 00 00 12 98 89 00 00 01 11 EE 45
Block 2: 01 01 4C 55 55 0B 00 00 00 53 07 01 00 00 00 12
Block 3: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 4: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 5: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 6: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 7: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 8: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 9: 36 42 0F 00 C9 BD F0 FF 36 42 0F 00 09 F6 09 FE
```

Looking for: 0x0753 (07 53)

Finding the Value

Dump of Blocks 1 to 9

```
Block 1: 30 30 00 01 00 00 00 12 98 89 00 00 01 11 EE 45
Block 2: 01 01 4C 55 55 0B 00 00 00 53 07 01 00 00 00 12
Block 3: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 4: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 5: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 6: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 7: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 8: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 9: 36 42 0F 00 C9 BD F0 FF 36 42 0F 00 09 F6 09 FE
```

Looking for: 0x0753 (07 53)

Finding the Value

Dump of Blocks 1 to 9

```
Block 1: 30 30 00 01 00 00 00 12 98 89 00 00 01 11 EE 45
Block 2: 01 01 4C 55 55 0B 00 00 00 53 07 01 00 00 00 12
Block 3: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 4: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 5: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 6: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 7: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 8: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 9: 36 42 0F 00 C9 BD F0 FF 36 42 0F 00 09 F6 09 FE
```

Looking for: 0x5307 (53 07)

Finding the Value

Value Positions:

- Block 2, Bytes: 9 & 10 (Value at last top-up)
- Blocks 4 & 8 (mirrored), Bytes: 0 & 1 and 8 & 9

Labelling the variables and data structure (the technical stuff)

```
Block 1: 30 30 00 01 00 00 00 12 98 89 00 00 01 11 EE 45
Block 2: 01 01 4C 55 55 0B 00 00 00 53 07 01 00 00 00 12
Block 3: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 4: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 5: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 6: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 7: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 8: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 9: 36 42 0F 00 C9 BD F0 FF 36 42 0F 00 09 F6 09 FE
```

Card Number (printed on card): 0 12 98 89 - (not real CN) Redacted due to being tied to the owner.
Value At Last Topup (VALT): 53 07
Transaction ID: 4C 55 55
Incremental Mirror Byte: 0B
Terminal Byte (XOR): 12
Value Pad (Current Value + Incremental Mirror Byte): 53 07 0B
Middle Bytes: AC F8 F4
Under Value Byte (UVB): 36
Under Middle Byte (UMB): C9
Sector KeyA: 45 71 75 69 70 20 - (not real key value) Redacted due to being specific to site
Sector Access Conditions: 68 77 89
Sector KeyB: 4C 75 75 31 37 36 - (not real key value) Redacted due to being specific to site

1. Sector Trailer

```
Block 1: 30 30 00 01 00 00 00 12 98 89 00 00 01 11 EE 45
Block 2: 01 01 4C 55 55 0B 00 00 00 53 07 01 00 00 00 12
Block 3: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 4: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 5: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 6: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 7: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 8: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 9: 36 42 0F 00 C9 BD F0 FF 36 42 0F 00 09 F6 09 FE
```

Sector KeyA: 45 71 75 69 70 20 - (not real key value) Redacted due to being specific to site
Sector Access Conditions: 68 77 89
Sector KeyB: 4C 75 75 20 20 20 - (not real key value) Redacted due to being specific to site

2. Card Number

```
Block 1: 30 30 00 01 00 00 00 12 98 89 00 00 01 11 EE 45
Block 2: 01 01 4C 55 55 0B 00 00 00 53 07 01 00 00 00 12
Block 3: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 4: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 5: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 6: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 7: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 8: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 9: 36 42 0F 00 C9 BD F0 FF 36 42 0F 00 09 F6 09 FE
```

Card Number (printed on card): 0 12 98 89 - (not real CN) Redacted due to being tied to the owner.
Actual CN: 0129889

3. Value At Last Topup (VALT)

