

GSM Security Overview

Mehdi Hassanzadeh

Mehdi.Hassanzadeh@ii.uib.no



Selmer Center, University of Bergen, Norway

FRISC Winter School, Finse, May 6-11, 2012

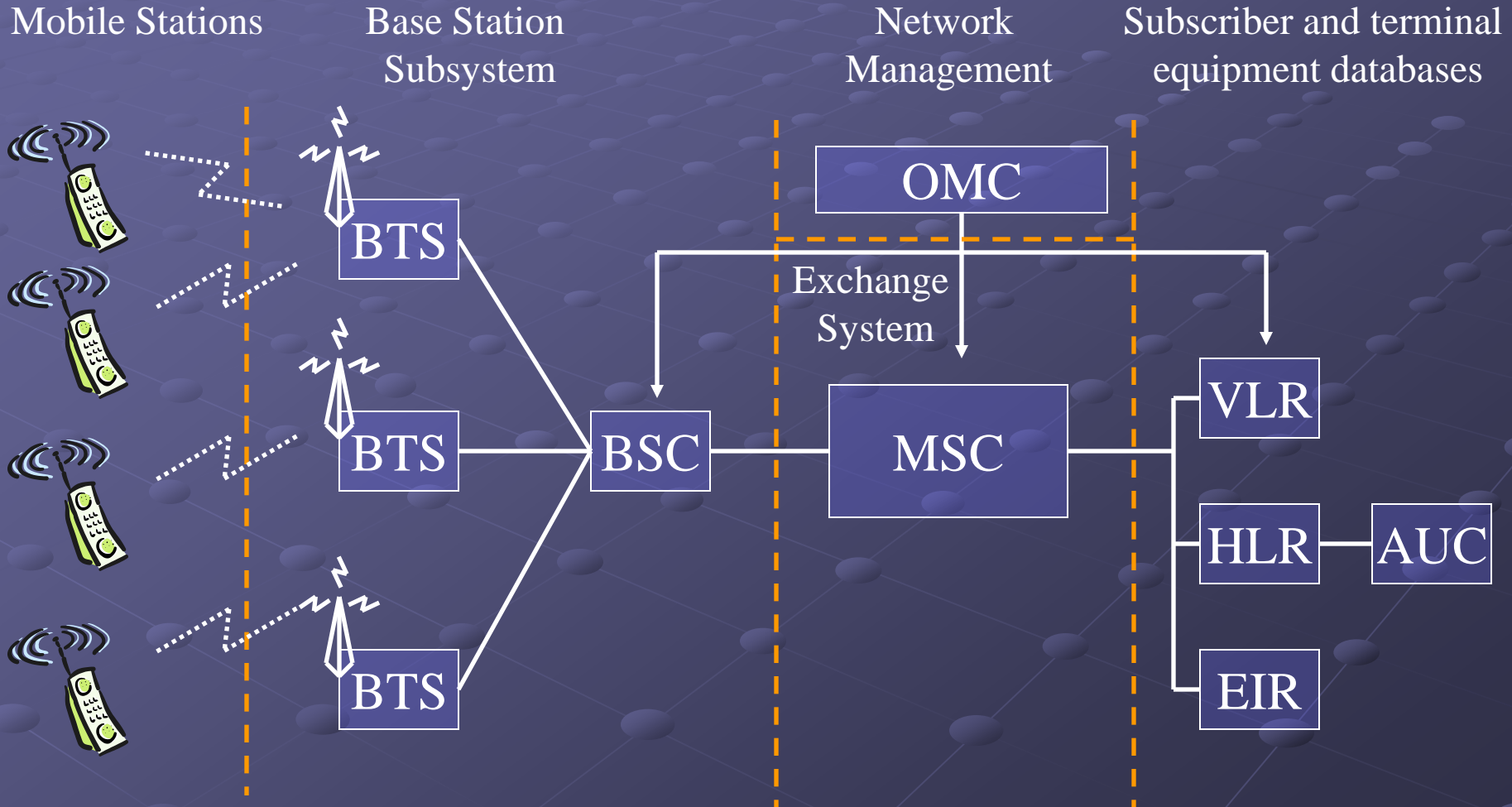
Agenda

- GSM structure
- A5/1 Overview
- Attack History on A5/1
- Rainbow Tables Attack on A5/1
- Current state of Tables
- Probability of successful attack
- Data Collection and Frequency Hopping

GSM is global and insecure

- 200+ countries
- 4 billion users!
- GSM encryption introduced in 1987
- Disclosed and shown insecure in 1994
- However, GSM is used in a growing number of sensitive applications:
 - Voice calls, obviously
 - SMS for banking
 - Seeding RFID/NFC secure elements for access control, payment and authentication
- GSM is constantly under attack .

