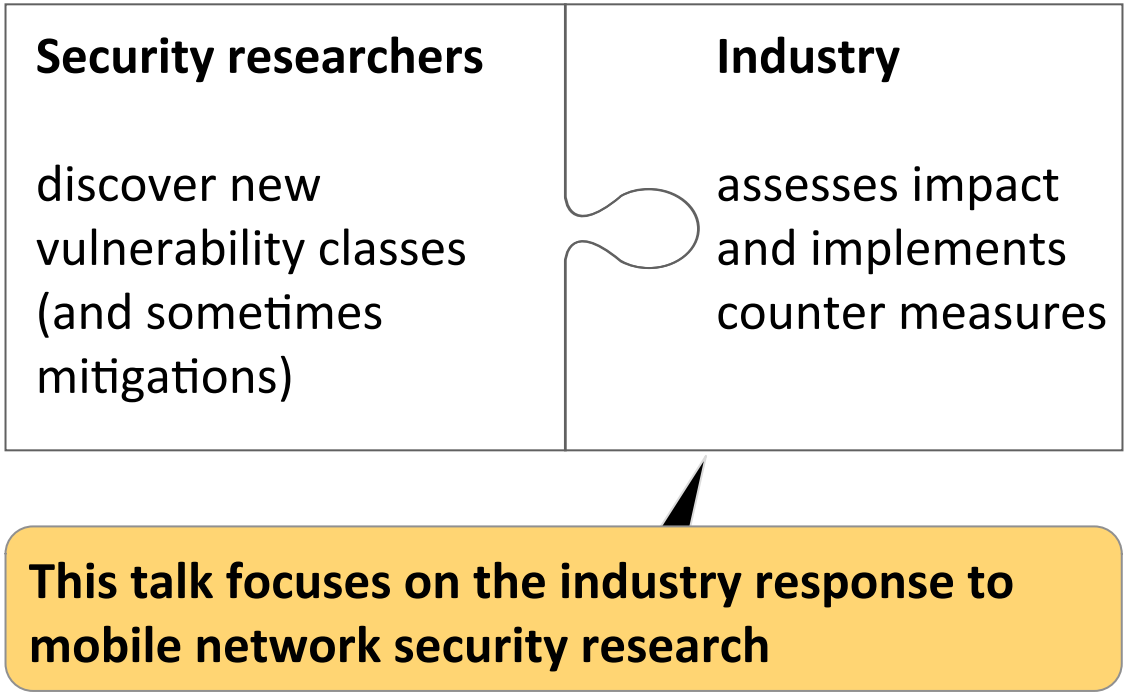
Mobile Network Attack Evolution

Karsten Nohl <nohl@srlabs.de>

Security research is successful if vulnerabilities get removed

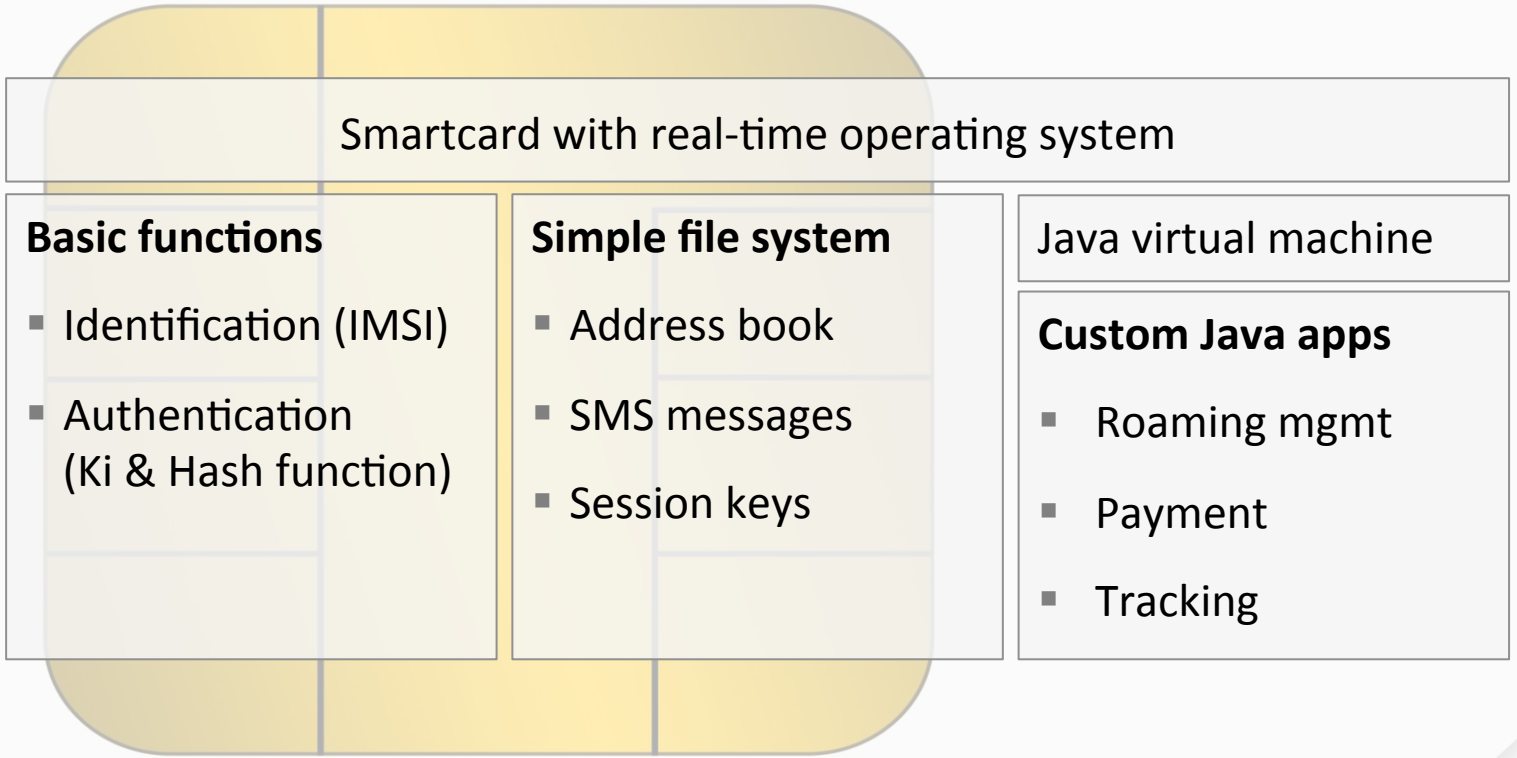


Agenda

- **Advanced SIM card attacks**
- Advanced GSM intercept
- Keeping network operators honest

SIM cards are fully programmable computer systems

Applications on modern SIM card



SIM have many security layers from smartcards to cryptography and Java process separation

SIM card includes various protection mechanisms

User authentication

by simple comparison

PIN/PUK numbers

SIM authentication

by cryptographic hash function (often Comp128 in GSM; Milenage in 3G/4G)

Ki

Secure Java deployment

using DES/3DES/AES signature + encryption

OTA keys

Storage protection

through proprietary smartcard security mechanisms

Application separation: Java VM sand boxing

Individual protection logic for banking applets, identification applets, etc.

…

Java crypto API: DES/3DES/AES; sometimes RSA

OTA security level is chosen by server while SIM enforces mandatory minimum level

ILLUSTRATIVE

OTA server initiates remote transaction

Binary SMS communication

Target app / key set #

Command – possibly encrypted and/or signed | Used security level | Requested security level

Response protected according to request, but not below minimum level stored on card

SIM card stores multiple key sets, possibly with different protection levels

Key set 3

Key set 2

Key set 1

	DES	3DES	AES	Mandatory
Encryption	◉	○	○	☐
Signature	◉	○	○	☑

OTA error handling is underspecified, possibly opening attack surface

Binary SMS communication

Attacker probes cards to gain material for DES key cracking

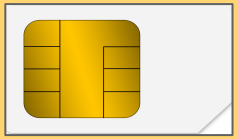
Command with wrong signature | Use: DES signature | Request: DES signature

Response to mal-signed request differs by card type

- a. (25%* of cards) — (No response)
- b. (50%*) — Error message | Sometimes with all-zeros signatures
- c. (25%*) — Error message | DES signature

Data useable for key cracking

SIM card with DES key (prevalence of DES keys varies between operators; can be up to 100%)



