

In-depth crypto attacks
It always takes two bugs

Karsten Nohl <nohl@srlabs.de>



SECURITY
RESEARCH
LABS

Agenda

A risk perspective on cryptography

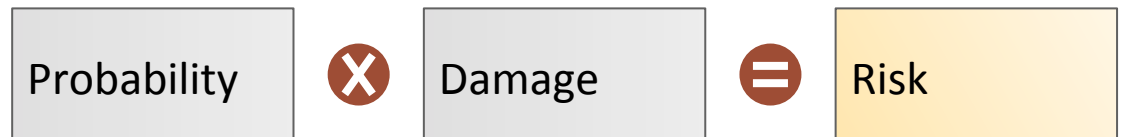
- Breaking silence –
Algebraic attacks on RFIDs
 - Ciphering the predictable –
Rainbows against mobile crypto
-

Risks summarize hacker, research, and corporate viewpoints

Research perspective



Infosec perspective



Risks summarize hacker, research, and corporate viewpoints



Agenda

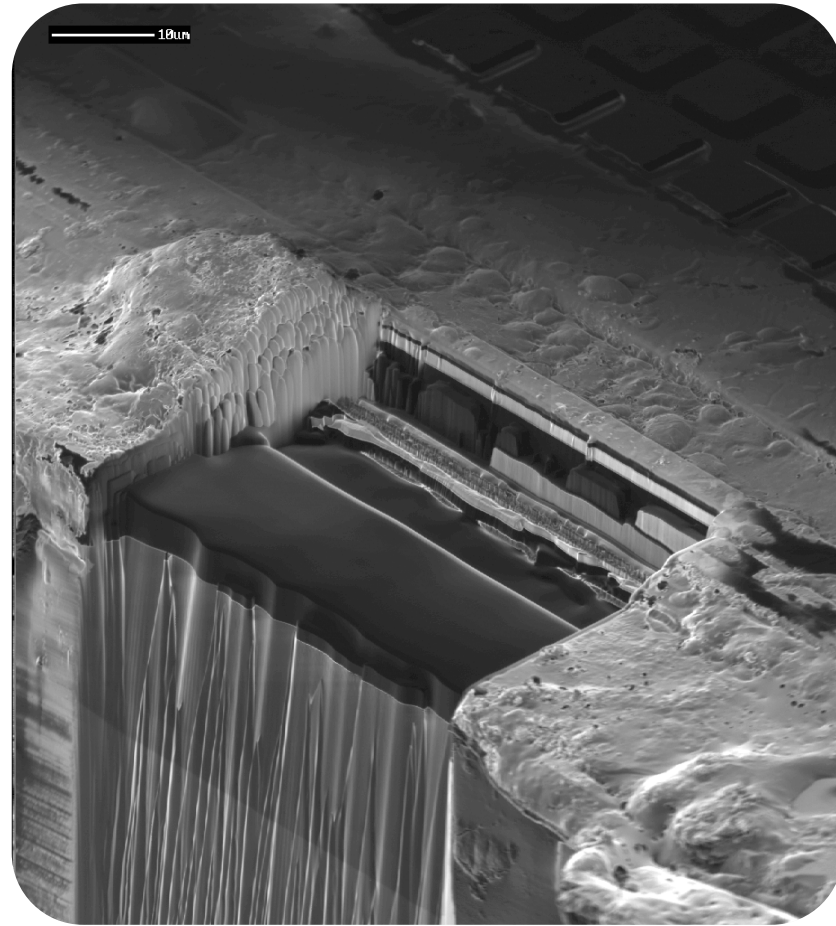
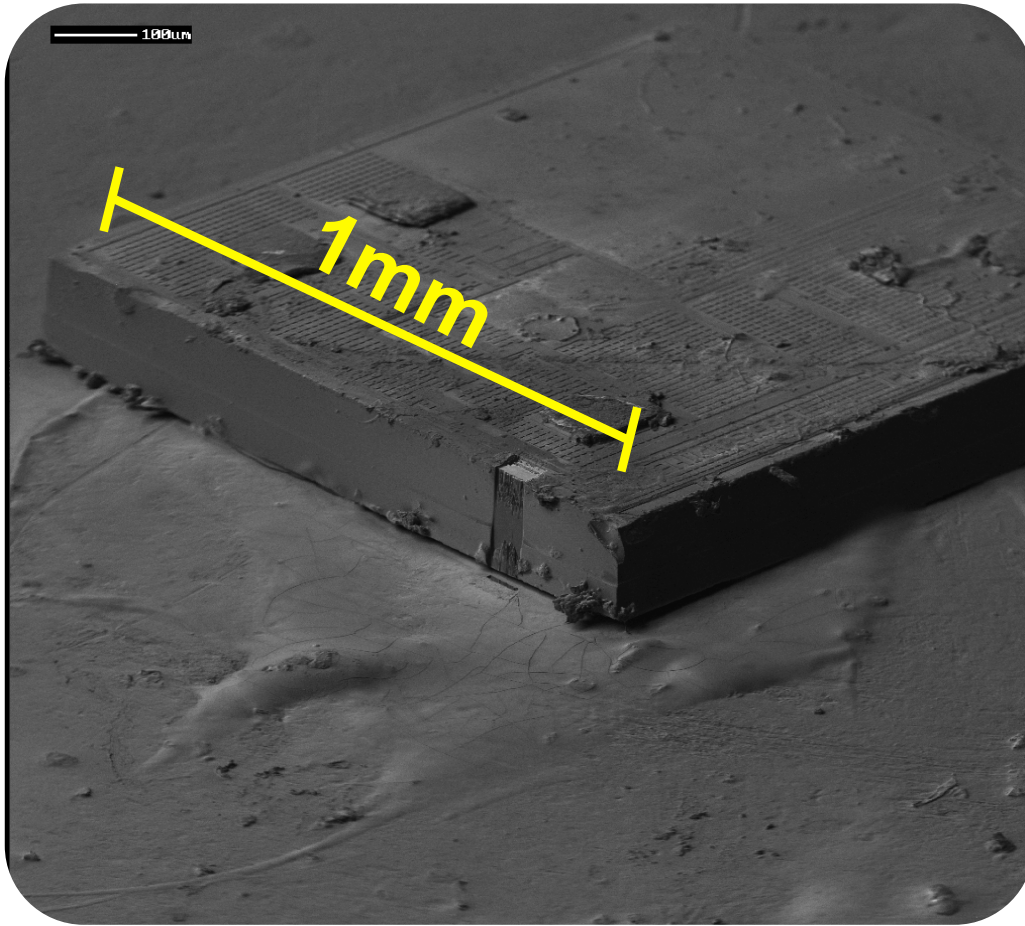
-
- A risk perspective on cryptography

