Cryptanalysis of White Box DES Implementations

Louis Goubin Jean-Michel Masereel
Michaël Quisquater

PRiSM
University of Versailles St-Quentin-en-Yvelines
{Louis.Goubin,Jean-Michel.Masereel,Michael.Quisquater}@uvsq.fr

CrySCoE Workshop - June 5th, 2009

Outline

Outline

Outline

State of the Art.

- Why do we need obfuscation?
- Commercial Obfuscation.
- Security relies on heuristic methods.
- We need to continue research in this domain.

State of the Art.

- Why do we need obfuscation?
- Commercial Obfuscation.
- Security relies on heuristic methods.
- We need to continue research in this domain.

State of the Art.

- Why do we need obfuscation?
- Commercial Obfuscation.
- Security relies on heuristic methods.
- We need to continue research in this domain.

State of the Art.

- Why do we need obfuscation?
- Commercial Obfuscation.
- Security relies on heuristic methods.
- We need to continue research in this domain.

DES Obfusction Historic.

- DES obfuscation methods proposed by Chow, Eisen, Johnson and van Oorschot in 2002.
- Naked-DES cryptanalysed
  - by Chow, Eisen, Johnson and van Oorschot, and by Jacob, Boneh and Felten, in 2002.
  - by Link and Neuman in 2004.
- Nonstandard-DES cryptanalysed by Wyseur, Michiels, Gorissen and Preneel, in parallel of our work.

DES Obfusction Historic.

- DES obfuscation methods proposed by Chow, Eisen, Johnson and van Oorschot in 2002.
- Naked-DES cryptanalysed
  - by Chow, Eisen, Johnson and van Oorschot, and by Jacob, Boneh and Felten, in 2002.
  - by Link and Neuman in 2004.
- Nonstandard-DES cryptanalysed by Wyseur, Michiels, Gorissen and Preneel, in parallel of our work.

DES Obfusction Historic.

- DES obfuscation methods proposed by Chow, Eisen, Johnson and van Oorschot in 2002.
- Naked-DES cryptanalysed
  - by Chow, Eisen, Johnson and van Oorschot, and by Jacob, Boneh and Felten, in 2002.
  - by Link and Neuman in 2004.
- Nonstandard-DES cryptanalysed by Wyseur, Michiels, Gorissen and Preneel, in parallel of our work.

Outline

DES Principle.

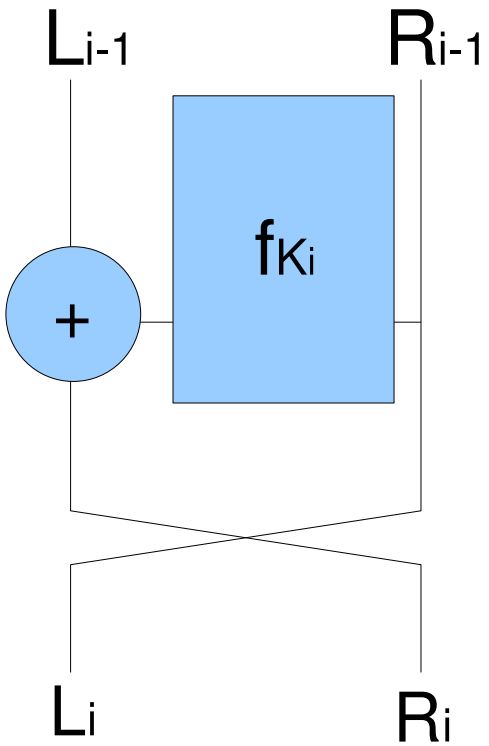

Figure: One round of DES.

DES Principle.

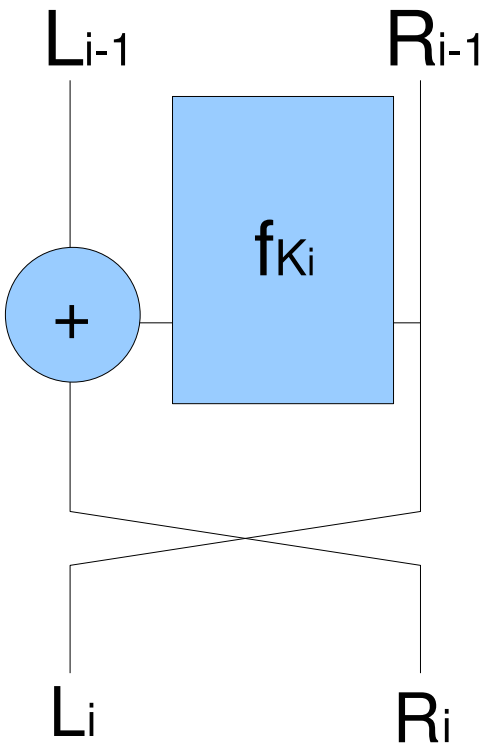

Figure: One round of DES.

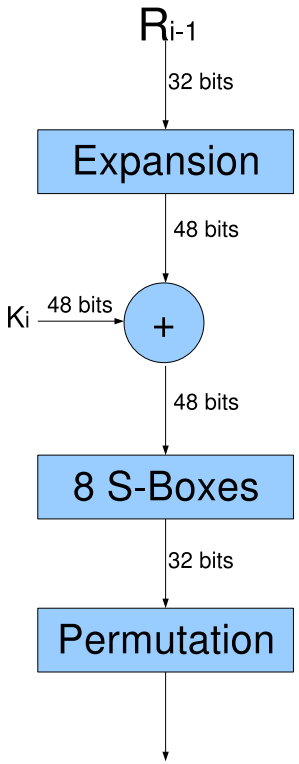

Figure: f_{K_i} function.