* Estimated from a geographically skewed measurement set

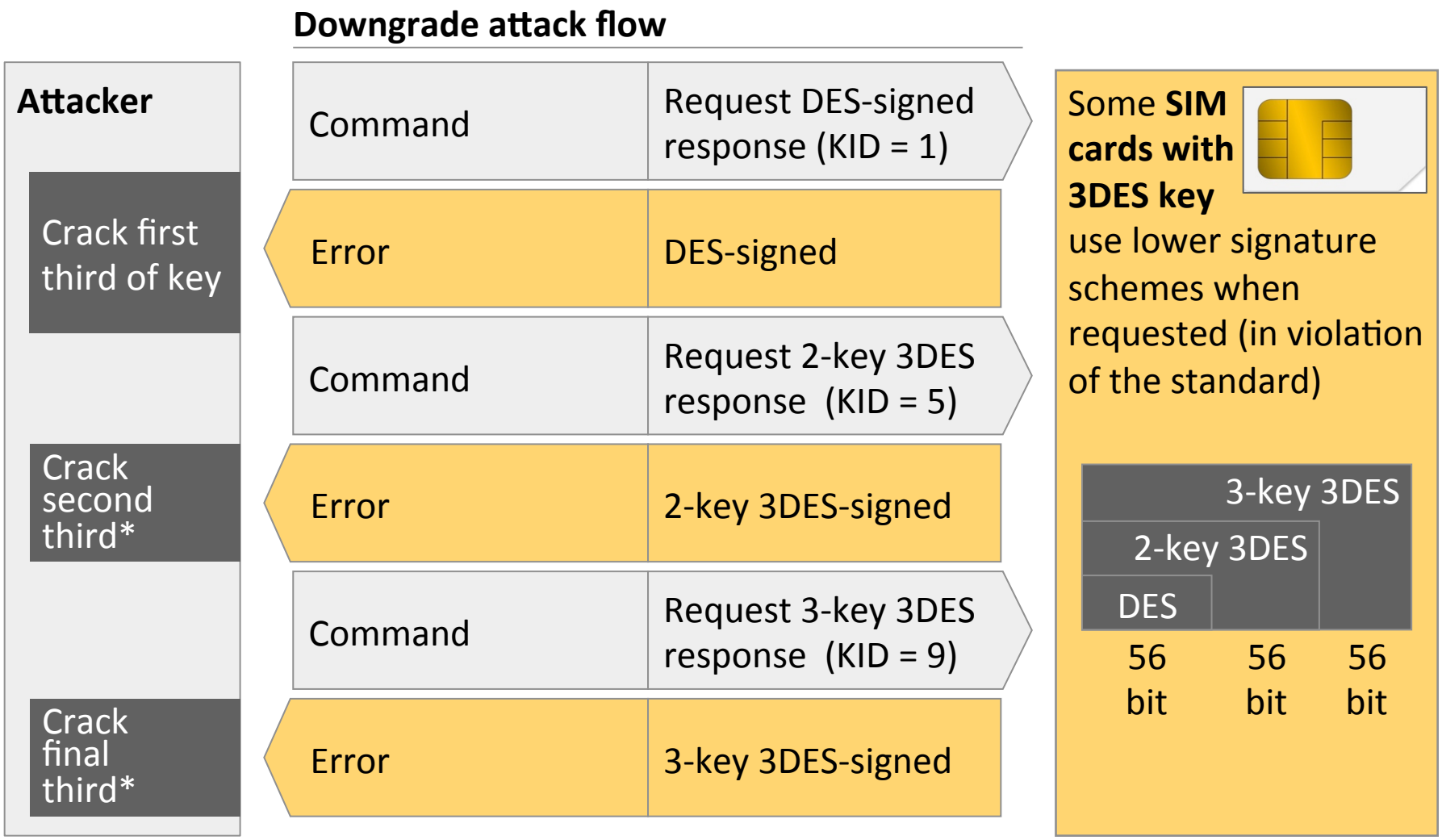
OTA DES do not withstand key cracking

Challenge: Derive 56 bit DES key from OTA response signature

Cracking strategies	Investment	Cracking time
Be patient Brute force on GPU	EUR 1.000	6 months
Throw money at it Brute force on FPGA cluster	EUR 50.000	1 day
Ride the rainbow Time-memory trade-off using large hard disks & GPU	EUR 1.500 + 1 year pre-computation	1 minute (but <100% success rate)

