

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

Orr Dunkelman, Nathan Keller, and Adi Shamir

Faculty of Mathematics and Computer Science
Weizmann Institute of Science
P.O. Box 26, Rehovot 76100, Israel

Presenter: Eun-kyung Kim

News

Tougher GSM, 3G security cracked

Newer GSM encryption hit using sandwich

The security of GSM phone calls dropped again late Monday with [word](#) that the standard's second, more guarded encryption standard has been broken. Following a [first crack](#) of the simpler standard, researchers at the Weizmann Institute of Science say they have cracked the A5/3 security cipher (nicknamed Kasumi) by using what's known as a "sandwich" attack. The group accomplished its goal by creating a distinguishing trait for the key and using just four related keys to determine the key for Kasumi itself.

While breaking the security takes time, the approach theoretically leaves GSM more directly exposed to call interceptions and other threats. Most cellular carriers still use the lower-grade GSM quality (A5/1) and the discovery makes switching to Kasumi impractical. Kasumi and is potentially exposed as well.

New GSM encryption cracked

Mobile algorithm broken by simple dual-core Linux PC

By Jaikumar Vijayan | [Computerworld US](#)

Published: 06:14 GMT, 15 January 10



An encryption algorithm designed to protect calls on GSM phones has been broken by three cryptographers using only a dual-core, Intel-based Dell Latitude

In a just-released paper, the Weizmann Institute of Science researchers developed called a complete 128-bit key for otherwise known as communications of

Orr Dunkelman, or improves on research that is theoretically attack

"What the researchers found was that computing time you need to find the key, Dunkelman says, is less than the computing power it takes to find the key. Now it can be done on a laptop computer. We can all agree that's a bit disturbing."

The two other researchers involved in the report are Nathan Keller and Adi Shamir, who is one of the inventors of the RSA encryption algorithm.

'Sandwich attack' busts new cellphone crypto

Kasumi cipher cracked (in theory)

By [Dan Goodin in San Francisco](#) • [Get more from this author](#)

Posted in [Security](#), 13th January 2010 00:17 GMT


[Free whitepaper – Taking control of your data demons: Dealing with unstructured content](#)

A new encryption scheme for protecting 3G phone networks hasn't even gone into commercial use and already cryptographers have cracked it - at least theoretically.

In a paper published Tuesday, the cryptographers showed that the Kasumi cipher, which is also referred to as A5/3, can be broken using what's known as a related-key attack, in which a message encrypted with one key is later changed to one or more different keys.

The team dubbed the technique a sandwich attack because it was broken into three parts: two thick slices at the top and bottom and a thin slice in the middle.

Abstract

- History of Cryptosystem of GSM phone
 - A5/1 & A5/2 stream cipher
 - A5/3 block cipher (KASUMI)
 - Modified version of MISTY
 - Attack on the A5/3
 - **sandwich attack**
 - construct a simple distinguisher for 7 of the 8 rounds of KASUMI with 2^{-14} prob.
 - analyzing the single remaining round: can derive the complete 128 bit key of the full KASUMI by
 - using 4 related keys
 - 2^{26} data
 - 2^{30} bytes of memory
 - 2^{32} time
 - MISTY: cannot break in less than 2^{128} complexity of exhaustive search
 - KASUMI is a much weaker cryptosystem
-  small complexity: 2h on a single PC

