GSM: WE CAN HEAR EVERYONE NOW!

DEFCON 2019

Campbell Murray, Eoin Buckley

James Kulikowski, Bartek Piekarski

BlackBerry

Biographical Information

Campbell Murray

Global Head of BlackBerry Cybersecurity Delivery

Eoin Buckley

Senior Cybersecurity Consultant

James Kulikowski

Senior Cybersecurity Consultant

Bartek Piekarski

Senior Cybersecurity Consultant

AGENDA

Topic: Vulnerability in GSM and generating an indicator to exploit it

- Section 1: Intro to GSM
- Section 2: Concept Overview
- Section 3: Test Lab Setup & Demonstration
- Section 4: Cellular Security Discussion

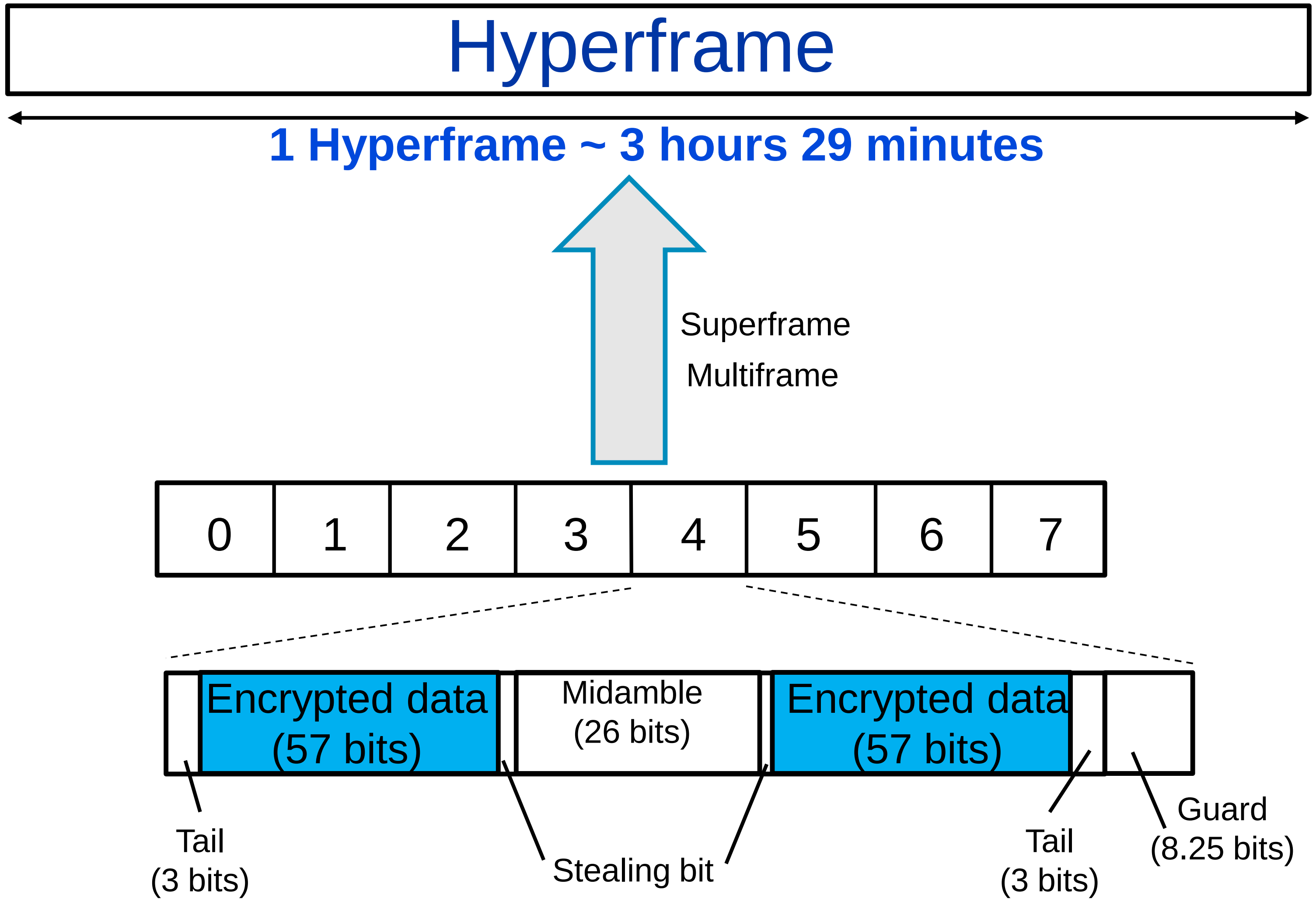
Introduction To GSM

GSM Introduction

- Concept for GSM (digital) started in the late 1980s
- Major improvement over AMPS (analogue)

- GSM Security has several design issues
  - Support for key sizes <= 64 bits
  - Encrypted data contains redundancy
    - Error control coding before ciphering