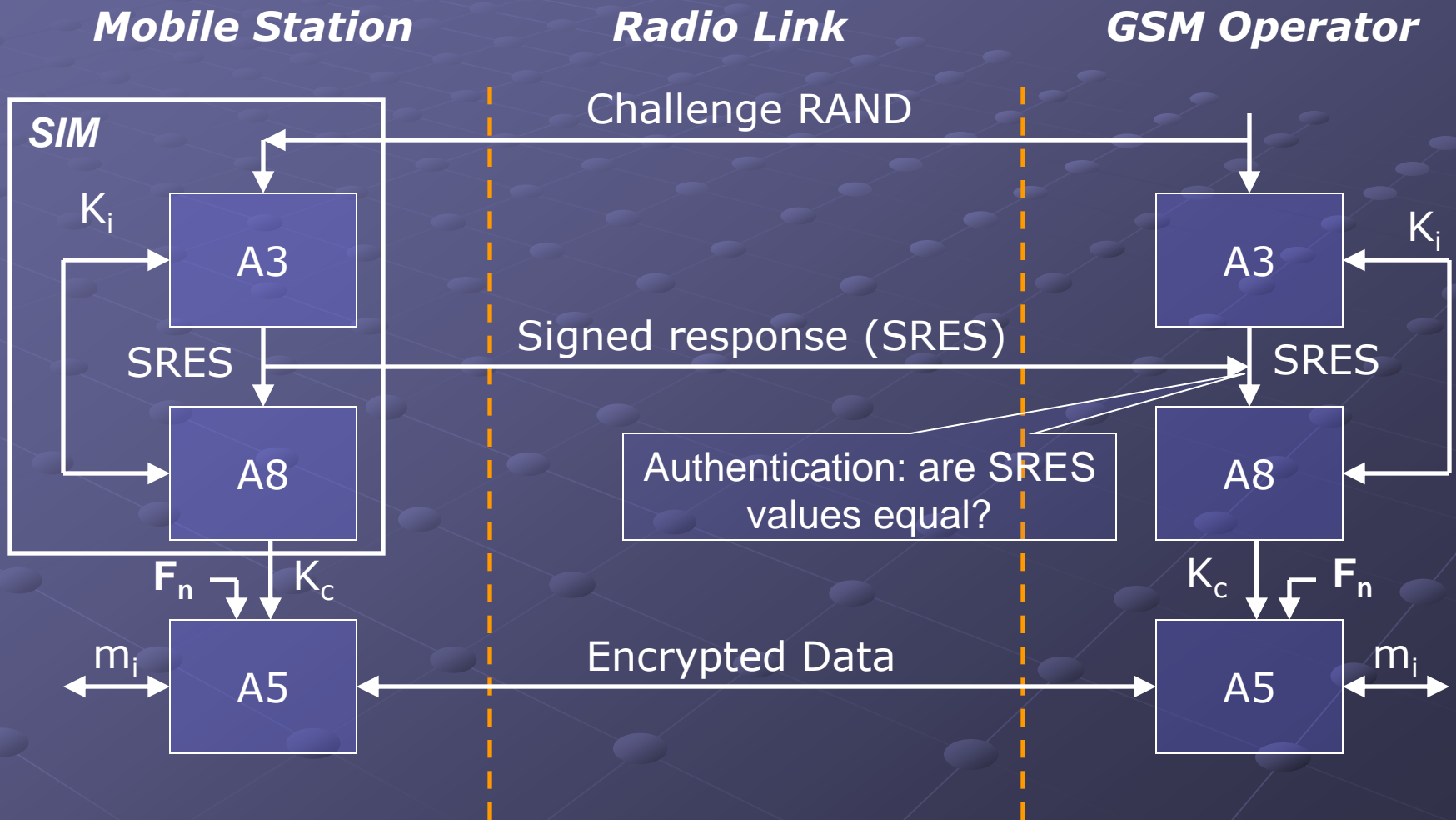
GSM Architecture



GSM cryptographic component

- Authentication → A8
- Key agreement → A3
- Stream cipher → A5
 - A5/1 the stronger version
 - A5/2 the weaker version
 - A5/3 the 3th generation (Block cipher)