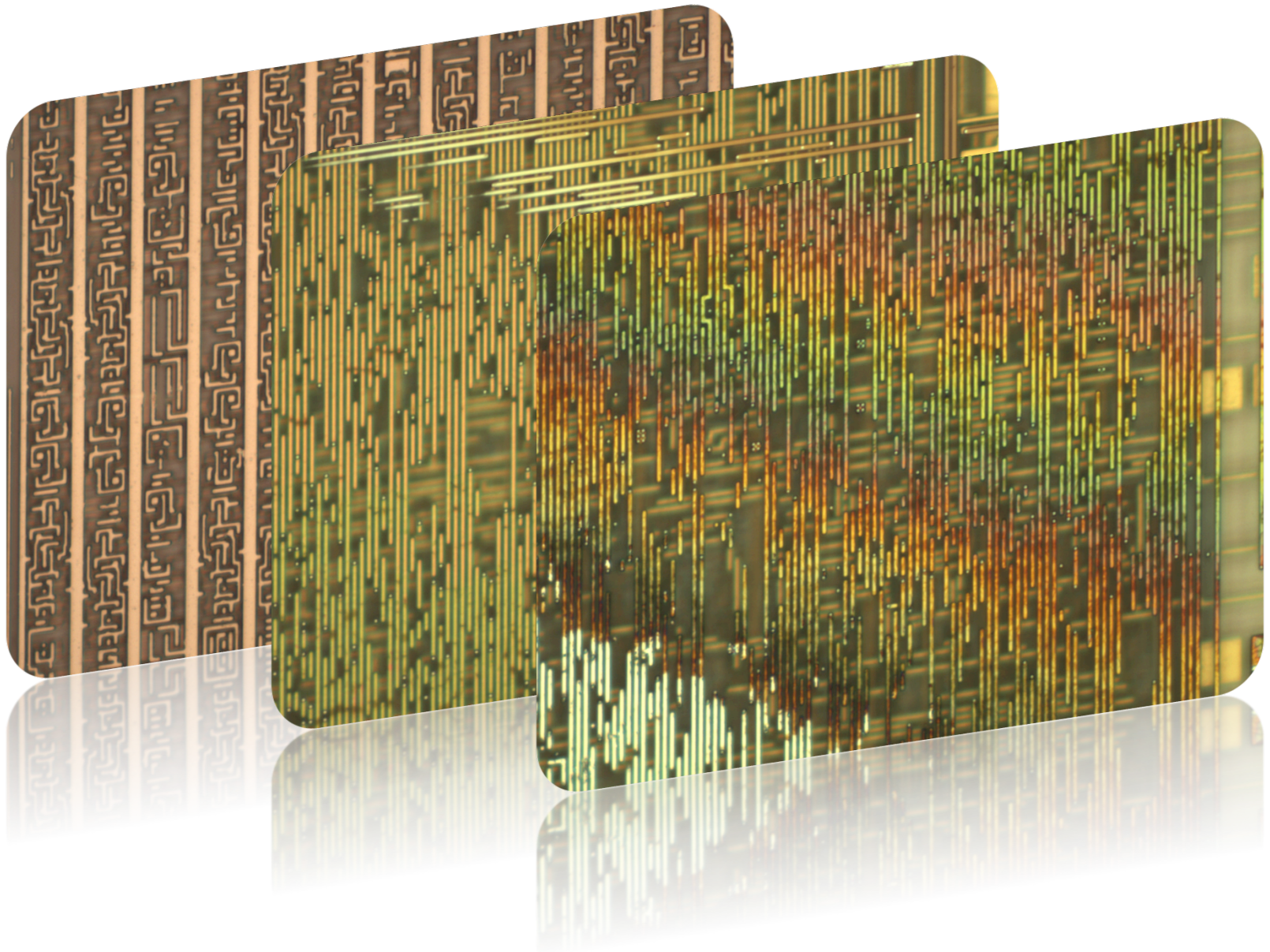


**Breaking silence –
Algebraic attacks on RFIDs**

- Ciphering the predictable –
Rainbows against mobile crypto
-

Mifare Classic RFID tags try to hide secret cipher in silicon die





Reverse-engineering is supported by *degate* software

The screenshot displays the *degate* software interface. At the top, the title bar reads "degate -- [home/martin/Development/degate3/2011-02-06_0045_degat3] [1/3]". Below the title bar is a menu bar with "Project", "View", "Tools", "Layer", "Logic", "Gate", "Recognition", and "Help". A toolbar with various icons is positioned below the menu bar. A horizontal ruler at the top of the main workspace shows coordinates from 5300 to 7000. The main workspace contains a complex logic circuit diagram with numerous gates and interconnections. A "Logic gates" panel is open on the left, displaying a table of available gates:

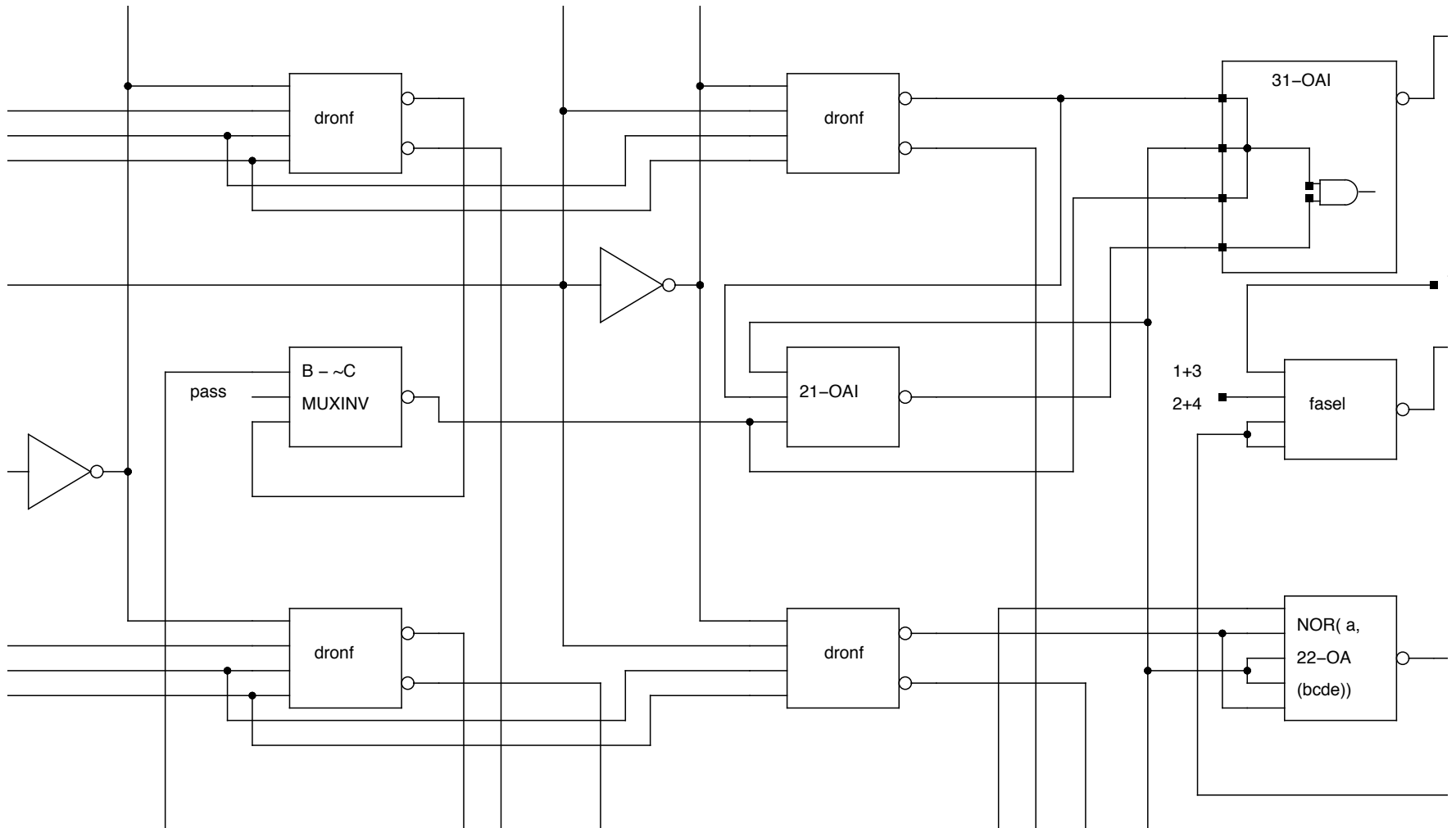
Short Name	#	Width	Height	Fill color	Frame color	Description
01-FF	58	272	134	Black	Red	D-Q-FlipFlop
02-FF	11	343	134	Black	Black	D-Q-FlipFlop with rst
03-FF	17	343	133	Black	Black	D-Q-FlipFlop with rts
04-BUF	5	226	128	Black	Black	
05-3XOR	13	249	133	Black	Black	
06-2-XNOR	12	133	133	Black	Black	

An "Edit gate" window is open in the foreground, showing the configuration for the "02-FF" gate. The "Entity" tab is selected, and the "Short name" is "02-FF" and the "Description" is "D-Q-FlipFlop with rst". The "Logic Class" is "flipFlop (generic)". The "Ports" section is as follows:

Port ID	Port Name	Port Description	In	Out
2384	!Q	!Q	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2380	Q	Q	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2388	clk	clk	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2386	D	D	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2382	rst	rst	<input checked="" type="checkbox"/>	<input type="checkbox"/>

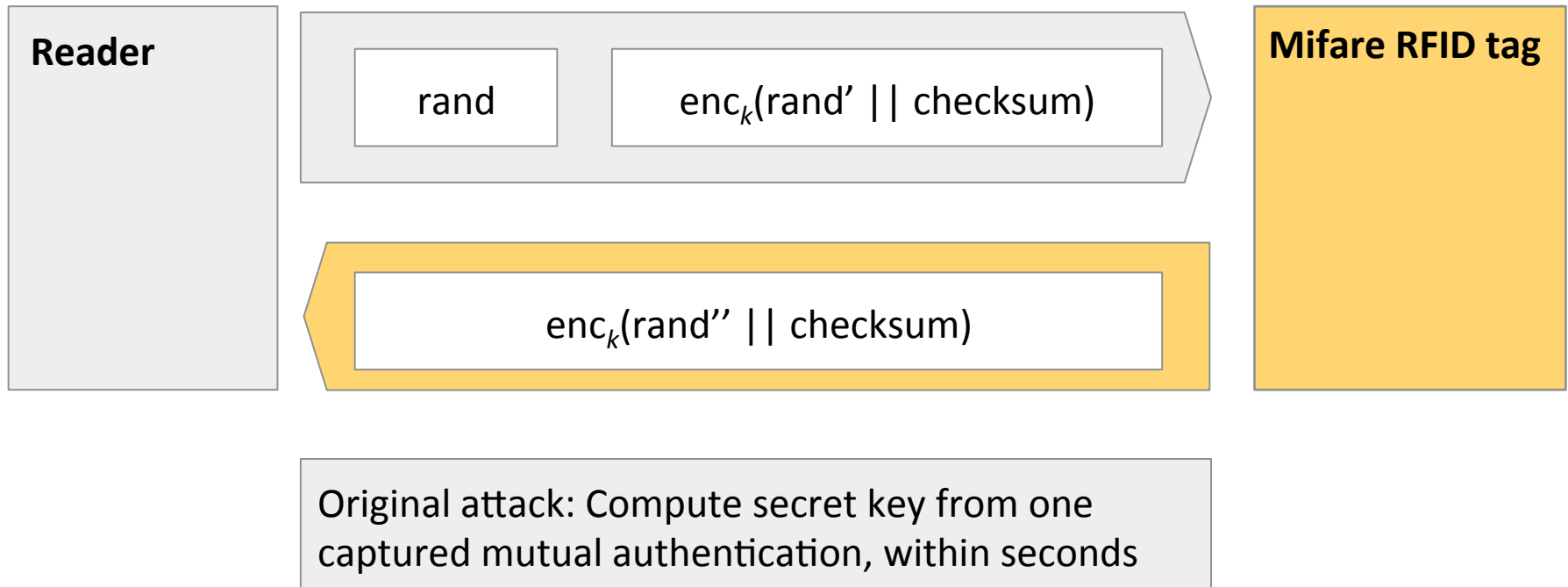
The "Fill color" and "Frame color" sections each have a color selection button and a "Reset Color" button. The "Edit gate" window has "Cancel" and "OK" buttons at the bottom. At the bottom left of the main workspace, a status bar reads "Autosaving project data ... done."

degate outputs synthesizable code that can be visualized and emulated with standard chip design tools