Outline

Obfuscation Method.
General case.

- Ideal=Replacing the function by a look-up table.
- Input and output of the DES have 64 bits.
- Concatenation and composition of many small look-up tables.
- Each subfonction preceeded and followed by randomly chosen encoding bijections.

Obfuscation Method.
General case.

- Ideal=Replacing the function by a look-up table.
- Input and output of the DES have 64 bits.
- Concatenation and composition of many small look-up tables.
- Each subfonction preceeded and followed by randomly chosen encoding bijections.

Obfuscation Method.
General case.

- Ideal=Replacing the function by a look-up table.
- Input and output of the DES have 64 bits.
- Concatenation and composition of many small look-up tables.
- Each subfonction preceeded and followed by randomly chosen encoding bijections.

Obfuscation Method.

General case.

- Ideal=Replacing the function by a look-up table.
- Input and output of the DES have 64 bits.
- Concatenation and composition of many small look-up tables.
- Each subfonction preceeded and followed by randomly chosen encoding bijections.

Obfuscation Method.

General case.

The composition of two obfuscated functions must be the obfuscation of the composition of the two functions.

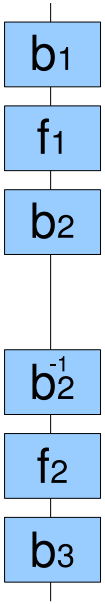

Figure: Composition of two functions.

DES Obfuscation.
S-boxes.

- The key is hidden in the S-boxes.
- Modification of the DES : 64 bits->96 bits.
- 12 T-boxes=bijections of 8 bits.
- Output of the T-boxes=output of the S-boxes and the copy of the input.
- Each T-box is preceded and followed by encoding bijections.

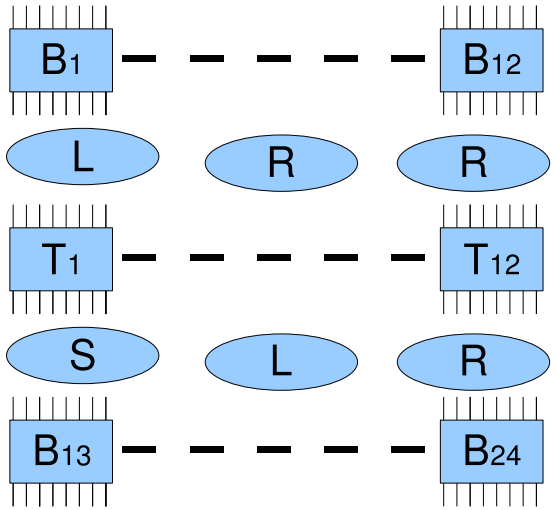

Figure: T-boxes.

DES Obfuscation.
S-boxes.

- The key is hidden in the S-boxes.
- Modification of the DES : 64 bits->96 bits.
- 12 T-boxes=bijections of 8 bits.
- Output of the T-boxes=output of the S-boxes and the copy of the input.
- Each T-box is preceded and followed by encoding bijections.

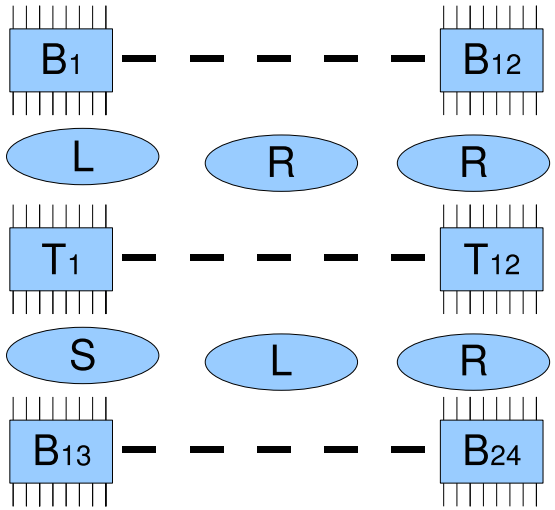

Figure: T-boxes.

DES Obfuscation.
S-boxes.

- The key is hidden in the S-boxes.
- Modification of the DES : 64 bits->96 bits.
- 12 T-boxes=bijections of 8 bits.
- Output of the T-boxes=output of the S-boxes and the copy of the input.
- Each T-box is preceded and followed by encoding bijections.

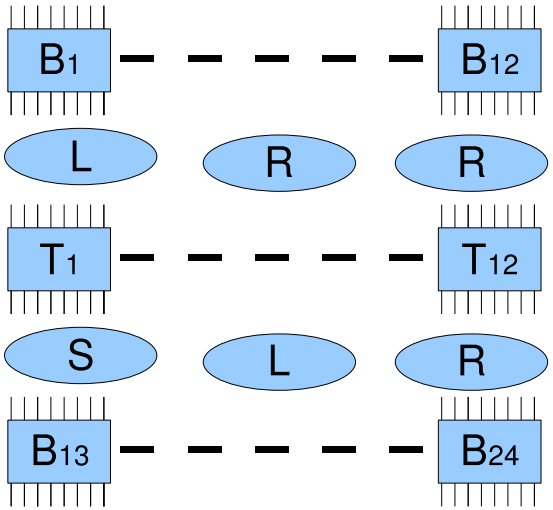

Figure: T-boxes.