Authentication and Encryption Scheme



A5 – Encryption Algorithm

● A5 is a stream cipher

- Implemented very efficiently on hardware
- Consists of 3 LFSRs of 19,22,23 bits length
- 64 bits section key
- 22 bits frame number
- Registers are clocked in a stop/go fashion using the majority rule.

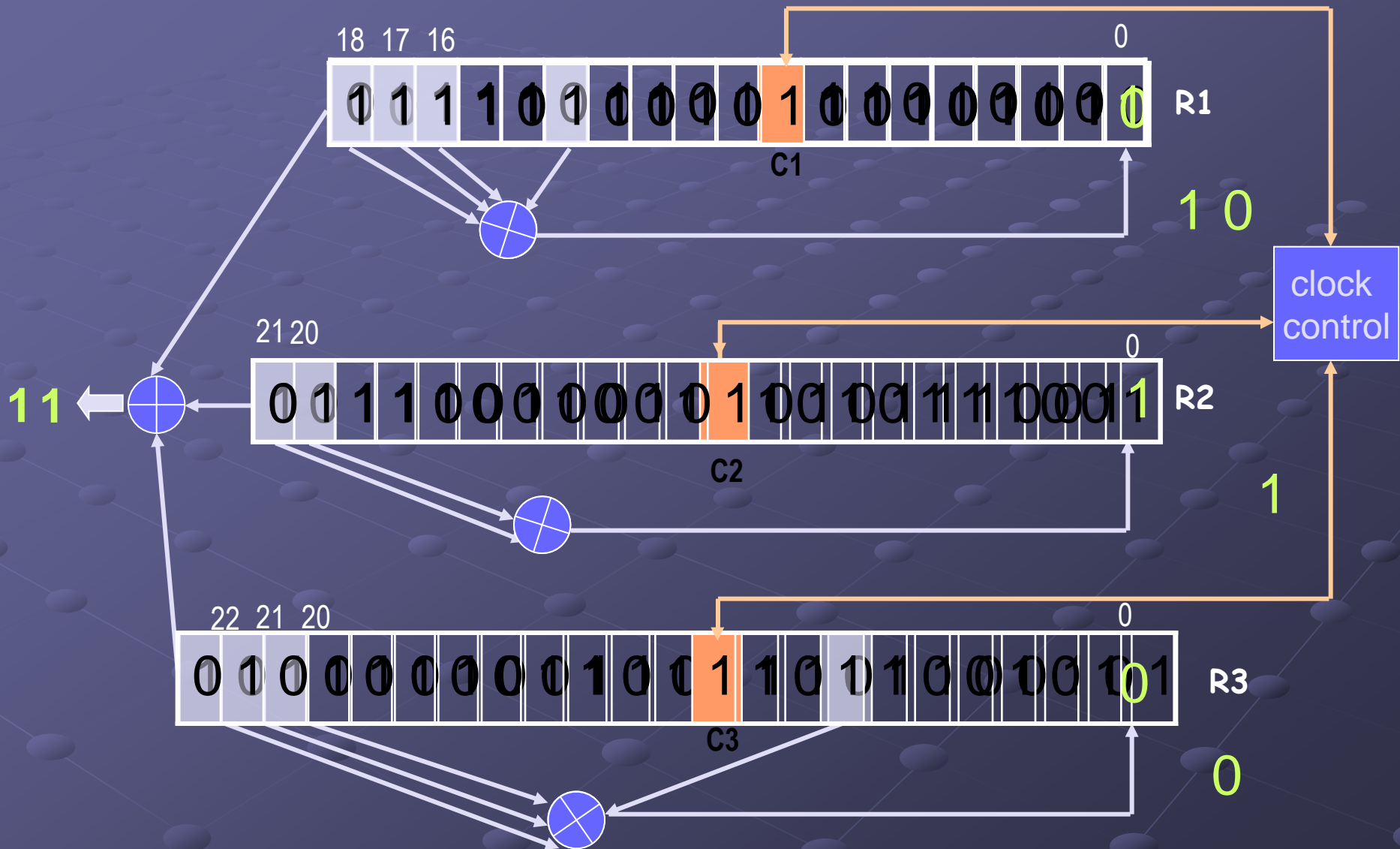
● Variants

- A5/1 – the strong version
- A5/2 – the weak version
- A5/3
 - GSM Association Security Group and 3GPP design
 - Based on Kasumi algorithm used in 3G mobile systems

A5/1 – Operation

- All 3 registers set to zero
- 64 cycles (without the stop/go clock) :
 - Each bit of K_c (lsb to msb) is XOR'ed in parallel into the lsb's of the registers
- 22 cycles (without the stop/go clock) :
 - Each bit of F_n (lsb to msb) is XOR'ed in parallel into the lsb's of the registers
- 100 cycles with the stop/go clock control, discarding the output
- 228 cycles with the stop/go clock control which produce the output bit sequence.

A5/1 – Structure



Attack History

- 1991: First GSM implementation.
- 1997:
 - Golic presented an attack based on solving sets of linear equations.
 - Time complexity of $2^{40.16}$
- 1999:
 - Alex Biryukov, Adi Shamir and David Wagner
 - Two minutes of intercepted call
 - The attack time was only 1 second.
 - 2^{48} Steps
 - 300 GB of data
- 1999:
 - Eli Biham and Orr Dunkelman
 - total work complexity of $2^{39.91}$ A5/1 clockings
 - $2^{20.8}$ given bits of known plaintext
 - 32 GB of data storage after a precomputation stage of 2^{38}

Attack History

● 2003:

- Ekdahl and Johansson published an attack on the initialization procedure
- Using two to five minutes of conversation plaintext.
- No preprocessing stage.
- In 2004, Maximov et al. improved this attack
 - Less than one minute of computations
 - A few seconds of known conversation
- in 2005, improved by Elad Barkan and Eli Biham