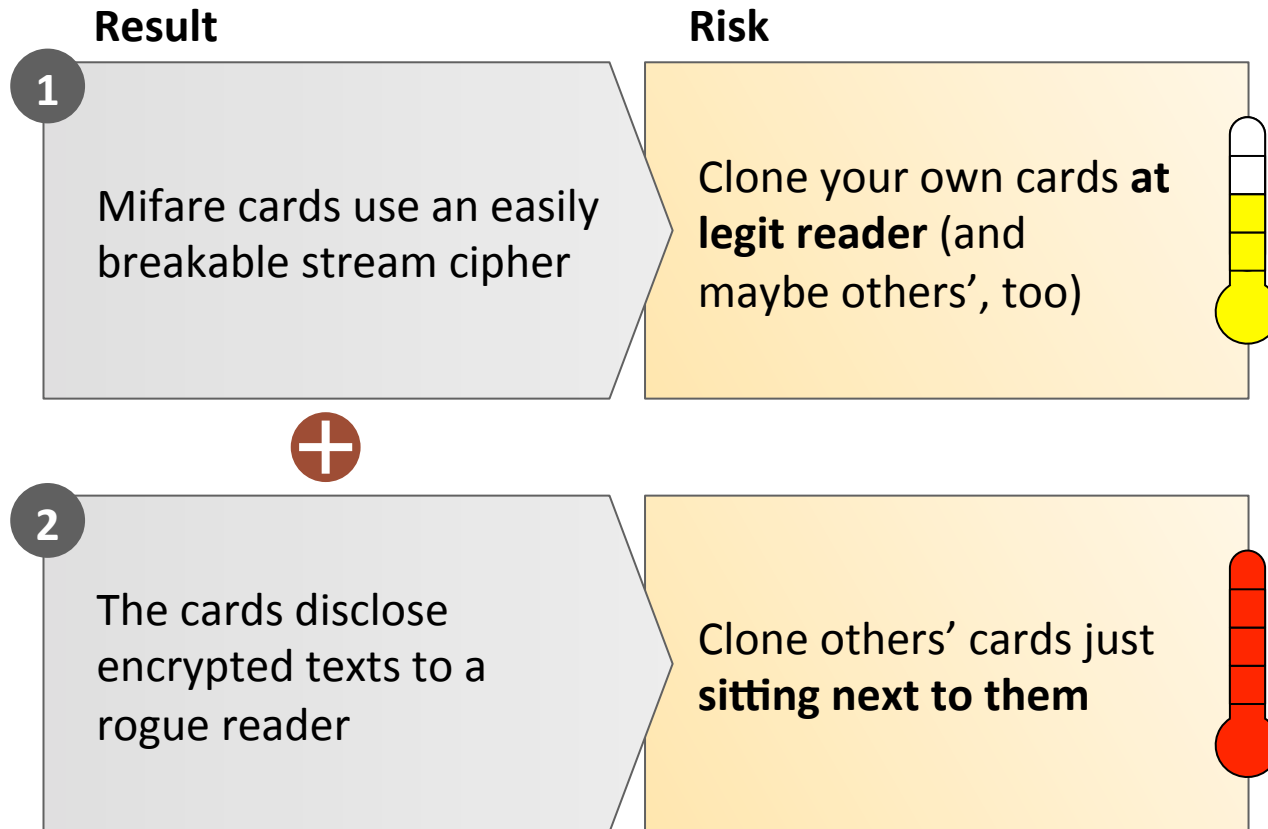


Mifare reader is authenticated first

Mutual authentication attests knowledge of k



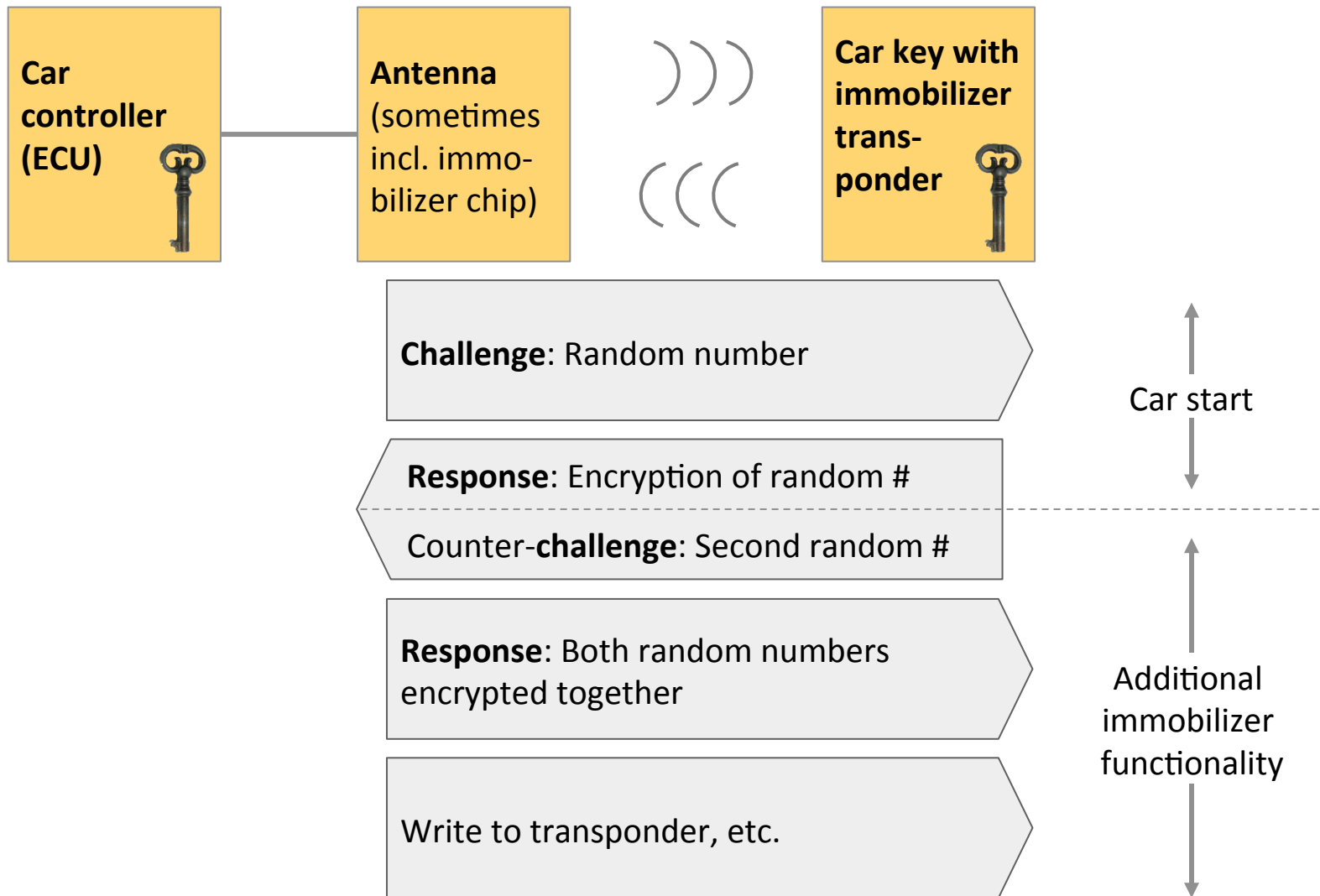
Mifare RFIDs are insecure based on two bugs



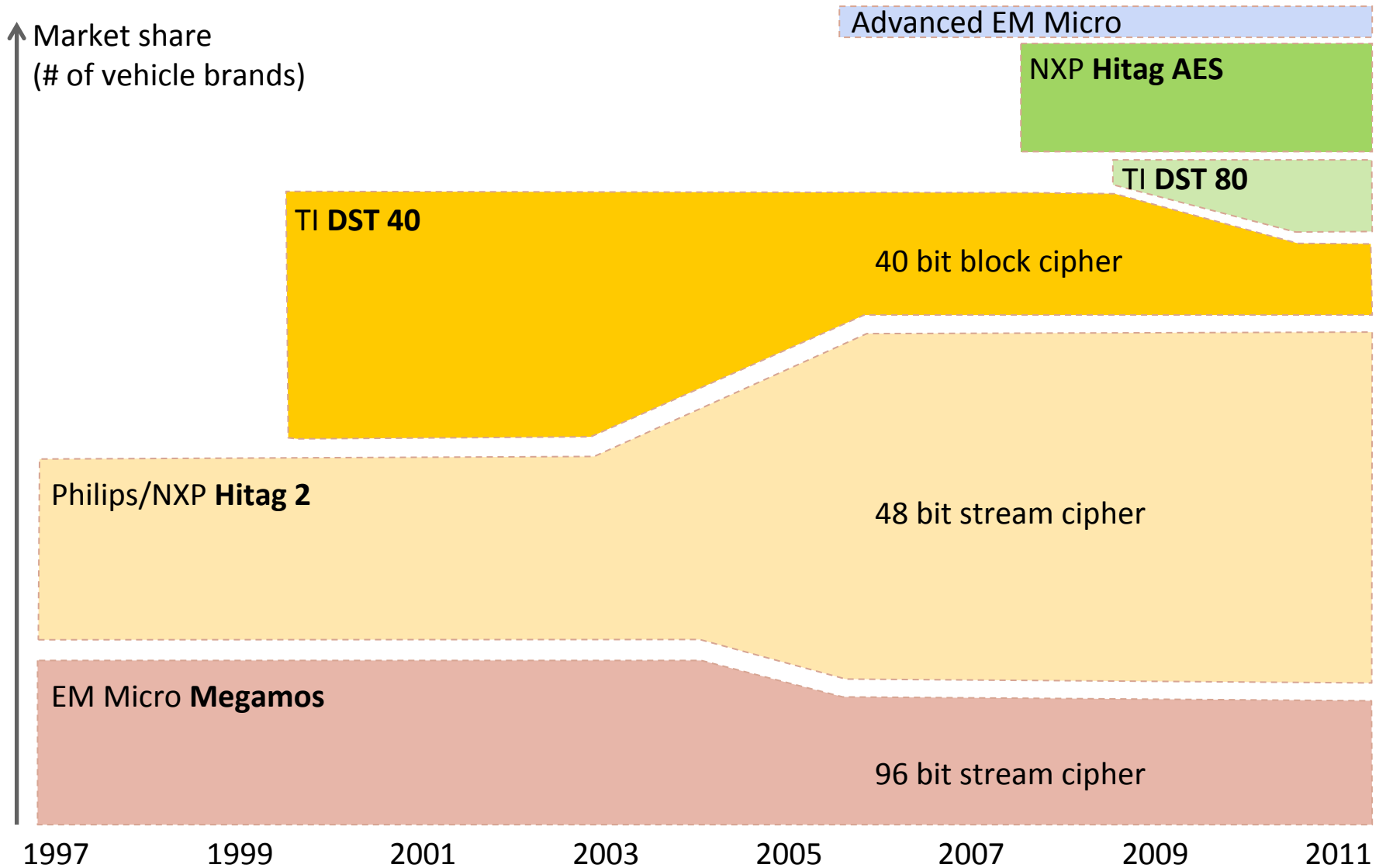
Immobilizers are the first application of IT security to cars