GSM Introduction



Hyperframe

- Contains 2,715,648 frames

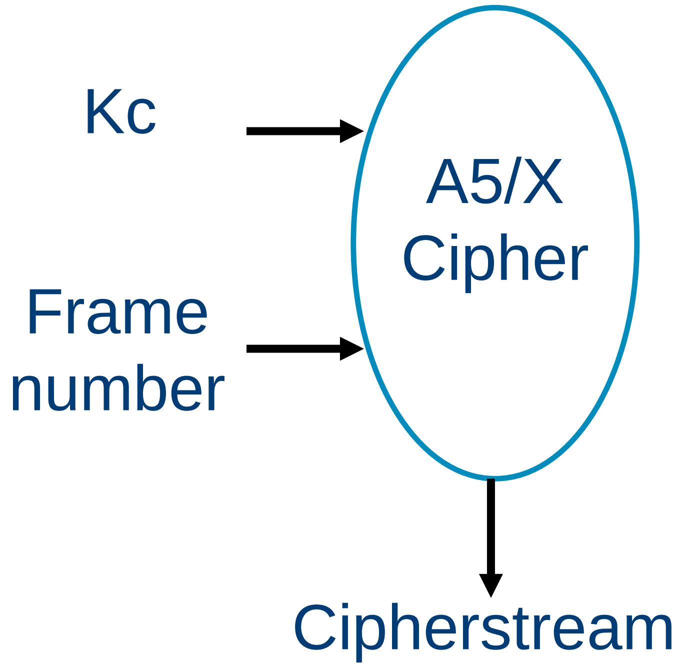
Frame

- Contains 8 timeslots

Timeslot

- Contains 114 encrypted data

GSM Introduction



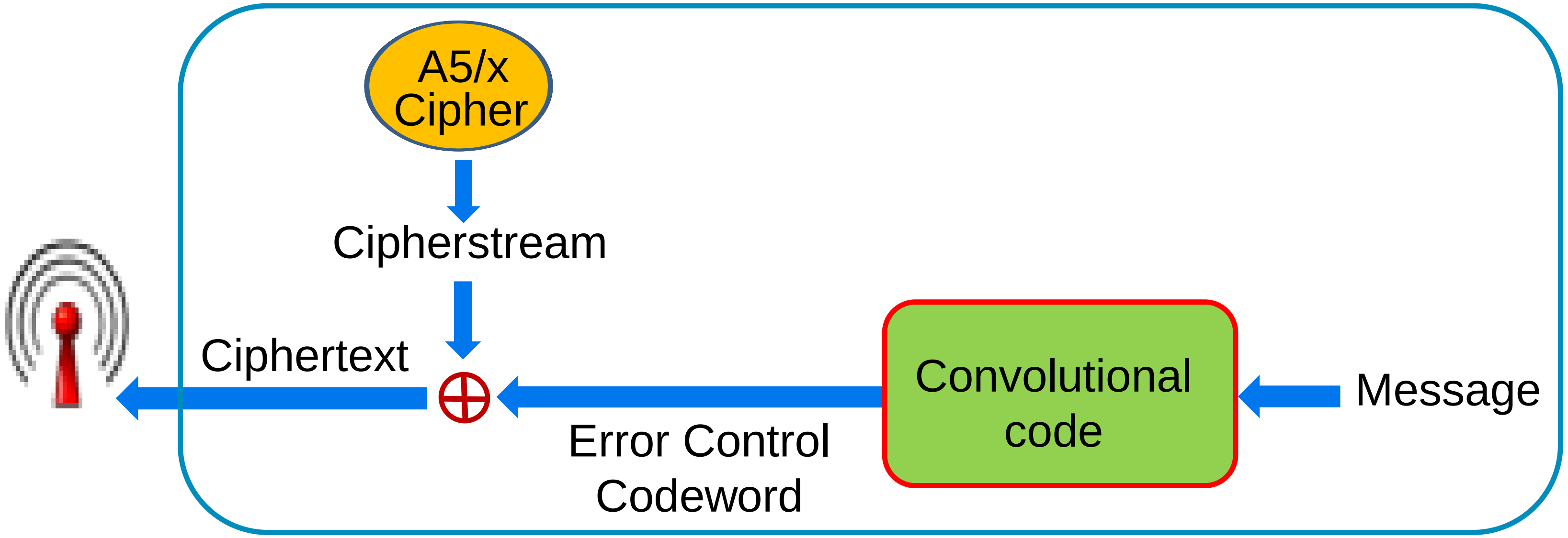
GSM based on symmetric encryption

- Specified ciphers: A5/1, A5/3 & A5/4
  - A5/1 up to 64 bit key
  - A5/3 up to 64 bit key
  - A5/4 up to 128 bit key
  - Note: A5/2 disallowed in 2000's

- NIST guidance: 112 bit security strength
  - "The use of keys that provide less than 112 bits of security strength for key agreement is now disallowed"