Rainbow Tables

● Presented

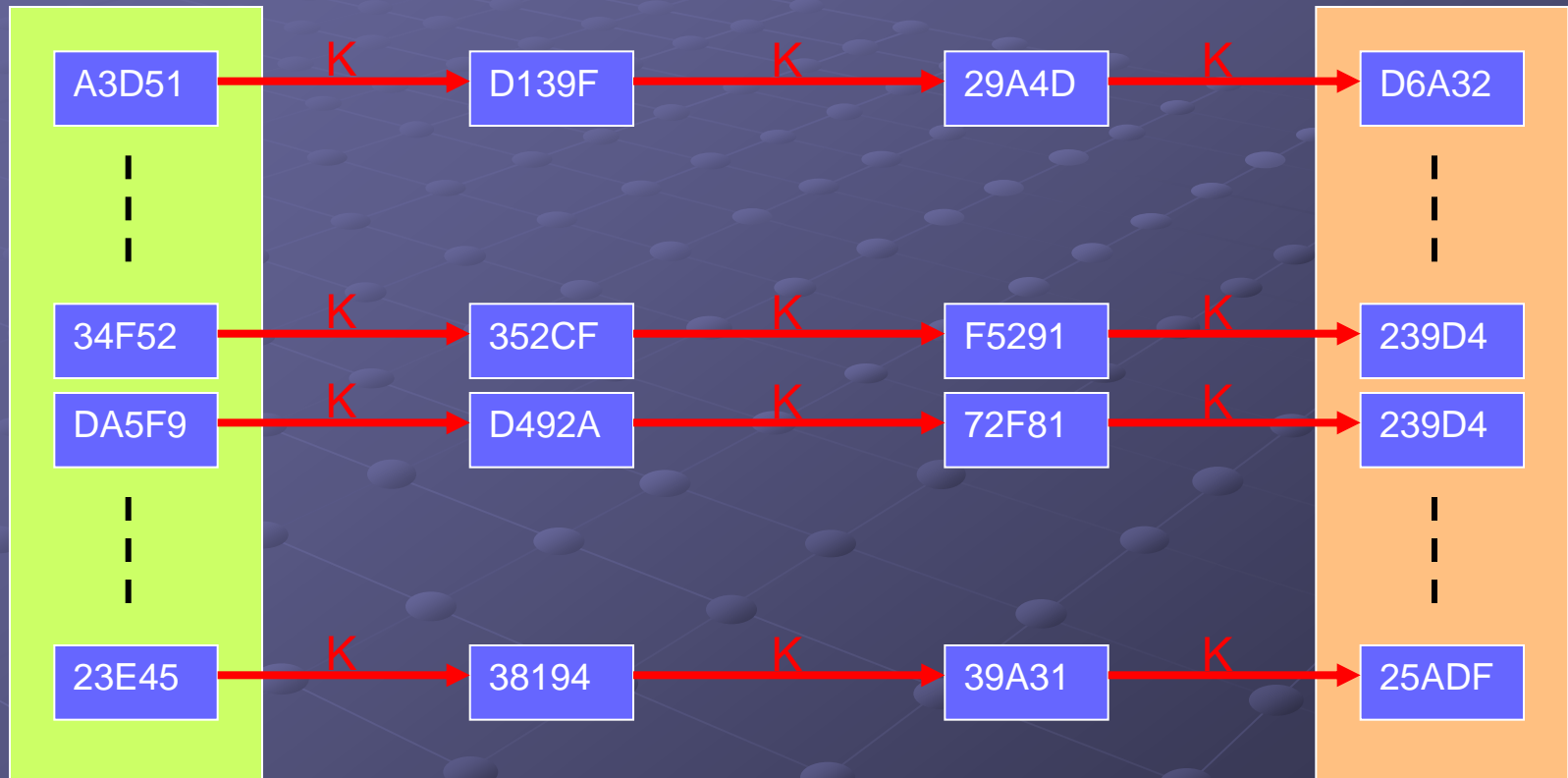
- In 2009
- The 2009 Black Hat security conference
- By cryptographers Karsten Nohl and Sascha Krißler
- Using look-up tables
- 2TB hard disk
- ATI GPU

Code book attacks

- For ciphers with **small keys**, code books allow decryption
- Code book provides a mapping from known output to secret state
- An A5/1 code book is **128 Petabyte** and takes 100,000+ years to be computed on a PC