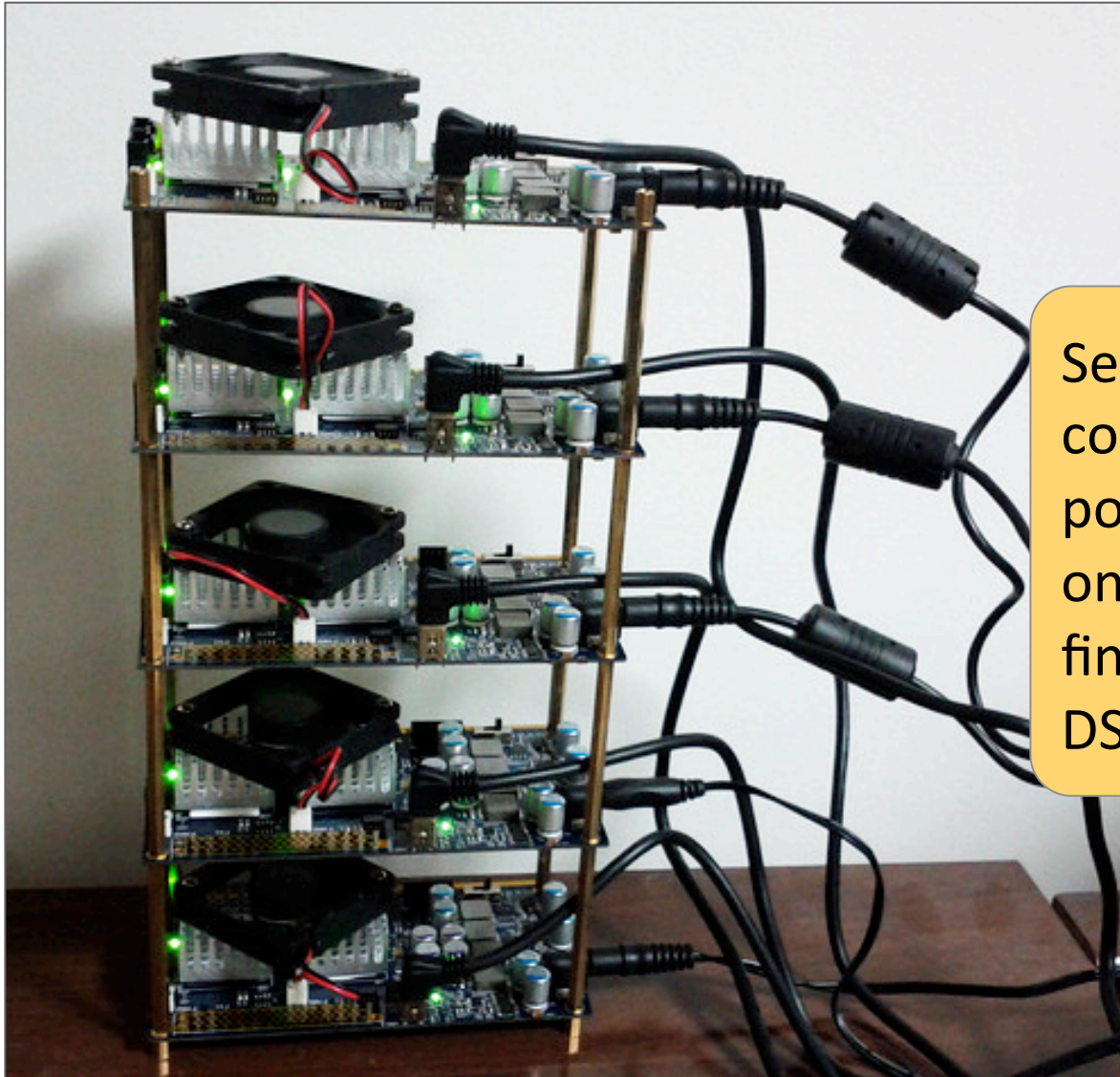
Immobilizers are simple challenge-response tokens



Three technologies dominate the immobilizer market



Immobilizer victim 1: DST40 keys are vulnerable to brute-force



Serious FPGA computing power takes one hour to find secret DST40 key

The crux of most weak ciphers is too little non-linearity

- Algebraic weaknesses in proprietary ciphers are often caused by insufficient *non-linearity*
- At the heart of the problem: *LFSRs* (linear feedback shift register)



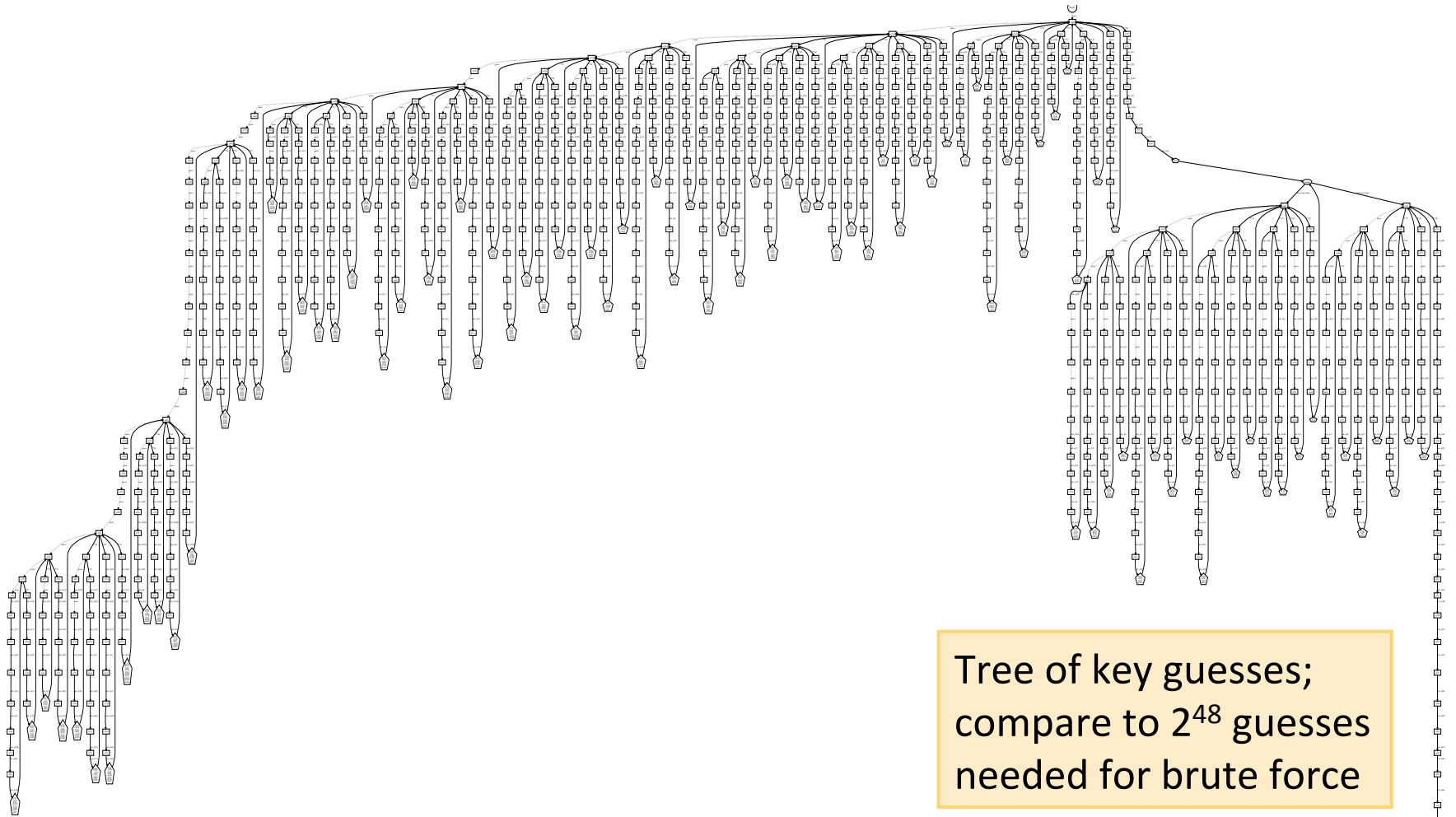
```
tmp = x[12]^x[15]^x[16]^x[17];  
for (i=17:-1:1) x[i]=x[i-1];  
x[0] = tmp;
```

Weak ciphers can be broken in three straightforward steps:

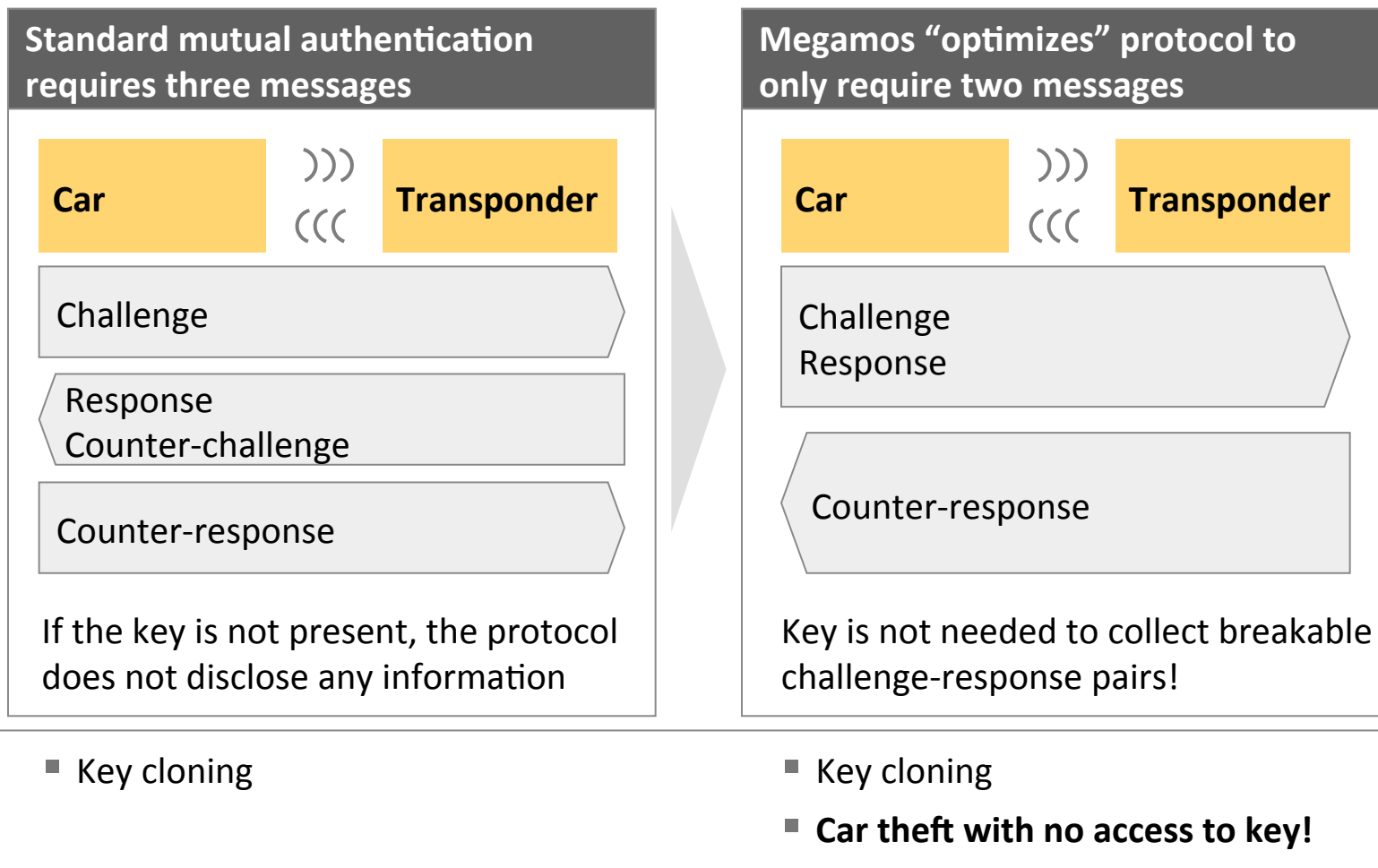
1. Describe weak parts of cipher as system of equations
 - Easiest way: Rewrite source-code to work on symbols instead of data
 - *A5/2*, for example, can be described in 656 quadratic variables
2. Brute-Force through complex parts: *Guess-and-Determine* attack.
3. Solve system of equations: *MiniSAT* is your friend

Immobilizer victim 2: Hitag2 is vulnerable to cryptanalysis

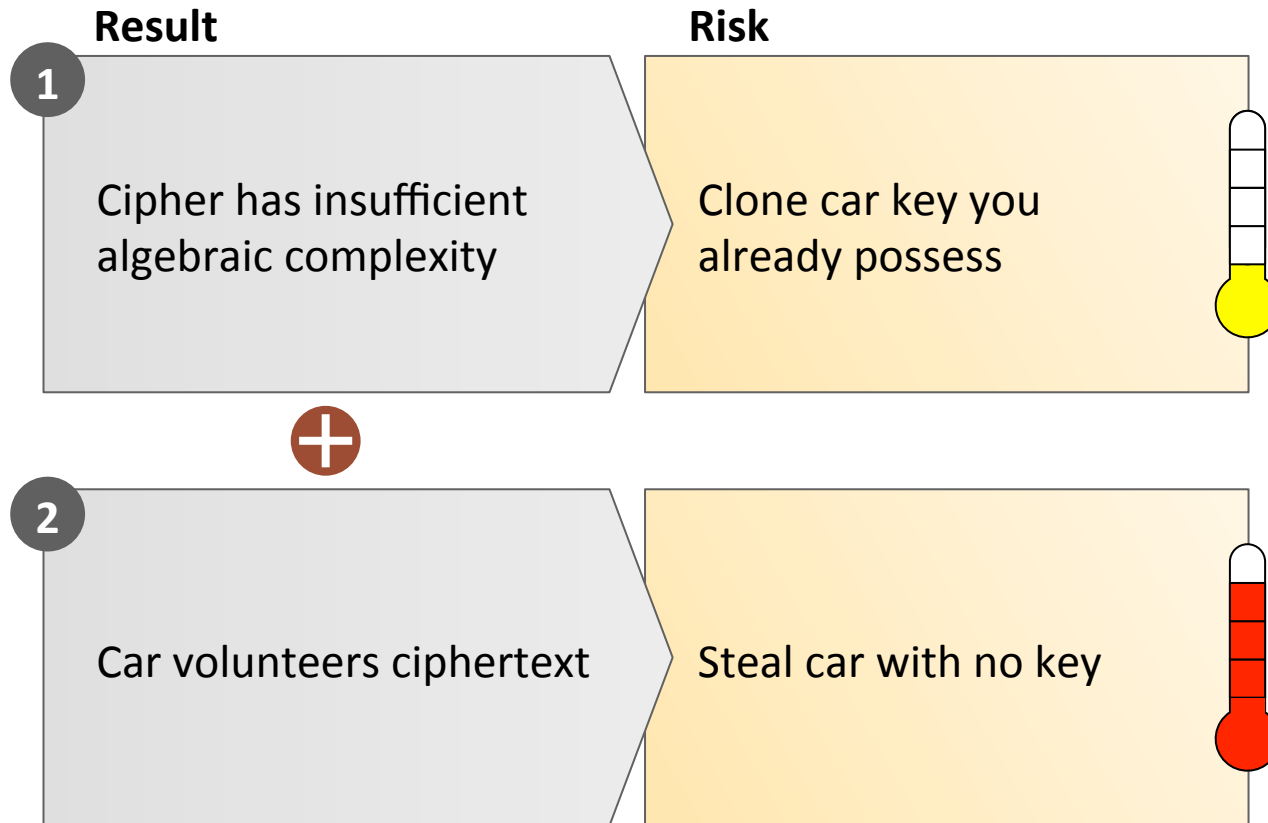
SAT solving (“smart brute force”) solves Hitag2 system of equations in minutes



Immobilizer victim 3: Megamos uses insecure authentication protocol



Megamos immobilizers are insecure based on two bugs

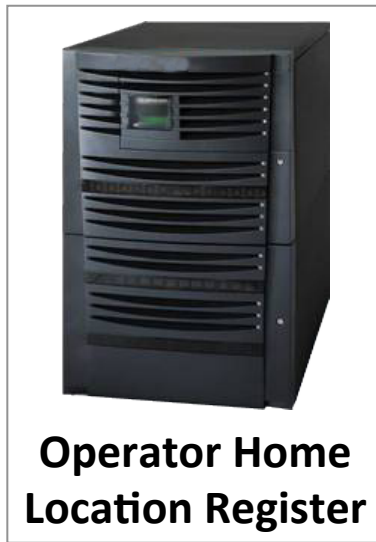
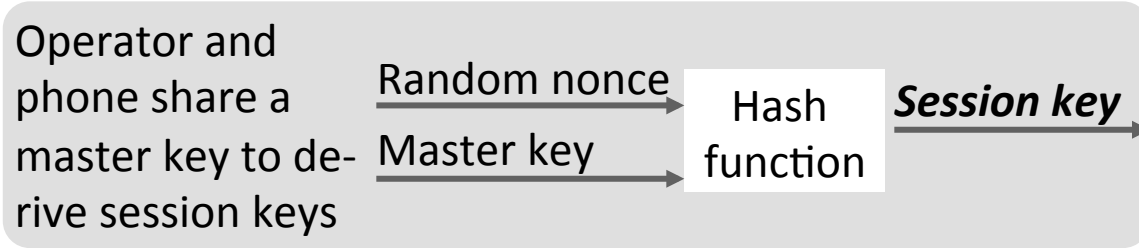


Agenda

-
- A risk perspective on cryptography
 - Breaking silence –
Algebraic attacks on RFIDs