DES Obfuscation.
S-boxes.

- The key is hidden in the S-boxes.
- Modification of the DES : 64 bits->96 bits.
- 12 T-boxes=bijections of 8 bits.
- Output of the T-boxes=output of the S-boxes and the copy of the input.
- Each T-box is preceded and followed by encoding bijections.

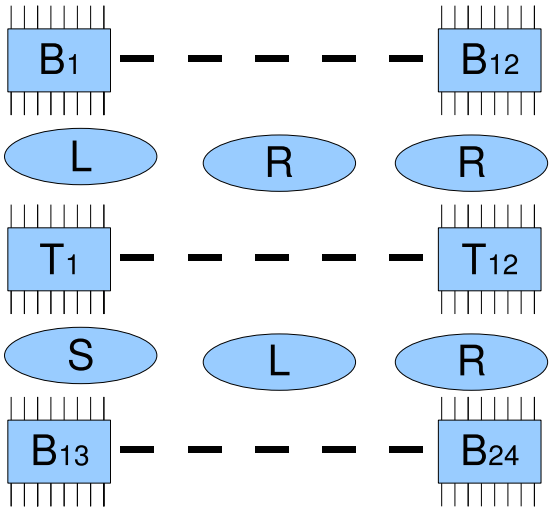

Figure: T-boxes.

DES Obfuscation.
S-boxes.

- The key is hidden in the S-boxes.
- Modification of the DES : 64 bits->96 bits.
- 12 T-boxes=bijections of 8 bits.
- Output of the T-boxes=output of the S-boxes and the copy of the input.
- Each T-box is preceded and followed by encoding bijections.

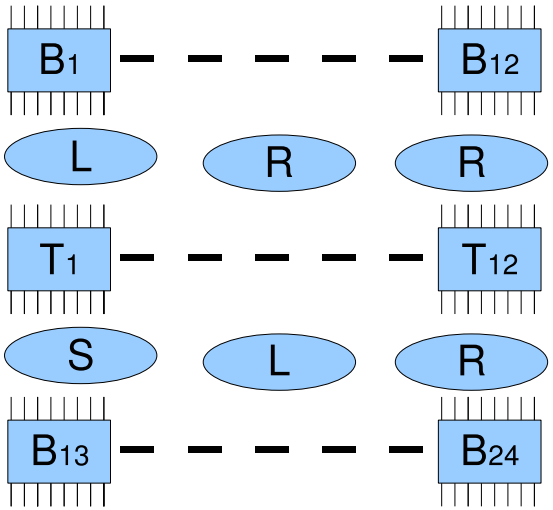

Figure: T-boxes.

DES Obfuscation.
Linear Parts.

- Input split into 24 parts.
- Each part enters into an encoding bijection.
- Outputs of bijections are the input of the linear function.
- Output of the function is split into 24 parts.
- Each part enters into an encoding bijection.

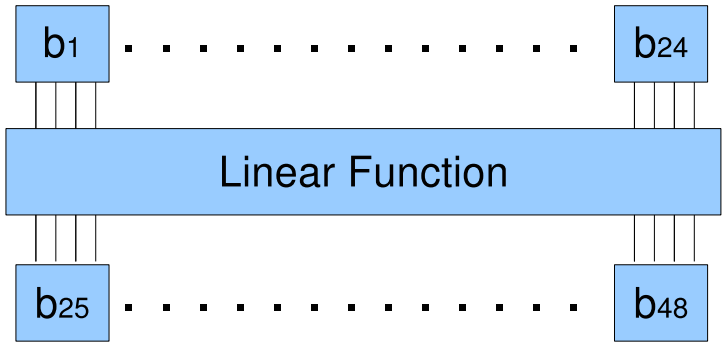

Figure: Linear Function.

DES Obfuscation.
Linear Parts.

- Input split into 24 parts.
- Each part enters into an encoding bijection.
- Outputs of bijections are the input of the linear function.
- Output of the function is split into 24 parts.
- Each part enters into an encoding bijection.

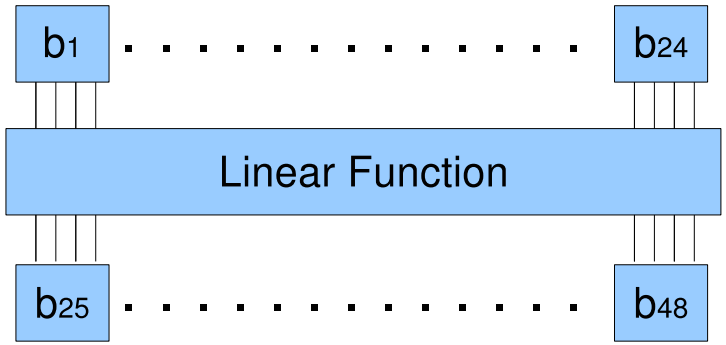

Figure: Linear Function.

DES Obfuscation.
Linear Parts.

- Input split into 24 parts.
- Each part enters into an encoding bijection.
- Outputs of bijections are the input of the linear function.
- Output of the function is split into 24 parts.
- Each part enters into an encoding bijection.