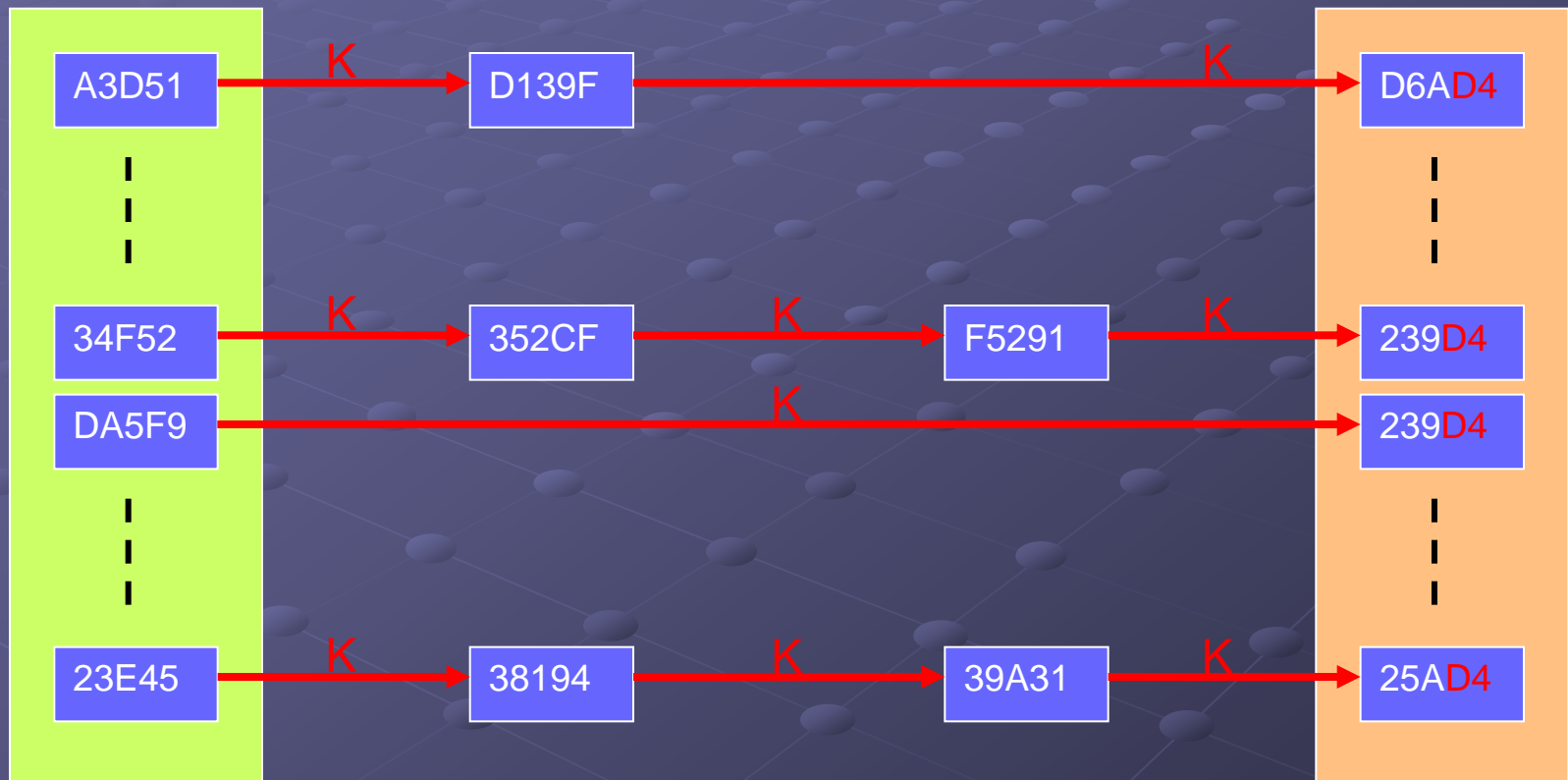
Secret state	Output
A52F8C02	52E91001
62B9320A	52E91002
C309ED0A	52E91003