```
Block 1: 30 30 00 01 00 00 00 12 98 89 00 00 01 11 EE 45
Block 2: 01 01 4C 55 55 0B 00 00 00 53 07 01 00 00 00 12
Block 3: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 4: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 5: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 6: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 7: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 8: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 9: 36 42 0F 00 C9 BD F0 FF 36 42 0F 00 09 F6 09 FE
```

Value At Last Topup (VALT): 53 07 → 0x0753 → $18.75

4. Transaction Identifier

```
Block 1: 30 30 00 01 00 00 00 12 98 89 00 00 01 11 EE 45
Block 2: 01 01 4C 55 55 0B 00 00 00 53 07 01 00 00 00 12
Block 3: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 4: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 5: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 6: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 7: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 8: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 9: 36 42 0F 00 C9 BD F0 FF 36 42 0F 00 09 F6 09 FE
```

Transaction ID: 4C 55 55

5. Incremental Mirror Byte

```
Block 1: 30 30 00 01 00 00 00 12 98 89 00 00 01 11 EE 45
Block 2: 01 01 4C 55 55 0B 00 00 00 53 07 01 00 00 00 12
Block 3: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 4: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 5: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 6: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 7: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 8: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 9: 36 42 0F 00 C9 BD F0 FF 36 42 0F 00 09 F6 09 FE
```

Incremental Mirror Byte: 0B

6. Terminal Byte (XOR)

```
Block 1: 30 30 00 01 00 00 00 12 98 89 00 00 01 11 EE 45
Block 2: 01 01 4C 55 55 0B 00 00 00 53 07 01 00 00 00 12
Block 3: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 4: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 5: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 6: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 7: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 8: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 9: 36 42 0F 00 C9 BD F0 FF 36 42 0F 00 09 F6 09 FE
```

Terminal Byte (XOR): 12

7. Value Pad

```
Block 1: 30 30 00 01 00 00 00 12 98 89 00 00 01 11 EE 45
Block 2: 01 01 4C 55 55 0B 00 00 00 53 07 01 00 00 00 12
Block 3: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 4: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 5: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 6: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 7: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 8: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 9: 36 42 0F 00 C9 BD F0 FF 36 42 0F 00 09 F6 09 FE
```

Value Pad (Current Value + Incremental Mirror Byte): 53 07 0B

8. Middle Bytes

```
Block 1: 30 30 00 01 00 00 00 12 98 89 00 00 01 11 EE 45
Block 2: 01 01 4C 55 55 0B 00 00 00 53 07 01 00 00 00 12
Block 3: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 4: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 5: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 6: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 7: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 8: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 9: 36 42 0F 00 C9 BD F0 FF 36 42 0F 00 09 F6 09 FE
```

Middle Bytes: AC F8 F4

9. Under Value Byte

```
Block 1: 30 30 00 01 00 00 00 12 98 89 00 00 01 11 EE 45
Block 2: 01 01 4C 55 55 0B 00 00 00 53 07 01 00 00 00 12
Block 3: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 4: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 5: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 6: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 7: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 8: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 9: 36 42 0F 00 C9 BD F0 FF 36 42 0F 00 09 F6 09 FE
```

Under Value Byte (UVB): 36

10. Under Middle Byte

```
Block 1: 30 30 00 01 00 00 00 12 98 89 00 00 01 11 EE 45
Block 2: 01 01 4C 55 55 0B 00 00 00 53 07 01 00 00 00 12
Block 3: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 4: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 5: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 6: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 7: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 8: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 9: 36 42 0F 00 C9 BD F0 FF 36 42 0F 00 09 F6 09 FE
```

Under Middle Byte (UMB): C9

Encoding Pattern Discovery

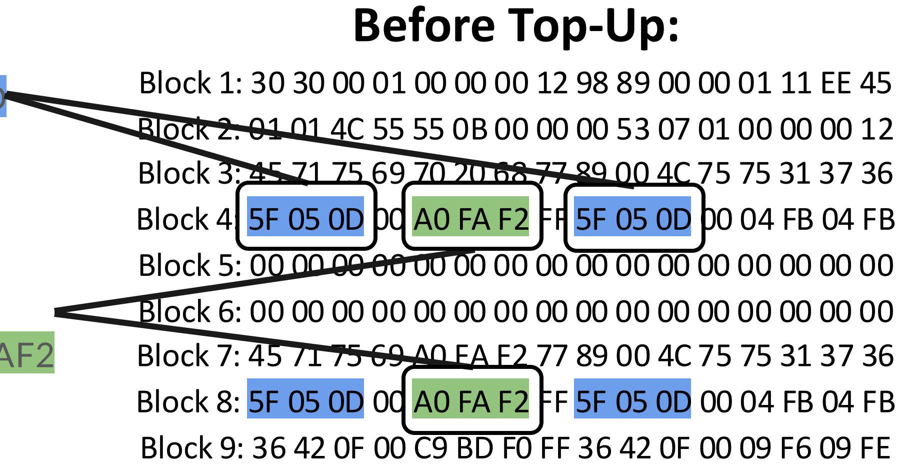
Let’s gather some data..

Encoding Pattern Discovery

- Value Pad: 0x5F050D
  (**95**, 5, 13) → $13.75

  and top up counter:
  0x0D

- Middle Bytes: 0xA0FAF2
  (**160**, 250, 242)



Before Top-Up:

```
Block 1: 30 30 00 01 00 00 00 12 98 89 00 00 01 11 EE 45
Block 2: 01 01 4C 55 55 0B 00 00 00 53 07 01 00 00 00 12
Block 3: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 4: 5F 05 0D 00 A0 FA F2 FF 5F 05 0D 00 04 FB 04 FB
Block 5: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 6: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 7: 45 71 75 69 A0 FA F2 77 89 00 4C 75 75 31 37 36
Block 8: 5F 05 0D 00 A0 FA F2 FF 5F 05 0D 00 04 FB 04 FB
Block 9: 36 42 0F 00 C9 BD F0 FF 36 42 0F 00 09 F6 09 FE
```

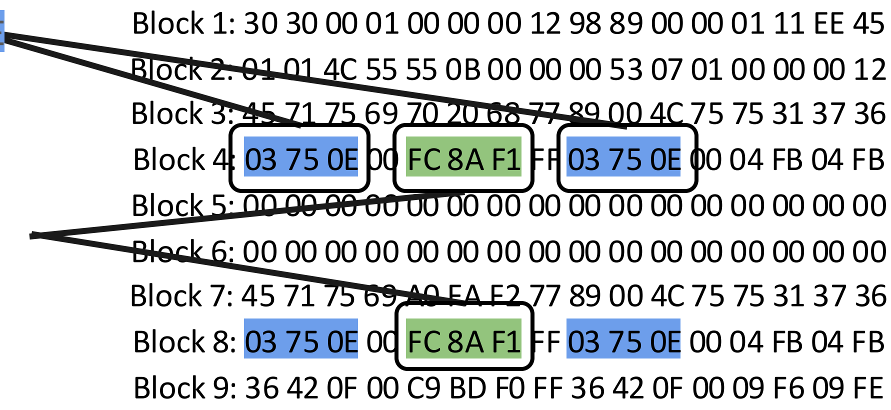
Encoding Pattern Discovery

- Value Pad: 0x03750E (**3**, 117, 14) → $18.75

  and top up counter: 0x0E

- Middle Bytes: 0xFC8AF1 (**252**, 138, 241)

After $5 Top-Up:



```
Block 1: 30 30 00 01 00 00 00 12 98 89 00 00 01 11 EE 45
Block 2: 01 01 4C 55 55 0B 00 00 00 53 07 01 00 00 00 12
Block 3: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 4: 03 75 0E 00 FC 8A F1 FF 03 75 0E 00 04 FB 04 FB
Block 5: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 6: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 7: 45 71 75 69 A0 FA F2 77 89 00 4C 75 75 31 37 36
Block 8: 03 75 0E 00 FC 8A F1 FF 03 75 0E 00 04 FB 04 FB
Block 9: 36 42 0F 00 C9 BD F0 FF 36 42 0F 00 09 F6 09 FE
```

Other parts have not changed for demonstration purposes*

Encoding Pattern Discovery

Before top-up:

- Value Pad: 0x5F050D (**95**, 5, 13) → $13.75 and top up counter: 0x0D
- Middle Bytes: 0xA0FAF2 (**160**, 250, 242)

After $5 top-up:

- Value Pad: 0x03750E (**3**, 117, 14) → $18.75 and top up counter: 0x0E
- Middle Bytes: 0xFC8AF1 (**252**, 138, 241)

Encoding Pattern Discovery

$$\frac{y_2 - y_1}{x_2 - x_1} = m$$

Where:

x_2 = First middle byte after adding $5

x_1 = First middle byte before adding money

y_2= First value byte after adding $5

y_1 = First value byte before adding money

Encoding Pattern Discovery

$$\frac{y_2 - y_1}{x_2 - x_1} = m$$

x_2 = 252

x_1 = 160

y_2 = 3

y_1 = 95

Encoding Pattern Discovery

$$\frac{y_2 - y_1}{x_2 - x_1} = m$$

$x_2 = 252$

$x_1 = 160$

$y_2 = 3$

$y_1 = 95$