Only possible when OTA response is fully predictable

For some cards, even 3DES keys are crackable



* Must be brute-forced; Rainbow table attack no longer possible

Java virus does not automatically have access to all SIM assets

OTA-deployed SIM virus can access SIM Toolkit API

Standard STK function	Abuse potential
Send SMS	▪ Premium SMS fraud
Dial phone numbers, send DTMF tones	▪ Circumvent caller-ID checks ▪ Mess with voice mail
Send USSD numbers	▪ Redirect incoming calls; sometimes also SMS ▪ Abuse USSD-based payment schemes
Query phone location and settings	▪ Track victim
Open URL in phone browser	▪ Phishing ▪ Malware deployment to phone ▪ Any other browser-based attack

Java sand box should protect critical data on SIM

Data access on SIM would enable further abuse

Protected function	Abuse potential
Read Ki	▪ SIM cloning ▪ Decrypt all 2G/3G/4G traffic
Read hash function	▪ Reverse-engineer proprietary authentication functions; perhaps find weaknesses
Read OTA keys	▪ Lateral attacks
Read Java processes	▪ Clone NFC payment takers and other future SIM applications
Write to Flash or EEPROM	▪ Alter OS to prevent vulnerability patching

Possible on some SIMs due to bug in their Java VM

SIM security research motivated some technology upgrades

Security researchers published several **SIM card attacks**

Industry reacted swiftly but **not thoroughly**

Finding	Response
1 Anybody can send management SMS to SIM cards	Many networks started filtering the most obvious attack messages
2 The OTA app mgmt interface is not always protected with good crypto	Some operators phased out DES keys in favor of 3DES
3 SIM applications can break out of their JavaCard sandbox	The vulnerability has not been addressed yet in affected cards

1 Binary SMS can take many forms to circumvent filters

		SMS field			
		PID	**DCS**	**UDHI**	**User data**
Best practice filters	**Several message types may go to the SIM**	127	*	*	*
		*	246 or 22	*	*
		*	*	1	027000…
	Some phones also forward other types	*	*	0	027000…
vs.					
Imple-mented filters	**Many networks only filter one type**	127	*	*	*

2 Misconfigurations in SIMs go well beyond DES keys

ILLUSTRATIVE

SIM configurations need to be assessed in two dimensions

2. Verify that all SIM applications enforce cryptography

1. Verify that all keys are 3DES or AES

		Application (TAR)			
Keyset		**000000**	**000001**	**…**	**FFFFFF**
1:	3DES	Sign + encrypt	Unprotected (MSL=0)		…
2:	3DES	Sign + encrypt	Sign		…
…					
16:	DES	Sign	Sign		…

Attack example– Persistent infection of modern SIM card

Target —
New nano-SIM (October 2013) in iPhone 5s from major European carrier

Attack steps

- **A** Lure the phone onto fake base station to circumvent network filters
- **B** Scan the SIM remotely for configuration issues (on the SIM in this demo: discover TAR with MSL=0)
- **C** Install Java virus through vulnerable TAR
- **D** Let phone connect back to normal network, maintain persistent access through SMS-C&C

Self-assessment tool: Find bugs in your SIM card's configuration

Tool name	**SIMtester**
Purpose	▪ Find cryptographic attack surface: – Signature disclosure – 3DES downgrade ▪ Enumerate logical attack surface: Detect hidden application TARs and test their security level ▪ Upload traces to gsmmap.org for further analysis (Thank you.)
Requirements	PC/SC smartcard reader –or– Osmocom phone
Source	opensource.srlabs.de

Agenda

- Advanced SIM card attacks
- **Advanced GSM intercept**
- Keeping network operators honest

GSM intercept attacks are still under addressed

The majority of mobile phone calls worldwide still uses 2G GSM frequencies

To protect customers, mobile networks must support and harden two encryption standards

1

Older phones only support **A5/1 encryption**

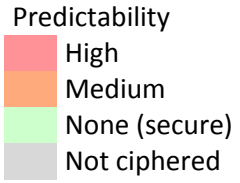
Protection status: Available strengthening measures are rarely seen

