RFID Security

Edward Farrell, **stratsec**

Ruxcon, Melbourne, November 2010

Overview

- Intro
- Cards currently in use (YAY STANDARDS!)
- Weaknesses and exploits
- Implementations
- Implications…

Me

- Consultant @ **stratsec**
- Researching RFID in spare time

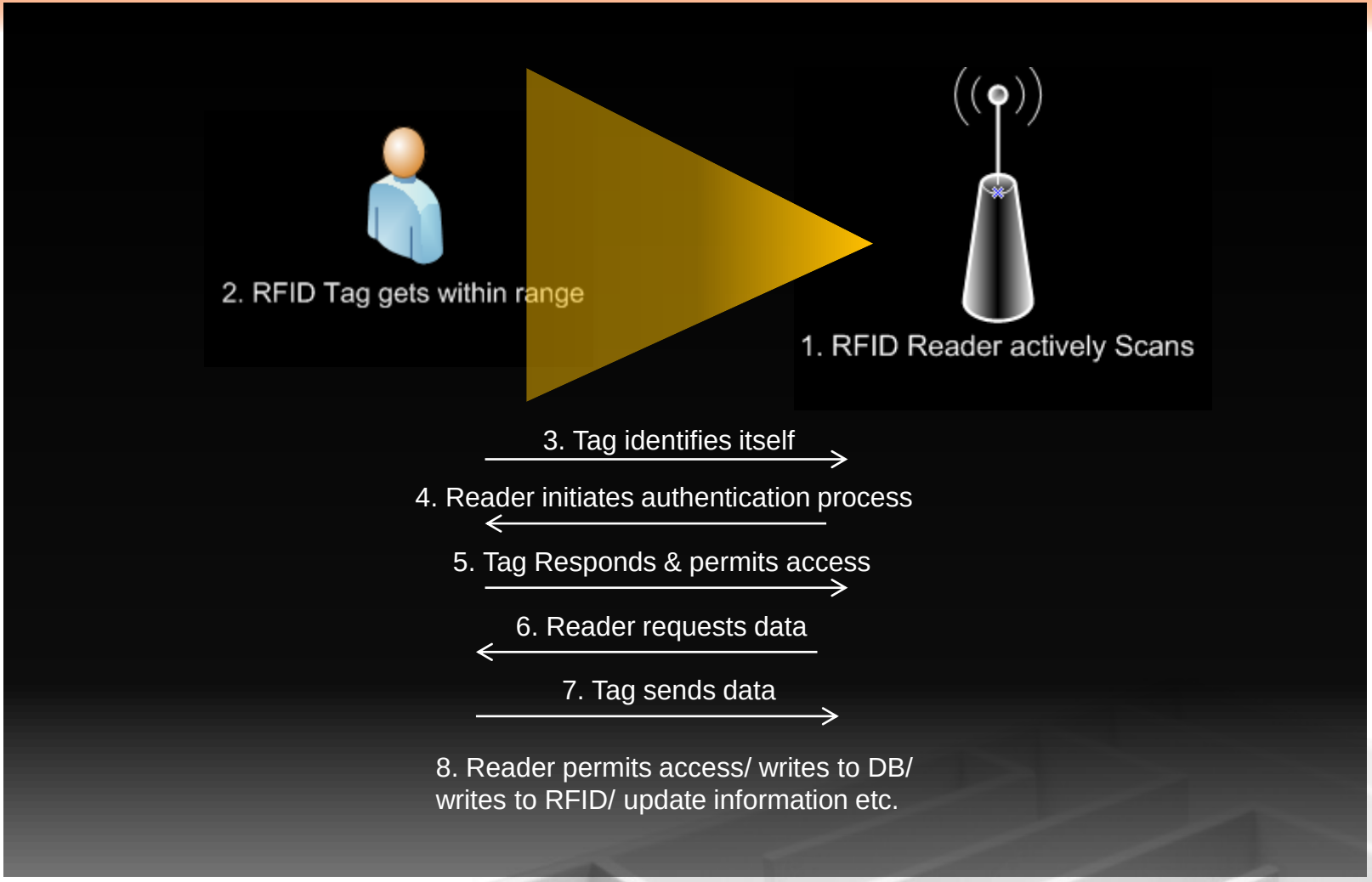
Small print legal stuff

Cybercrime Act 2001 (cth)

- DIV 477-Unauthorised access, modification or impairment with intent to commit a serious offence
- DIV 478- Unauthorised access to data on Commonwealth property (such as an asset tag or access pass).

