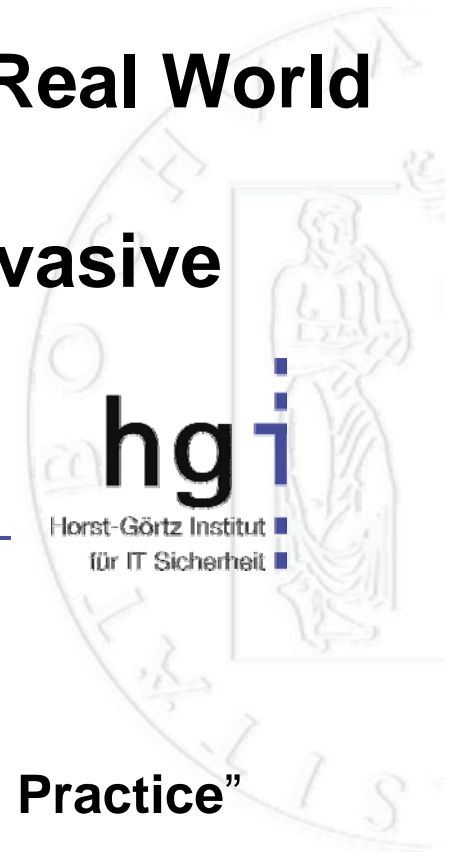


On the Power of Power Analysis in the Real World or Do we need SCA-resistance for pervasive computing?



Dagstuhl Workshop „Security Hardware in Theory and Practice”

July 20, 2008

Christof Paar

www.crypto.rub.de

Contents

- 1. Brief look at SCA history**
2. A case study: Brekaing temote keyless entry systems
3. Consequences and research problems

History of Side-Channel Attacks (1-slide version)

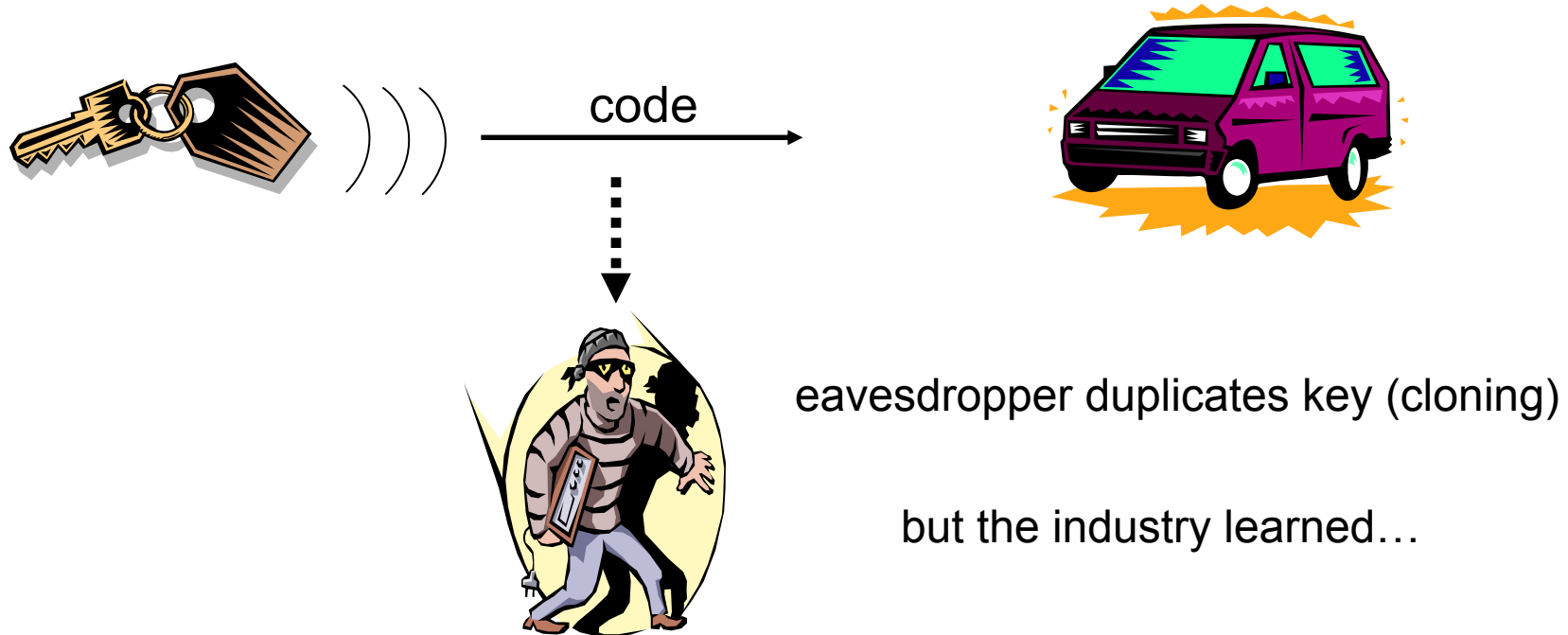
- Existence of side-channels for crypto devices known for several decades, (e.g., „Tempest“)
- Little concrete results / poor understanding prior to 1996 (at least outside intelligence community)
- 2nd half of 1990s: golden years of SCA
 - RSA CRT attack, 1996
 - Timing attacks, 1996
 - SPA, DPA, 1998
- Since 1999: 100es of SCA research papers, e.g. in CHES
- But: very few (if any) documented real-world attacks to date

Contents

1. Brief look at SCA history
2. **A case study: Brekaing remote keyless entry systems**
3. Consequences and research problems

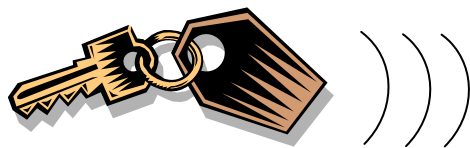
How do Keyless Entry Systems work?

early access controls: fixed code ("password")



Modern Keyless Entry Systems

advanced theft control: rolling code



$$\text{code} = e_k(n_i)$$



rolling code (or hopping code) protects against replay attacks:

1. $\text{code} = e_k(n)$
2. $\text{code} = e_k(n+1)$
3. $\text{code} = e_k(n+2)$

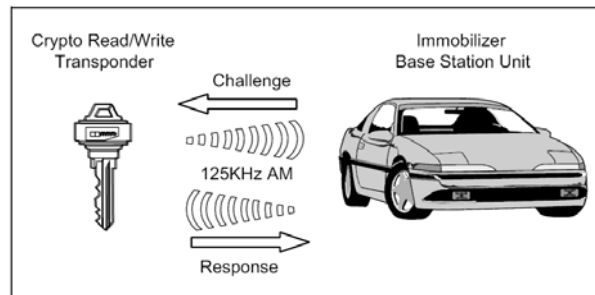
.....