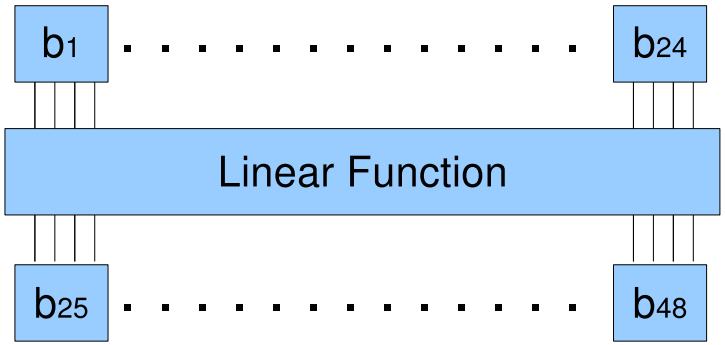

Figure: Linear Function.

DES Obfuscation.
Linear Parts.

- Input split into 24 parts.
- Each part enters into an encoding bijection.
- Outputs of bijections are the input of the linear function.
- Output of the function is split into 24 parts.
- Each part enters into an encoding bijection.

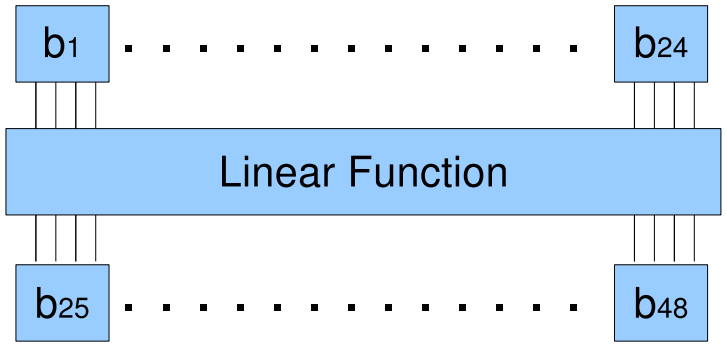

Figure: Linear Function.

DES Obfuscation.
Linear Parts.

- Input split into 24 parts.
- Each part enters into an encoding bijection.
- Outputs of bijections are the input of the linear function.
- Output of the function is split into 24 parts.
- Each part enters into an encoding bijection.

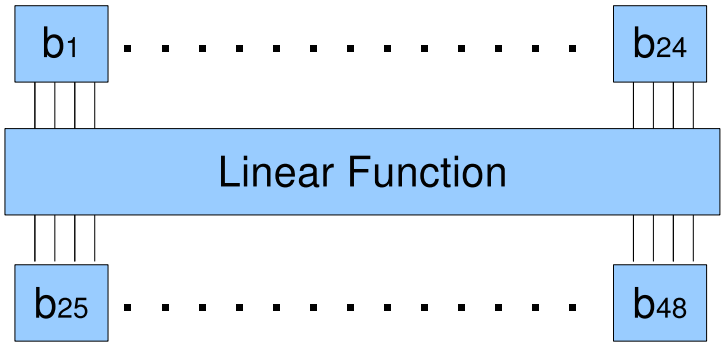

Figure: Linear Function.

Naked-DES.

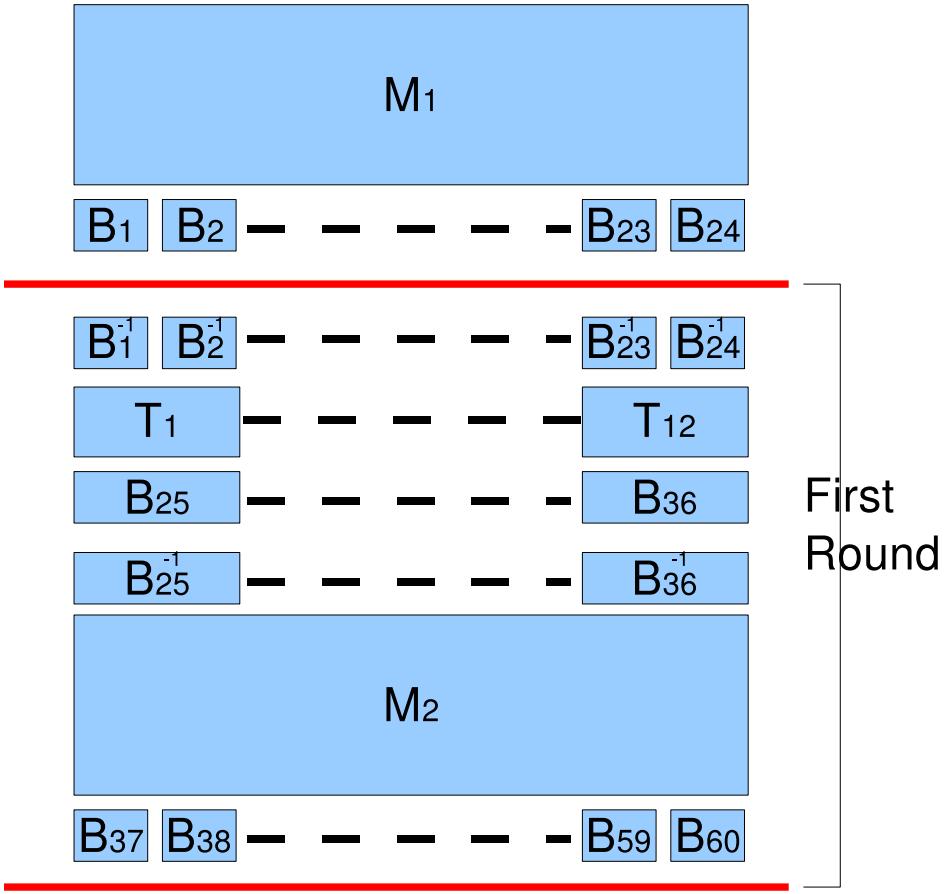

Figure: Naked-DES.