**Ciphering the predictable –
Rainbows against mobile crypto**

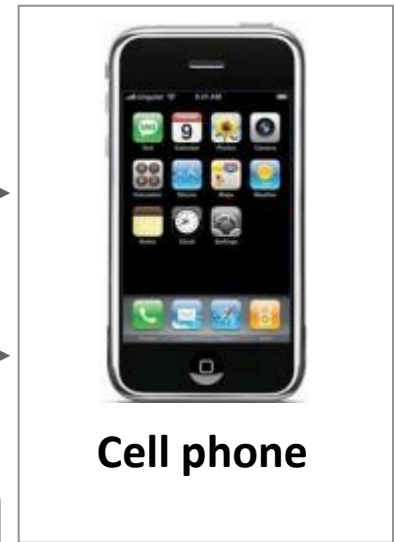
GSM uses symmetric 64-bit A5/1 session keys for call privacy



Random nonce and **session key**

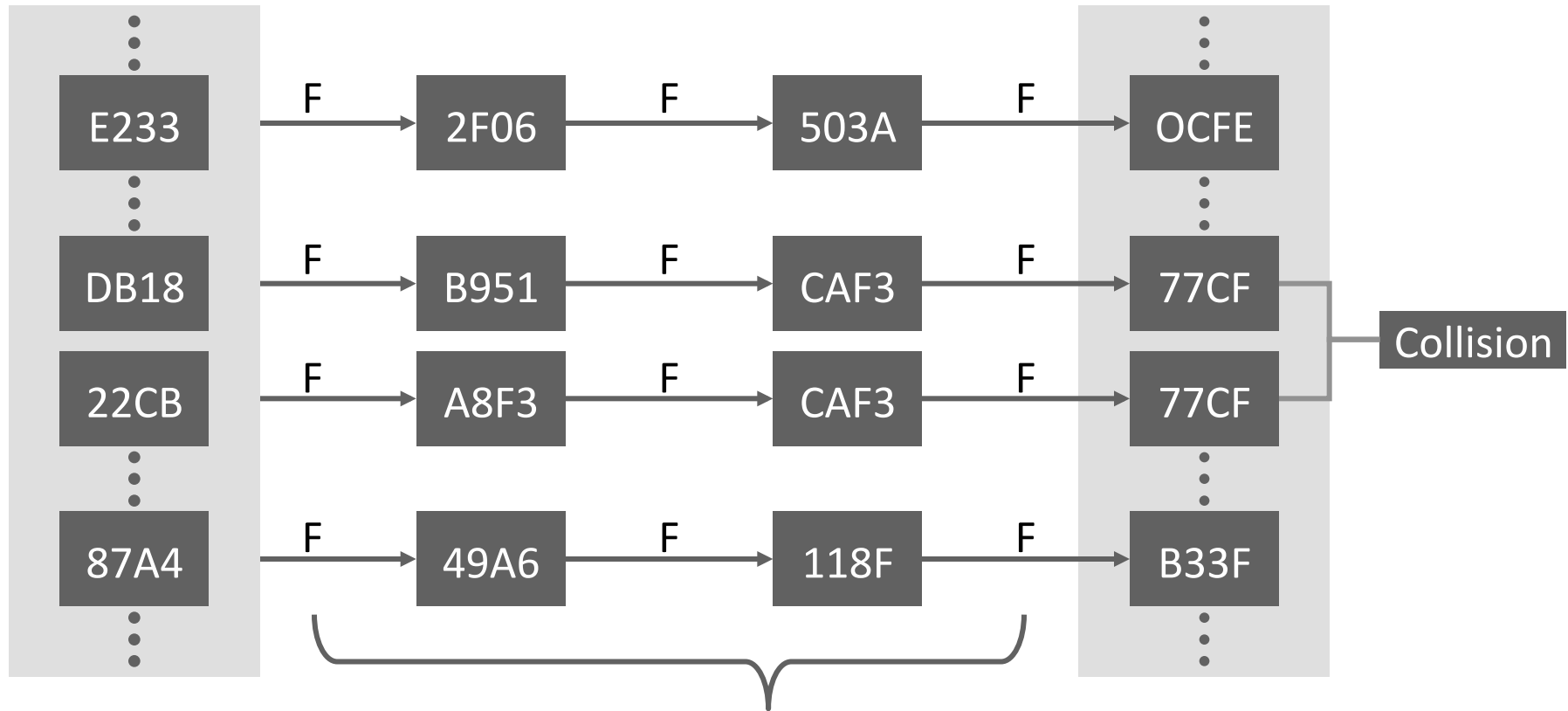


Random nonce
Communication A5/1-encrypted with **session key**



We want to crack this key

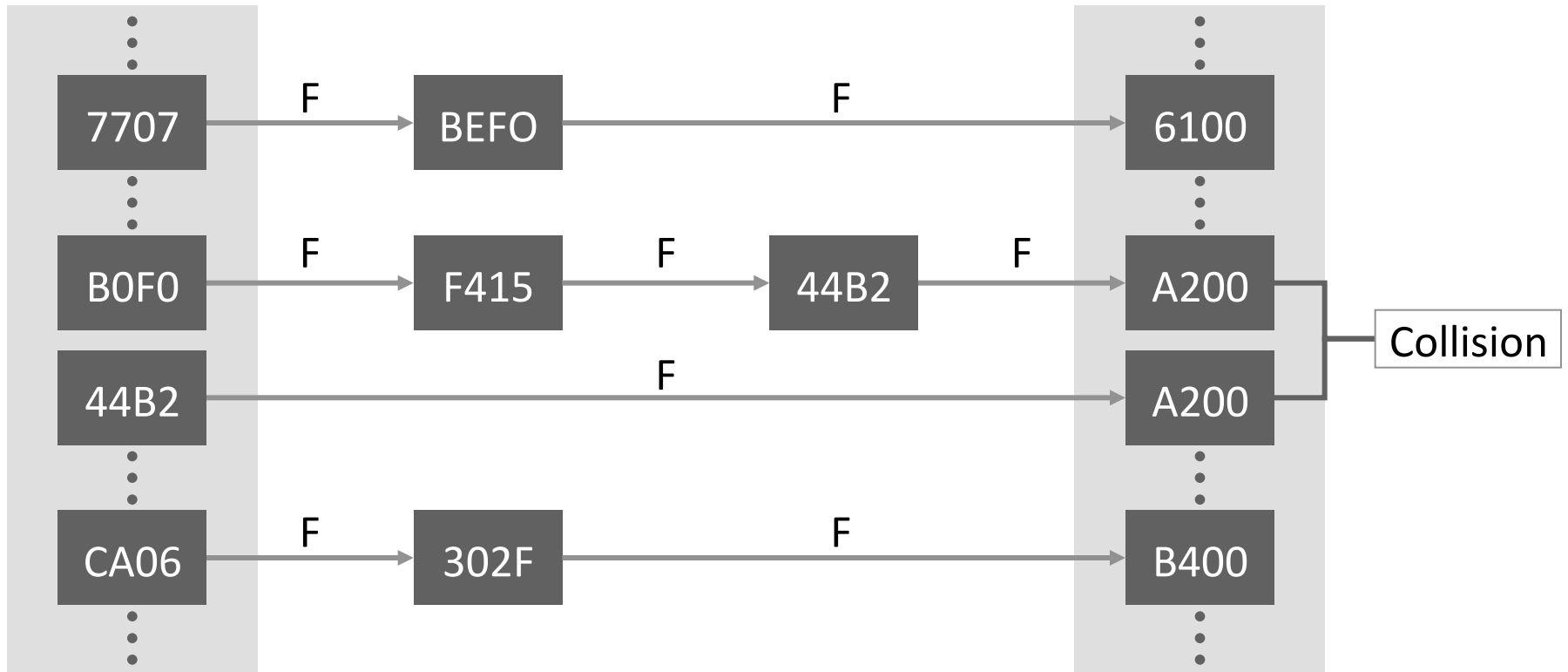
Pre-computation tables store the computed A5/1 code book in condensed form



The uncondensed code book is 100's of Petabyte. Tables provide a **trade-off**: Longer chains := a) less storage, b) longer attack time

Table optimization 1:

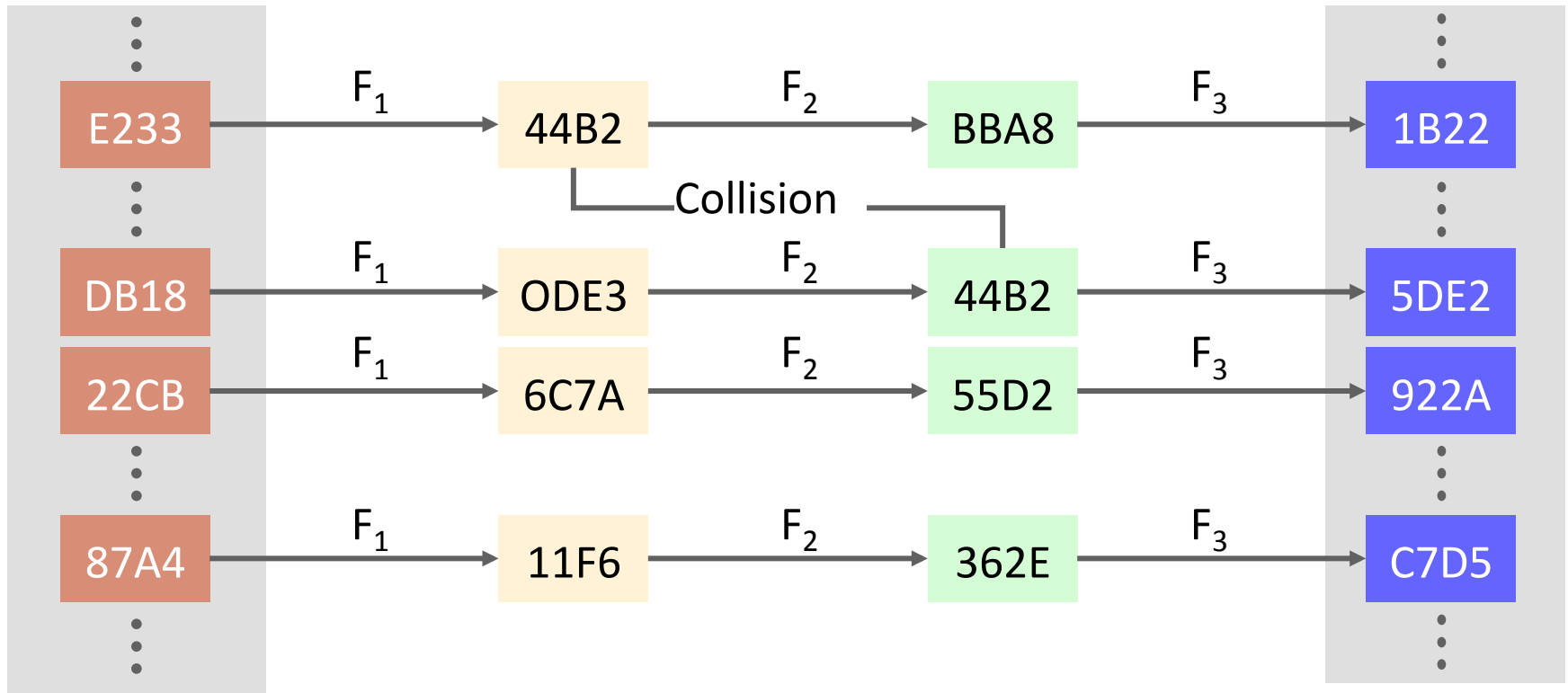
Distinguished point tables save hard disk lookups