- This presentation is PCI-DSS compliant

RFID overview

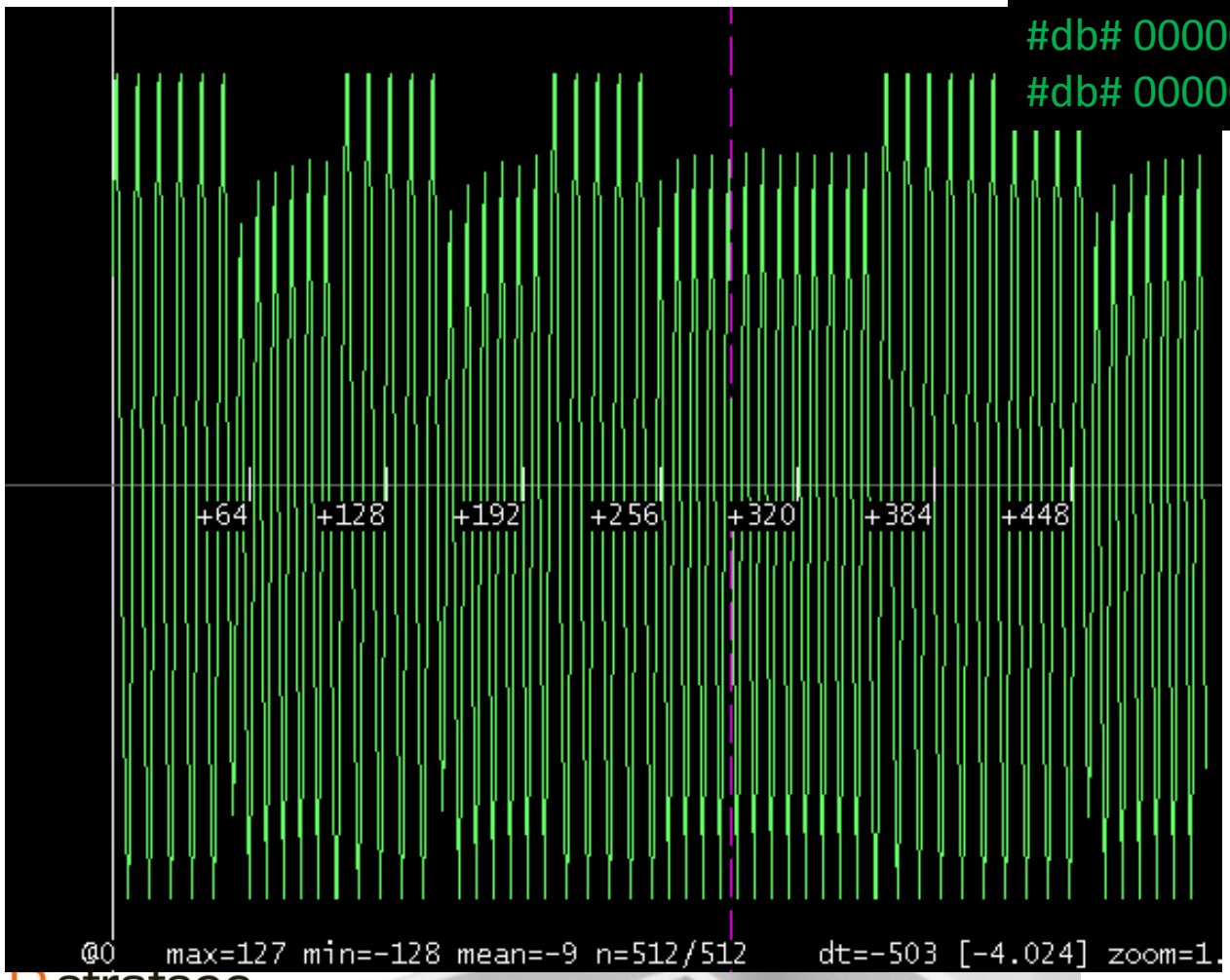


Toolz

- Reader
- Proxmark

125kHz- a RECAP

facID, LOCID , card ID
#db# 00000020, 04e20c38, 0000061c
#db# 00000024, ea6802bf, 0000015f
#db# 00000020, 05b57642, 0000bb21



ISO14443- overview

- Introduced 2001
- …it's a standard
- Widely used
- 13.56MHz freq
- Card variants

ISO14443- Logical characteristics

- MIFARE1k- 64 blocks divided into 16 Sectors
- Block 0
  - Specifies the Unique ID (UID)
  - Manufacturer data
  - Read only…mostly ☺
- Every 4th block
  - 2x Keys
  - Access conditions

0	UID, BCC, Manufacturers data
1	Data
2	Data
3	Key A, Access conditions, Key B
4	Data
5	Data
6	Data
7	Key A, Access conditions, Key B

Other deviations- MIFARE4k, MIFARE DESFIRE

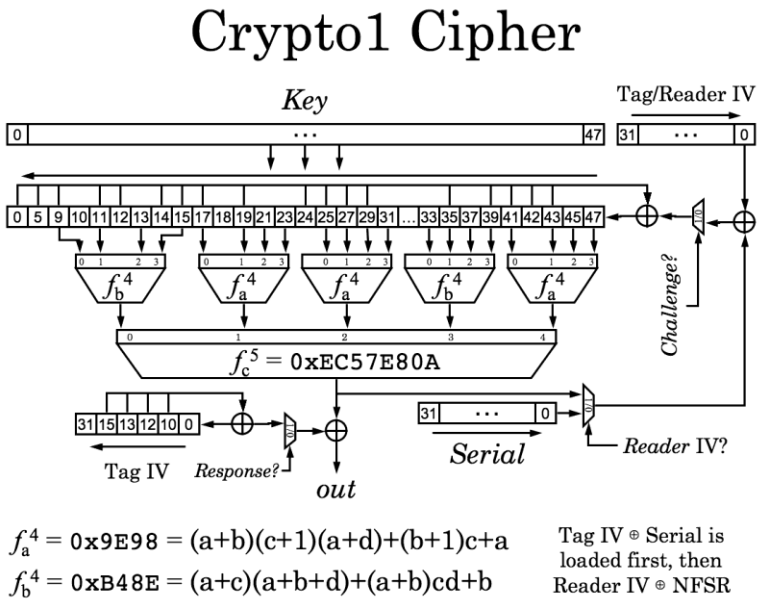
ISO14443- Physical characteristics

- distance of >10cm for power
- Upon being powered up the UID can be transmitted
- Power transferred on 13.56 MHz freq
- ISO14443-1 standard not particularly relevant anymore.
- Custom designs, physically and logically.

- Anti-collision

ISO14443 Cryptography

- Symmetric encryption
- The crypto-1 cypher
  - Used on the majority of implementations
  - Attempts to phase out…
- MIFARE plus- 128bit AES… and crypto-1
- DESFire EV1- DES/3DES/3KDES/AES



Crypto-1… its just a little broken

- 48bit keystream limits us to a potential keyspace of 2^48 (or about 281 trillion possible keys)
- Key exchange process visible and completely reversible

- 

Demotime

ISO14443 communication w/Crypto-1

Reader	26								Requesting
TAG	04	00							request answered
Reader	93	20							Select
TAG	7d	99	47	0b	a8				tag gives UID
Reader	93	70	7d	99	47	0b	a8	2e	e0	Reader selects UID
TAG	08	b6	dd						Respond as a mifare1k
Reader	50	00	57	cd					ACK.
Reader	52								Terminate

