A New Attack Against Khazad

Frédéric Muller

DCSSI Crypto Lab

Summary

$\rightarrow$ Khazad is a recent block cipher

$\rightarrow$ Proposed for the NESSIE project by P. Barreto and V. Rijmen

$\rightarrow$ Byte-oriented structure

$\rightarrow$ Some similarities with AES

Characteristics

- Block size = 64 bits
- Key size = 128 bits
- Number of rounds = 8

Results in this paper

⇒ New observations about the structure of the cipher

⇒ New attack against 5 rounds of the cipher
(the full version has 8 rounds)

The attack has

- Time complexity of 2^{91} steps
- Data complexity of 2^{64} blocks

⇒ Faster than exhaustive search

Outline of the presentation

- Quick description of Khazad
- Previous attacks against Khazad
- New observations : cascade eliminations
- A first attempt of cryptanalysis
- The actual attack
- Conclusions about the structure of Khazad
- Conclusion

Facts about Khazad

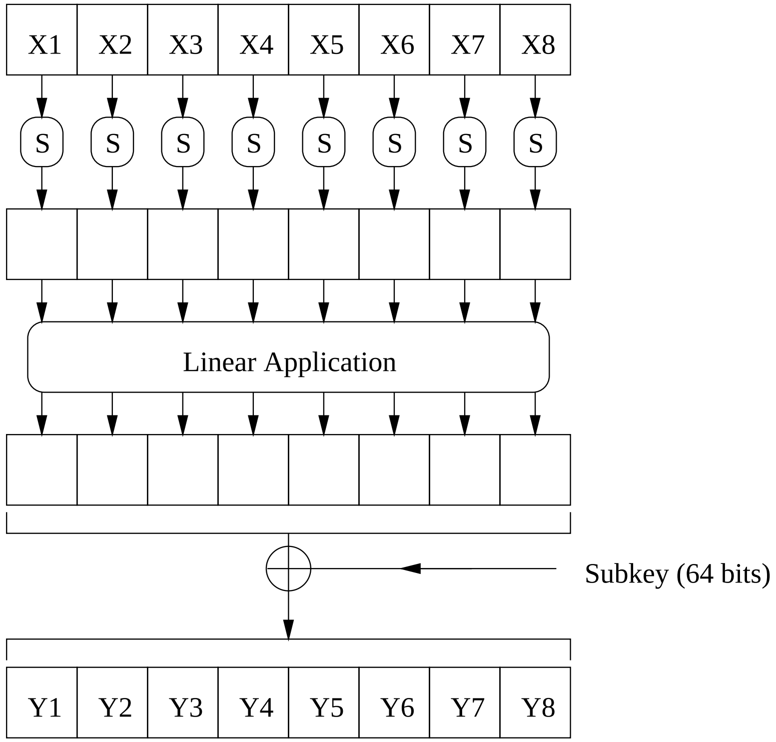
- It is a Substitution-Permutation Network
- Round function consists in

  $\rightarrow$ a layer of S-box

  $\rightarrow$ a linear layer

  $\rightarrow$ a subkey addition layer
- All components are self-inverse
- The cipher himself is an involution, except the key schedule

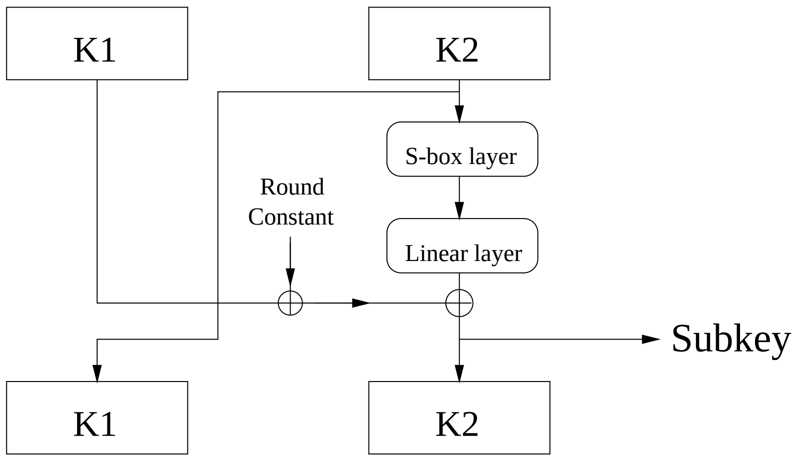
The round function



Our attack only uses the high-level structure of the cipher

The subkey derivation

Initially, split the secret key (128 bits) into two parts of 64 bits. Then, a Feistel network is applied



Basically, the new subkey is obtained by applying the round function using the two previous subkeys

Known Attacks

Best known attack :

$\rightarrow$ Integral (aka "Square") attack proposed by the authors of Khazad

- Focus on the Byte Distributions over a set of 256 plaintexts
- Notations
  - A = "All" possible values for one byte
  - C = "Constant" value for one byte
  - S = "Sum" of the values for one byte is 0 (balanced byte)

Integral Attacks against Khazad

It is based on a two-round distinguisher.

