One Construction of a Backdoored AES-like Block Cipher and How to Break it

Arnaud Bannier & **Eric Filiol**
filiol@esiea.fr

ESIEA
Operational Cryptology and Virology Lab $(C+V)^O$

конференция РусКрипто

esiea ECOLE D'INGENIEURS DU MONDE NUMERIQUE

Agenda

Summary of the talk

Introduction

- Encryption systems have always been under export controls (ITAR, Wassenaar...). Considered as weapons and dual-use means.

Introduction

- Encryption systems have always been under export controls (ITAR, Wassenaar...). Considered as weapons and dual-use means.
- Implementation backdoors
  - Key escrowing, key management and key distribution protocols weaknesses (refer to recent CIA leak)
  - Hackers are likely to find and use them as well

Introduction

- Encryption systems have always been under export controls (ITAR, Wassenaar...). Considered as weapons and dual-use means.
- Implementation backdoors
  - Key escrowing, key management and key distribution protocols weaknesses (refer to recent CIA leak)
  - Hackers are likely to find and use them as well
- Mathematical backdoors
  - Put a secret flaw at the design level while the algorithm remains public
  - Finding the backdoor must be an untractable problem while exploiting it must be "easy"
  - Historic cases: Crypto AG and Buehler's case (1995)
  - Extremely few open and public research in this area
  - Known existence of NSA and GCHQ research programs

Introduction

- Encryption systems have always been under export controls (ITAR, Wassenaar...). Considered as weapons and dual-use means.
- Implementation backdoors
  - Key escrowing, key management and key distribution protocols weaknesses (refer to recent CIA leak)
  - Hackers are likely to find and use them as well
- Mathematical backdoors
  - Put a secret flaw at the design level while the algorithm remains public
  - Finding the backdoor must be an untractable problem while exploiting it must be "easy"
  - Historic cases: Crypto AG and Buehler's case (1995)
  - Extremely few open and public research in this area
  - Known existence of NSA and GCHQ research programs
- Sovereignty issue: can we trust foreign encryption algorithms?

Aim of our Research

- Try to answer to the key question
  - *"How easy and feasible is it to design and to insert backdoors (at the mathematical level) in encryption algorithms?"*

Aim of our Research

- Try to answer to the key question
  - *"How easy and feasible is it to design and to insert backdoors (at the mathematical level) in encryption algorithms?"*
- Explore the different possible approaches
  - The present work is a first step
  - We consider a particular case of backdoors here (linear partition of the data spaces)

Aim of our Research

- Try to answer to the key question
  - *"How easy and feasible is it to design and to insert backdoors (at the mathematical level) in encryption algorithms?"*
- Explore the different possible approaches
  - The present work is a first step
  - We consider a particular case of backdoors here (linear partition of the data spaces)
- For more details on backdoors and the few existing works, please refer to our ForSE 2017 paper
  - Available on `https://arxiv.org/abs/1702.06475`

Summary of the talk

Partition-based Trapdoors

- Based on our theoretical work (Bannier, Bodin & Filiol, 2016; Bannier & Filiol, 2017)
  - Generalization of Paterson's work (1999)

Partition-based Trapdoors

- Based on our theoretical work (Bannier, Bodin & Filiol, 2016; Bannier & Filiol, 2017)
  - Generalization of Paterson's work (1999)
- BEA-1 is inspired from the *Advanced Encryption Standard* (AES)
  - BEA-1 is a Substitution-Permutation Network (SPN)
  - BEA-1 stands for *Backdoored Encryption Algorithm* version 1

Linear Partitions

Definition (Linear Partition)

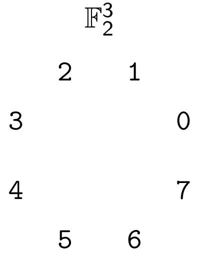
A partition of $\mathbb{F}_2^n$ made up of all the cosets of a linear subspace is said to be *linear*.

Linear Partitions

Definition (Linear Partition)

A partition of $\mathbb{F}_2^n$ made up of all the cosets of a linear subspace is said to be *linear*.

Example of a linear partition over $\mathbb{F}_2^3$:



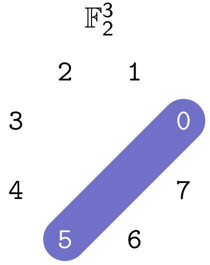
Linear Partitions

Definition (Linear Partition)

A partition of $\mathbb{F}_2^n$ made up of all the cosets of a linear subspace is said to be *linear*.

Example of a linear partition over $\mathbb{F}_2^3$:

- $$V = \{000, 101\} = \{0, 5\},$$



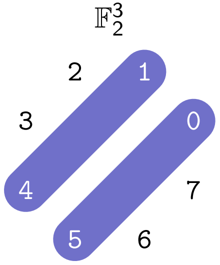
Linear Partitions

Definition (Linear Partition)

A partition of $\mathbb{F}_2^n$ made up of all the cosets of a linear subspace is said to be *linear*.

Example of a linear partition over $\mathbb{F}_2^3$:

- $V = \{000, 101\} = \{0, 5\}$,
- $001 + V = \{001, 100\} = \{1, 4\}$,



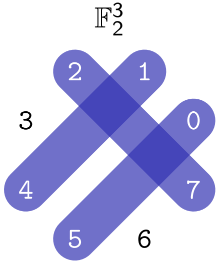
Linear Partitions

Definition (Linear Partition)

A partition of $\mathbb{F}_2^n$ made up of all the cosets of a linear subspace is said to be *linear*.

Example of a linear partition over $\mathbb{F}_2^3$:

- $V = \{000, 101\} = \{0, 5\},$
- $001 + V = \{001, 100\} = \{1, 4\},$
- $010 + V = \{010, 111\} = \{2, 7\},$



Linear Partitions

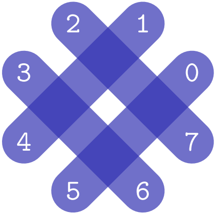
Definition (Linear Partition)

A partition of $\mathbb{F}_2^n$ made up of all the cosets of a linear subspace is said to be *linear*.

Example of a linear partition over $\mathbb{F}_2^3$:

- $V = \{000, 101\} = \{0, 5\}$,
- $001 + V = \{001, 100\} = \{1, 4\}$,
- $010 + V = \{010, 111\} = \{2, 7\}$,
- $011 + V = \{011, 110\} = \{3, 6\}$,

$\mathbb{F}_2^3$



Linear Partitions

Definition (Linear Partition)

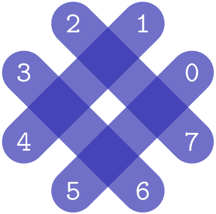
A partition of $\mathbb{F}_2^n$ made up of all the cosets of a linear subspace is said to be *linear*.