TAG	04	00							answer req
Reader	93	70	7d	99	47	0b	a8	2e	e0	Reader selects UID
TAG	08	b6	dd						select mifare1k
Reader	60	00	f5	7b					Auth (block 0)
TAG	00	1e	b0	9a					Nt
Reader	2f	3e	d3	27	a9	8c	68	4c	nR ⊕ ks1, aR ⊕ ks2
TAG	36	e3	63	6e					aT ⊕ ks3
Reader	0e	44	64	56					

TAG	34!	f5	75	d2!	46	0c!	be	a7!	41!	fa!	91	bb!	52	10	42	e5!	84	79!
Reader	1d	fb	71	84														
TAG	1c	6b!	44!	bd	68	f2	f3!	cd!	f0!	11	e4!	29	a5!	8f	2d	58	e2	81
Reader	90	fd	68	2d														
TAG	5d!	ec	d9	b9!	94	7a!	f8	8f!	f0!	4f!	ee!	95	2e	b2	ea	f9!	cf	55
Reader	9c	d6	4c	f4														
TAG	43!	78	62!	ce!	18	32!	c2!	ab	31	92!	01	eb!	eb!	89!	fa!	6b	62!	04
Reader	97	2c	4d	8d														
Reader	52																	

Code available at:
code.google.com/p/crapto1/

Crypto-1 keys… why bother changing?

- **Applicable to CRYPTO-1 cards**
- **Default Keys**

```
FFFFFFFFFFFF
(Phillips)
A0A1A2A3A4A5
(Infineon)
4D3A99C351DD
D3F7D3F7D3F7
AABBCCDDEEFF
B0B1B2B3B4B5
4d3a99c351dd
1a982c7e459a
000000000000
```

- **One key to rule them all!**
- **Default/weak access conditions**

```
787788E9
FF078069
FF078069
```

Crypto-1… its just a little broken

- 48bit keystream limits us to a potential keyspace of 2^48 (or about 281 trillion possible keys)
- Key exchange process visible and completely reversible
- Other attacks
  - Radboud University Nijmegen
  - Nicolas Courtois
- Our own brute forcing tool

GPU brute forcing

Graphics Cards

- Demand for processing power
- First seen against MD5 hash
- Leveraging power for brute forcing with NVIDIA CUDA

Vs other attacks

- Courtois and Nijmegen attacks
- Future applications…

GPU brute forcing (2)- where I'm up to

GPU brute forcing (2)- where I'm up to

```
Enter the UID: 7d99470b
Enter the Tag Challenge: 6ddb00f0
Cracking UID 0x7d99470b and Tag Challenge 0x6ddb00f0
**********

Unique value found:                    FFFA3597
Tag Response:              5342
Reader Response:                 0
Reader Challenge:                 0
**********
```

```
**********

Unique value found:                    24F1D049
Tag Response:              28CDA
Reader Response:                 0
Reader Challenge:                 0
**********
```

```
Unique value found:                    5AF7544B
Tag Response:              2DD3F
Reader Response:                 0
Reader Challenge:                 0
**********
```

GPU brute forcing (2)- where I'm up to

```
Enter the UID: 7d99470b
Enter the Tag Challenge: 6ddb00f0
Cracking UID 0x7d99470b and Tag Challenge 0x6ddb00f0
**********

Unique value found:                                    FFFA3597
Tag Res
Reader
Reader
********
```