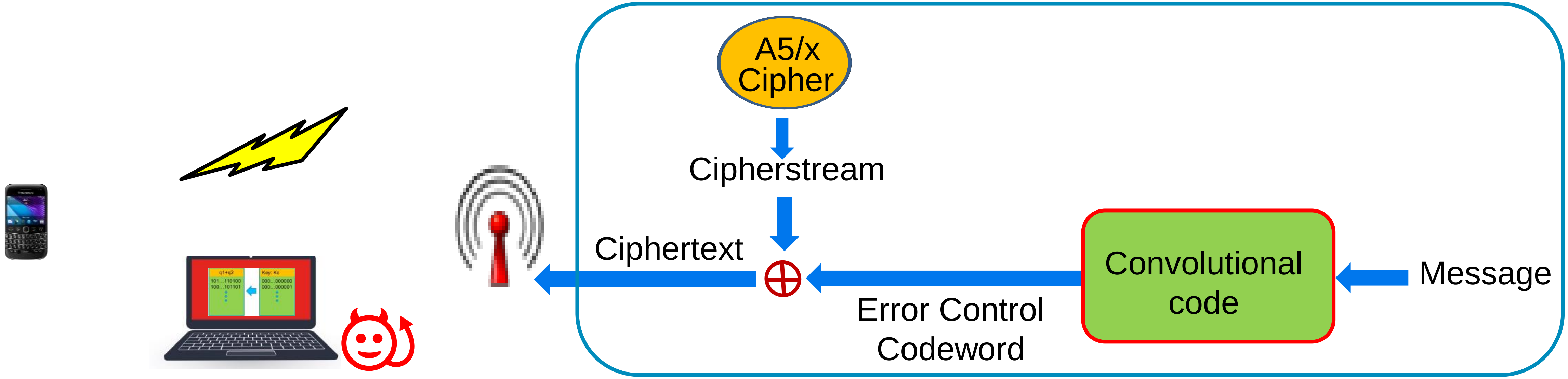
Concept Overview

Concept Overview



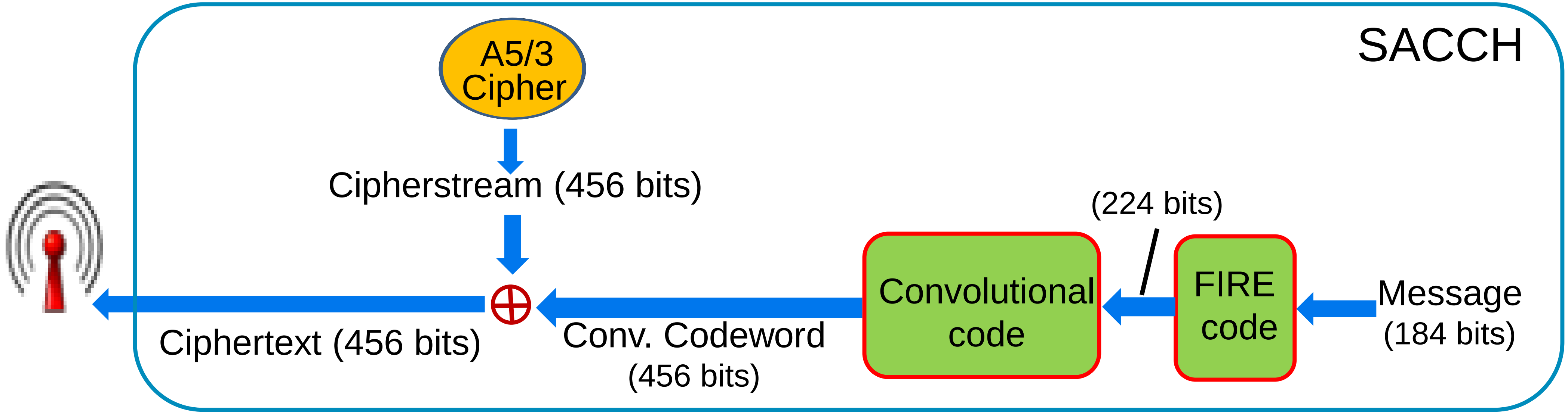
- Typical GSM Channel Structure with A5/1
  - Encryption
    - Key maximum 64 bits length
  - Convolutional error control code
    - Intended to combat noise from wireless channel
    - Attack uses code to identify cipherstream “noise”

Concept Overview



- High level view of attack
  - Capture GSM packet
  - Compute a cipherstream/key indicator
    - Use convolutional code parameters
  - Use indicator with a Rainbow table to identify ciphering key
    - Use indicator as a fingerprint for ciphering key