Optimization 1: Chain



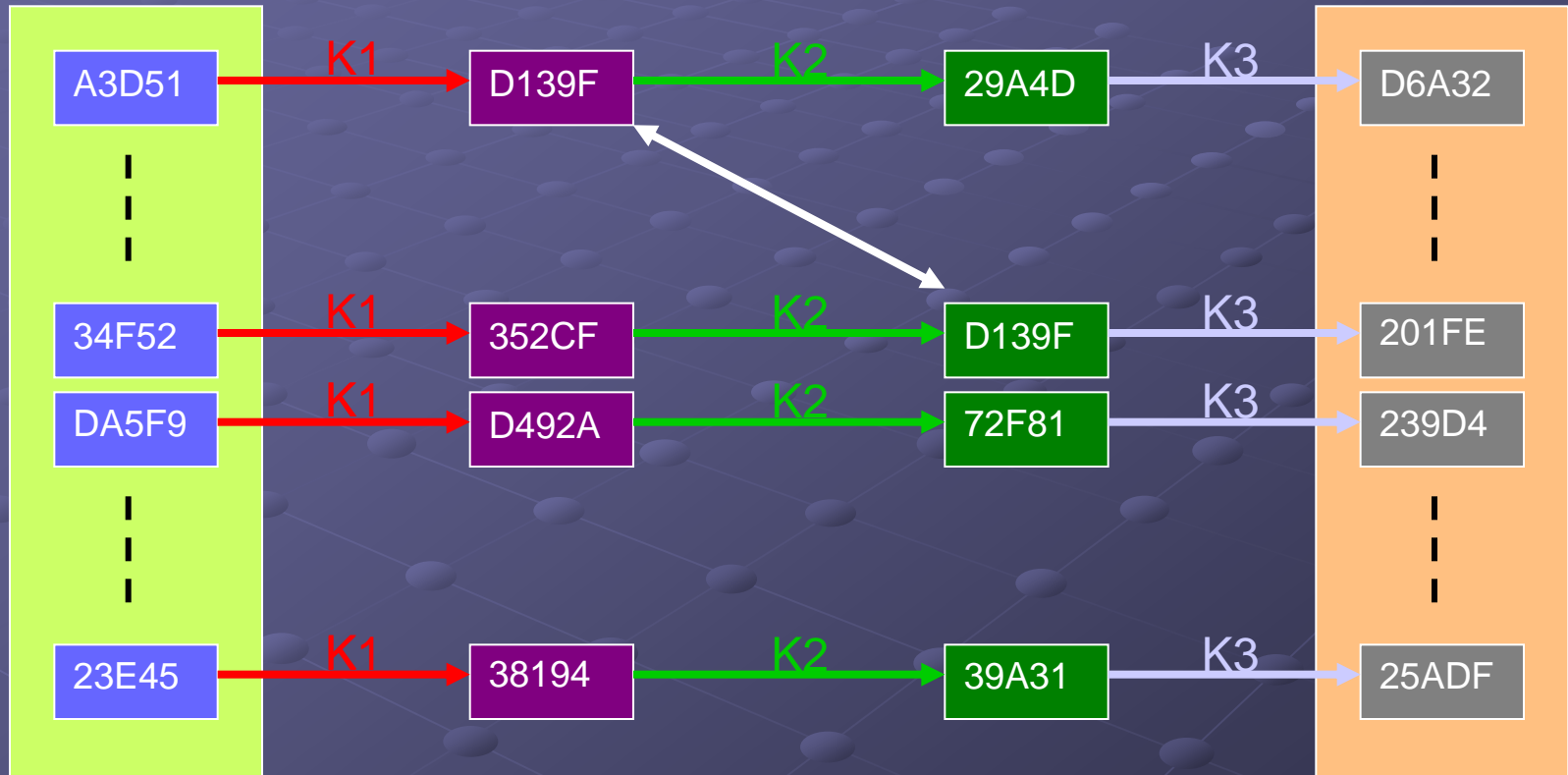
Longer chains := a) less storage, b) longer attack time