A5 - Encryption Algorithm

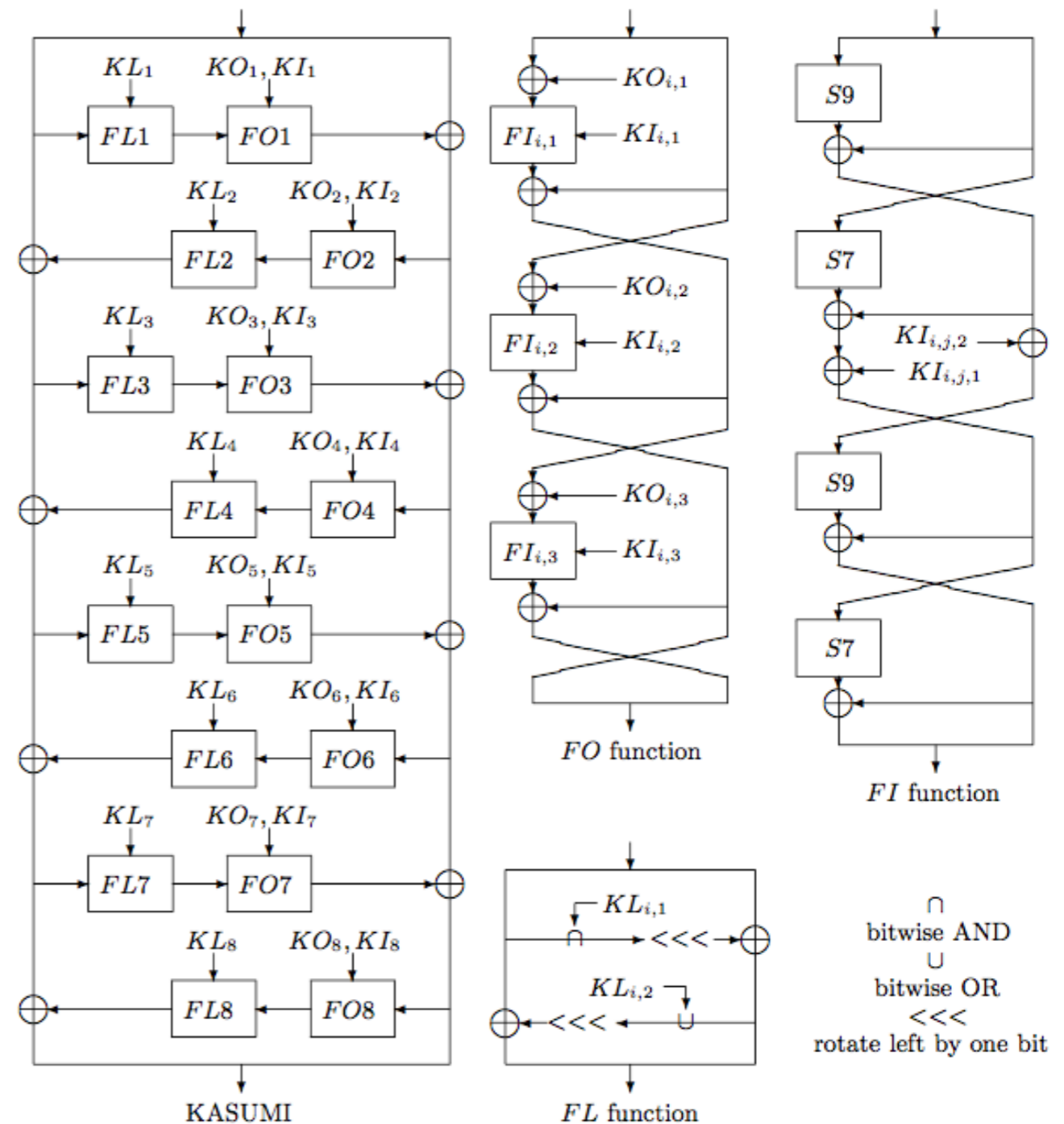
- A5 is a stream cipher
 - Used to encrypt voice communication
 - Provides privacy to calls against eavesdroppers
- Variants
 - A5/1: original A5 algorithm
 - employed in Europe and US
 - A5/2
 - employed outside of Europe and US
 - Attack in December 2009
 - 2 terabyte rainbow table for A5/1 [by Karsten Nohl]
 - easy to derive the session key of any particular conversation with minimal hardware support
 - A5/3
 - not stream cipher, but Block cipher
 - GSM Association Security Group and 3GPP design
 - based on MISTY block cipher in 3G mobile system

MISTY

- Block cipher, published in 1997 by Matsui
- Features
 - 64 bit blocks
 - 128 bit keys
 - complex recursive Feistel structure with 8 rounds
- Attack
 - No attack known its full version
 - The best published attack
 - can be applied to a 6-round reduced variant of the 8-round MISTY
 - Time complexity of more than 2^{123}

KASUMI

- Block cipher, based on MISTY
- Features
 - 64 bit blocks
 - 128 bit keys
 - complex recursive Feistel structure with 8 rounds
- Purpose
 - More faster
 - More hardware-friendly
- Goal
 - Simplify the key schedule