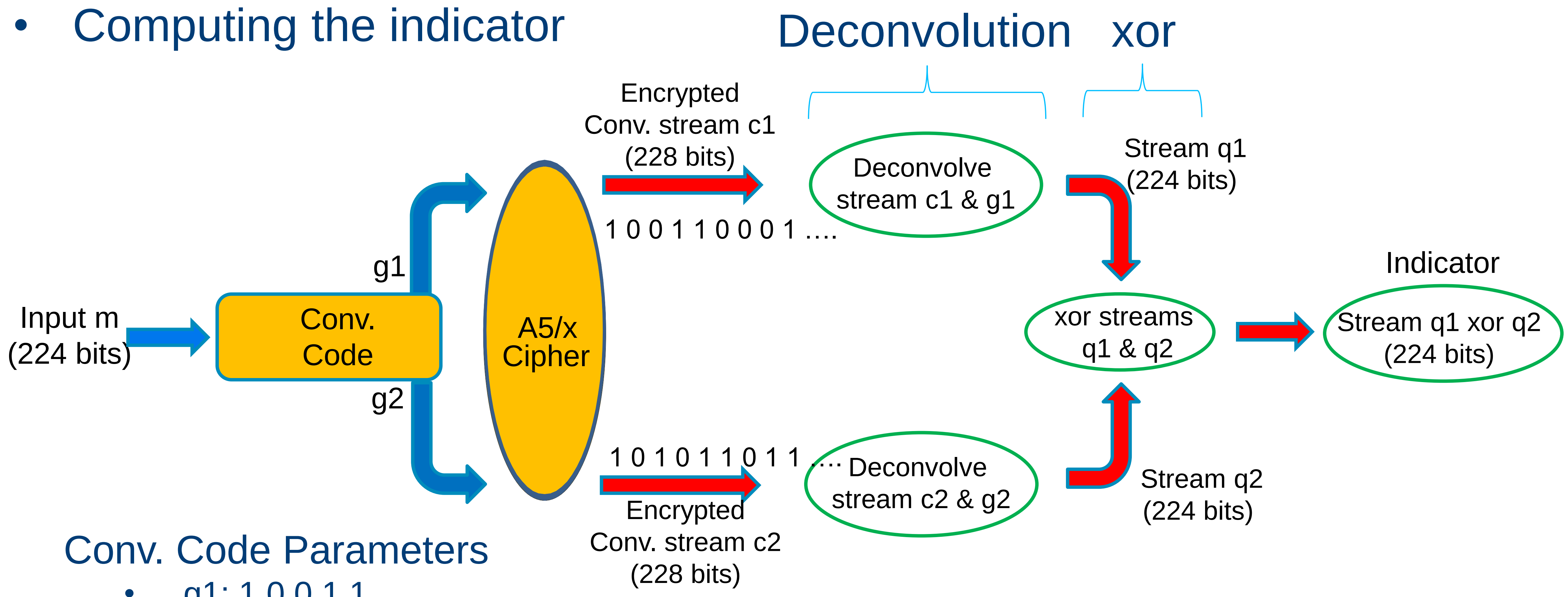
Concept Overview



- Demonstration uses SACCH control channel
  - Compromise of SACCH also compromises voice (same key)
  - Works for any SACCH message
    - Indicator/fingerprint is independent of the message
    - Knowledge of plaintext is not needed

Concept Overview

- Computing the indicator



Conv. Code Parameters

- g1: 1 0 0 1 1
- g2: 1 1 0 1 1

Concept Overview

- The indicator “q1 xor q2” is
  - Computed using the full convolutional codeword
    - Need 4x114 bursts
  - Independent of SACCH message
    - Fully determined by 1)The cipher stream and 2) Convolutional code
  - Full indicator length 224 bits
    - More than sufficient to identify a 64 bit key

Test Lab Setup & Demonstration

Hardware

- Various unlocked cellular devices
- 2G compliant Programable SIM cards
- PC SmartCard reader/writer
- Ettus Research N210 WBX
- Mini GPS ref. clock
- AirSpy SDR
- Various antennas