2

A5/3 protects much better

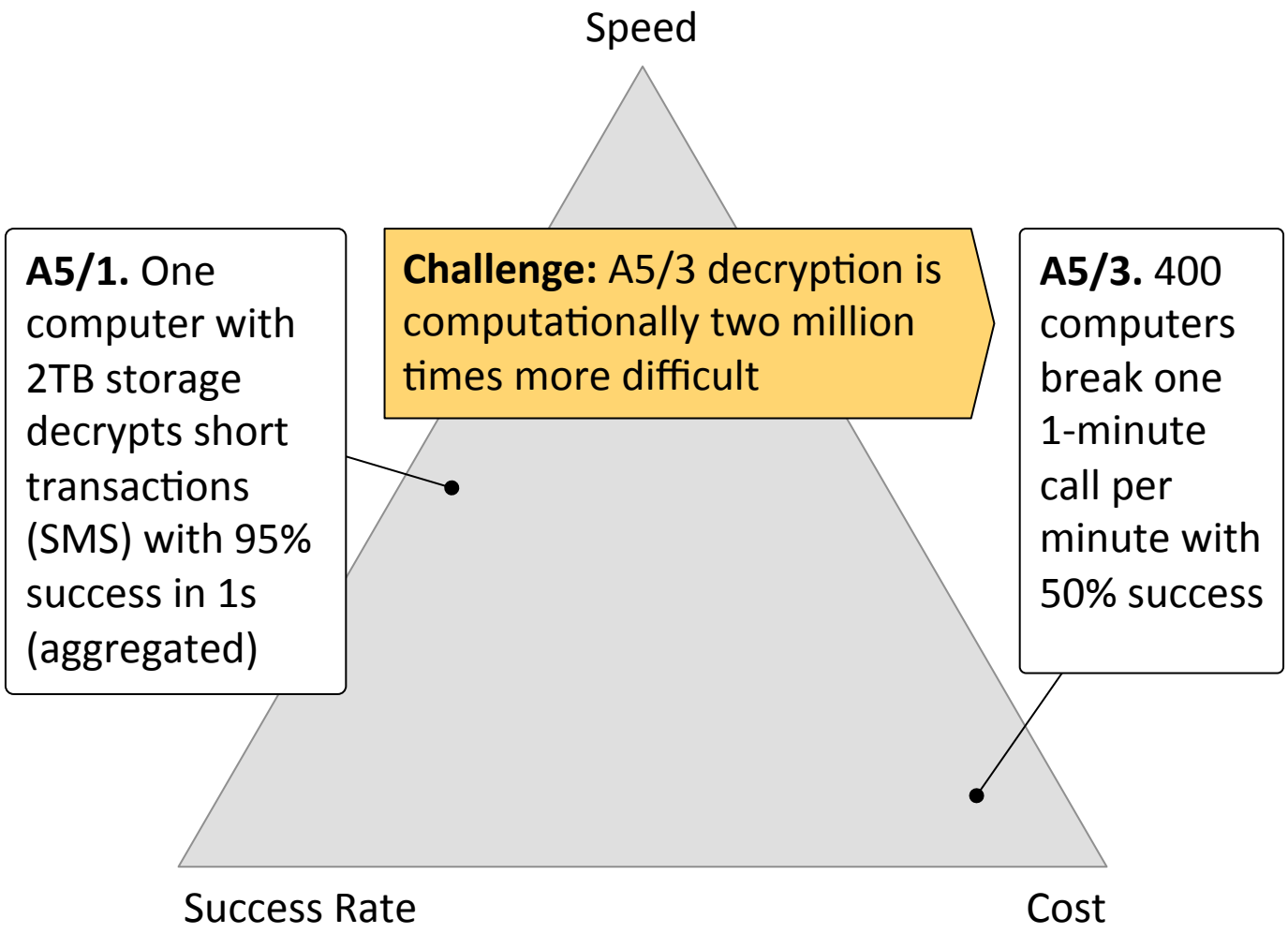
Protection status: Still only a minority of networks support A5/3

1 A5/1 decryption can mostly be prevented through randomization



	Features to decrease cryptographic attack surface		
Example call setup trace	**Unprotected**	**Padding randomization**	**+ SI5 randomization**
Ciphering Mode Cmd			
TMSI Reallocation Cmd			
Null Frame			
System Information 5			
Call Proceeding			
System Information 6			
Null Frame			
Fragment			
Assignment Command			
System Information 5ter			