Example of a linear partition over $\mathbb{F}_2^3$:

- $V = \{000, 101\} = \{0, 5\}$,
- $001 + V = \{001, 100\} = \{1, 4\}$,
- $010 + V = \{010, 111\} = \{2, 7\}$,
- $011 + V = \{011, 110\} = \{3, 6\}$,

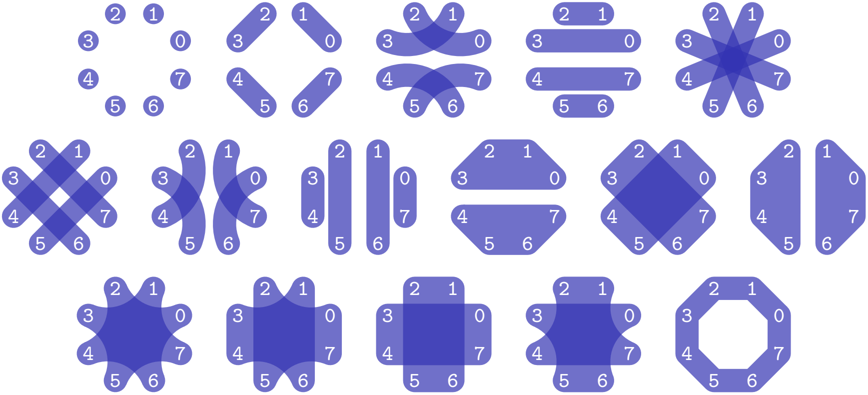
$\mathcal{L}(V) = \{\{0, 5\}, \{1, 4\}, \{2, 7\}, \{3, 6\}\}$.

$\mathbb{F}_2^3$



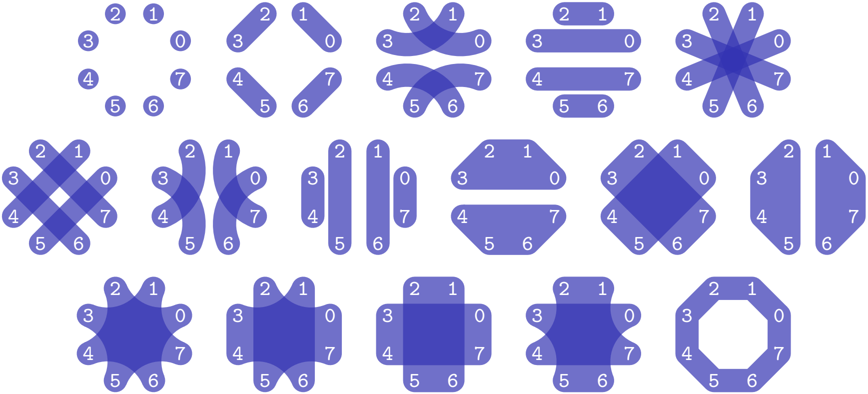
Linear Partitions

The 16 linear partition over $\mathbb{F}_2^3$:



Linear Partitions

The 16 linear partition over $\mathbb{F}_2^3$:



There are 229 755 605 linear partitions over $\mathbb{F}_2^{10}$.

Partition-Based Backdoor SPN

Assumption

The SPN maps $\mathcal{A}$ to $\mathcal{B}$, no matter what the round keys are.

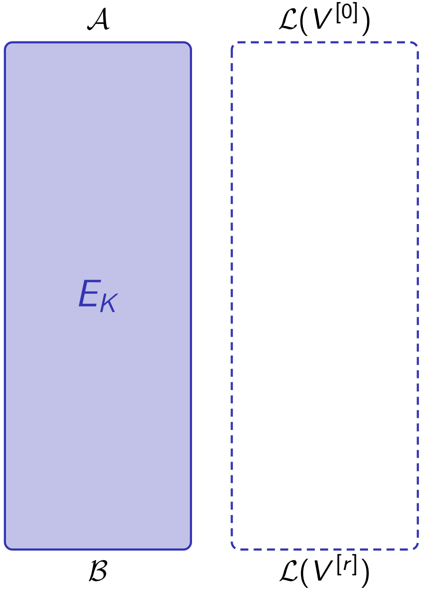
Partition-Based Backdoor SPN

Assumption

The SPN maps $\mathcal{A}$ to $\mathcal{B}$, no matter what the round keys are.

Theoretical results :

- $\mathcal{A}$ and $\mathcal{B}$ are linear,



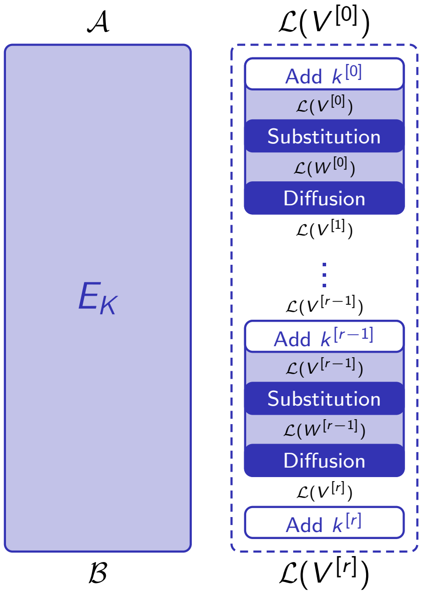
Partition-Based Backdoor SPN

Assumption

The SPN maps $\mathcal{A}$ to $\mathcal{B}$, no matter what the round keys are.

Theoretical results :

- $\mathcal{A}$ and $\mathcal{B}$ are linear,
- $\mathcal{A}$ is transformed through each step of the SPN in a deterministic way,



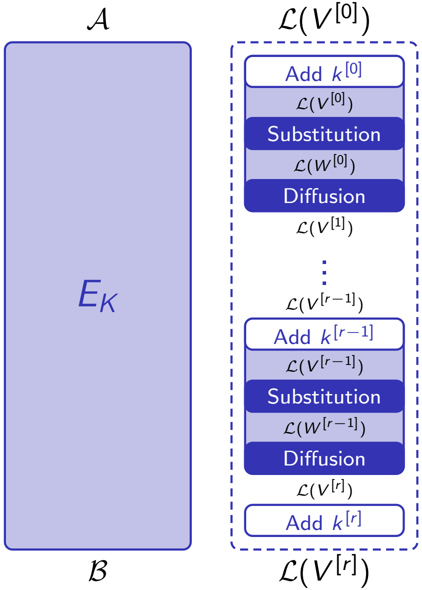
Partition-Based Backdoor SPN

Assumption

The SPN maps $\mathcal{A}$ to $\mathcal{B}$, no matter what the round keys are.

Theoretical results :

- $\mathcal{A}$ and $\mathcal{B}$ are linear,
- $\mathcal{A}$ is transformed through each step of the SPN in a deterministic way,
- At least one S-box maps a linear partition to another one.