Optimization 2: Distinguishing points



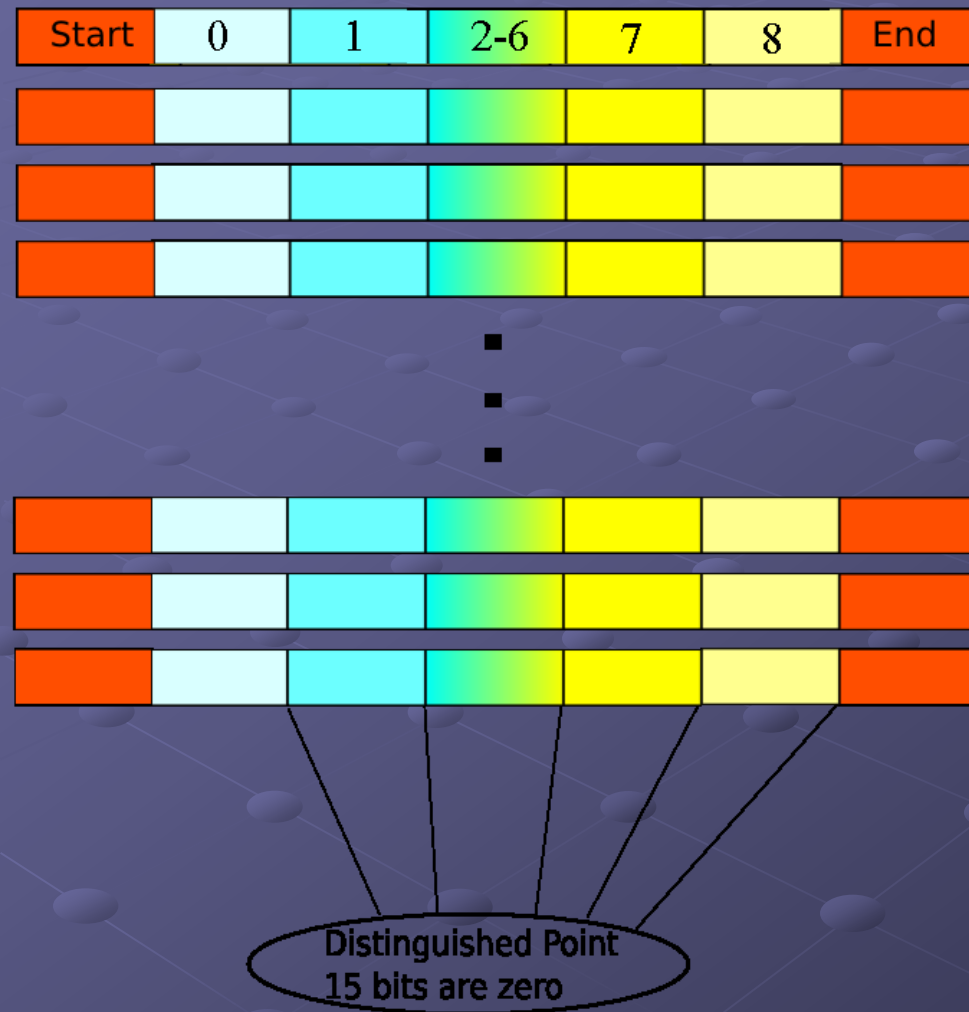
Hard disk access only needed at distinguished points