$$\frac{252 - 160}{3 - 95} = -1$$

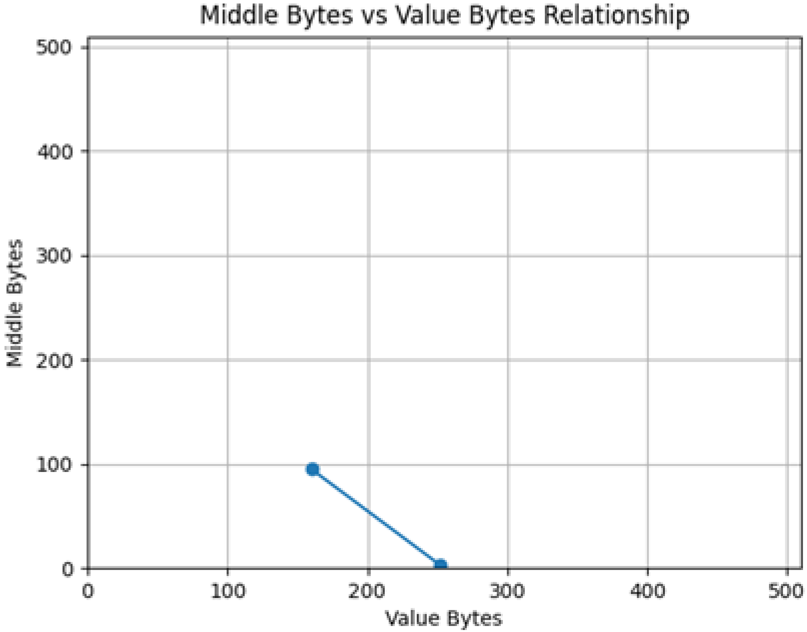
Encoding Pattern Discovery

$x_2 = 252$

$x_1 = 160$

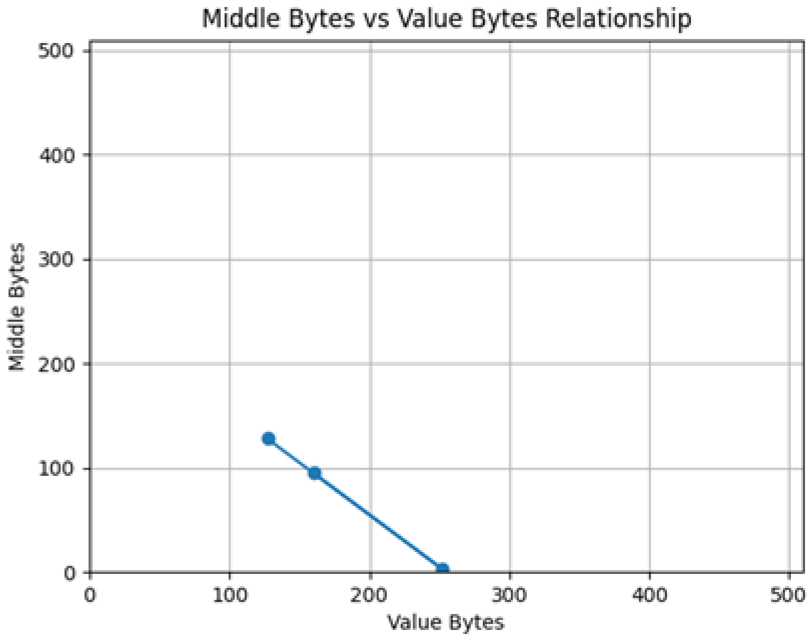
$y_2 = 3$

$y_1 = 95$



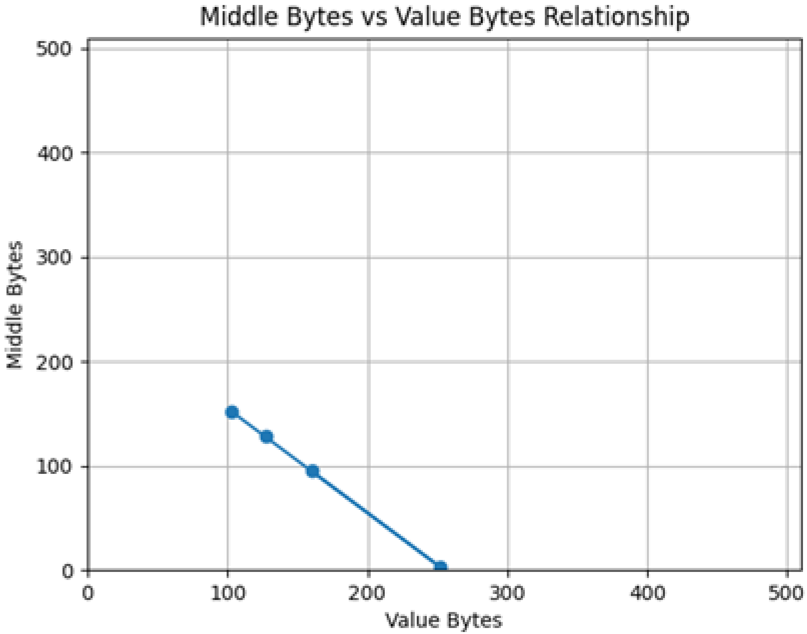
Encoding Pattern Discovery

+$3

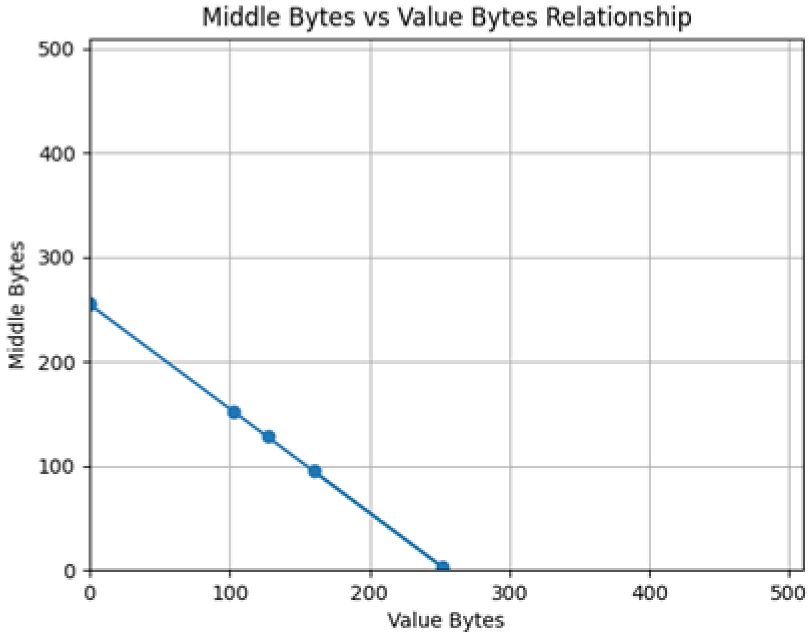


Encoding Pattern Discovery

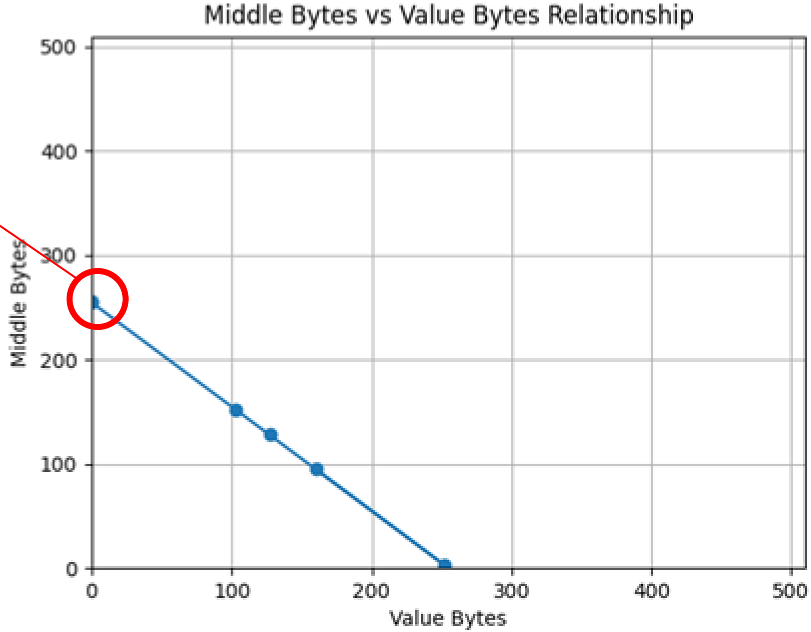
+$10



Encoding Pattern Discovery



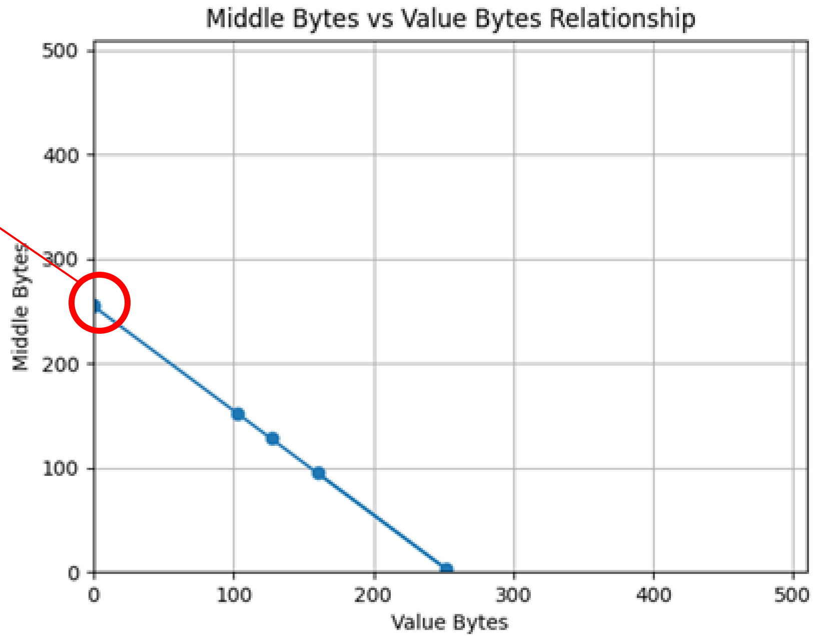
Encoding Pattern Discovery



Y-int: 0, 255

Encoding Pattern Discovery

Y-int: 0, 255



New Formula:

$$y = -x + 255$$

Encoding Pattern Discovery

Well..

It’s just an XOR of 0xFF…

AKA Bitwise Negation

Function Mirrors