BEA-1 Key Features

- Parameters
  - BEA-1 operates on 80-bit data blocks
  - 120-bit master key and twelve 80-bit round keys
  - 11 rounds (the last round involves two round keys)

BEA-1 Key Features

- Parameters
  - BEA-1 operates on 80-bit data blocks
  - 120-bit master key and twelve 80-bit round keys
  - 11 rounds (the last round involves two round keys)
- Primitives & base functions
  - Key schedule & key addition (bitwise XOR)
  - Substitution layer (involves four S-Boxes over $\mathbb{F}_2^{10}$)
  - Diffusion layer (`ShiftRows` and `MixColumns` operations)
  - Linear map $M : (\mathbb{F}_2^{10})^4 \rightarrow (\mathbb{F}_2^{10})^4$

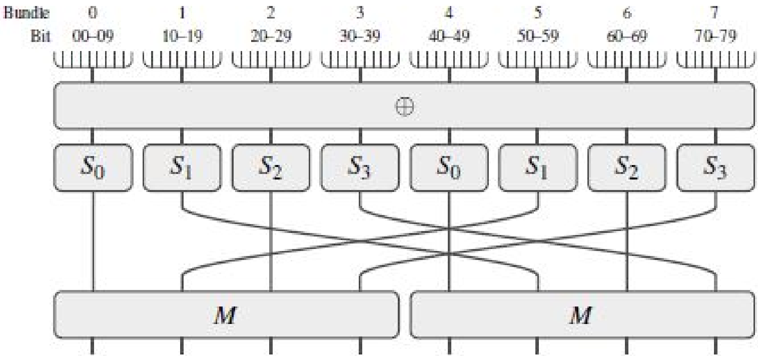
BEA-1 Key Features

- Parameters
  - BEA-1 operates on 80-bit data blocks
  - 120-bit master key and twelve 80-bit round keys
  - 11 rounds (the last round involves two round keys)
- Primitives & base functions
  - Key schedule & key addition (bitwise XOR)
  - Substitution layer (involves four S-Boxes over $\mathbb{F}_2^{10}$)
  - Diffusion layer (`ShiftRows` and `MixColumns` operations)
  - Linear map $M : (\mathbb{F}_2^{10})^4 \rightarrow (\mathbb{F}_2^{10})^4$
- S-Boxes, linear map M and pseudo-codes for the different functions are given in the ForSE 2017 paper

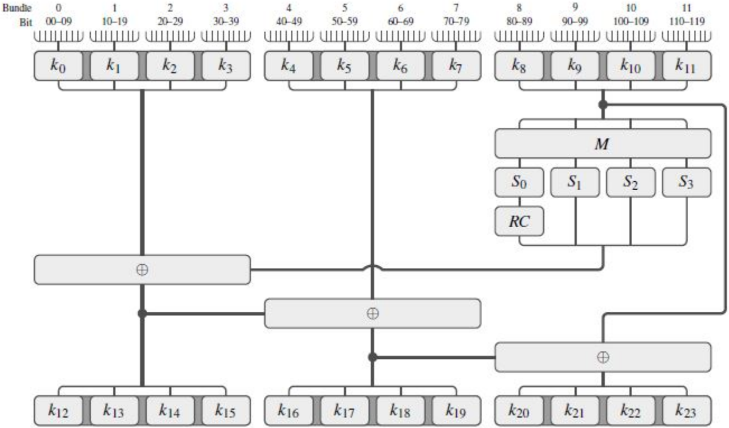
BEA-1 Key Features

- Parameters
  - BEA-1 operates on 80-bit data blocks
  - 120-bit master key and twelve 80-bit round keys
  - 11 rounds (the last round involves two round keys)
- Primitives & base functions
  - Key schedule & key addition (bitwise XOR)
  - Substitution layer (involves four S-Boxes over $\mathbb{F}_2^{10}$)
  - Diffusion layer (`ShiftRows` and `MixColumns` operations)
  - Linear map $M : (\mathbb{F}_2^{10})^4 \rightarrow (\mathbb{F}_2^{10})^4$
- S-Boxes, linear map M and pseudo-codes for the different functions are given in the ForSE 2017 paper
- BEA-1 is statically compliant with FIPS 140 (US NIST standard) and resists to linear/differential attacks.

BEA-1 Round Function

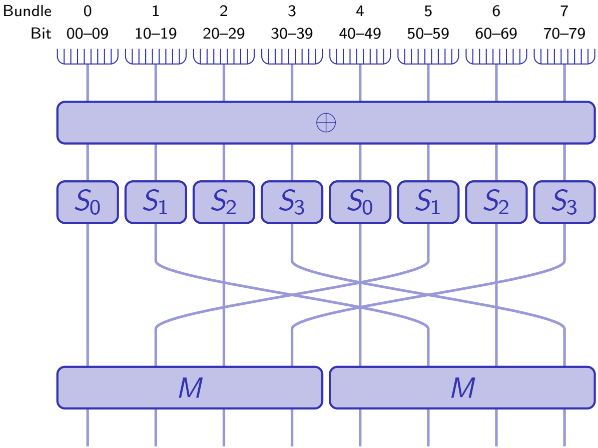


BEA-1 Key Schedule

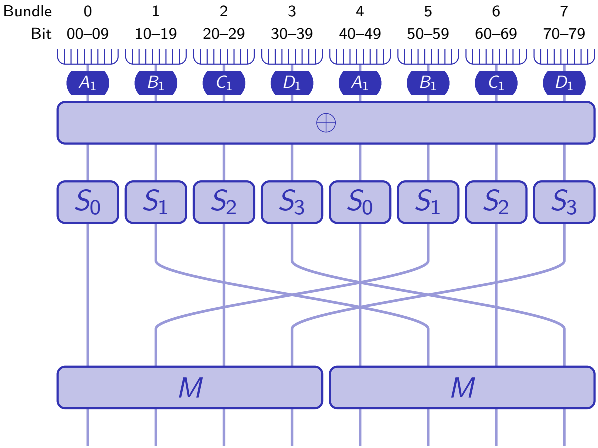


Summary of the talk

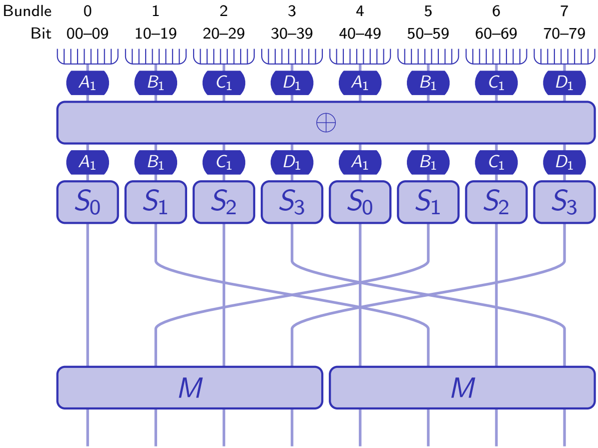
Linear Partitions and the Round Function



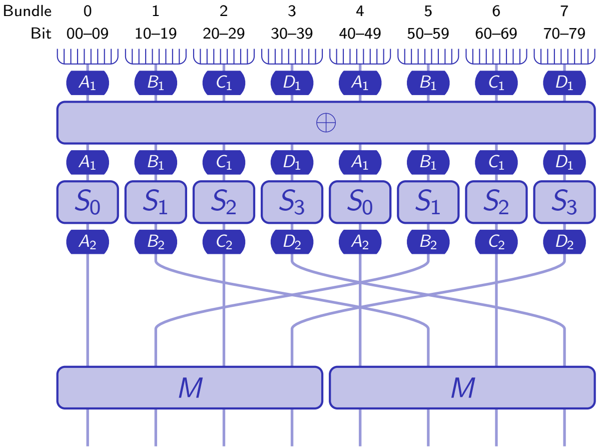
Linear Partitions and the Round Function