2 A5/3 makes intercept much harder, but decryption is still possible for well-funded spy agencies



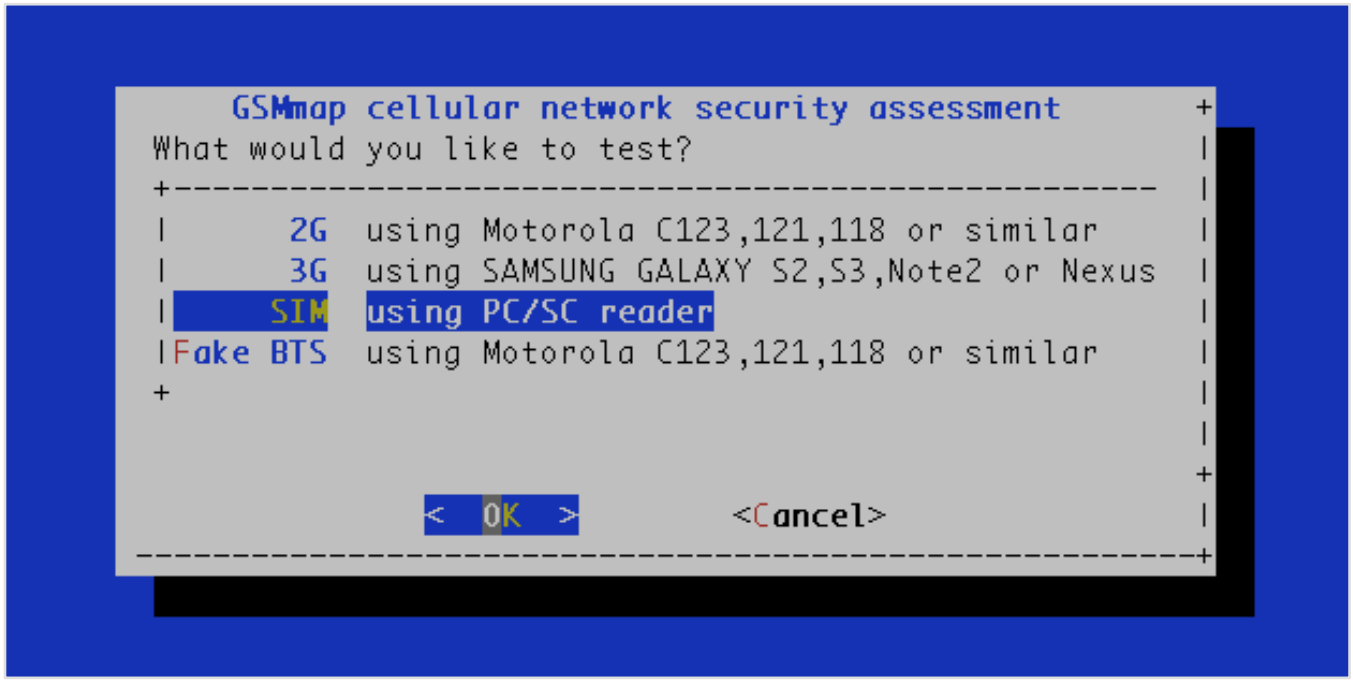
Agenda

- Advanced SIM card attacks
- Advanced GSM intercept
- **Keeping network operators honest**

You can help:
Measuring mobile network security from Android or Linux

Tool name	GSMmap.apk	xgoldscanner	OsmocomBB
Purpose	Collect network traces on Android phone and upload for analysis to gsmmap.org	Record network traces for analysis in Linux	Update to Sylvain's burst_ind setup to capture network traces for analysis in Linux
Requirements	Rooted Samsung Galaxy S2/S3	Samsung Galaxy S2, S3, Note 2, or Nexus	An older Motorola phone (C123, …)
Source	Google Play: GSMmap	opensource.srlabs.de	OsmocomBB git: *gsmmap* branch