Software

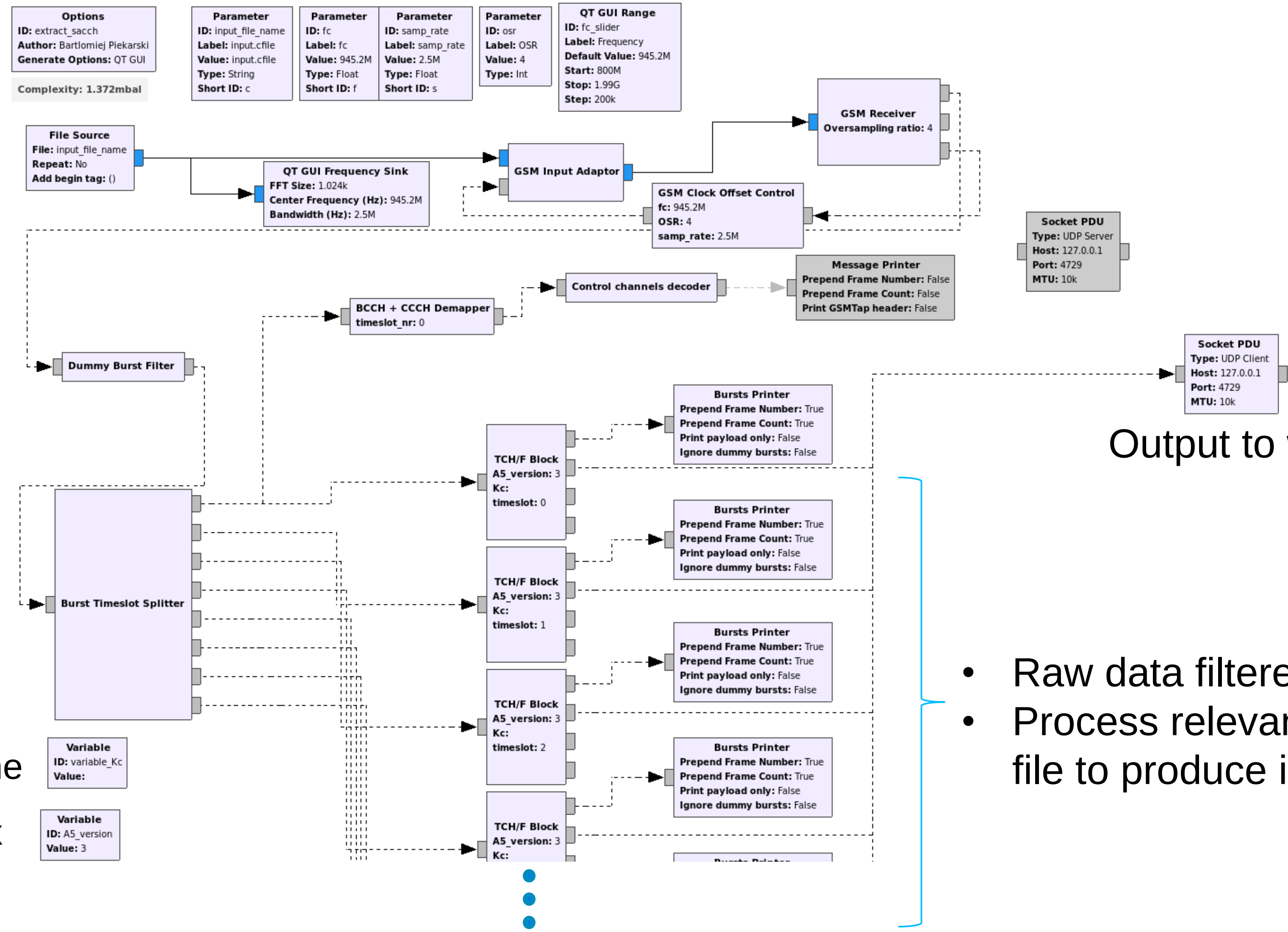
- RangeNetworks SDMN 7.0.4
- RangeNetworks OpenBTS 7.0.4
- PySIM
- GNU Radio 3.7.0
- GR-GSM
- GNU Octave 5.1

Lab Configuration

- All GSM testing run under RF isolation
- Cipher Mode configured for A5/3
- SACCH random neighbor protection enabled
- Random padding filler protection enabled

GR-GSM Configuration

Opening file, parsing, GSM signal processing



Output to wireshark

Kc/None

A5/x

- Raw data filtered into files.
- Process relevant timeslot file to produce indicator

Considerations in Mounting Attack

- Small key size: A5/1 & A5/3
  - Key length up to 64 bits
  - NIST guidance: 112 bit security strength for key agreement

- Rainbow table computation
  - Estimates based on 1 rig (4x NVIDIA GTX1080) proof-of-concept using opencl
  - A5/1, 64 bit key: Obtain 10% coverage using 500 rigs for 200 days
  - A5/3, 1 epoch, 64 bit key: Obtain 10% key coverage using 500 rigs for 263 days

Cellular Security Discussion

Cellular Security

- Indicator attack possible in GSM voice:
  - Small key size (e.g. At most 64 bits for A5/1 & A5/3)
    - Up to 64 bit key size for A5/1 & A5/3
    - NIST guidance: 112 bit security strength for key agreement
  - Ciphering performed after error control coding

- Additional attacks on GSM include:
  - Karsten Nohl (DEFCON 2010) “Attacking phone privacy”
  - Barkan et al. 2006 “Instant Ciphertext-only Cryptanalysis of GSM encrypted communication”
  - False Basestation attacks

