Strengths and Weaknesses of KASUMI: Analysis and Testing of the Cipher

Candidate: Teresa Tanzi
Supervisor: Andrea Visconti

April 21, 2022

Context

Purpose:

- analysis of the encryption algorithm used in 3G
- implementation of an attack

Mobile telecommunication system:

- 7.1 bilion of smartphone users → 89.9% of the global population
- 600 PB of voice call and packet data transmitted everyday
- wireless channel → necessity of encryption

KASUMI

KASUMI:

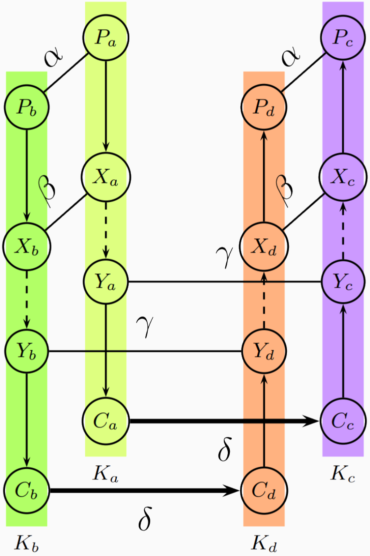
- 64-bit block cipher
- 128-bit key K
- 8 rounds
- recursive Feistel structure
- linear key schedule

$\Longrightarrow$ used in OFB mode as a stream cipher in $f8$, A5/3 and GEA3

State of the art

Attacks on KASUMI:

- single-key differential attacks
- related-key differential attacks



Sandwich and rectangle+ attacks

Sandwich:

- adaptive chosen plaintext and ciphertext
- 4 related keys
- 2^{26} data
- 2^{32} encryptions
- 76% success rate

$\Longrightarrow$ practically feasible

Rectangle+:

- chosen plaintext
- 4 related keys
- $2^{41.6}$ data
- $2^{41.6}$ encryptions
- 95% success rate

Both based on the same 7-round related-key differential

Sandwich attack implementation (1)

1. Data collection:

Sandwich attack implementation (1)

1. Data collection:
   a. Generate 2^{24} pairs (C_a, C_b) s.t. $C_a^R = A$ and $P_b = P_a \oplus \alpha$

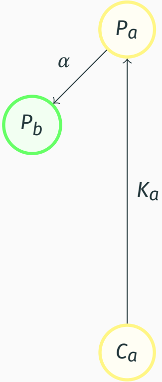
C_a

Sandwich attack implementation (1)

1. Data collection:
   a. Generate 2^{24} pairs (C_a, C_b) s.t. $C_a^R = A$ and $P_b = P_a \oplus \alpha$

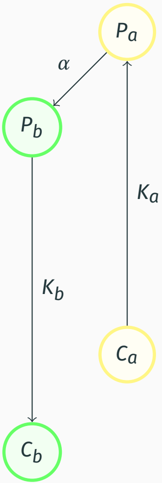
Sandwich attack implementation (1)

1. Data collection:
    a. Generate 2^{24} pairs (C_a, C_b) s.t. $C_a^R = A$ and $P_b = P_a \oplus \alpha$



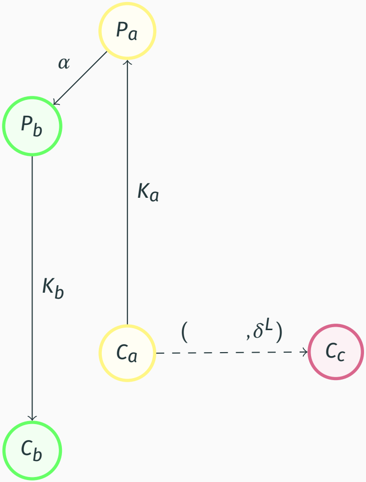
Sandwich attack implementation (1)

1. Data collection:
   a. Generate 2^{24} pairs (C_a, C_b) s.t. $C_a^R = A$ and $P_b = P_a \oplus \alpha$

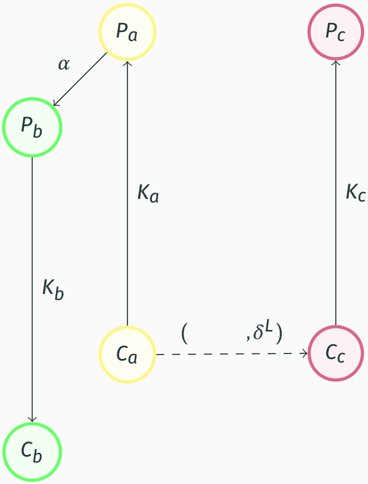


Sandwich attack implementation (1)