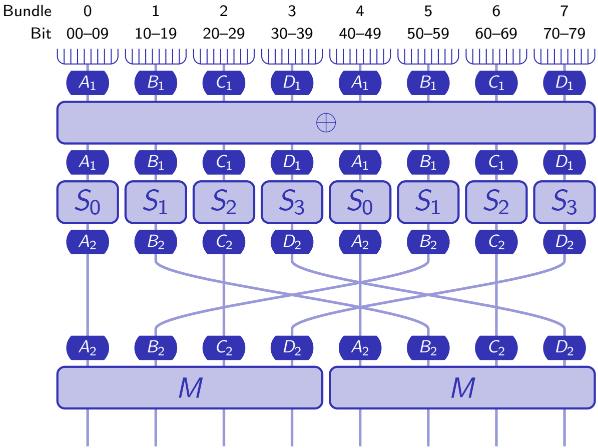
Linear Partitions and the Round Function



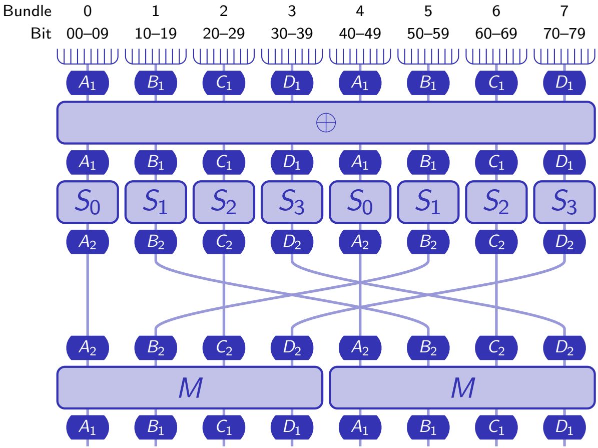
Linear Partitions and the Round Function



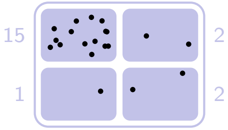
Linear Partitions and the Round Function



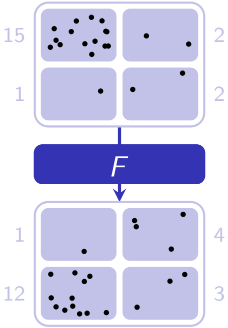
Linear Partitions and the Round Function