Key schedule of KASUMI

- Much simpler than the original key schedule of MISTY
- 128-bit key K is divided into eight 16-bit sub keys

$$K = K_1 || K_2 || K_3 || K_4 || K_5 || K_6 || K_7 || K_8$$

- Modified key is divided into 16-bit sub keys

$$K'_i = K_i \oplus C_i$$

- Key-schedule function (2D table lookup)
 - 16-bit XOR
 - eight 1-bit cyclic left shift
 - eight 5-bit cyclic left shift
 - eight 8-bit cyclic left shift
 - eight 13-bit cyclic left shift

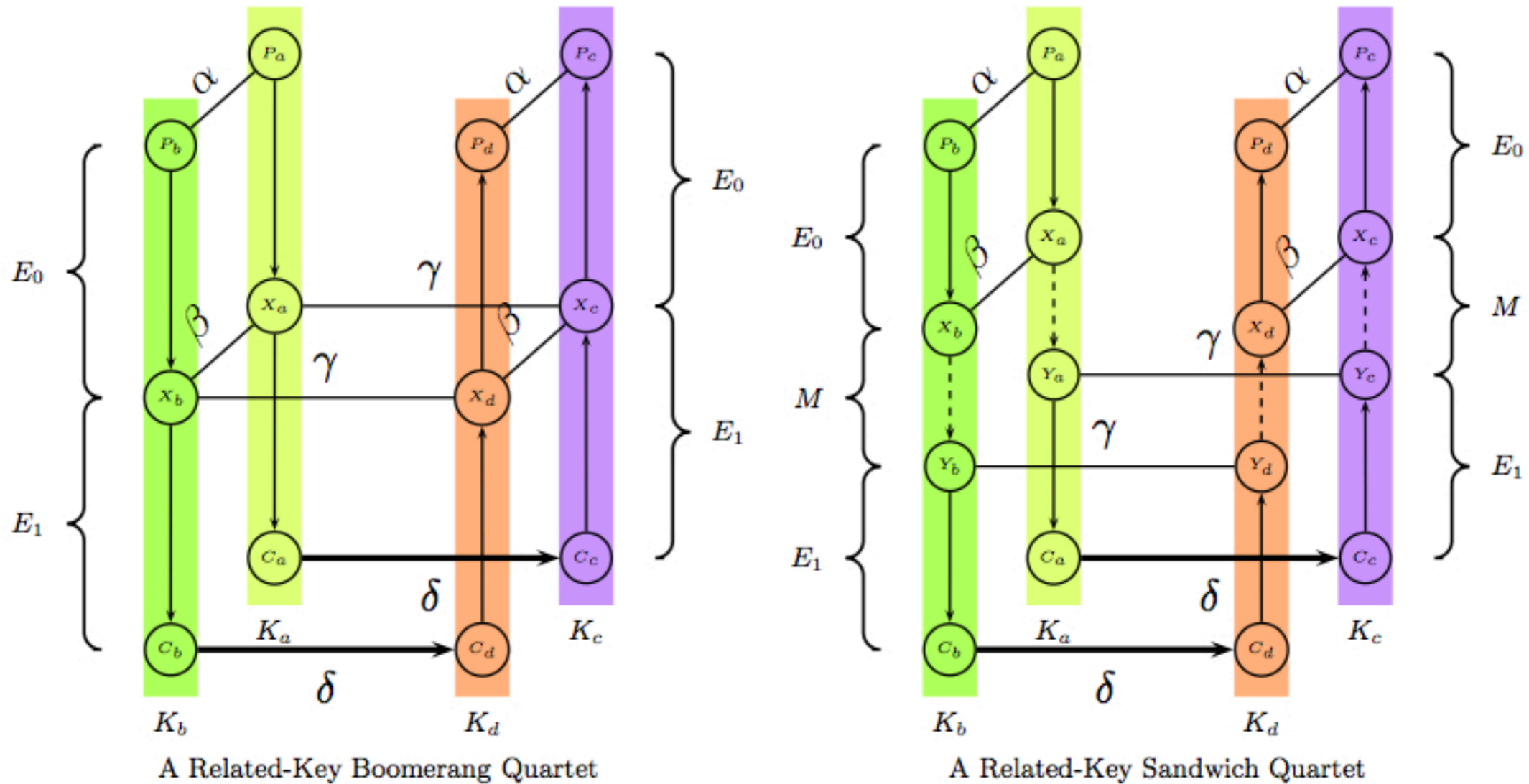
Overview of Sandwich Attacks

- An improved of the Boomerang Attack
- Distinguisher structure
 - “Bread” + “Meat” + “Bread”
 - Top & Bot parts have high probability differential characteristics
 - It can be combined into consistent quartet structures by the standard boomerang attack
 - Additional step, versus boomerang attack, is middle slice
 - It significantly reduce the probability of the resulting boomerang structure