1. Data collection:
   a. Generate 2^{24} pairs (C_a, C_b) s.t. $C_a^R = A$ and $P_b = P_a \oplus \alpha$
   b. Generate 2^{24} pairs (C_c, C_d) s.t. $C_c^R = A \oplus \delta^L$ and $P_d = P_c \oplus \alpha$

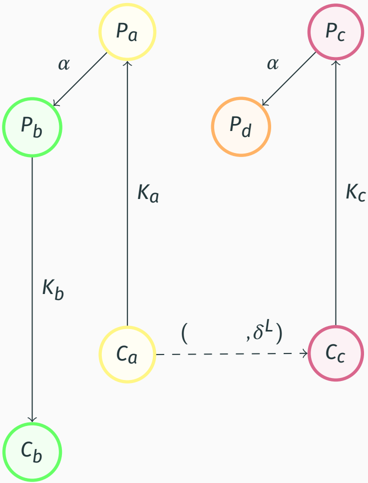


1. Data collection:
   a. Generate 2^{24} pairs (C_a, C_b) s.t. $C_a^R = A$ and $P_b = P_a \oplus \alpha$
   b. Generate 2^{24} pairs (C_c, C_d) s.t. $C_c^R = A \oplus \delta^L$ and $P_d = P_c \oplus \alpha$



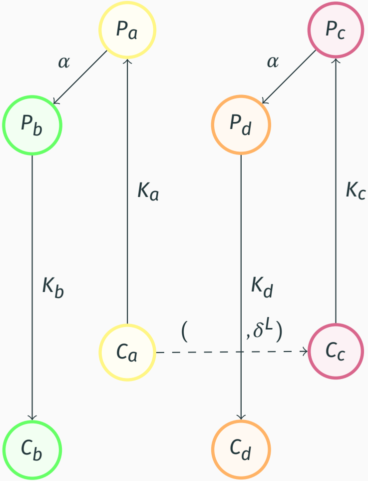
Sandwich attack implementation (1)

1. Data collection:
   a. Generate 2^{24} pairs (C_a, C_b) s.t. $C_a^R = A$ and $P_b = P_a \oplus \alpha$
   b. Generate 2^{24} pairs (C_c, C_d) s.t. $C_c^R = A \oplus \delta^L$ and $P_d = P_c \oplus \alpha$



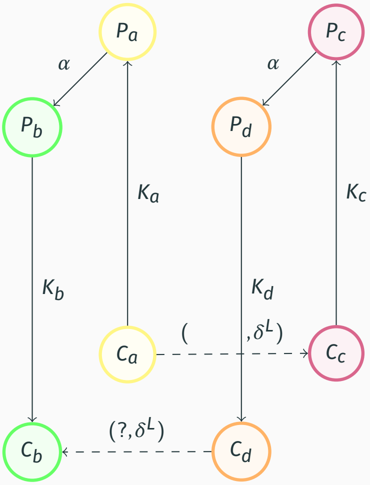
Sandwich attack implementation (1)

1. Data collection:
   a. Generate 2^{24} pairs (C_a, C_b) s.t. $C_a^R = A$ and $P_b = P_a \oplus \alpha$
   b. Generate 2^{24} pairs (C_c, C_d) s.t. $C_c^R = A \oplus \delta^L$ and $P_d = P_c \oplus \alpha$



Sandwich attack implementation (1)

1. Data collection:
   a. Generate 2^{24} pairs (C_a, C_b) s.t. $C_a^R = A$ and $P_b = P_a \oplus \alpha$
   b. Generate 2^{24} pairs (C_c, C_d) s.t. $C_c^R = A \oplus \delta^L$ and $P_d = P_c \oplus \alpha$
   c. Keep quartets (C_a, C_b, C_c, C_d) for which $C_b^R \oplus C_d^R = \delta^L$