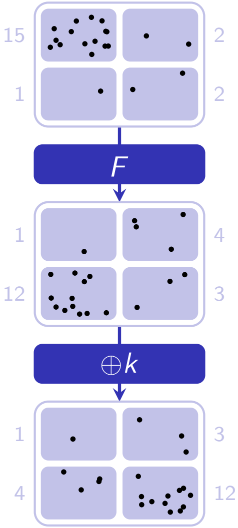
Principle of the Cryptanalysis



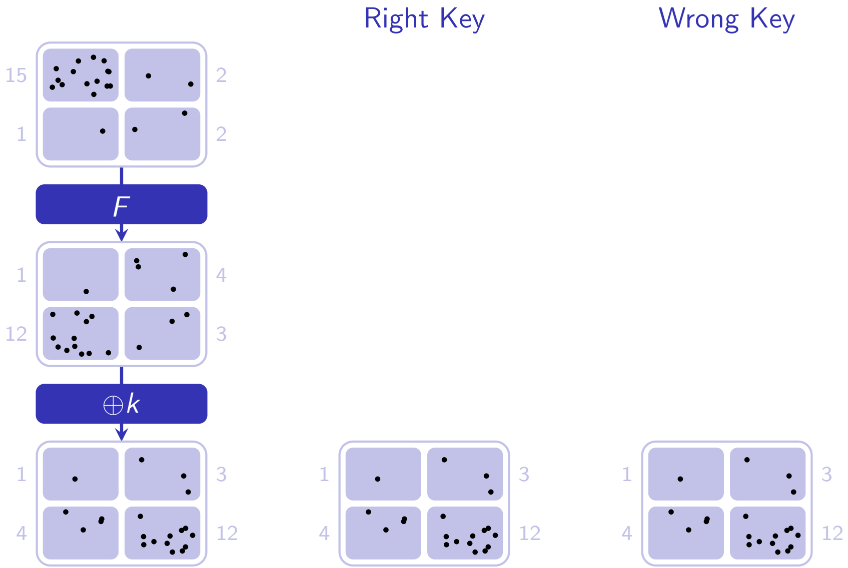
Principle of the Cryptanalysis



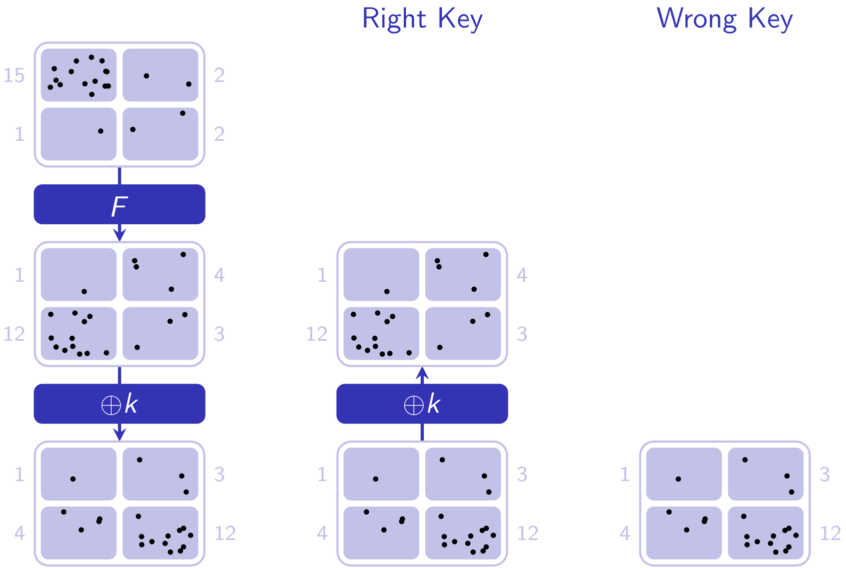
Principle of the Cryptanalysis



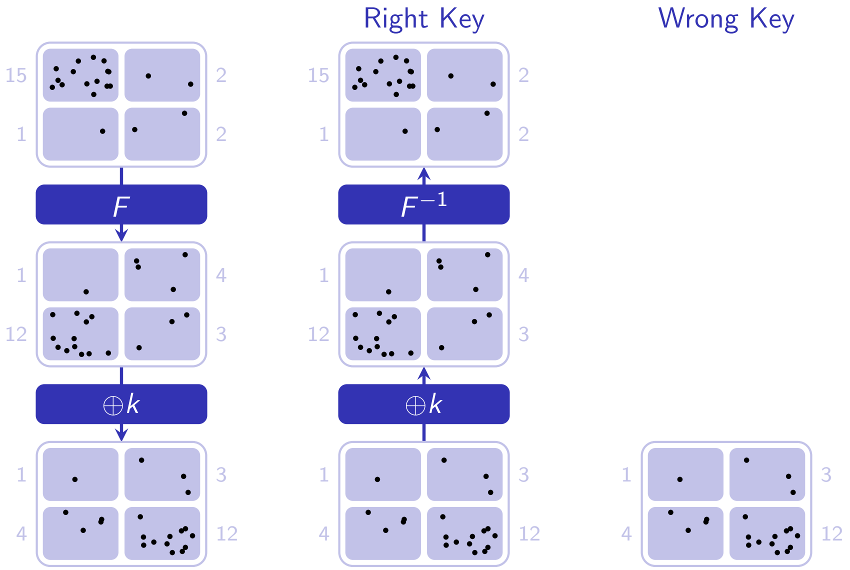
Principle of the Cryptanalysis



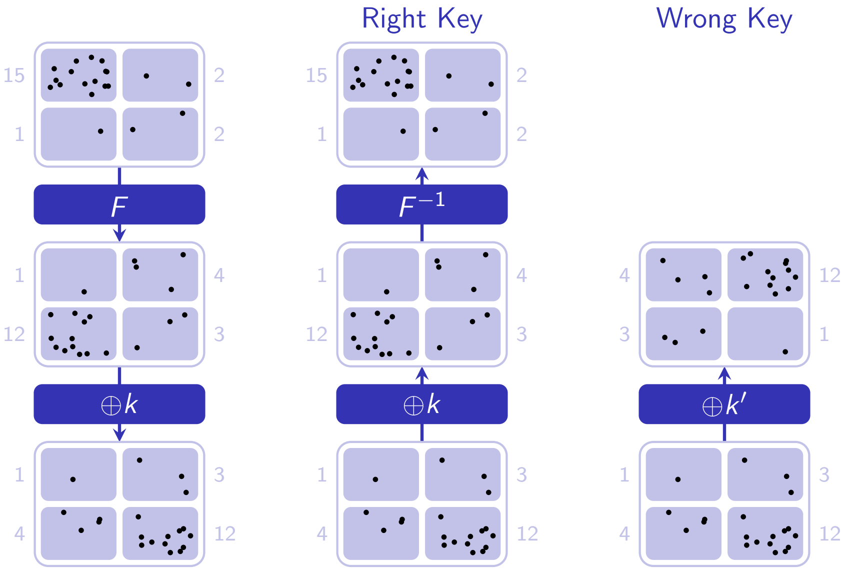
Principle of the Cryptanalysis



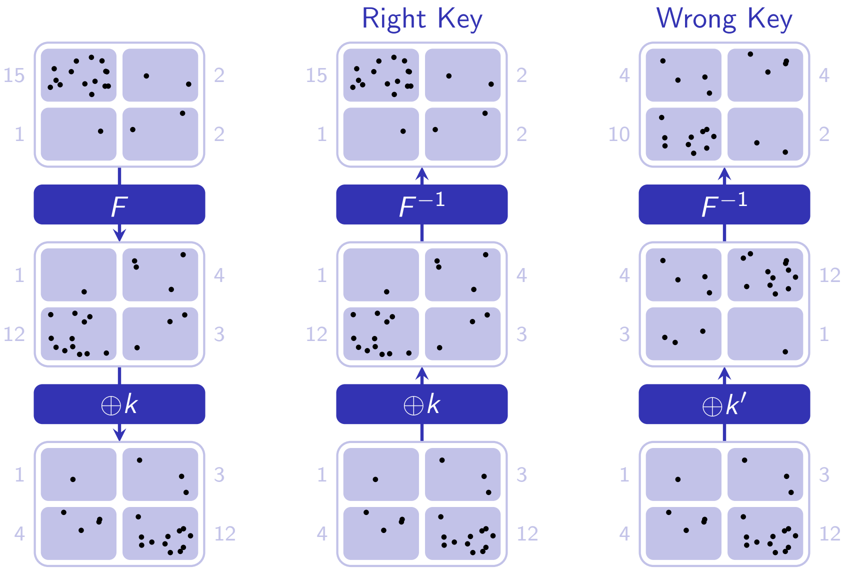
Principle of the Cryptanalysis



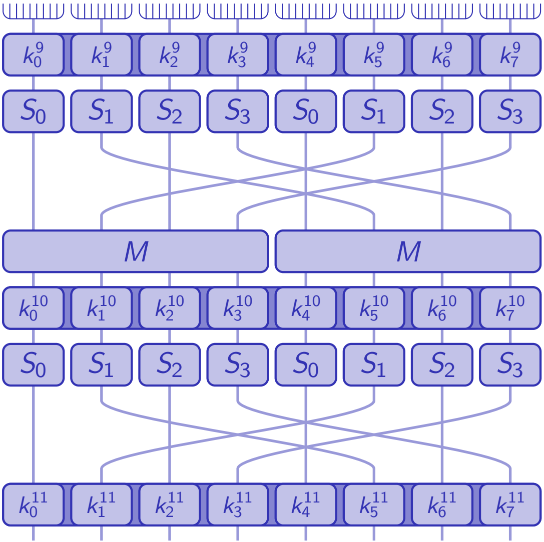
Principle of the Cryptanalysis



Principle of the Cryptanalysis

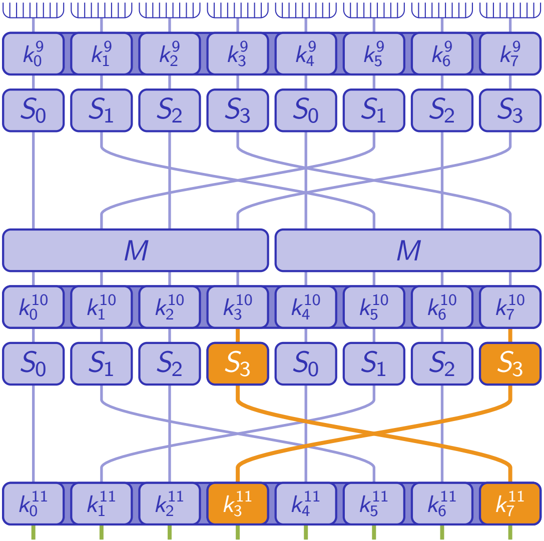


Overview of the Cryptanalysis



Find the output coset of $(A_2 \times B_2 \times C_2 \times D_2)^2$. There are 2^{40} possibilities.

Overview of the Cryptanalysis



Brute force:

$(k_0^{11}, k_1^{11}, k_2^{11}, k_3^{11}, k_4^{11}, k_5^{11}, k_6^{11}, k_7^{11})$

Test the 2^{15} saved keys:

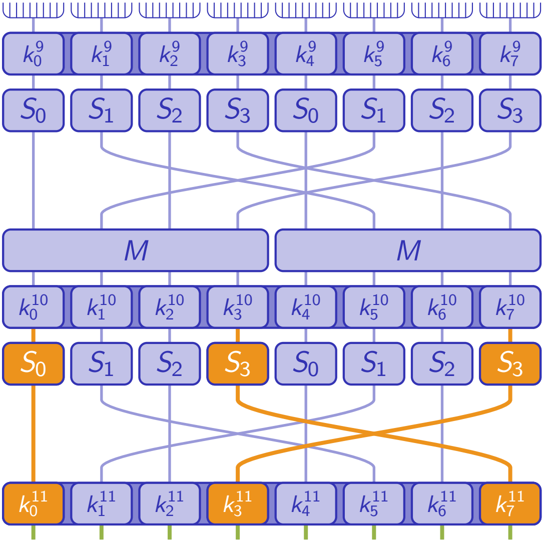
$(k_0^{11}, k_1^{11}, k_2^{11}, k_3^{11}, k_4^{11}, k_5^{11}, k_6^{11}, k_7^{11})$

Save the 2^{15} best keys:

$(k_0^{11}, k_1^{11}, k_2^{11}, k_3^{11}, k_4^{11}, k_5^{11}, k_6^{11}, k_7^{11})$

At the end of this step, we keep 2^{15} 80-bit candidates from the 2^{80} possible.