Introduce 256 plaintexts with distribution :

A	**C**	**C**	**C**	**C**	**C**	**C**	**C**

$\downarrow$ 1 round

A	**A**	**A**	**A**	**A**	**A**	**A**	**A**

$\downarrow$ 1 round

S	**S**	**S**	**S**	**S**	**S**	**S**	**S**

A S distribution has probability 2^{-8} for random data

$\Rightarrow$ Resulting 3-round attack requires $2^8 \times 2^8 = 2^{16}$ operations

$\Rightarrow$ Extension to 4 rounds : $2^{64} \times 2^{16} = 2^{80}$ operations

Other cryptanalytic results against Khazad

- Differential-based attacks have been studied throughout the NESSIE project
- An impossible differential attack against 3 rounds has complexity 2^{64}

  $\Rightarrow$ Not a very promising direction
- All components are involutions
  - Biryukov (FSE'03) presented new observations concerning the cycle structure of Khazad
  - Class of weak keys that can be attacked with complexity 2^{43} for 5 rounds of Khazad

A new observation : Cascade eliminations

Starting point : redundancy between the key scheduling and the encryption function

- $\mathrm{Round}_x(\cdot)$ = round function with subkey x
- K_i = i-th subkey of Khazad
- C_i = the corresponding round constant

$$K_i = \mathrm{Round}_{K_{i-2} \oplus C_i}(K_{i-1})$$

Eliminations (II)

An attack framework

- $\rightarrow$ Guess the first subkey K_0
- $\rightarrow$ Introduce plaintext = a well chosen function of K_0
- $\rightarrow$ Write exact equations and obtain eliminations
- $\rightarrow$ Predict the ciphertext / intermediate values

Eliminations (III)

Let us choose

$$\text{Plaintext} = K_0 \oplus \text{Round}_0^{-1}(0)$$

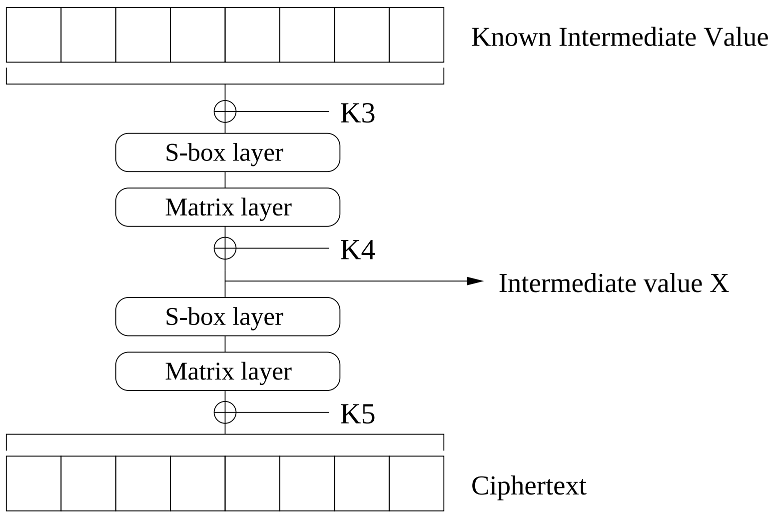
Then,

$$\begin{aligned}
\text{State}_0 &= K_0 \oplus \text{Plaintext} = \text{Round}_0^{-1}(0)\\
\text{State}_1 &= \text{Round}_{K_1}(\text{State}_0) = K_1\\
\text{State}_2 &= \text{Round}_{K_2}(\text{State}_1) = K_0 \oplus C_2\\
\text{State}_3 &= \text{Round}_{K_3}(\text{State}_2) = K_3 \oplus f(K_0)
\end{aligned}$$

$\Rightarrow$ We can predict the internal state after 2.5 rounds

A first attempt to break Khazad

What is left with 5 rounds-Khazad ?



a LC (Linear Combination) of the bytes of X can be computed

- By guessing one byte of K_3 and a proper LC of K_4's bytes
  $\Rightarrow$ 16 bits to guess
- As a polynomial in the ciphertext bits and K_5 bits

A first attempt (II)

Summary :

- Guess 64 bits of K_0
- Introduce a chosen block of plaintext
- Guess 16 bits from K_3 and K_4
- It provides 8 binary equations of degree ≤ 8 with unknowns the bits of K_5

