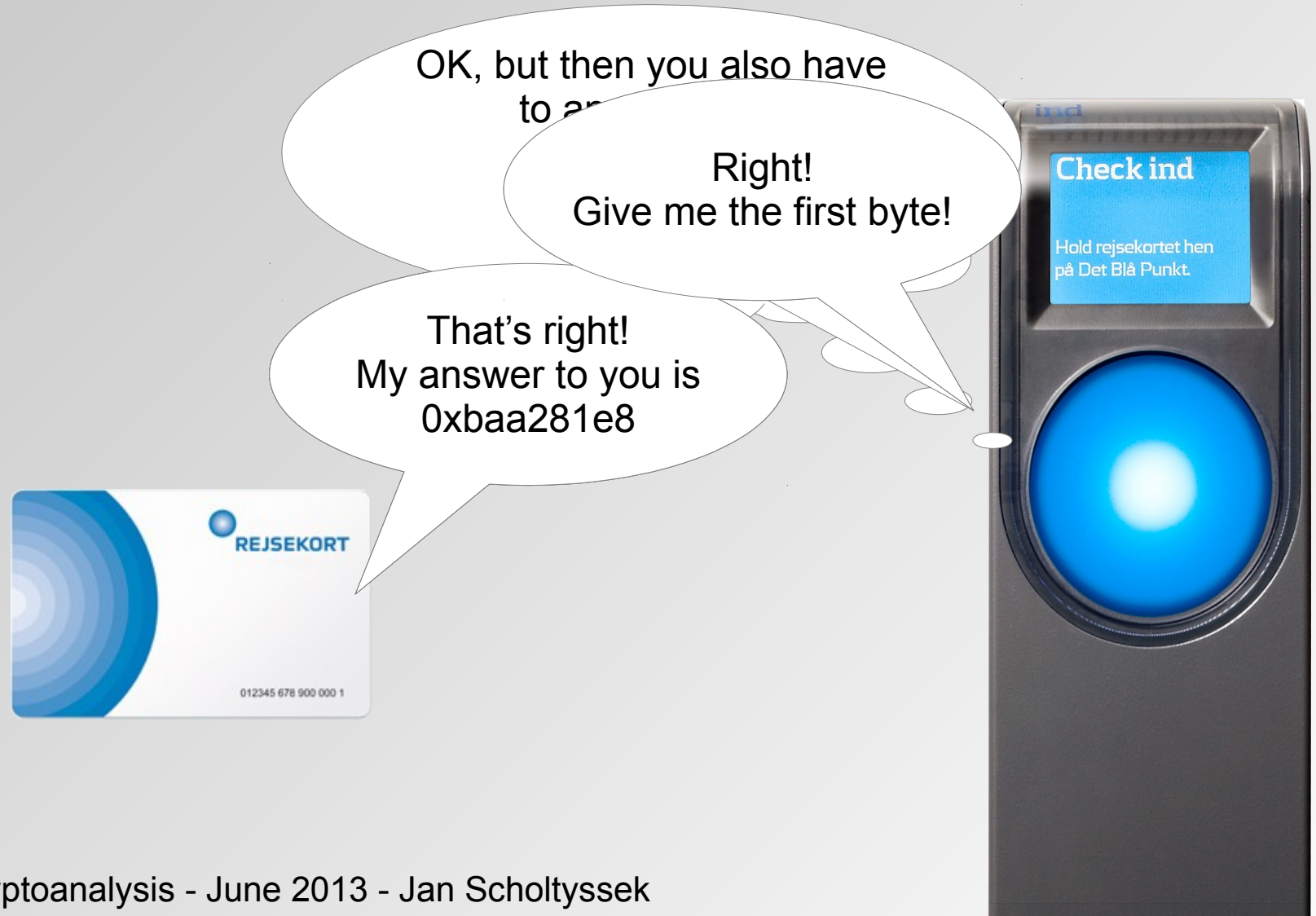
CRYPTO1

- The Mifare Classic proximity card

AGENDA

- Authentication
- CRYPTO1 algorithm
- Weaknesses
- Attacks
- Demonstration
- What have we learned?

Authentication



CRYPTO1

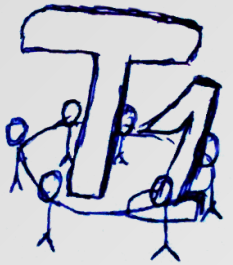
Is it secure?

Of course!
Nobody knows how it's work.