Only one hard disk access needed instead of one for each chain link

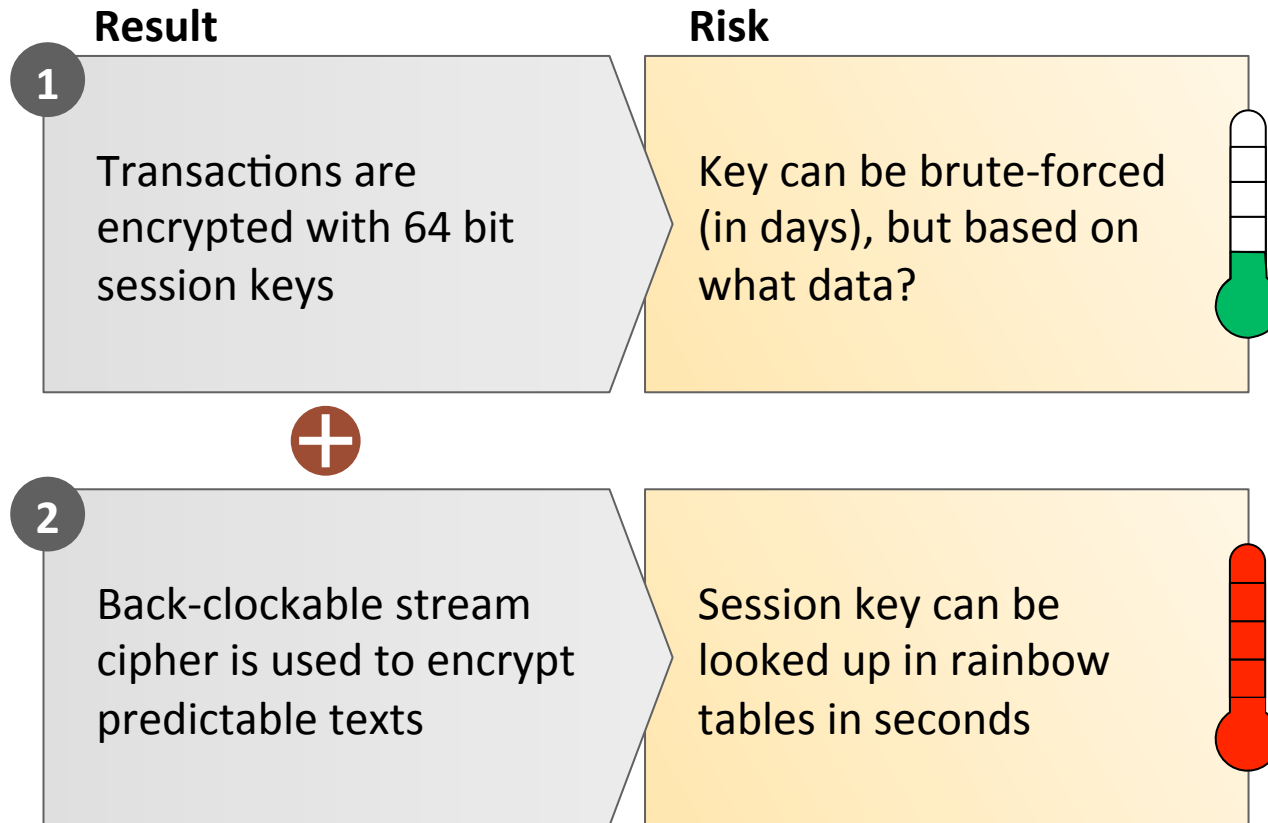
Table optimization 2:

Rainbow tables mitigate the effect of collisions



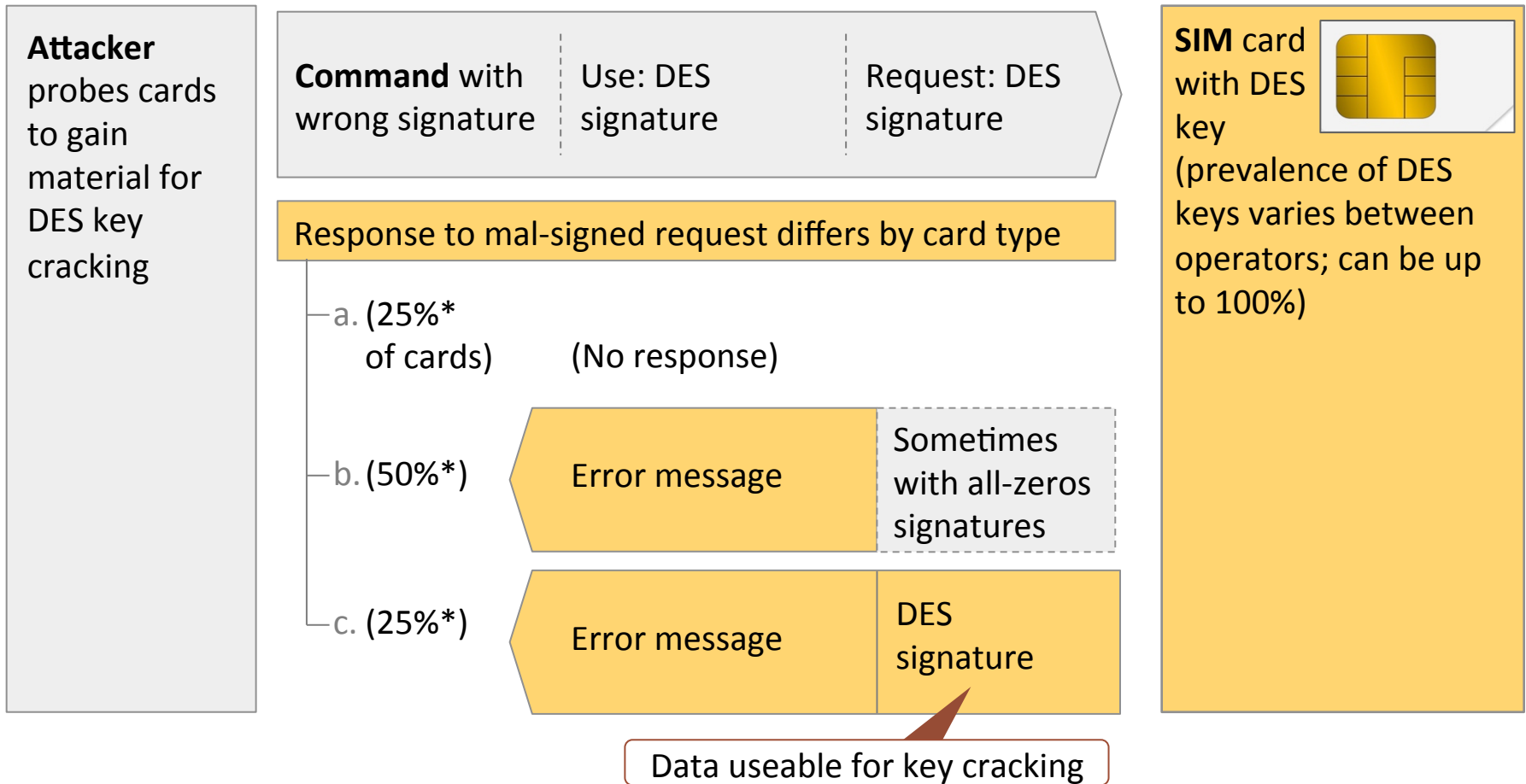
Rainbow tables have no mergers, but quadratically growing attack time

GSM calls and SMS are insecure based on two bugs



SIM cards are managed through OTA protocol, whose error handling is underspecified