$e_k()$ is often a block cipher

Popular Remote Keyless Entry Cipher: KeeLoq



HCS410 IMMOBILIZER
TRANSPONDER



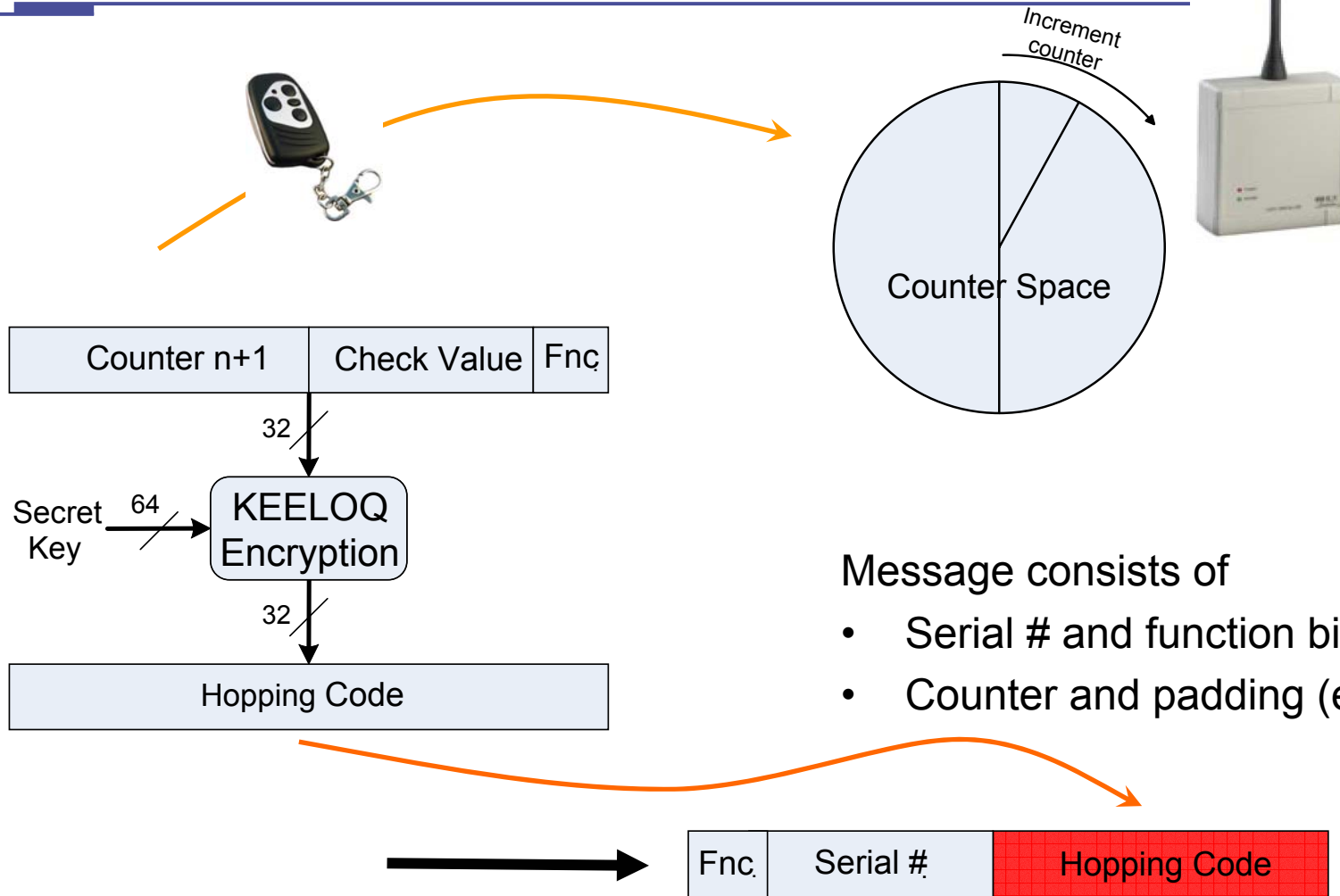
KEELOQ
CODE HOPPING



- automotive and building access authentication
- KeeLoq chip embedded in passive or active RFID transponder („car key“)
- can be used as rolling code or challenge-response
- very widely used for garage doors in US and Europe
- Wikipedia (?):
Chrysler, Daewoo, Fiat, GM, Honda, Toyota, Volvo, Jaguar, ...

Q: How secure is KeeLoq?