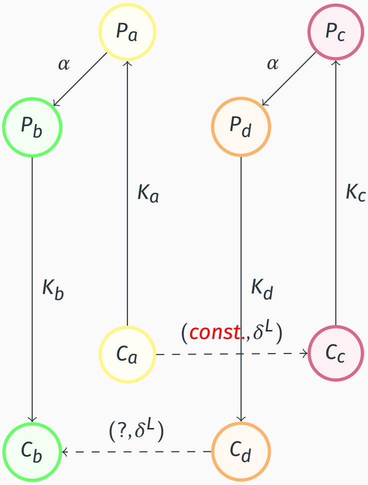


Sandwich attack implementation (1)

1. Data collection:
   a. Generate 2^{24} pairs (C_a, C_b) s.t. $C_a^R = A$ and $P_b = P_a \oplus \alpha$
   b. Generate 2^{24} pairs (C_c, C_d) s.t. $C_c^R = A \oplus \delta^L$ and $P_d = P_c \oplus \alpha$
   c. Keep quartets (C_a, C_b, C_c, C_d) for which $C_b^R \oplus C_d^R = \delta^L$
2. Identifying the right quartets:

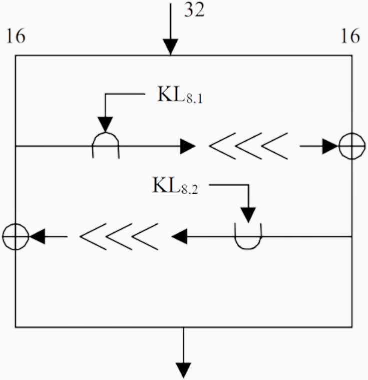
   Keep quartets (C_a, C_b, C_c, C_d) for which $C_a^L \oplus C_c^L = const.$

   $\Longrightarrow$ right quartets



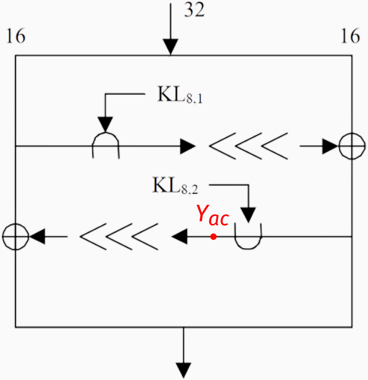
Sandwich attack implementation (2)

3. Analyzing right quartets:



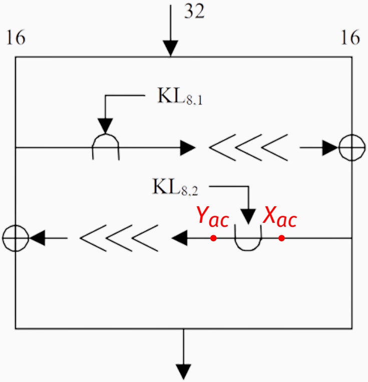
Sandwich attack implementation (2)

3. Analyzing right quartets:
   a. Guess $KO_{8,1}$ and $KI_{8,1} \rightarrow KL_{8,2}$



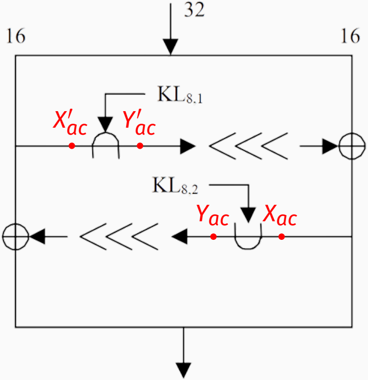
Sandwich attack implementation (2)

3. Analyzing right quartets:
   a. Guess $KO_{8,1}$ and $KI_{8,1} \rightarrow KL_{8,2}$



Sandwich attack implementation (2)

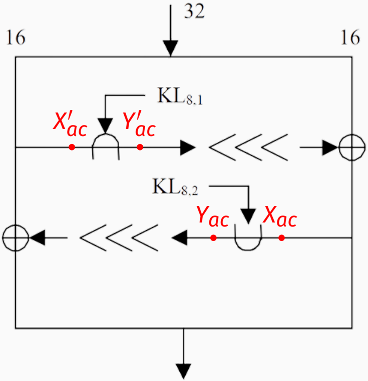
3. Analyzing right quartets:
    a. Guess $KO_{8,1}$ and $KI_{8,1} \rightarrow KL_{8,2}$
    b. Guess $KO_{8,3}$ and $KI_{8,3} \rightarrow KL_{8,1}$