```
Block 1: 30 30 00 01 00 00 00 12 98 89 00 00 01 11 EE 45
Block 2: 01 01 4C 55 55 0B 00 00 00 53 07 01 00 00 00 12
Block 3: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 4: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 5: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 6: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 7: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 8: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 9: 36 42 0F 00 C9 BD F0 FF 36 42 0F 00 09 F6 09 FE
```

Value Byte #1: 53

Middle Byte #1: AC

Function Mirrors

```
Block 1: 30 30 00 01 00 00 00 12 98 89 00 00 01 11 EE 45
Block 2: 01 01 4C 55 55 0B 00 00 00 53 07 01 00 00 00 12
Block 3: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 4: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 5: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 6: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 7: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 8: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 9: 36 42 0F 00 C9 BD F0 FF 36 42 0F 00 09 F6 09 FE
```

Value Byte #2: 07
Middle Byte #2: F8

Function Mirrors

```
Block 1: 30 30 00 01 00 00 00 12 98 89 00 00 01 11 EE 45
Block 2: 01 01 4C 55 55 0B 00 00 00 53 07 01 00 00 00 12
Block 3: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 4: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 5: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 6: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 7: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 8: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 9: 36 42 0F 00 C9 BD F0 FF 36 42 0F 00 09 F6 09 FE
```

Incremental Mirror Byte: 0B
Middle Byte #3: F4

Function Mirrors