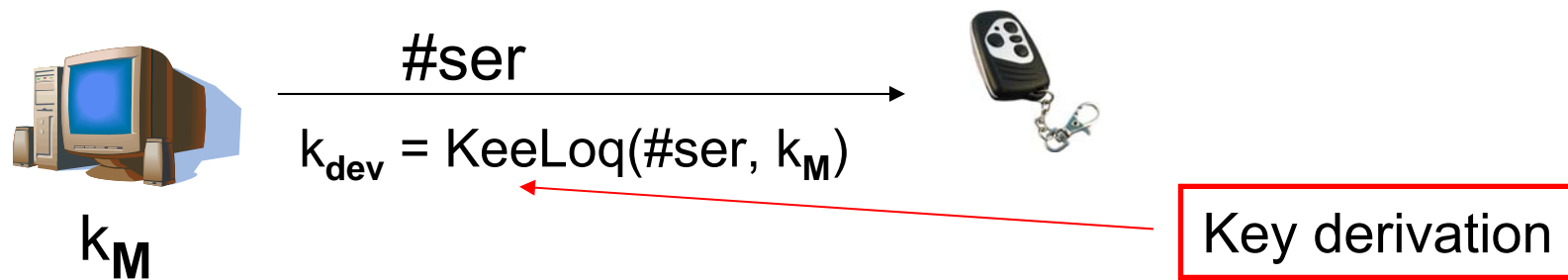
KeeLoq Rolling Code Scheme



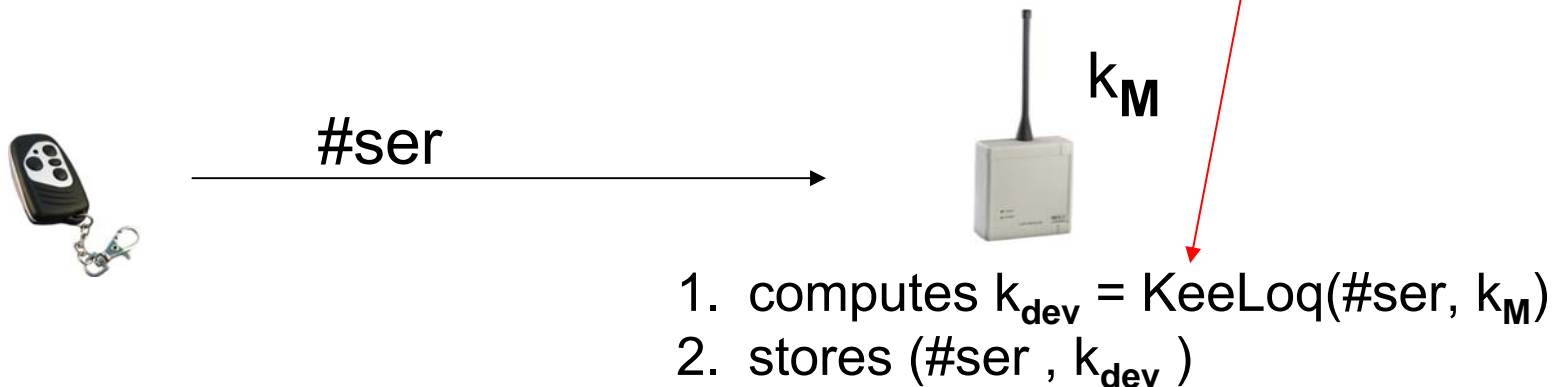
Key Management

OEM gets *Manufacturer Key* k_M assigned (burned in all its receivers)

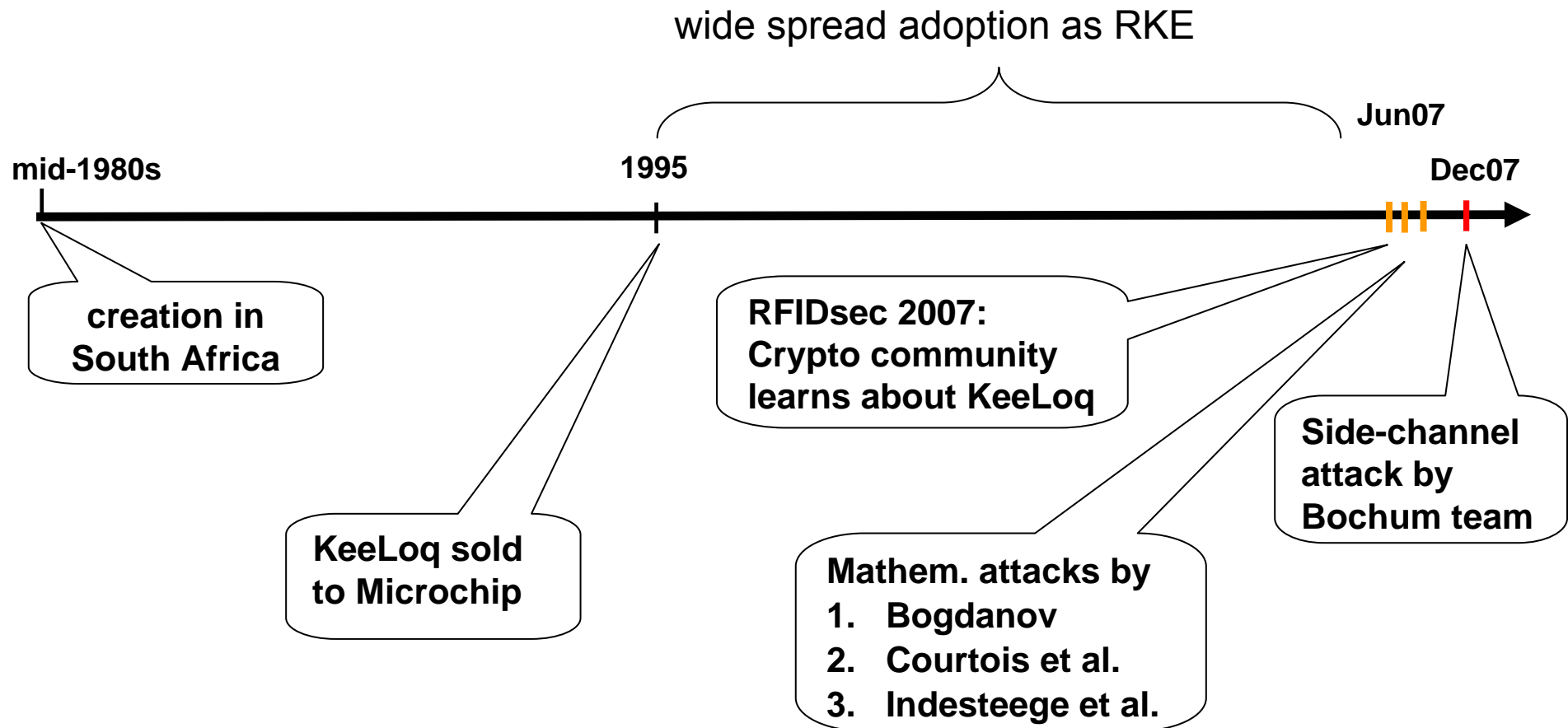
1) Creation of new remote (in secure environment)



2) Key Learning Phase of receiver (keys are never sent in clear)