Live ISO puts mobile security tools on ready-to-use USB stick

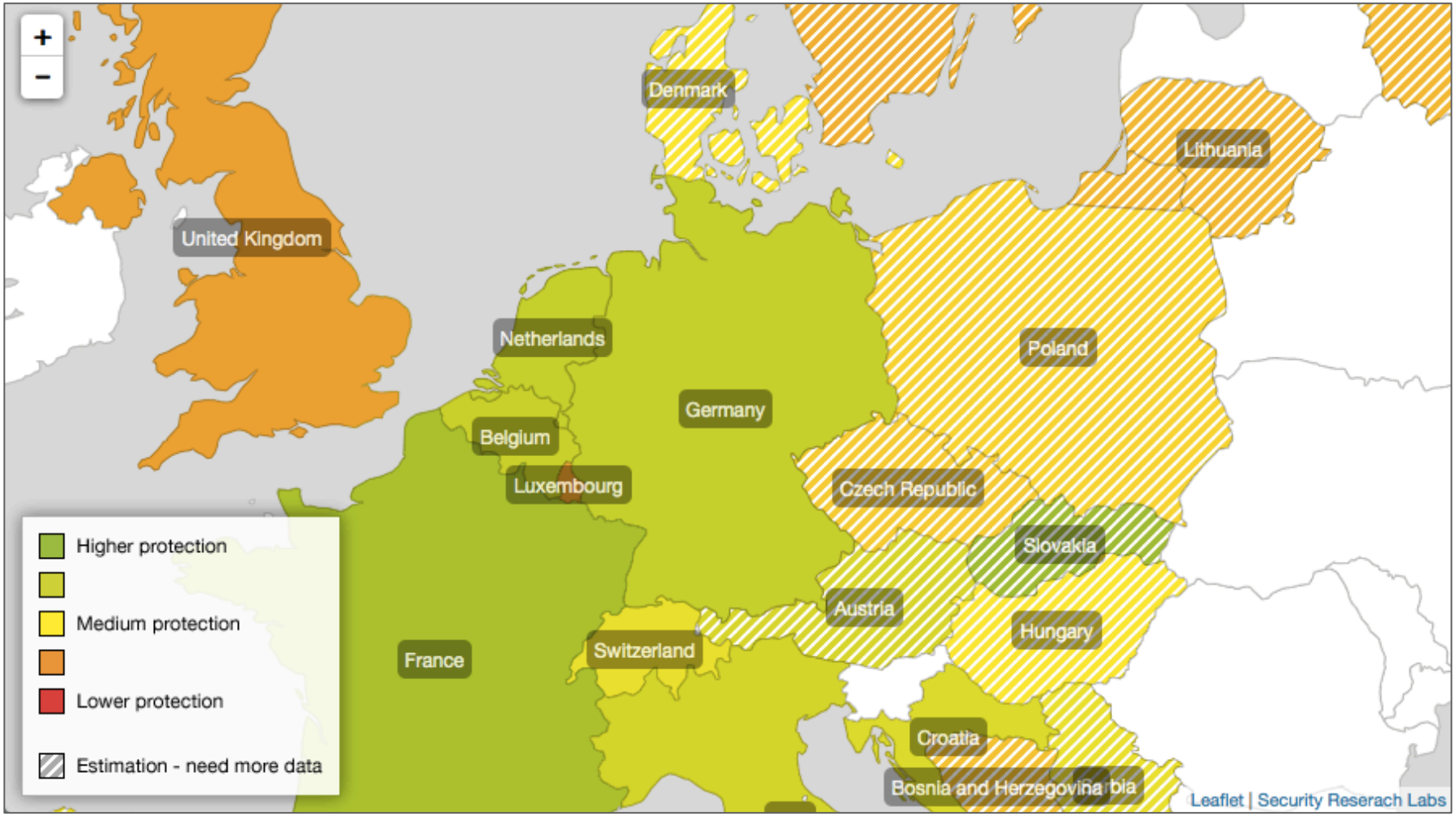


GSM map live ISO bundles mobile security tools

Network measurement with Galaxy S2/S3	Network measurement & IMSI catcher detection with Osmocom BB phone	SIM card assessment with PC/SC reader or Osmocom BB phone

Download and How-Tos	opensource.srlabs.de

gsmmap.org – Tracking mobile network evolution online



Thank you!

Research supported by

OPEN TECHNOLOGY FUND

Many thanks to **Lukas Kuzmiak, Luca Melette, and Linus Neumann** for creating and supporting our research tools!

Questions?

Karsten Nohl <nohl@srlabs.de>