Nonstandard-DES

We add an extra protection : two linear functions M_0 and M_4.

Figure: Nonstandard-DES.

Outline

Attack on the Naked-DES

Differential Attack : If we know the key, we can choose 2 inputs with almost the same outputs.

- Just one S-box is touched -> only 6 bits of sub-key.
- We modify only one middle bit and four left bits.
- Using the key, we find two inputs such that only one bit changes at the end of the first round.

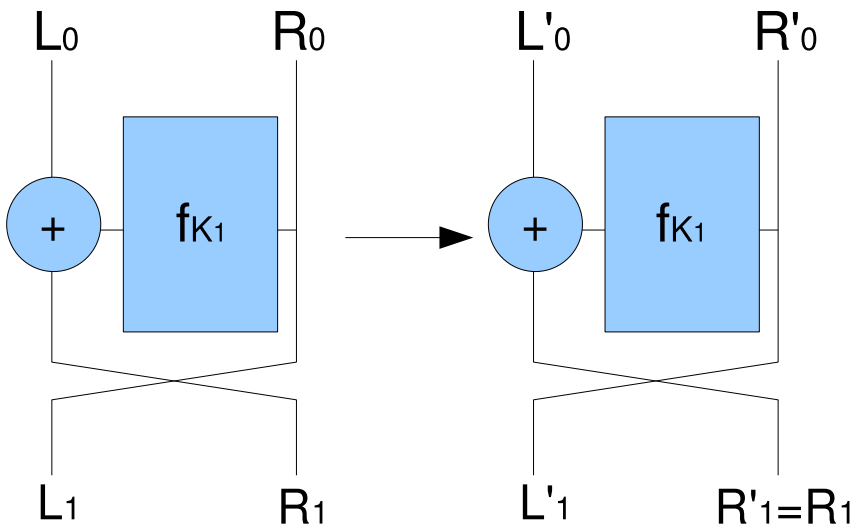

Figure: Principle.

Attack on the Naked-DES

Differential Attack : If we know the key, we can choose 2 inputs with almost the same outputs.

- Just one S-box is touched -> only 6 bits of sub-key.
- We modify only one middle bit and four left bits.
- Using the key, we find two inputs such that only one bit changes at the end of the first round.

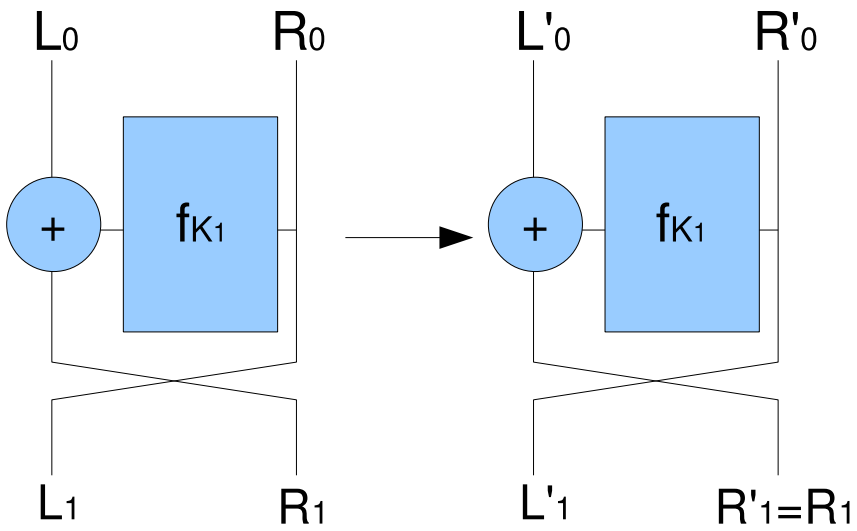

Figure: Principle.

Attack on the Naked-DES

Differential Attack : If we know the key, we can choose 2 inputs with almost the same outputs.

- Just one S-box is touched -> only 6 bits of sub-key.
- We modify only one middle bit and four left bits.
- Using the key, we find two inputs such that only one bit changes at the end of the first round.