Rise and Fall of KeeLoq

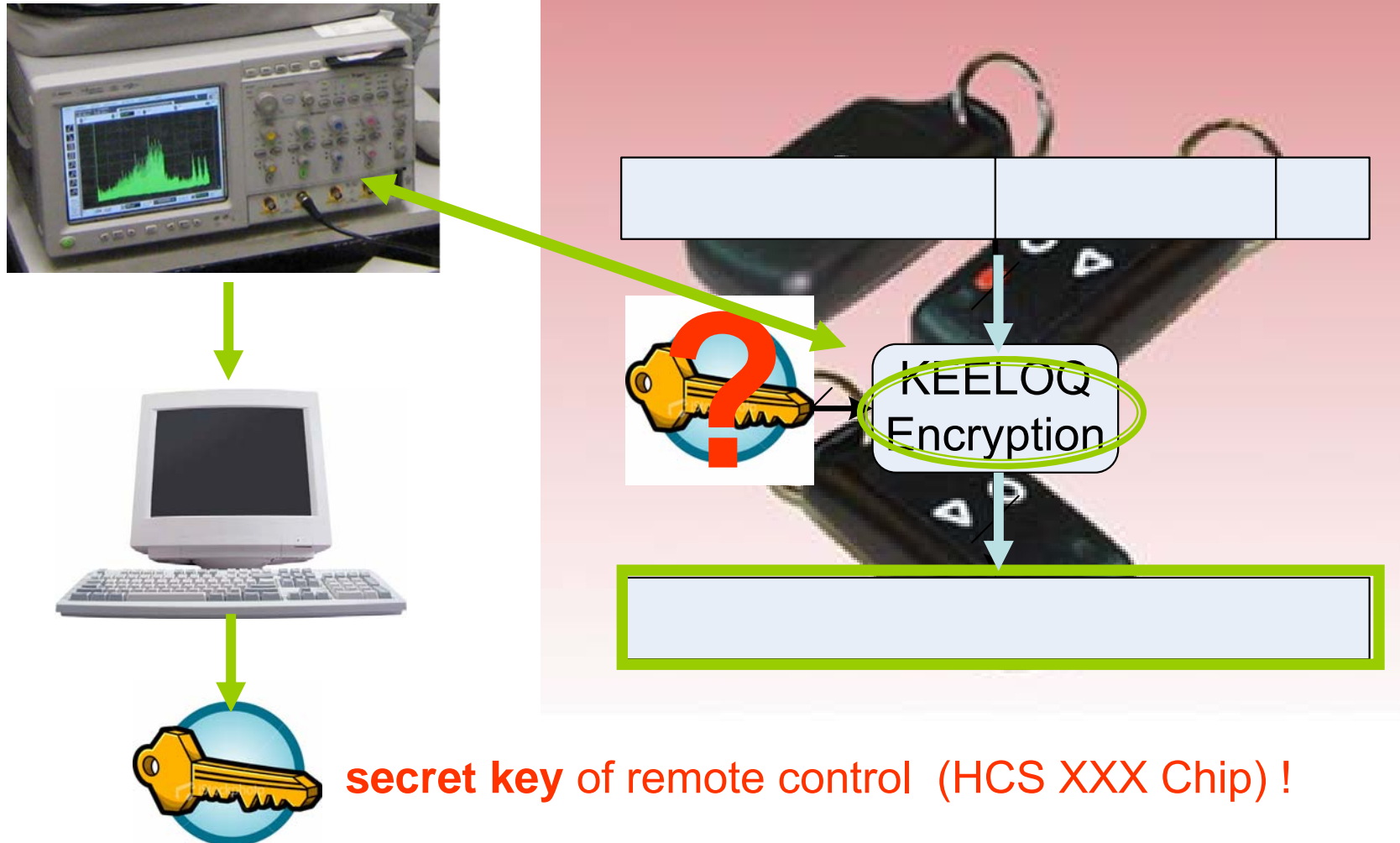


Mathematical Attacks: Recover of Manufacturer Key

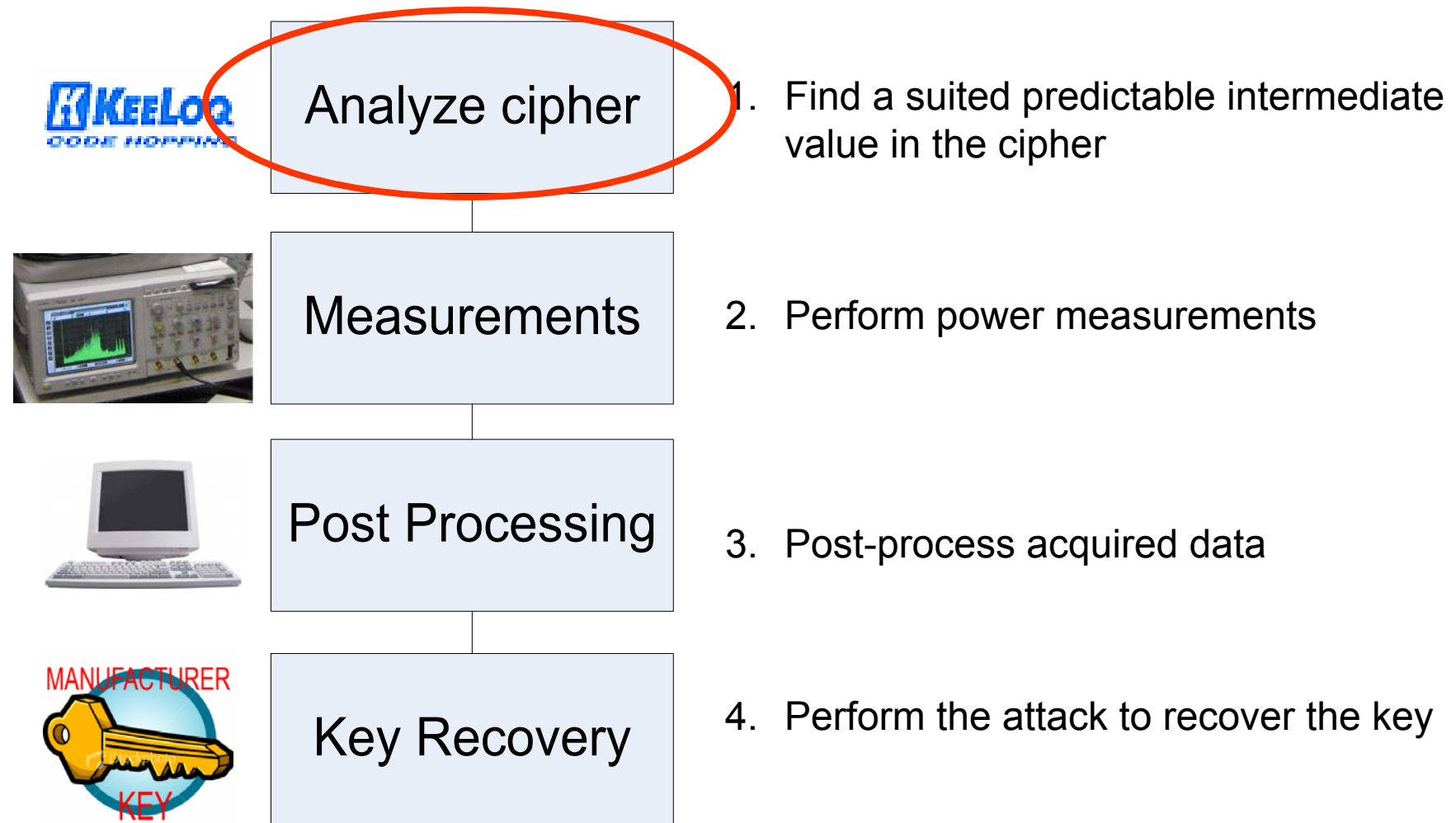
	XOR Key Derivation	KeeLoq Kee Derivation
Challenge- Response	Y	N
Rolling Code	N	N

- Mathematical attack (sliding attack) is cryptanalytically very impressive!
Device Key is recovered from 2^{16} known plain/ciphertext pairs
- **Q: How dangerous are physical attacks?**