Optimization 3: Rainbow tables



Rainbow tables have no merges, but
an exponentially higher attack time

Solution



- 2^{12} elements per color
- 8 color per chain
- 40 tables
- 2^{33} chains per table
- In total, we have $2^{53.334}$ Keys pre-calculated
- Total space: $2^{61.3561}$

Optimizations

- **Original format**: 2x64 bits for each chain
- **Start values** are 34 bits instead of 64 bit
- Tables are sorted by end values. Only the first values in each block are stored in full. Since neighboring values only differ in the last bits, a complete chain can be encoded in 52 to 72 bits.
- An **average**: 54 bits per chain
- Effectively compressing the tables by **42%**
- **40 tables** for a total of 2TB

Probability of Attack

- 2 plaintext messages = 4 bursts = 4*114 bits
- We apply this attack on each 64 bits
- For each burst the probability of cracking:
$$1 - (1 - 2^{53.334} / 2^{61.3561})^{(114 - 64 + 1)} = 17.85\%$$

- Total probability:

$$1 - (1 - 2^{53.334} / 2^{61.3561})^{2 * (2 * 114 - 64 + 1)} = 71.97\%$$

Key Cracking

- Given **two** encrypted known **plaintext** messages (ie. *Cipher mode complete* and a *System Information* message)
- The table set finds a secret key with almost **72% probability**.
- The cracking takes about **five seconds** on two GPUs using SSD storage device.
- Roughly **100,000 look-ups** are required which fast SSD disks can provide within five seconds.

Data Collection

- *Cipher Mode Complete* message is available:
 - which is the first encrypted message in an encrypted transaction and usually contains the same data, mostly empty padding bytes
- *Cipher Mode Complete* message is not available:
 - the *System Information* messages also carry highly-predictable data that can be used for known plaintext where.

Frequency Hopping

● Hardware: USRP2



- Capture 25 MHz of GSM spectrum
- Typically enough for one operator
- Need separate boards for up-and down-link

● Software: OpenBTS

- Decode and record one GSM channel
- Interpret control channel data

What We Need?

● Signal Collection

- USRP
- GNU Radio software
- OpenBTS



● Online Phase

- Download Rainbow table (2TB)
- ATI Radeon HD (GPU)
- Download Kraken to crack the keys of the encrypted parts
- Wireshark displays the decoded data

The background is a dark blue gradient. Overlaid on this is a 3D grid of small, rounded cubes or spheres. The grid is composed of thin, light blue lines that recede into the distance, creating a sense of depth. The cubes themselves are slightly darker than the background, giving them a three-dimensional appearance.

Thank You !