Overview of the Cryptanalysis



Brute force:

$(k_0^{11}, k_1^{11}, k_2^{11}, k_3^{11}, k_4^{11}, k_5^{11}, k_6^{11}, k_7^{11})$

Test the 2^{15} saved keys:

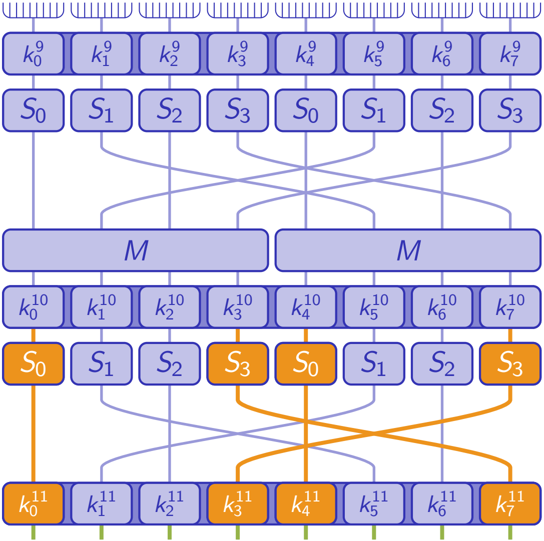
$(k_0^{11}, k_1^{11}, k_2^{11}, k_3^{11}, k_4^{11}, k_5^{11}, k_6^{11}, k_7^{11})$

Save the 2^{15} best keys:

$(k_0^{11}, k_1^{11}, k_2^{11}, k_3^{11}, k_4^{11}, k_5^{11}, k_6^{11}, k_7^{11})$

At the end of this step, we keep 2^{15} 80-bit candidates from the 2^{80} possible.

Overview of the Cryptanalysis



Brute force:

$(k_0^{11}, k_1^{11}, k_2^{11}, k_3^{11}, k_4^{11}, k_5^{11}, k_6^{11}, k_7^{11})$

Test the 2^{15} saved keys:

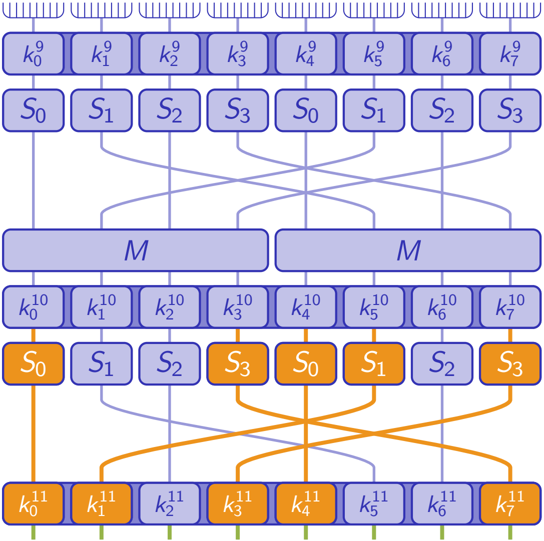
$(k_0^{11}, k_1^{11}, k_2^{11}, k_3^{11}, k_4^{11}, k_5^{11}, k_6^{11}, k_7^{11})$

Save the 2^{15} best keys:

$(k_0^{11}, k_1^{11}, k_2^{11}, k_3^{11}, k_4^{11}, k_5^{11}, k_6^{11}, k_7^{11})$

At the end of this step, we keep 2^{15} 80-bit candidates from the 2^{80} possible.

Overview of the Cryptanalysis



Brute force:

$(k_0^{11}, k_1^{11}, k_2^{11}, k_3^{11}, k_4^{11}, k_5^{11}, k_6^{11}, k_7^{11})$

Test the 2^{15} saved keys:

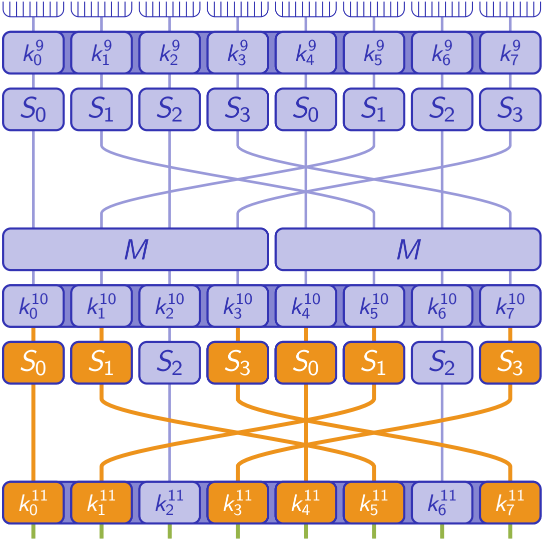
$(k_0^{11}, k_1^{11}, k_2^{11}, k_3^{11}, k_4^{11}, k_5^{11}, k_6^{11}, k_7^{11})$

Save the 2^{15} best keys:

$(k_0^{11}, k_1^{11}, k_2^{11}, k_3^{11}, k_4^{11}, k_5^{11}, k_6^{11}, k_7^{11})$

At the end of this step, we keep 2^{15} 80-bit candidates from the 2^{80} possible.

Overview of the Cryptanalysis



Brute force:

$(k_0^{11}, k_1^{11}, k_2^{11}, k_3^{11}, k_4^{11}, k_5^{11}, k_6^{11}, k_7^{11})$

Test the 2^{15} saved keys:

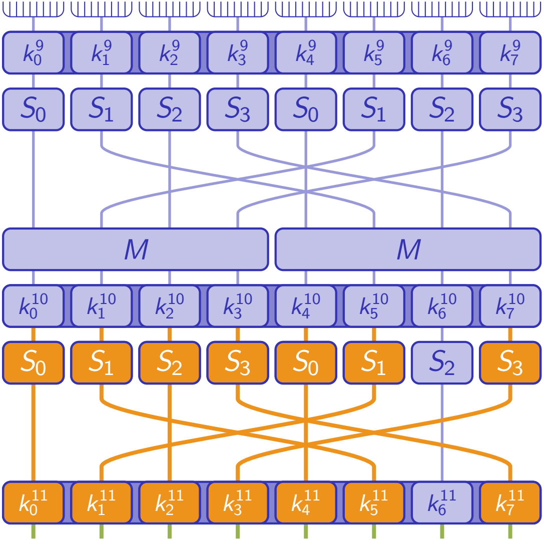
$(k_0^{11}, k_1^{11}, k_2^{11}, k_3^{11}, k_4^{11}, k_5^{11}, k_6^{11}, k_7^{11})$

Save the 2^{15} best keys:

$(k_0^{11}, k_1^{11}, k_2^{11}, k_3^{11}, k_4^{11}, k_5^{11}, k_6^{11}, k_7^{11})$

At the end of this step, we keep 2^{15} 80-bit candidates from the 2^{80} possible.