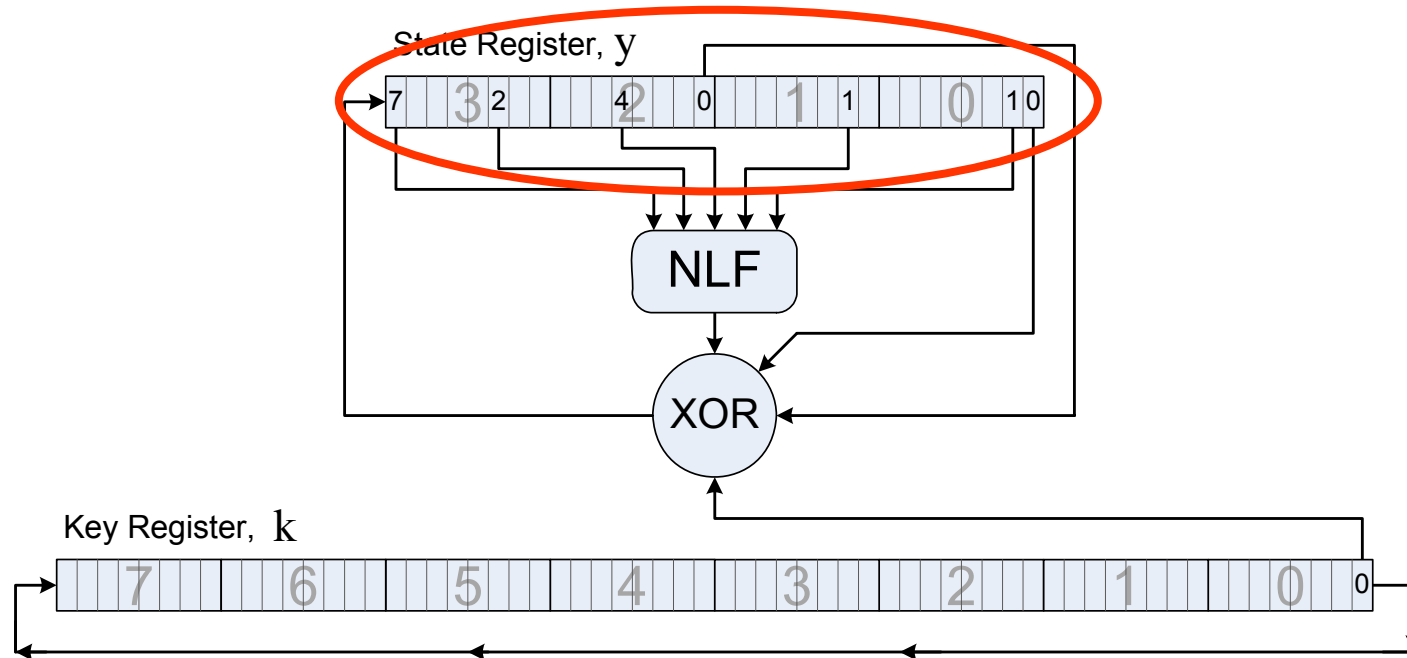
Side Channel Analysis



Performing the Side Channel Attack



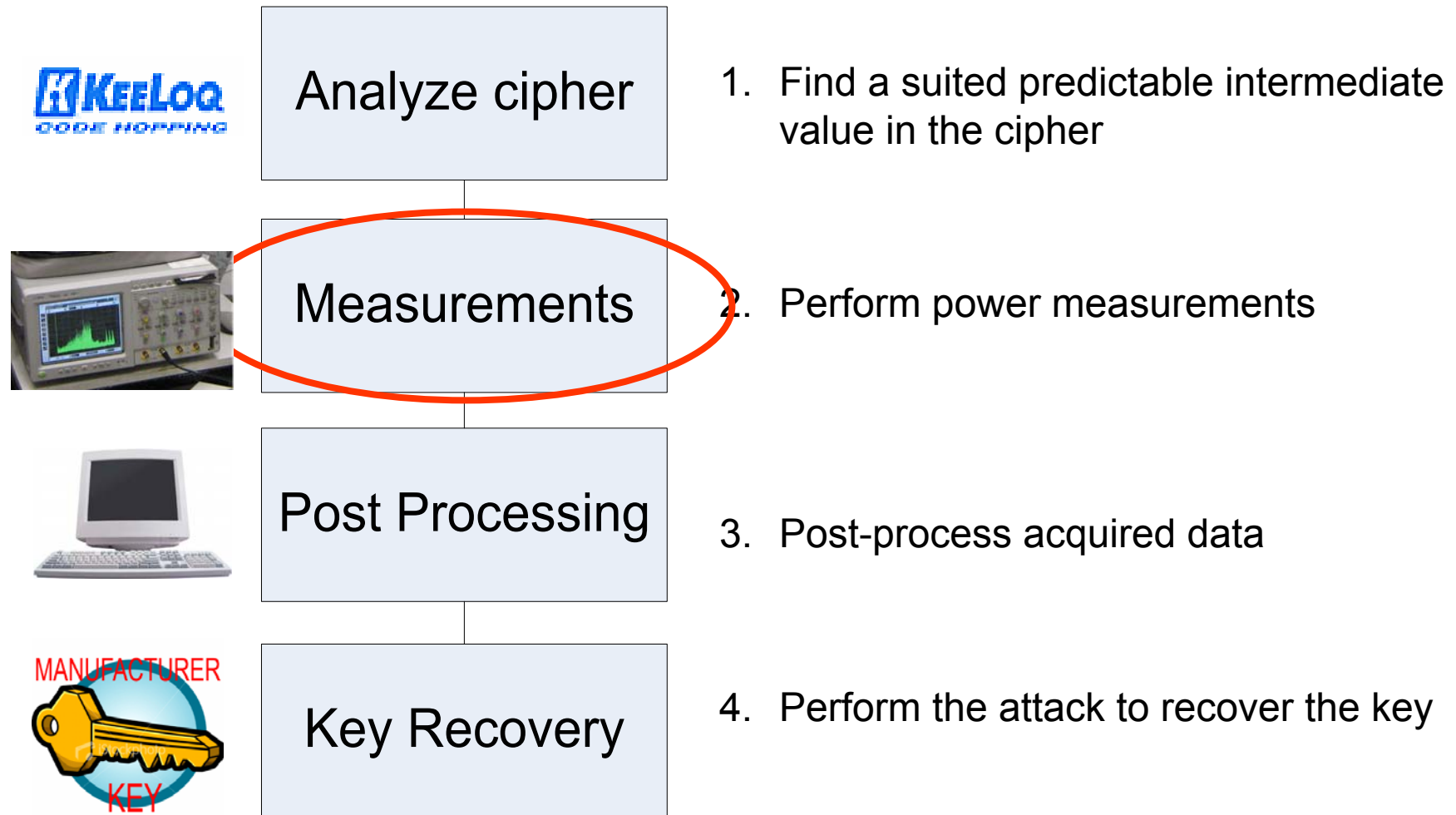
KeeLoq – The Algorithm



Power consumption is a function of the state