A simple analysis shows that there are 2048 monomials

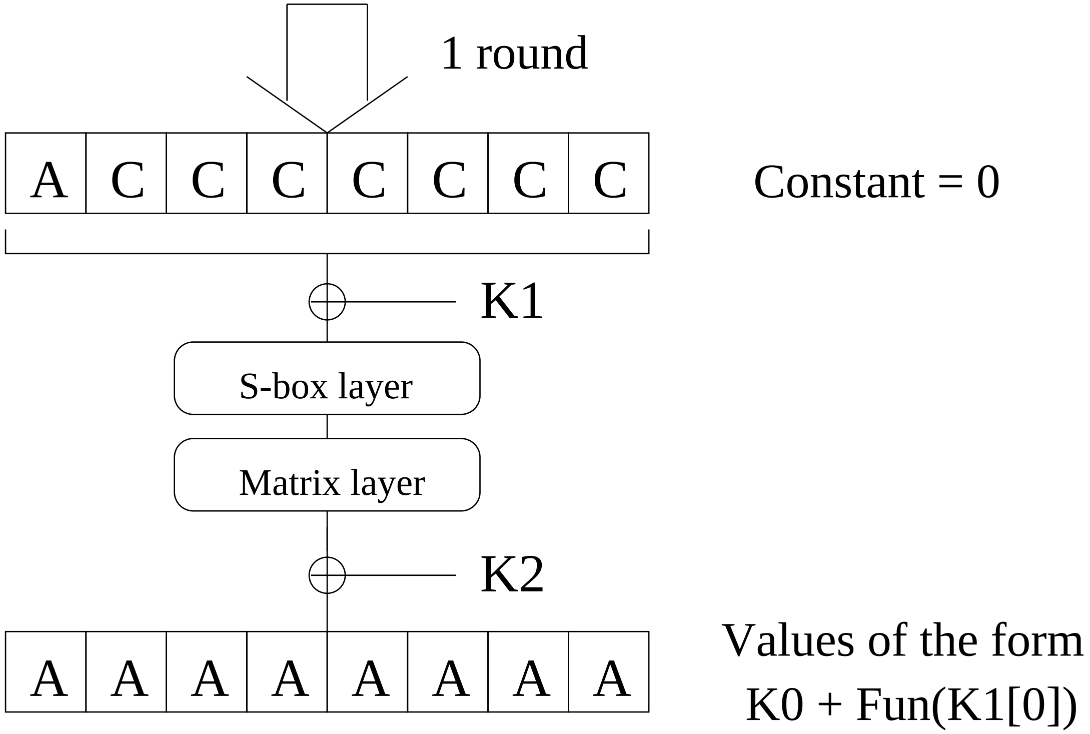
$\Rightarrow$ Not enough equations to solve

$\Rightarrow$ Try to work with more chosen plaintexts

Using more plaintexts

Combine the Square attack and the Cascade elimination

First, guess K_0, then choose 256 plaintexts such that



The actual attack

- Guess $K_0 \rightarrow$ 64 bits
- Introduce 256 plaintexts
  $\rightarrow$ with distributions (A,C,C,C,C,C,C,C)
- Guess 8 appropriate bits of K_1
- 256 known intermediate values after 2.5 rounds
- Guess 16 extra bits of K_3 and K_4
- Algebraic observations

Linearization

We have a polynomial system with unknowns the bits of K_5
The number of monomials m_i is 2048

Besides, there are 256 known intermediate values giving

$$256 \times 8 = 2048$$

binary relations :

$$\begin{pmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_{2048} \end{pmatrix} = \begin{bmatrix} a_{1,1} & \cdot & \cdot & \cdot & a_{1,2048} \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ a_{2048,1} & \cdot & \cdot & \cdot & a_{2048,2048} \end{bmatrix} \begin{pmatrix} m_1 \\ \cdot \\ \cdot \\ \cdot \\ m_{2048} \end{pmatrix}$$

x_i's are known internal state bits

$$\vec{x} = A\,\vec{m}$$

→ A is not invertible (kernel of rank ≥ 64)

→ Build ≥ 64 linear relations on the lines of A

→ If the x_i's do not verify all of them, the system has no solution

→ A is independent of the guess on K_1, K_3, K_4

So, after our initial guess of K_0 bits

$\Rightarrow$ Build the linear system

Build ≥ 64 linear relations verified by the x_i's

$\Rightarrow$ Extra guess (24 bits)

Verify the linear relations

$\Rightarrow$ Extra guess ($\rightarrow$ full secret key)

Analysis

Bottleneck = Linear Algebra

It is performed only 2^{64} times.

$$\text{Expected Runtime } \simeq 2^{91} \text{ S-box lookups}$$

$$\text{Data Complexity } \simeq 2^{64} \text{ blocks}$$

because we need to introduce different plaintexts for each K_0

Weaknesses of Khazad

The attack described is independent of

- the description of the S-box
- the coefficients of the matrix layer

$\Rightarrow$ The problem is the high-level structure of Khazad

- The use of the round function in the subkey derivation is a problem
- The number of rounds (8) does not offer a large security margin

Improvements and Extensions

→ Using cascade eliminations

- Possible in case of redundancy between encryption and key schedule (as for Khazad)
- Useful mostly when block size $<$ key size ($64 < 128$)
- Extension to partial cascade effects ?
  → Data Complexity $<$ Full Dictionary
- Time/Data tradeoff

→ Cryptanalysis using algebraic observations

Conclusion

Our contribution :

→ The best known attack against Khazad

→ 5 rounds vs 8 rounds for the full cipher

→ $T = 2^{91} \ll 2^{128}$

→ $D \simeq 2^{64}$

Open Problems :

- Decrease complexity of 5 round attacks
- Attacks against 6 rounds ?