Overview of the Cryptanalysis



Brute force:

$(k_0^{11}, k_1^{11}, k_2^{11}, k_3^{11}, k_4^{11}, k_5^{11}, k_6^{11}, k_7^{11})$

Test the 2^{15} saved keys:

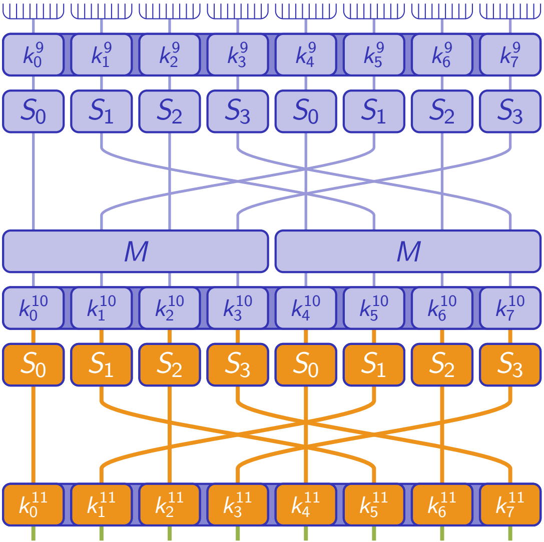
$(k_0^{11}, k_1^{11}, k_2^{11}, k_3^{11}, k_4^{11}, k_5^{11}, k_6^{11}, k_7^{11})$

Save the 2^{15} best keys:

$(k_0^{11}, k_1^{11}, k_2^{11}, k_3^{11}, k_4^{11}, k_5^{11}, k_6^{11}, k_7^{11})$

At the end of this step, we keep 2^{15} 80-bit candidates from the 2^{80} possible.

Overview of the Cryptanalysis



Brute force:

$(k_0^{11}, k_1^{11}, k_2^{11}, k_3^{11}, k_4^{11}, k_5^{11}, k_6^{11}, k_7^{11})$

Test the 2^{15} saved keys:

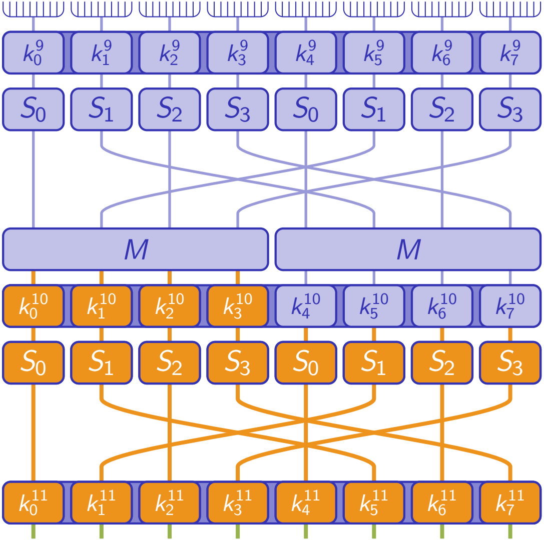
$(k_0^{11}, k_1^{11}, k_2^{11}, k_3^{11}, k_4^{11}, k_5^{11}, k_6^{11}, k_7^{11})$

Save the 2^{15} best keys:

$(k_0^{11}, k_1^{11}, k_2^{11}, k_3^{11}, k_4^{11}, k_5^{11}, k_6^{11}, k_7^{11})$

At the end of this step, we keep 2^{15} 80-bit candidates from the 2^{80} possible.

Overview of the Cryptanalysis



According to the key schedule:

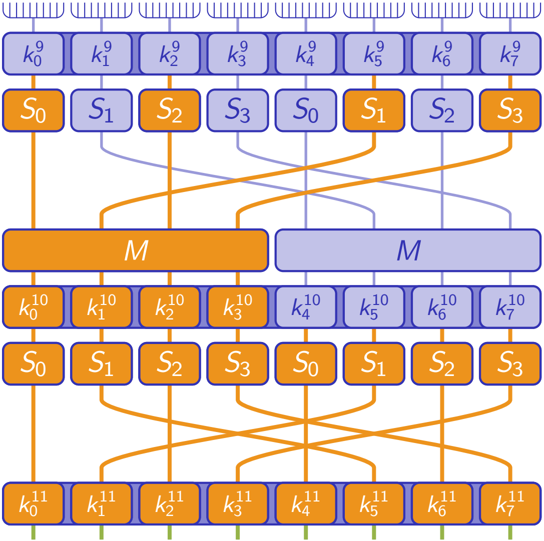
$k_0^{10} = k_0^{11} \oplus k_4^{11}$

$k_1^{10} = k_1^{11} \oplus k_5^{11}$

$k_2^{10} = k_2^{11} \oplus k_6^{11}$

$k_3^{10} = k_3^{11} \oplus k_7^{11}$

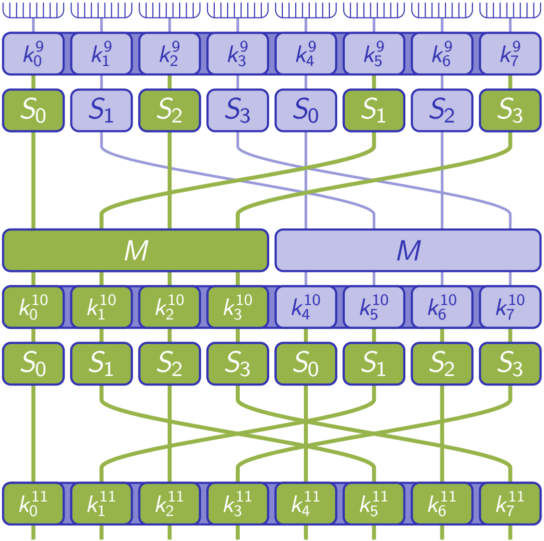
Overview of the Cryptanalysis



Test the 2^{15} saved keys:

$(k_0^{11}, k_1^{11}, k_2^{11}, k_3^{11}, k_4^{11}, k_5^{11}, k_6^{11}, k_7^{11})$

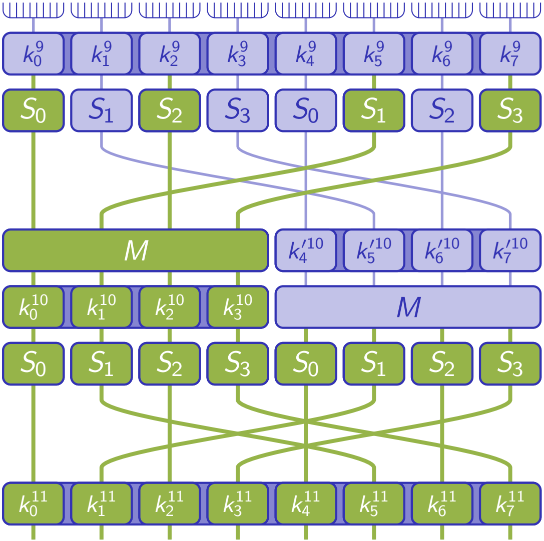
Overview of the Cryptanalysis



Save the best key:

$(k_0^{11}, k_1^{11}, k_2^{11}, k_3^{11}, k_4^{11}, k_5^{11}, k_6^{11}, k_7^{11})$