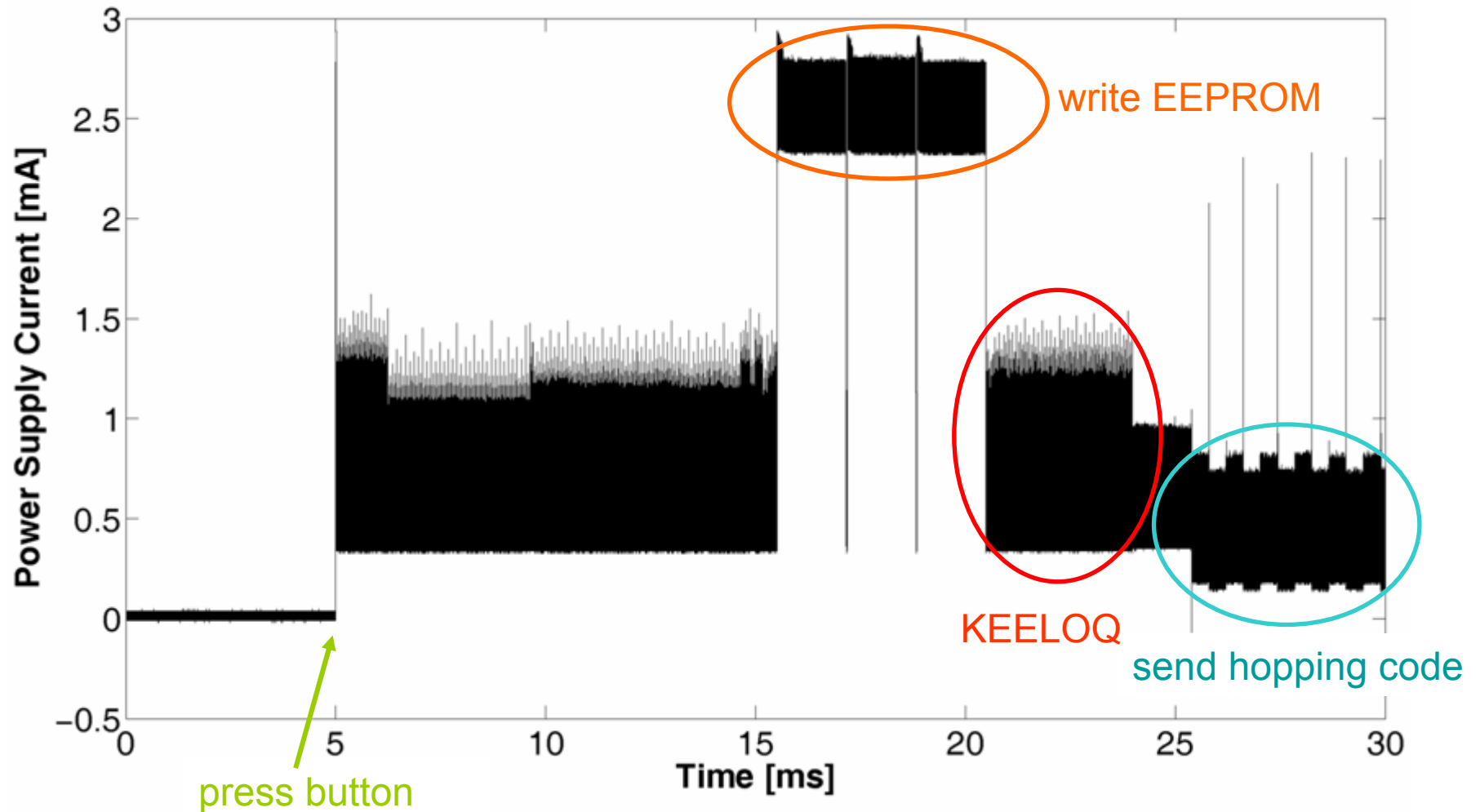
- NLF is a 5x1 non-linear function
- Simple key management (only 0s and 1s of data)
- 528 words, each word of Key Register is constant
- Changes in power consumption correspond to state