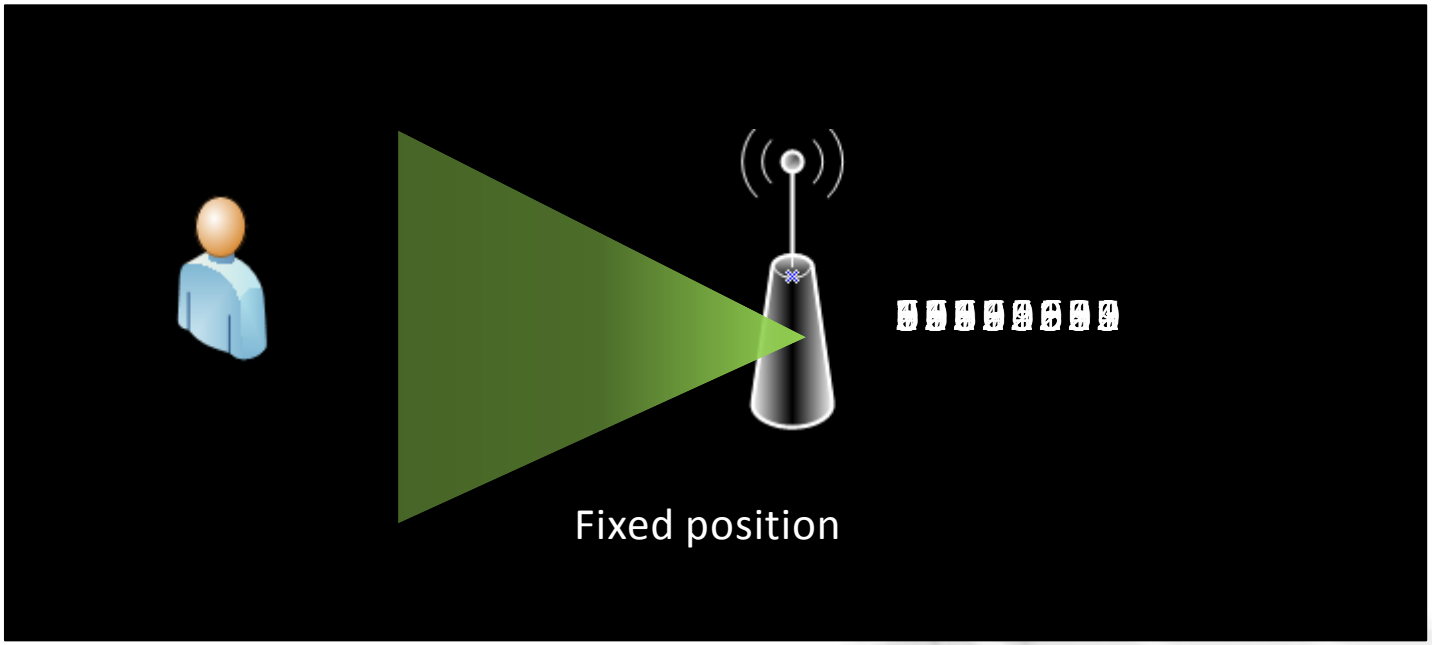
```
If this is the first time you've seen this Stop error screen,
restart your computer. If this screen appears again, follow
these steps:

Check for viruses on your computer. Remove any newly installed
hard drives or hard drive controllers. Check your hard drive
to make sure it is properly configured and terminated.
Run CHKDSK /F to check for hard drive corruption, and then
restart your computer.

Refer to your Getting Started manual for more information on
troubleshooting Stop errors.
```

```
24F1D049
```

```
Unique va
Tag Respo
Reader Re
Reader Ch
**********
```

a matter of entropy…

- *"entropy is a measure of the uncertainty associated with a random variable"*



- How does a card get its randomisims?

So we've found our Target

- Physical security
- Design considerations
  - Purpose
  - lets come back to block 0 ☺
- Distribution
- DB/RFID duplication

Theoretical attack walkthrough (insert path to Prox Tool)

Whats on the cards?

- Data represented in hexadecimal

1

```
0: 8ABB78B7 F E8804004755565O5D002107
1: FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
```

2

```
0: A096470B7A08040023561234OO11FFFF
1: 800F00000000000000000000000000
2: 00000000000000000000000000001250
3: 000000000000787788E9000000000000
```

3

```
0  : 04A6B09AD9D20280894800F00000000
5  : 00000000000000000000000000680100
1C: 00000000000000000000000000006820
1D: 00000000000000000000000000A31C
```

Tying it all together

- Card IDs readable regardless of authentication… implications?
- More advanced implementations still relying on security by obscurity
- Back on design considerations

Questions/Queries/comments/ doubtful points?