Cellular Security

- Beyond GSM into 3G-to-5G:
  - Reduced security risk
  - Minimum encrypting key size of 128 bits
  - Error control coding applied **after** encryption not before

- Cellular industry actively studying solutions for GSM security
  - 3GPP TR 33.809 v0.5.0 “Study on 5G Security Enhancements against False Basestations”

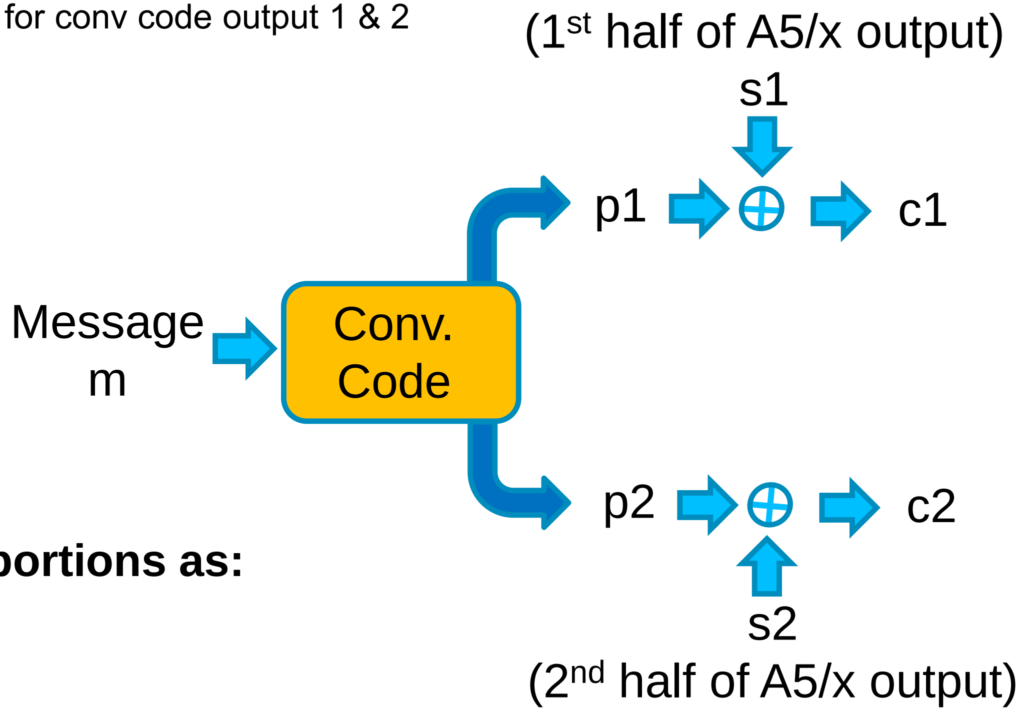
Q&A
Thank You

Appendix

Appendix

Consider the addition of the A5/x cipherstream to the codeword

- Separate cipherstream into portion xor'd for conv code output 1 & 2
- Let output 1 cipherstream be: s1
- Let output 2 cipherstream be: s2



Denote the resulting ciphertext portions as:

- c1 = s1 + p1 = s1 + m*g1
- c2 = s1 + p2 = s2 + m*g2

Appendix

The key to the attack is that the ciphertext portions can also be divided by g1 & g2 respectively for quotient q1 & q2

- C1 = s1 + p1 = s1 + m*g1 = (q1*g1 + r1) + m*g1
- C2 = s2 + p2 = s2 + m*g2 = (q2*g2 + r2) + m*g2

Rearranging c1&c2 we can now write

- C1 = (q1*g1 + r1) + m*g1 = (q1 + m)*g1 + r1
- C2 = (q2*g2 + r2) + m*g2 = (q2 + m)*g2 + r2

By deconvolving the ciphertext c1&c2 by g1&g2 respectively we can product the quotients

- (q1+m)
- (q2+m)

Adding these quotients generates (q1+q2) which is independent of the “m” :

- (q1+m) + (q2+m) = (q1+q2)