Performing the Side-Channel Attack

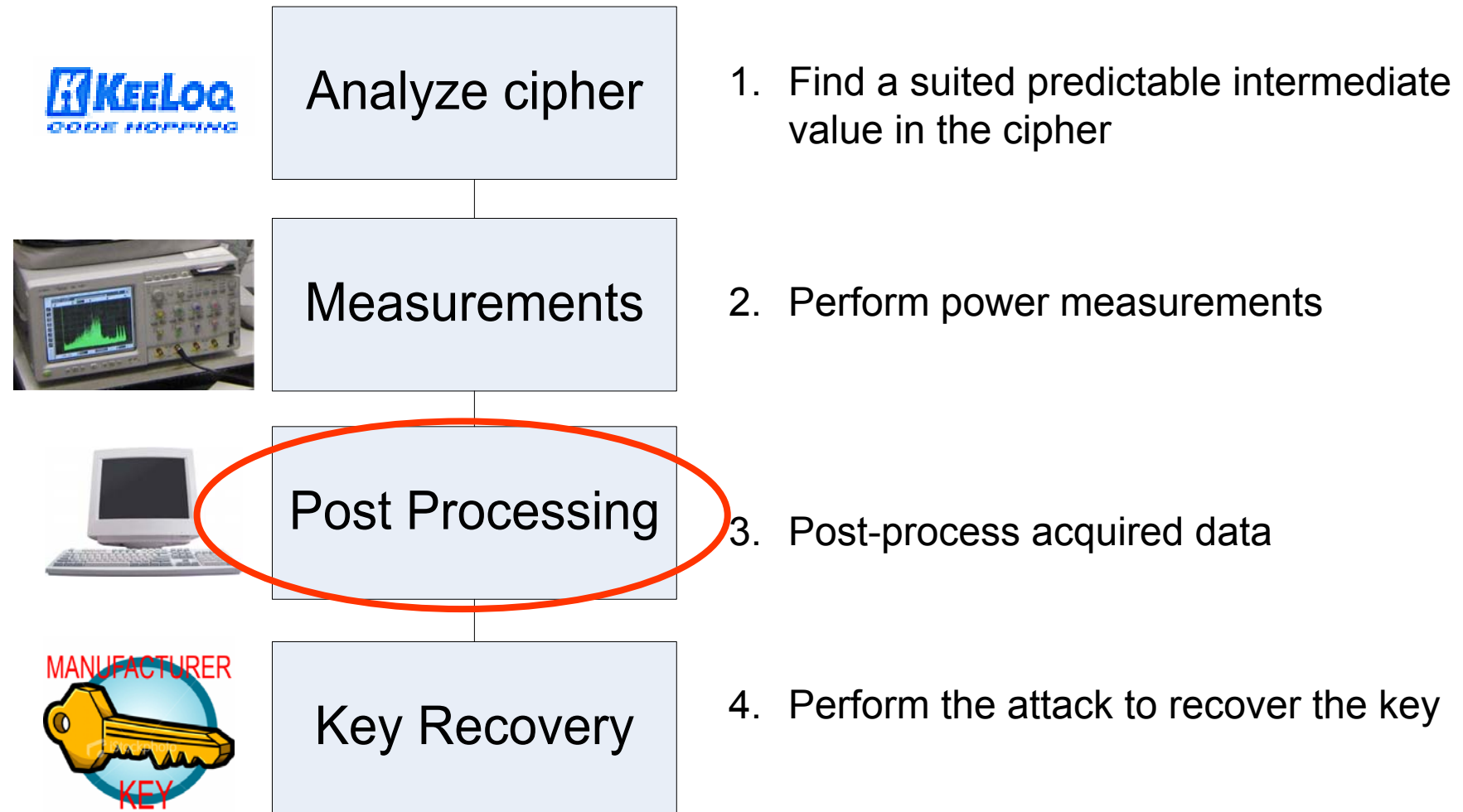


Performing the Side-Channel Attack

Finding the KEELOQ - Encryption

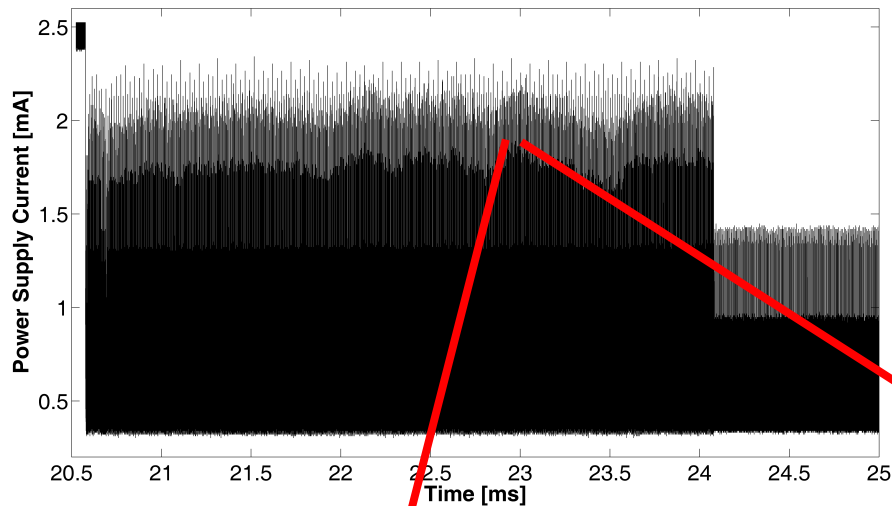


Performing the Side-Channel Attack



Performing the Side-Channel Attack

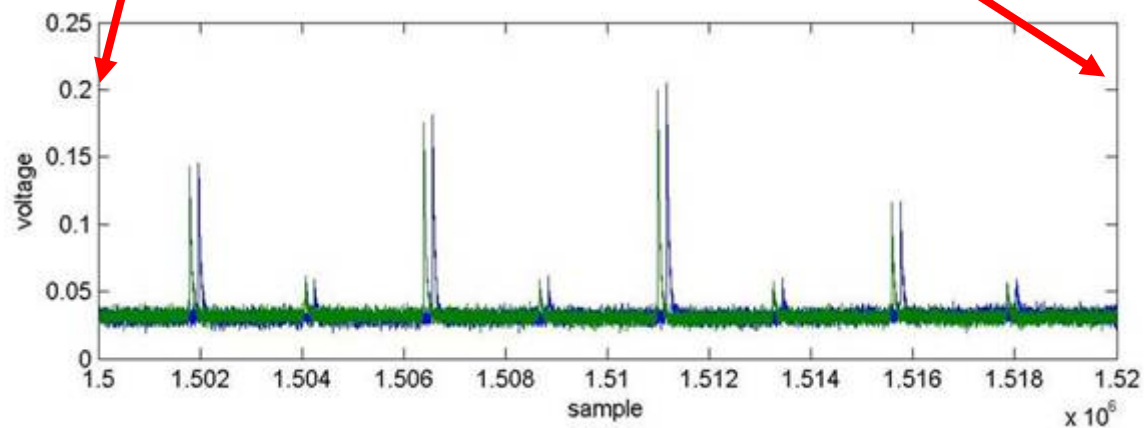
Post Processing



Problems:

- Clock jitter introduces noise
- Traces are very large

Peak detection takes care of alignment and reduces size of traces!

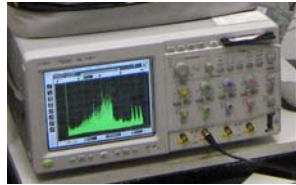


Performing the Side-Channel Attack



Analyze cipher

1. Find a suited predictable intermediate value in the cipher



Measurements

2. Perform power measurements



Post Processing

3. Post-process acquired data

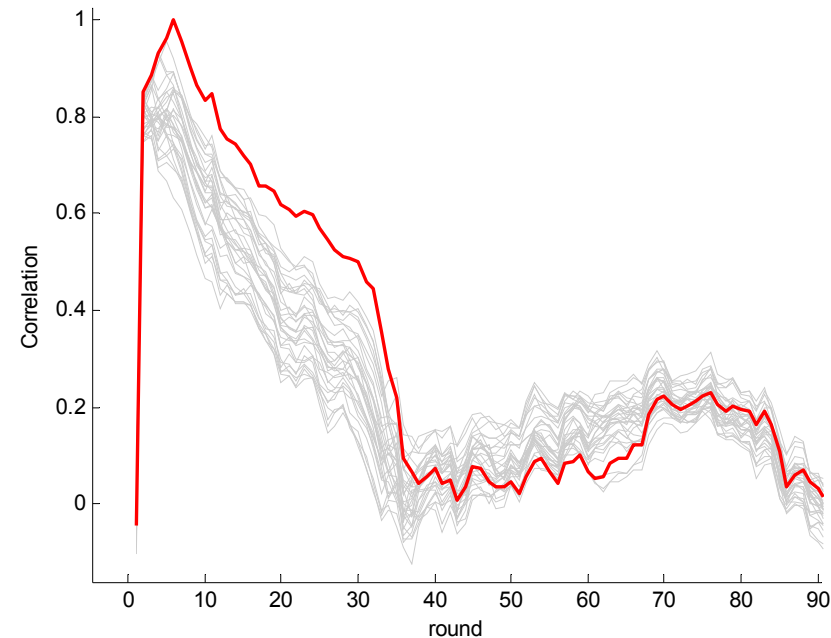


Key Recovery

4. Perform the attack to recover the key

Performing the Side-Channel Attack Key Recovery

- Correlate measured power consumption to predicted key-dependent value $y = f(x,k)$
- Divide and conquer approach
- Much off-line number crunching