Sandwich attack implementation (2)

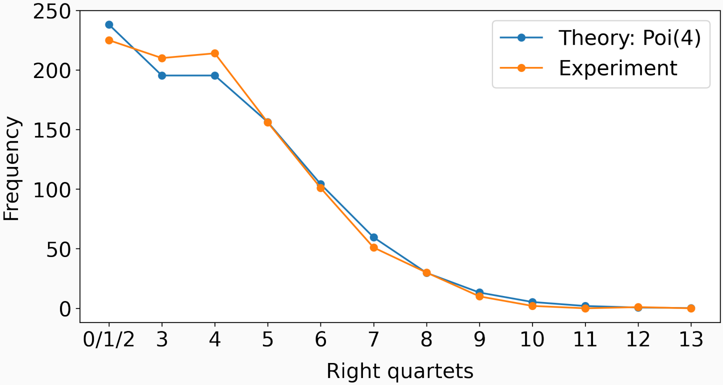
3. Analyzing right quartets:
   a. Guess $KO_{8,1}$ and $KI_{8,1} \rightarrow KL_{8,2}$
   b. Guess $KO_{8,3}$ and $KI_{8,3} \rightarrow KL_{8,1}$
4. Find the right key:

   Guess the remaining bits of K and perform trial encryptions $\Longrightarrow$ correct key



Right quartets distribution

Theoretical distribution: Poisson with rate 4
Empirical success rate: 77.5% (theoretical: 76.1%)



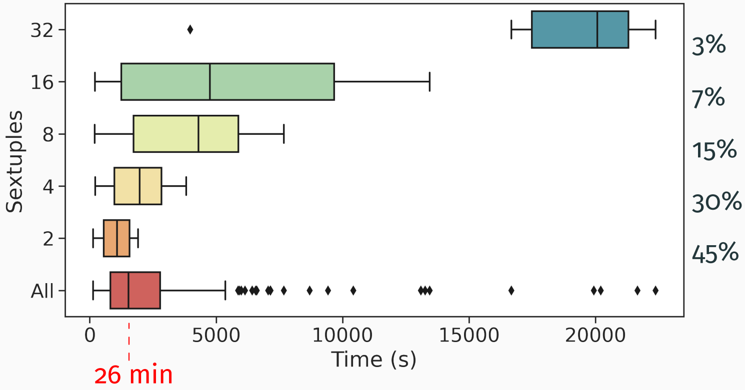
Execution time (1)

Phases 1-2: 30 s

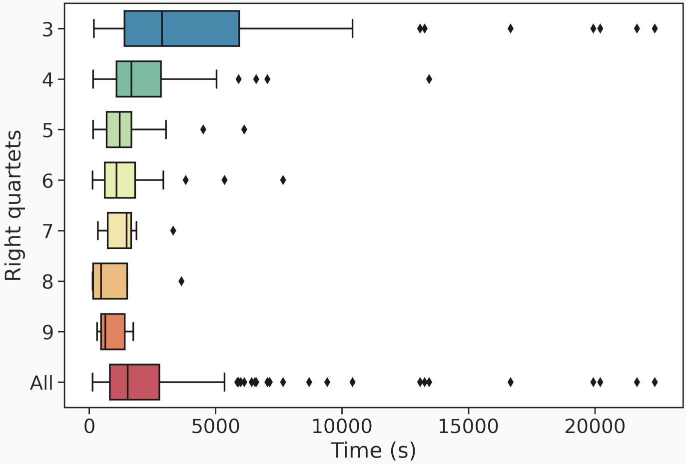
Phase 3: 1-2 min

Phase 4: ~ 15 min per exhaustive search

on a standard machine



Execution time (2)



Conclusions

Sandwich attack:

- good execution time
- faster and higher success rate with more initial data

Rectangle+ attack:

- unfeasible in practice
- implementable in a distributed fashion exploiting some big data processing framework

Not applicable to the 3G stream ciphers, but showing that KASUMI is not secure

Thank you for your attention

☺