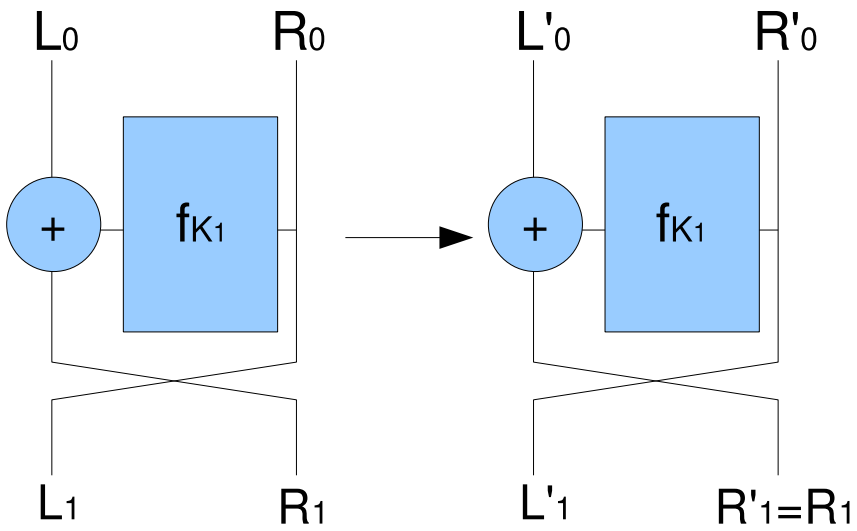

Figure: Principle.

Outline

General Principle of the Attack.

- Goal : retrieving the key, M_0 and M_4.
- Attack based on the one for the Naked-DES.
- Need to retrieve M_0 first.
- Columns of M_0^{-1} are preimages of canonical vectors.

General Principle of the Attack.

- Goal : retrieving the key, M_0 and M_4.
- Attack based on the one for the Naked-DES.
- Need to retrieve M_0 first.
- Columns of M_0^{-1} are preimages of canonical vectors.

General Principle of the Attack.

- Goal : retrieving the key, M_0 and M_4.
- Attack based on the one for the Naked-DES.
- Need to retrieve M_0 first.
- Columns of M_0^{-1} are preimages of canonical vectors.

General Principle of the Attack.

- Goal : retrieving the key, M_0 and M_4.
- Attack based on the one for the Naked-DES.
- Need to retrieve M_0 first.
- Columns of M_0^{-1} are preimages of canonical vectors.

Algorithm.

- Retrieve lists of candidates for preimages : lists are shorten step by step.
- Retrieve the key : attack on the Naked-DES.
- Retrieve M_4 : solve system of linear equations.

Algorithm.

- Retrieve lists of candidates for preimages : lists are shorten step by step.
- Retrieve the key : attack on the Naked-DES.
- Retrieve M_4 : solve system of linear equations.

Algorithm.

- Retrieve lists of candidates for preimages : lists are shorten step by step.
- Retrieve the key : attack on the Naked-DES.
- Retrieve M_4 : solve system of linear equations.

Shortening the lists of candidates.

Lists of candidates are shorten step by step.

- Find vectors that act on only one encoding bijection before the first round.
- Make the difference between left and right bits.
- Find middle bits by reseting some inputs.
- Reduce the lists of the left bits.
- Find the correspondance between T-boxes and S-boxes.
- Label left bits.

Shortening the lists of candidates.

Find vectors that act on only one encoding bijection before the first round.

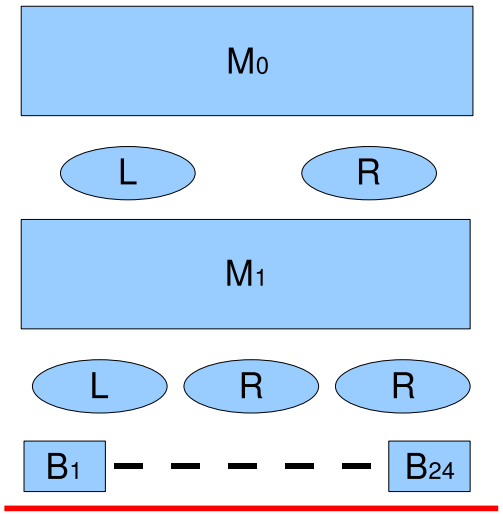

Figure: Before the first round.

Shortening the lists of candidates.

Lists of candidates are shorten step by step.

- Find vectors that act on only one encoding bijection before the first round.
- Make the difference between left and right bits.
- Find middle bits by reseting some inputs.
- Reduce the lists of the left bits.
- Find the correspondance between T-boxes and S-boxes.
- Label left bits.

Shortening the lists of candidates.

Make the difference between left and right bits.

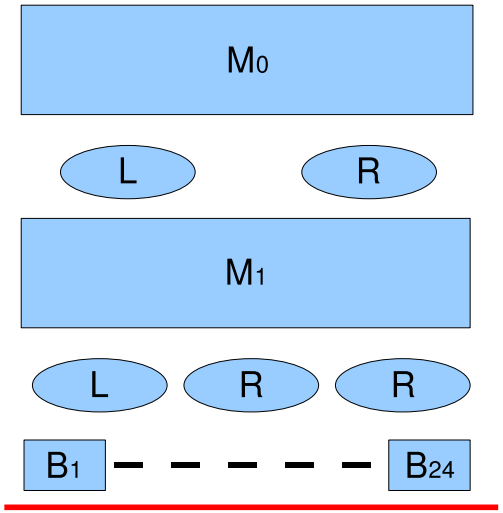

Figure: Before the first round.

Shortening the lists of candidates.

Lists of candidates are shorten step by step.

- Find vectors that act on only one encoding bijection before the first round.
- Make the difference between left and right bits.
- Find middle bits by reseting some inputs.
- Reduce the lists of the left bits.
- Find the correspondance between T-boxes and S-boxes.
- Label left bits.

Shortening the lists of candidates.

Find middle bits by reseting some inputs.