Related-Key Sandwich Attacks

- Combination of the sandwich attack + related-key differential attack
- Cipher as a cascade of 3 sub-ciphers $E = E_1 \circ M \circ E_0$
 - related-key differential $\alpha \rightarrow \beta$ for E_0 under key difference ΔK_{ab} with probability p
 - related-key differential $\gamma \rightarrow \delta$ for E_1 under key difference ΔK_{ac} with probability q
 - The pair (P_a, P_b) is a right pair wrt the 1st differential
 - (C_a, C_c) and (C_b, C_d) are right pairs wrt the 2nd differential

Related-Key Boomerang & Sandwich Attacks



$$(X_a \oplus X_b = \beta) \wedge (X_a \oplus X_c = \gamma) \wedge (X_b \oplus X_d = \gamma)$$

$$X_c \oplus X_d = (X_c \oplus X_a) \oplus (X_a \oplus X_b) \oplus (X_b \oplus X_d) = \beta \oplus \gamma \oplus \gamma = \beta$$

$$(X_a \oplus X_b = \beta) \wedge (Y_a \oplus Y_c = \gamma) \wedge (Y_b \oplus Y_d = \gamma)$$

Related-key Sandwich Attack on the Full KASUMI

- Apply the distinguisher to round 1-7
- Retrieve subkey material in round 8

$$\Delta K_{ab} = (0, 0, 8000_x, 0, 0, 0, 0, 0)$$

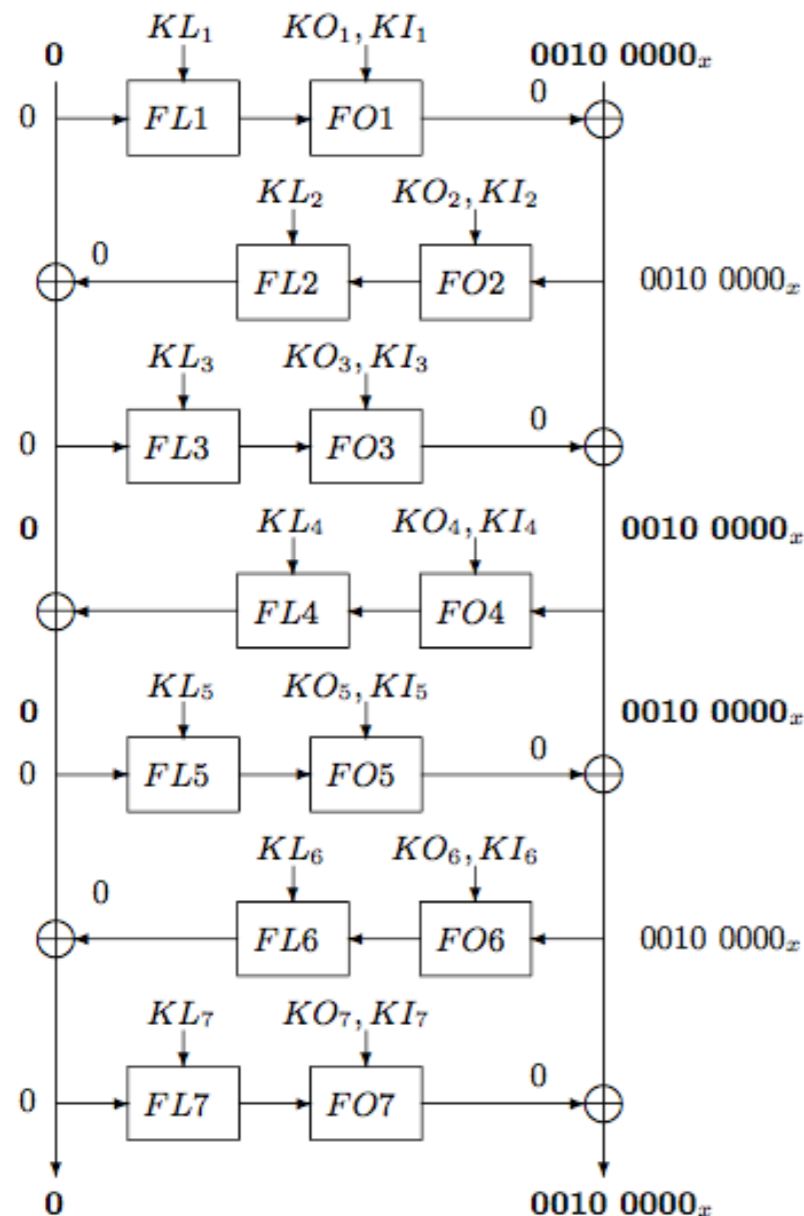
$$\Delta K_{ac} = (0, 0, 0, 0, 0, 0, 8000_x, 0)$$