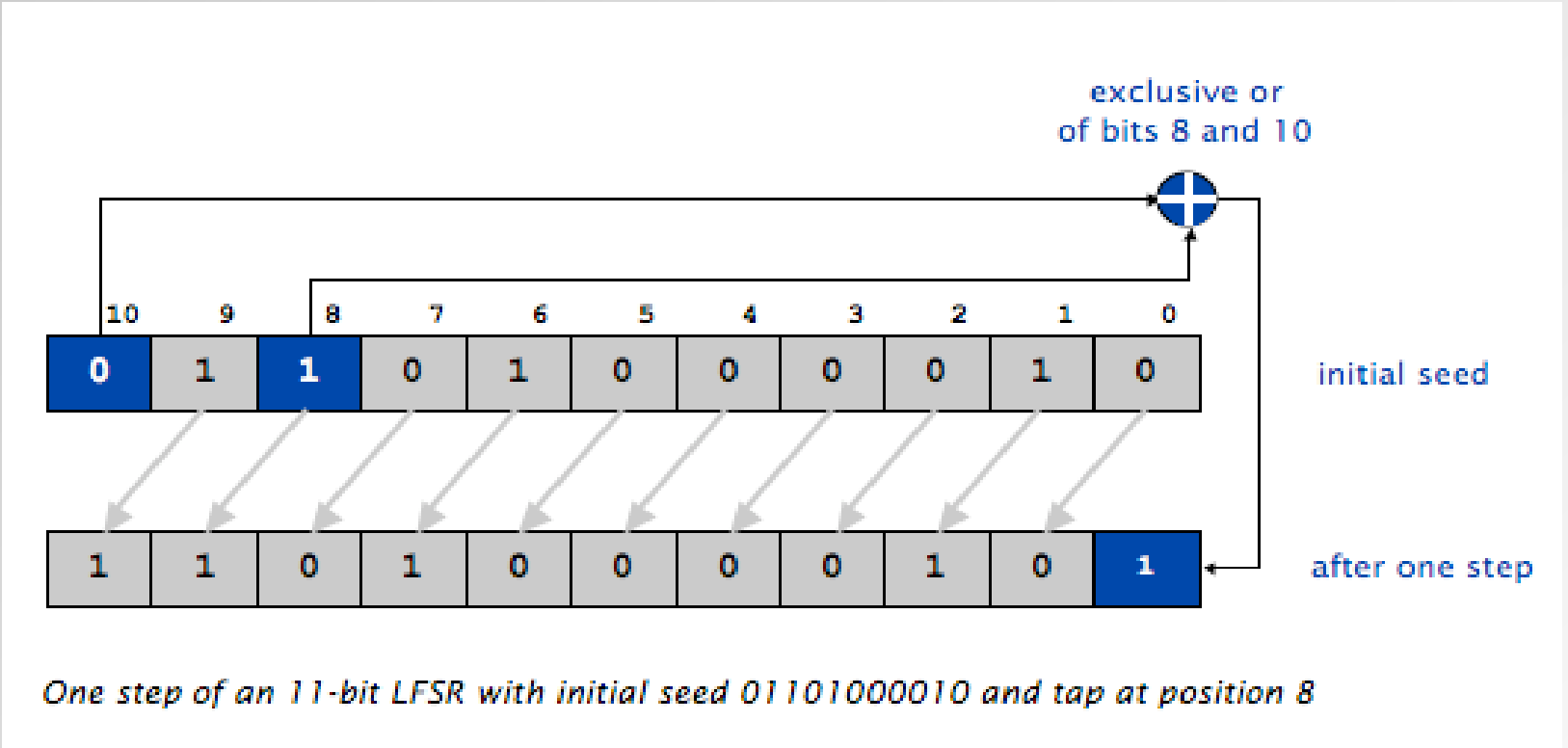
Crap!

Nothing to see here!
Go away!

Linear Feedback Shift Register (LFSR)

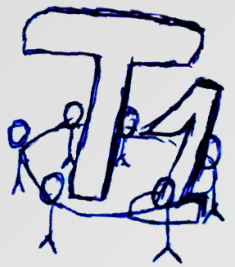


How it works:



One step of an 11-bit LFSR with initial seed 01101000010 and tap at position 8

Weaknesses
- Parity bits



Encrypted challenge:

0	0		6	8		2	6		1	5	

Parity bits

Encrypted answer:

8	2		d	b		e	a		9	2	

1	0	0	0	0	0	1	0

⊕

1	0	1	1	0	1	0	1	1	0	0	0	0	1	1	1	0	1	0	1	0	0

=

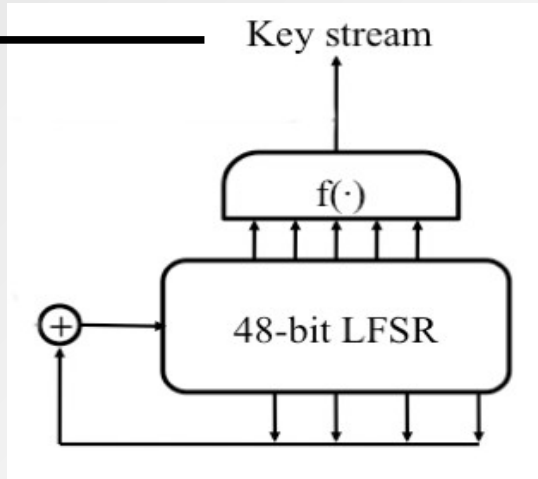
0	0	1	1	0	1	1	1

Decrypted answer:

3	7

Probability to guess right:

2^(-8)



Weaknesses
- Parity bits

Encrypted challenge:

0	0		6	8		2	6		1	5	

Parity bits

Encrypted answer:

8	2		d	b		e	a		9	2	

Answer my challenge:
0x492742de

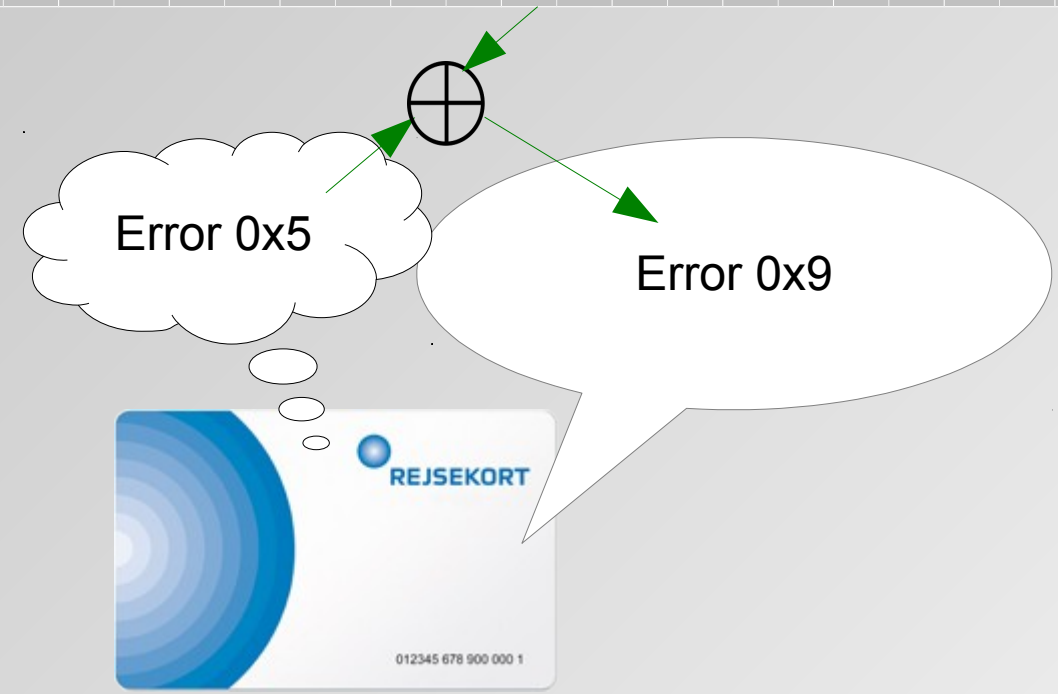
OK, but then you also have to answer mine:
0x00682615
My answer:
0x82dbea92

Weaknesses
- Parity bits

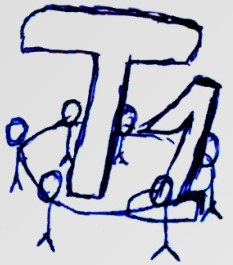
If we guess right:

Keystream

0	0	1	1	0	1	0	1	**1**	**0**	**0**	**0**	0	1	1	1	0	1	0	1	0	0



Attacks



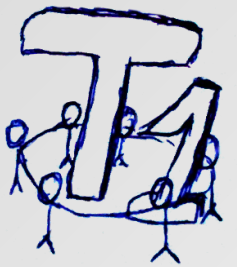
Brute force:

- Guess parity bits and store the challenges and answers when the tag answers
- Repeat several times
- Test all 2^48 keys and see if they give the same encrypted error

Strength:
- Short online time (< 1 sec)

Weaknesses:
- We have to test all keys
- No precomputation possible (tag challenge acts like salt)

Attacks



Vary the reader challenge:

- Start with a random reader challenge, guess parity bits until match
- Invert the last bit of each byte* and guess parity until match
- If all parity bits changes, there are only 436 possibilities for the odd LFSR bits

Strength:
- Keyspace reduced to ~2^33

Weaknesses:
- Long online time (several hours)
- No precomputation possible (tag challenge acts like salt)

* One at a time

Attacks

Recover more sector keys:

- In an already encrypted session, the new tag challenge is also encrypted!
- We know:
    - The first tag challenge
    - The time between the generation of the two challenges!
    - The cycle of the random number generator

Strength:
- Fast
- Can be eavedropped (no interaction required)

Weaknesses:
- None

What have we learned?

- Take Kerchhoffs principle serious
- 48 bits isn’t enough. Even brute force is feasable
- The crypto system is too weak and the random number generator not random.
- Don’t reuse keystream

Thank you!

Any challenges?