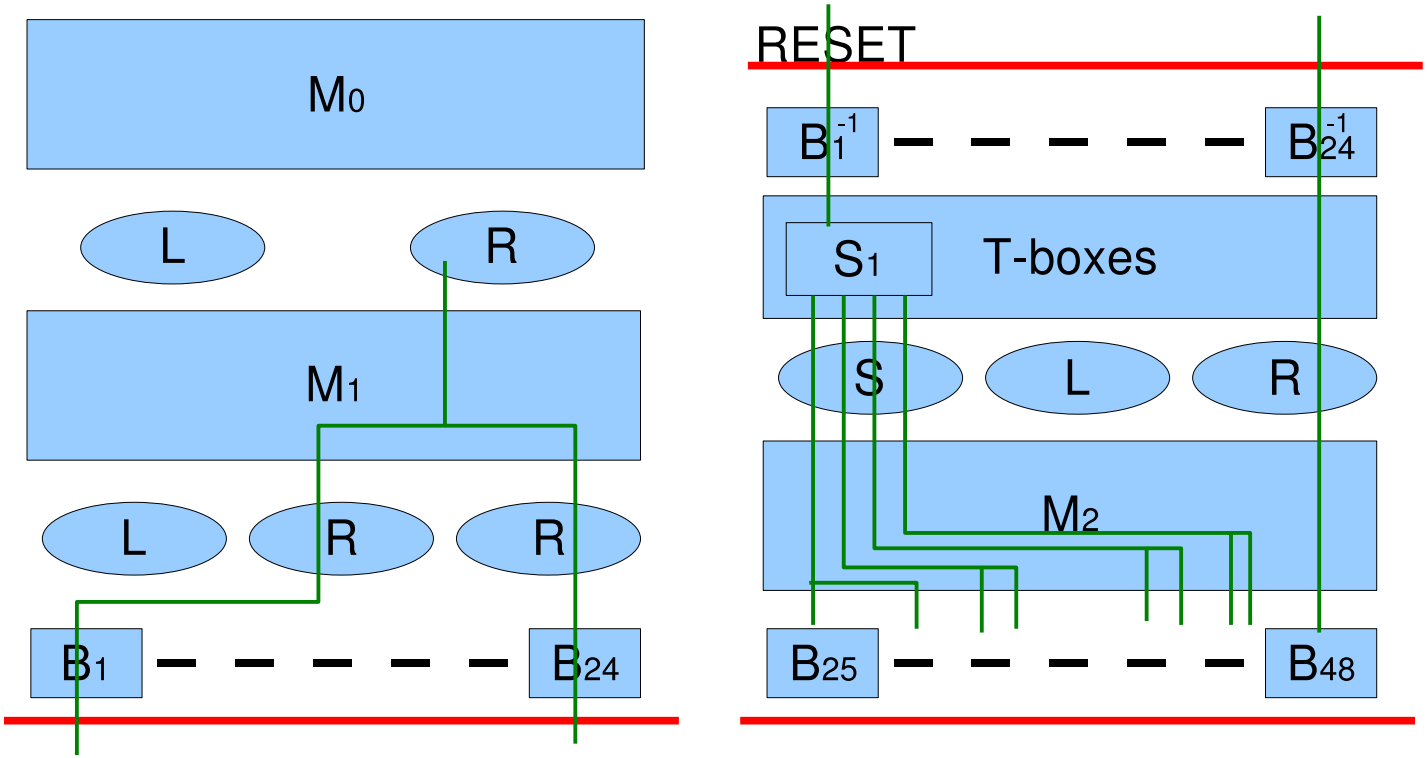

Figure: Middle bits.

Shortening the lists of candidates.

Lists of candidates are shorten step by step.

- Find vectors that act on only one encoding bijection before the first round.
- Make the difference between left and right bits.
- Find middle bits by reseting some inputs.
- Reduce the lists of the left bits.
- Find the correspondance between T-boxes and S-boxes.
- Label left bits.

Shortening the lists of candidates.

Reduce the lists of the left bits.

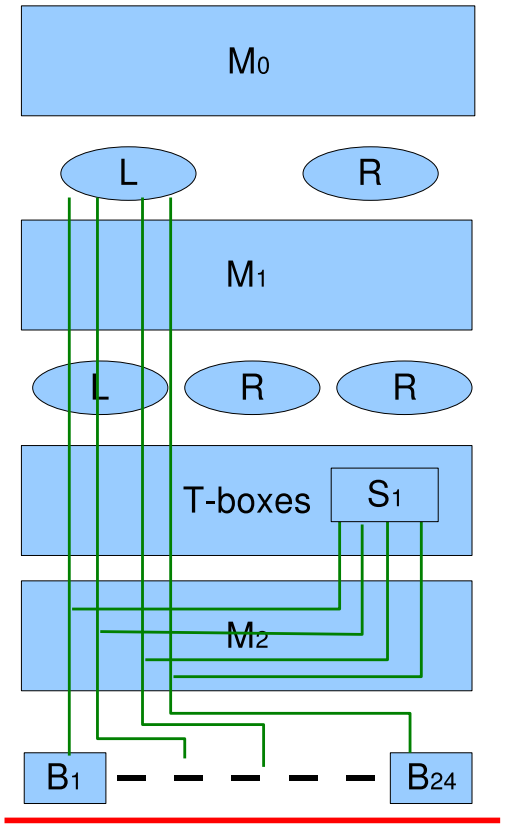

Figure: Left Bits.