Binary SMS communication



Java virus has have access to lots of abusable functionality

OTA-deployed SIM virus can access SIM Toolkit API

Standard STK function

Send SMS

Dial phone numbers, send DTMF tones

Send USSD numbers

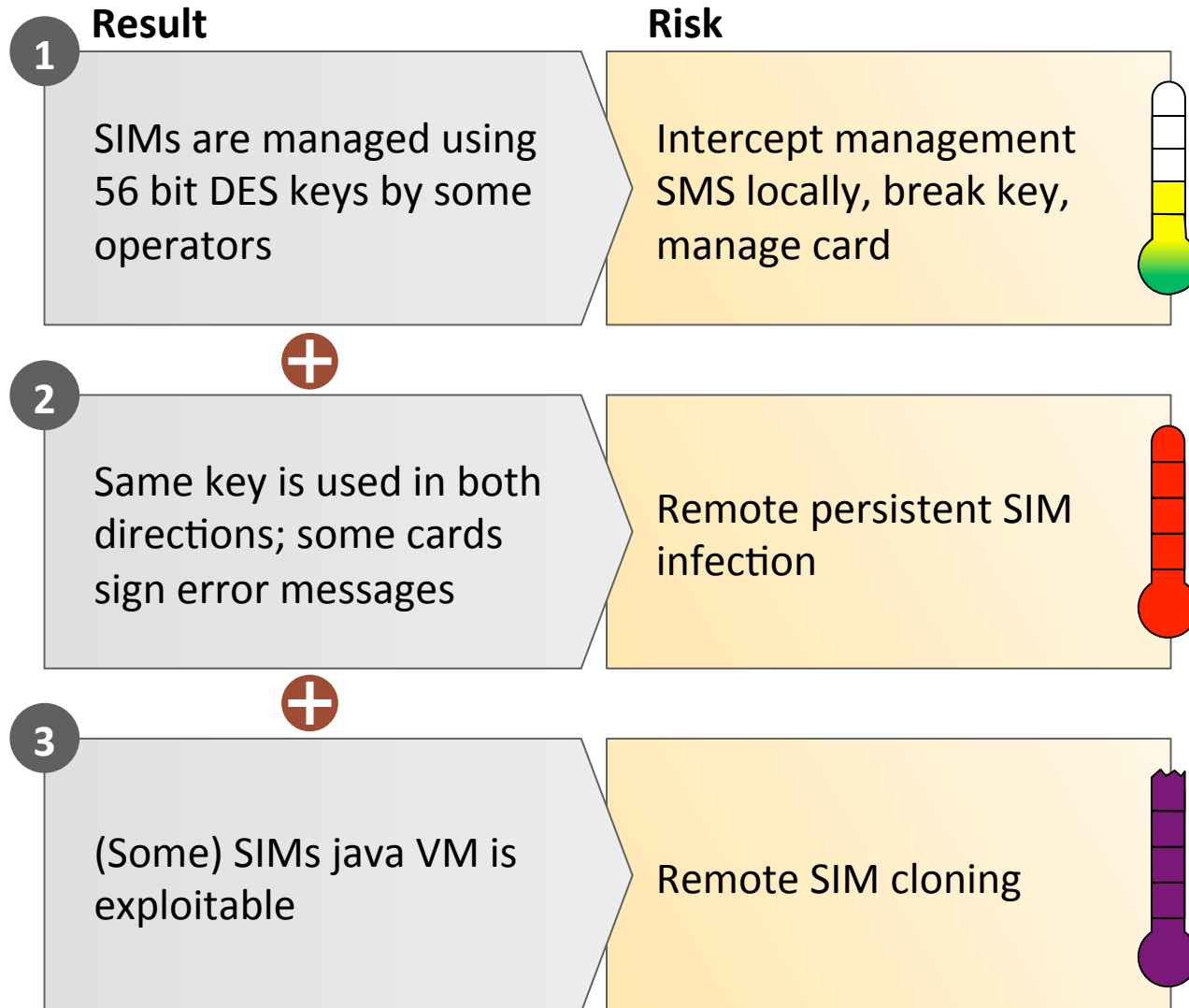
Query phone location and settings

Open URL in phone browser

Abuse potential

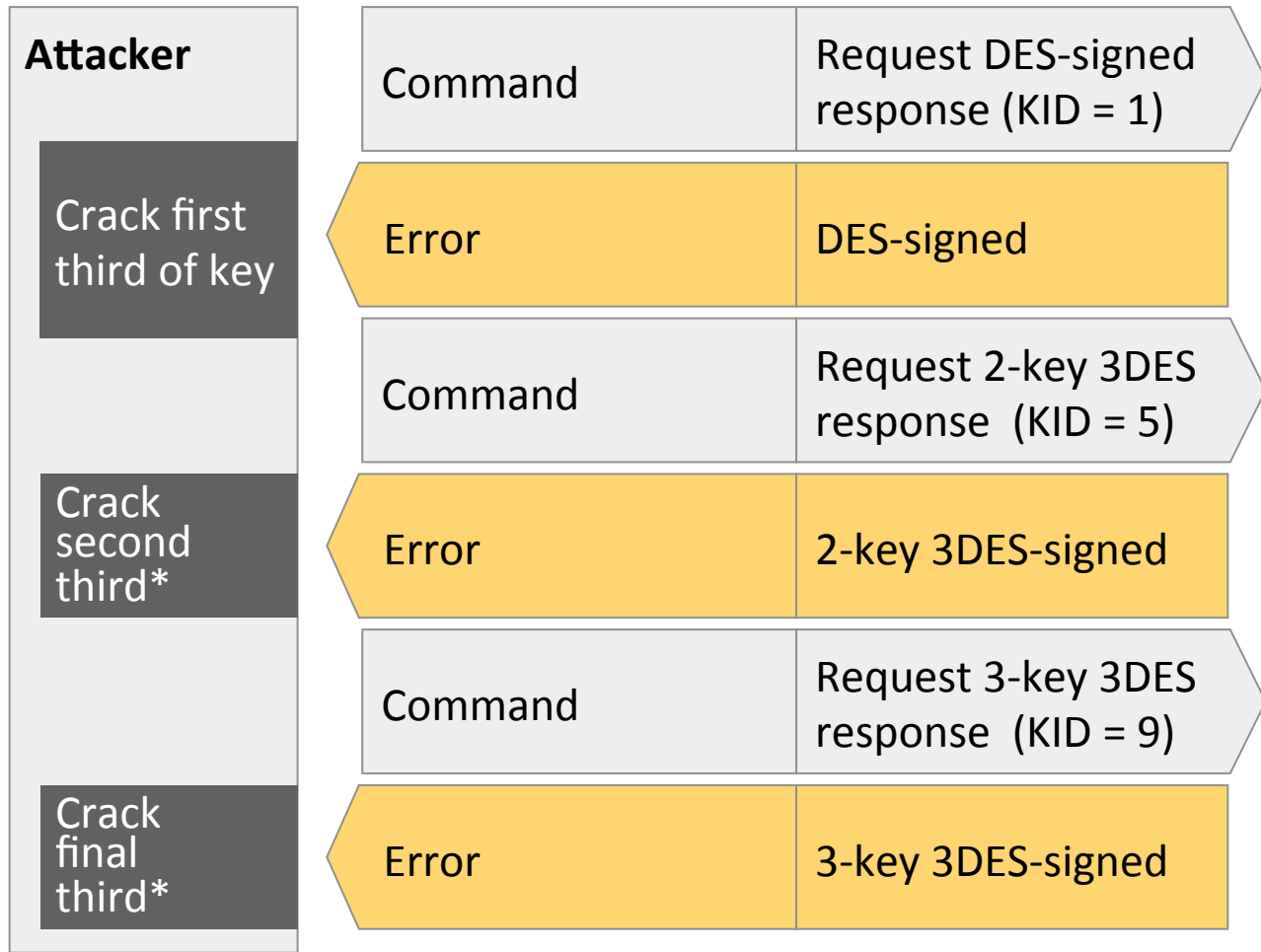
- Premium SMS fraud
- Circumvent caller-ID checks
- Mess with voice mail
- Redirect incoming calls; sometimes also SMS
- Abuse USSD-based payment schemes
- Track victim
- Phishing
- Malware deployment to phone
- Any other browser-based attack


SIM cards are insecure based on several bugs



4 For some cards, even 3DES keys are crackable

Downgrade attack flow



Some **SIM cards with 3DES key**  use lower signature schemes when requested (in violation of the standard)

3-key 3DES		
2-key 3DES		
DES		
56 bit	56 bit	56 bit

Is 3G broken yet?

Cryptology ePrint Archive: Report 2010/013

A Practical-Time Attack on the A5/3 Cryptosystem Used in Third Generation GSM Telephony

Orr Dunkelman and Nathan Keller and Adi Shamir

Slashdot

Second 3G GSM Cipher Cracked

engadget 

3G GSM encryption cracked in less than two hours

Result

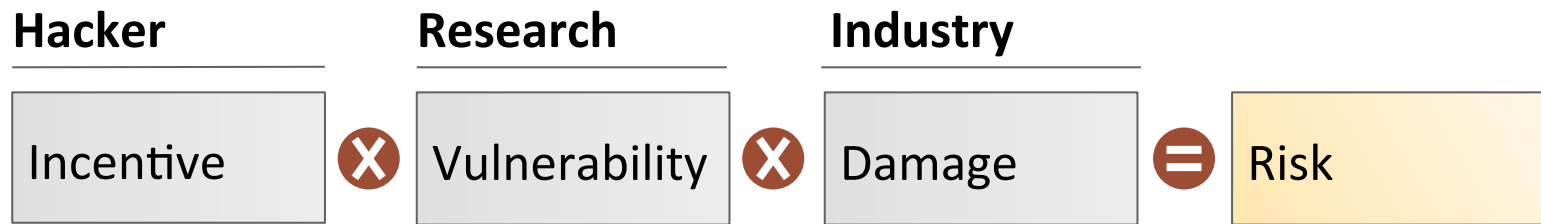
Cipher in 3G can be cracked based on large ciphertext samples

Risk

None? 3G only encrypts very short messages



Managing IT security risks should involve three perspectives



Questions?

 Karsten Nohl <nohl@srlabs.de>