```
Block 1: 30 30 00 01 00 00 00 12 98 89 00 00 01 11 EE 45
Block 2: 01 01 4C 55 55 0B 00 00 00 53 07 01 00 00 00 12
Block 3: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 4: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 5: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 6: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Block 7: 45 71 75 69 70 20 68 77 89 00 4C 75 75 31 37 36
Block 8: 53 07 0B 00 AC F8 F4 FF 53 07 0B 00 04 FB 04 FB
Block 9: 36 42 0F 00 C9 BD F0 FF 36 42 0F 00 09 F6 09 FE
```

Under Value Byte (UVB): 36

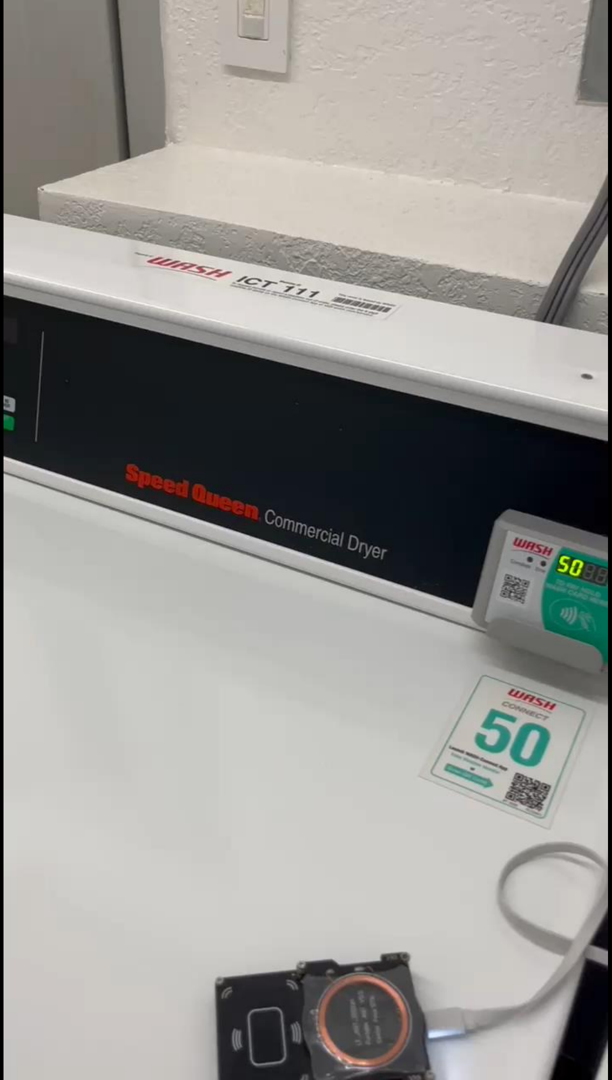
Under Middle Byte (UMB): C9

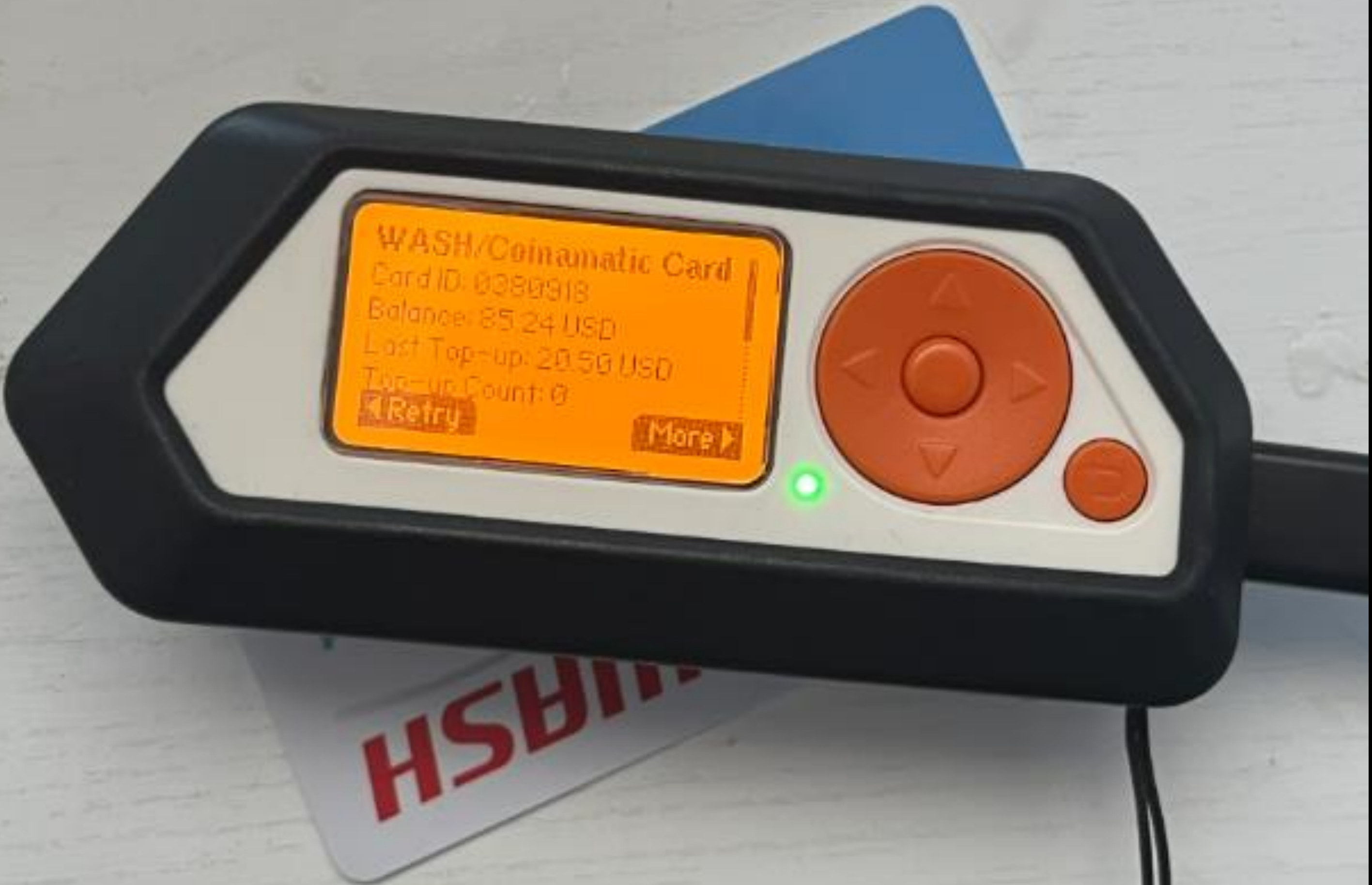
Why a valid card cannot be created from nothing

- Unknown deployment/site keys
- Non-valid card numbers

Challenges in System Upgrades

- 💸 Upgrade cost is high
- 📉 Low risk perception / market incentive
- 📊 Not profitable to fix


WASH ICT 111
Speed Queen Commercial Dryer
WASH
50
WASH
CONNECT
50



```
WASH/Coinamatic Card
Card ID: 0380918
Balance: 85.24 USD
Last Top-up: 20.50 USD
Top-up Count: 0
```

Acknowledgements

- Luu’s Parents
- GuruSteve
- Torron
- TheChamp
- Bettse, NVX, ZVE8

- **Everyone in ThePiratesPlunder discord!**