Shortening the lists of candidates.

Lists of candidates are shorten step by step.

- Find vectors that act on only one encoding bijection before the first round.
- Make the difference between left and right bits.
- Find middle bits by reseting some inputs.
- Reduce the lists of the left bits.
- Find the correspondance between T-boxes and S-boxes.
- Label left bits.

Shortening the lists of candidates.

Find the correspondance between T-boxes and S-boxes.

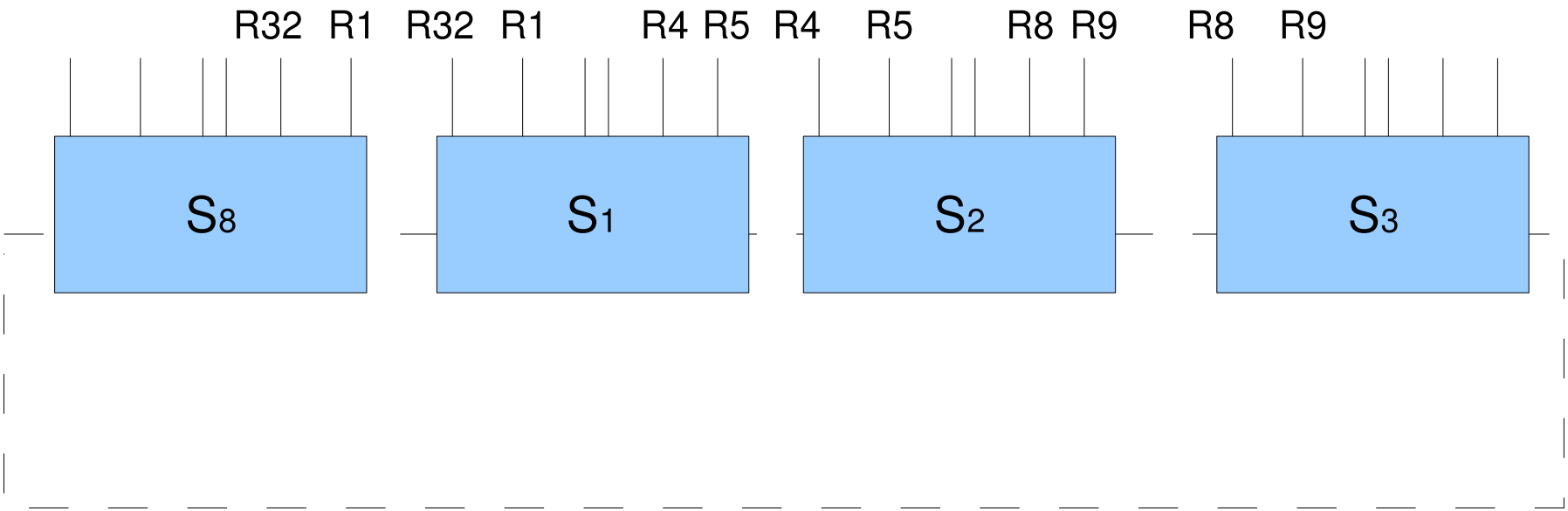

Figure: Chaining.

Shortening the lists of candidates.

Lists of candidates are shorten step by step.

- Find vectors that act on only one encoding bijection before the first round.
- Make the difference between left and right bits.
- Find middle bits by reseting some inputs.
- Reduce the lists of the left bits.
- Find the correspondance between T-boxes and S-boxes.
- Label left bits.

Shortening the lists of candidates.

Label left bits.

Out of the four Left bits that are xored with the output of a specified S-Box, exactly two become (in the second round) middle bits.

Summary

- We have found an attack on the nonstandard-DES.
- This attack has been implemented with a C code. Within seconds, we recover the key, and the two matrices M_0 and M_4.

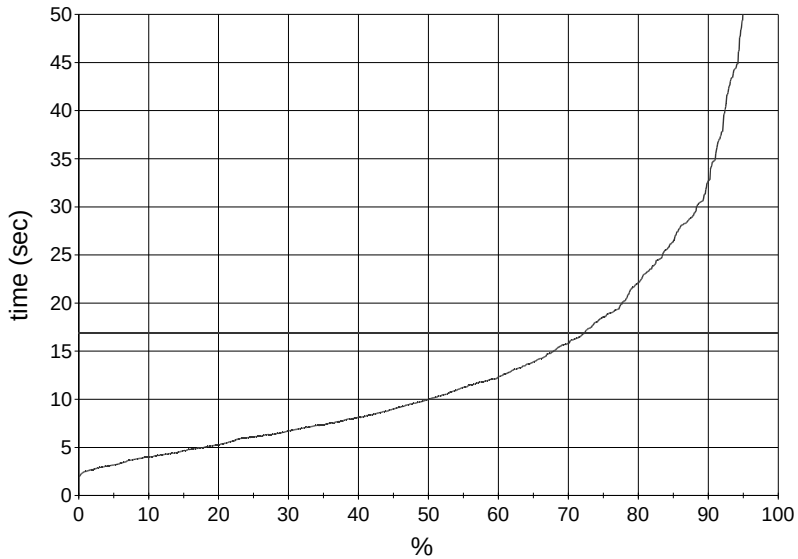

Figure: Results.

Ongoing Work: Consider the case where M_0 and M_4 are fully encoded (both side).