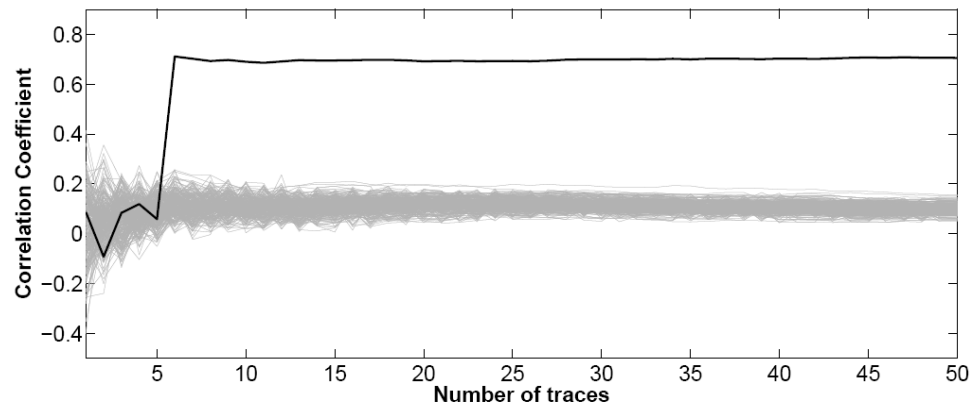


$$r(I_i(t), D(X_i, K_h)) = \frac{\sum_{i=1}^M I_i(t) \cdot D(X_i, K_h)}{\sqrt{\sum_{i=1}^M (I_i(t) - \overline{I_i(t)})^2 \cdot \sum_{i=1}^M (D(X_i, K_h) - \overline{D(X_i, K_h)})^2}}$$
$$= \frac{\frac{1}{M} \cdot \sum_{i=1}^M I_i(t) \cdot \sum_{i=1}^M D(X_i, K_h)}{\sqrt{\sum_{i=1}^M (I_i(t) - \overline{I_i(t)})^2 \cdot \sum_{i=1}^M (D(X_i, K_h) - \overline{D(X_i, K_h)})^2}}$$

Side Channel Attack Results for KeeLoq

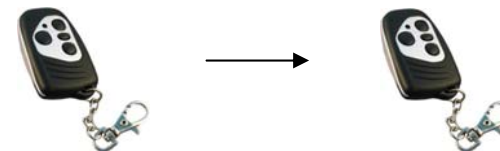
Hardware implementation („car key“)

- Total attack time (for known device family):
10-30 traces, \approx minutes



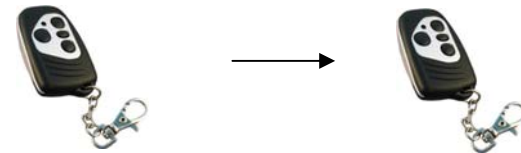
If we have access to a remote:

Recover device key and **clone the device**



So what can we do now (1) ?

1. If we have access to a remote:



Recover device key and clone the device

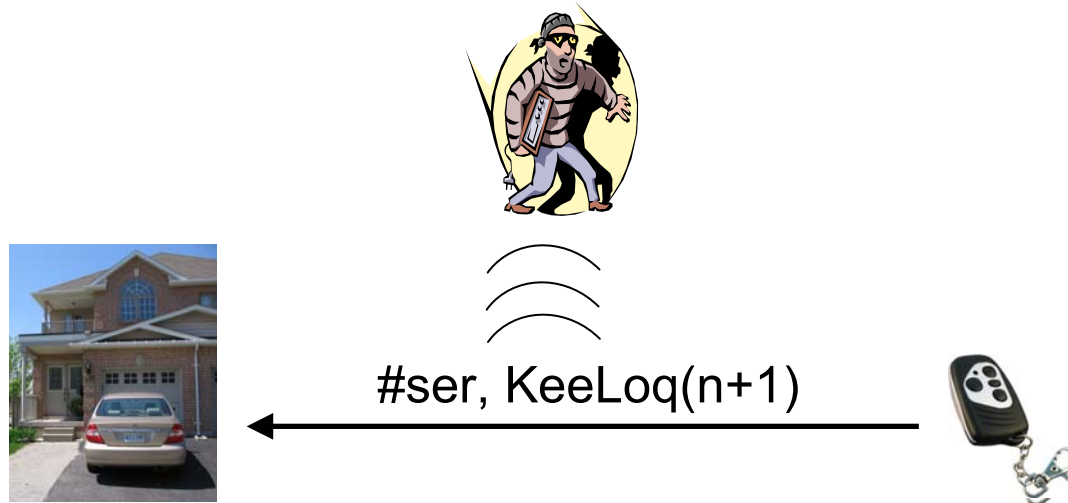
2. If we have access to a receiver:



Recover manufacturer key and generate new remotes

So what can we do now (2) ?

3. After Step 2: **Remotely eavesdrop on 1-2 communications & clone key!**



- works for all key derivation schemes
- might require a few hours of computation
(Rem: not necessary for any system we've analysed.)
- more info: www.crypto.rub/keeloq

!Side channel step (difficult) can be outsourced to criminal cryptographers!

Contents

1. Brief look at SCA history
2. A case study: Brekaing remote keyless entry systems
3. **Consequences and research problems**

Summary

Lessons learned (short version): DPA works against real systems.

- ⇒ We have to put SCA-resistance in many devices, including consumer-style applications
- ⇒ Scalability of an attack is crucial

Disclaimer: Our attack does NOT imply that real-world systems have actually be attacked via SCA by criminals (merely by researchers)

Research Problems

Easy and not-so-easy questions:

1. How do we deal with the patent situation?
2. Are there ciphers that are inherently SCA-resistant?
3. Can we construct protocols where lost of a few keys is not catastrophic? (yes?)
4. Different system design: Can PUFs protect against DPA? (no?)

More details on KeeLoq



**please see upcoming CRYPTO 2008 paper
(and current ePrint report)**

Thanks for your attention!

Prof. Dr.-Ing. Christof Paar
www.crypto.rub.de