$$K_a, K_b = K_a \oplus \Delta K_{ab}, K_c = K_a \oplus \Delta K_{ac}, \text{ and } K_d = K_c \oplus \Delta K_{ab}$$

Attack Algorithm

- Data Collection Phase
- Identifying the Right Quartets
- Analyzing Right Quartets
- Finding the Right Key

1. Data Collection



- Choose a structure of 2^{24} ciphertexts of the form $C_a = (X_a, A)$, where A is fixed and X_a assumes 2^{24} arbitrary different values. Ask for the decryption of all the ciphertexts under the key K_a and denote the plaintext corresponding to C_a by P_a . For each P_a , ask for the encryption of $P_b = P_a \oplus (0_x, 0010\ 0000_x)$ under the key K_b and denote the resulting ciphertext by C_b . Store the pairs (C_a, C_b) in a hash table indexed by the 32-bit value C_b^R (i.e., the right half of C_b).
- Choose a structure of 2^{24} ciphertexts of the form $C_c = (Y_c, A \oplus 0010\ 0000_x)$, where A is the same constant as before, and Y_c assumes 2^{24} arbitrary different values. Ask for the decryption of the ciphertexts under the key K_c and denote the plaintext corresponding to C_c by P_c . For each P_c , ask for the encryption of $P_d = P_c \oplus (0_x, 0010\ 0000_x)$ under the key K_d and denote the resulting ciphertext by C_d . Then, access the hash table in the entry corresponding to the value $C_d^R \oplus (0_x, 0010\ 0000_x)$, and for each pair (C_a, C_b) found in this entry, apply Step 2 on the quartet (C_a, C_b, C_c, C_d) .

2^{48} possible quartets \rightarrow 2^{16} quartets

2: Identifying the Right Quartets

- (a) Insert the approximately 2^{16} remaining quartets (C_a, C_b, C_c, C_d) into a hash table indexed by the 32-bit value $C_a^L \oplus C_c^L$, and apply Step 3 only to bins which contain at least three quartets.

- very high probability only the right quartets remain after this filtering

$$\binom{2^{16}}{3} \cdot 2^{-64} \leq 2^{-18}$$

3: Analyzing Right Quartets

- (a) For each remaining quartet (C_a, C_b, C_c, C_d) , guess the 32-bit value of $KO_{8,1}$ and $KI_{8,1}$. For the two pairs (C_a, C_c) and (C_b, C_d) use the value of the guessed key to compute the input and output differences of the OR operation in the last round of both pairs.

- (b) Guess the 32-bit value of $KO_{8,3}$ and $KI_{8,3}$, and use this information to compute the input and output differences of the AND operation in both pairs of each quartet.

4: Finding the Right Key

For each value of the 96 bits of $(KO_{8,1}, KI_{8,1}, KO_{8,3}, KI_{8,3}, KL_{8,1}, KL_{8,2})$ suggested in Step 3, guess the remaining 32 bits of the key, and perform a trial encryption.

Result of Attack

- Data complexity
 - 2^{25} chosen ciphertexts
 - 2^{25} adaptively chosen plaintexts encrypted/decrypted under one of four keys
- Time complexity
 - To find the last 32 bits of the key
 - Approximately equal to 2^{32} encryptions
- Success probability
 - 76% (this is the probability of having at least three right pairs in the data pool)
- Memory complexity
 - To store 2^{26} plaintext/ciphertext pairs (each pair: 16 bytes)
 - Total amount of memory: 2^{30} bytes (1 GByte)
- It was child play :)