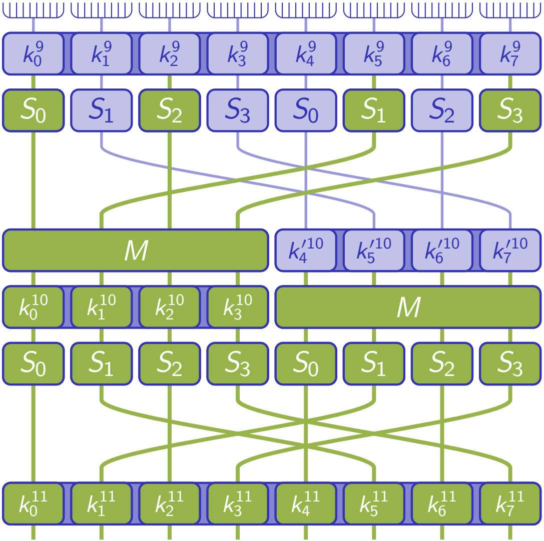
Overview of the Cryptanalysis



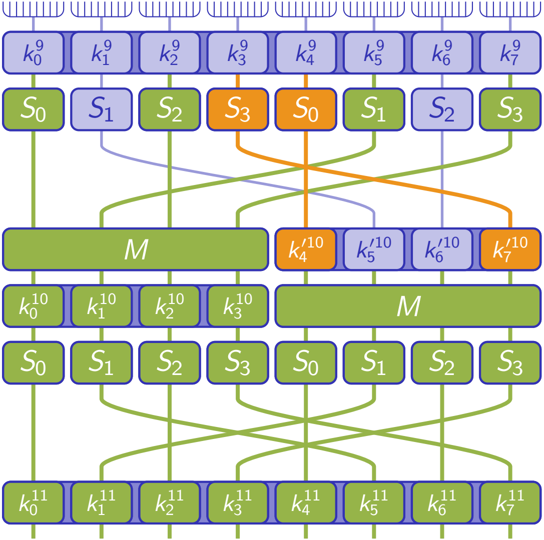
Observe that:

$(k_4^{10}, k_5^{10}, k_6^{10}, k_7^{10})$
$= M(k_4'^{10}, k_5'^{10}, k_6'^{10}, k_7'^{10})$

Overview of the Cryptanalysis



Overview of the Cryptanalysis



Brute force:

$(k_4'^{10}, k_5'^{10}, k_6'^{10}, k_7'^{10})$

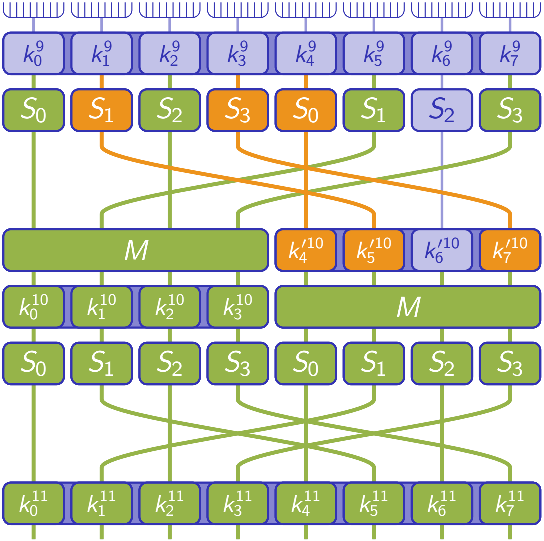
Test the 2^{15} saved keys:

$(k_4'^{10}, k_5'^{10}, k_6'^{10}, k_7'^{10})$

Save the 2^{15} best keys:

$(k_4'^{10}, k_5'^{10}, k_6'^{10}, k_7'^{10})$

Overview of the Cryptanalysis



Brute force:

$(k_4'^{10}, k_5'^{10}, k_6'^{10}, k_7'^{10})$

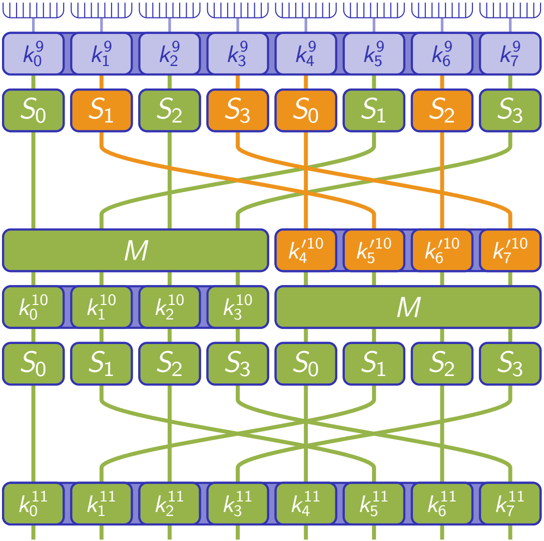
Test the 2^{15} saved keys:

$(k_4'^{10}, k_5'^{10}, k_6'^{10}, k_7'^{10})$

Save the 2^{15} best keys:

$(k_4'^{10}, k_5'^{10}, k_6'^{10}, k_7'^{10})$

Overview of the Cryptanalysis



Brute force:

$(k_4'^{10}, k_5'^{10}, k_6'^{10}, k_7'^{10})$

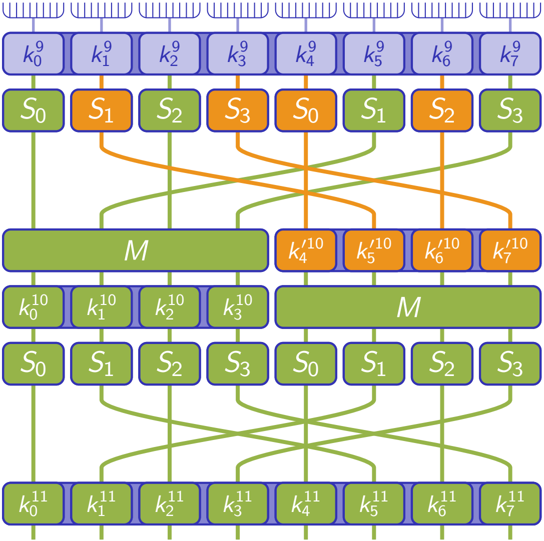
Test the 2^{15} saved keys:

$(k_4'^{10}, k_5'^{10}, k_6'^{10}, k_7'^{10})$

Save the 2^{15} best keys:

$(k_4'^{10}, k_5'^{10}, k_6'^{10}, k_7'^{10})$

Overview of the Cryptanalysis



For each saved key,

deduce the cipher key and test it

Cryptanalysis Summary

- Probabilities for the modified cipher
  - S_0, S_1, S_2: $944/1024$, S_3: $925/1024$

Cryptanalysis Summary

- Probabilities for the modified cipher
  - S_0, S_1, S_2: $944/1024$, S_3: $925/1024$
  - Round function: $(944/1024)^6 \times (925/1024)^2 \approx 2^{-1}$

Cryptanalysis Summary

- Probabilities for the modified cipher
  - S_0, S_1, S_2: 944/1024, S_3: 925/1024
  - Round function: $(944/1024)^6 \times (925/1024)^2 \approx 2^{-1}$
  - Full cipher: $(2^{-1})^{11} = 2^{-11}$

Cryptanalysis Summary

- Probabilities for the modified cipher
  - S_0, S_1, S_2: $944/1024$, S_3: $925/1024$
  - Round function: $(944/1024)^6 \times (925/1024)^2 \approx 2^{-1}$
  - Full cipher: $(2^{-1})^{11} = 2^{-11}$
  - If 30 000 plaintexts lie in the same coset, $30\,000 \times 2^{-11} \approx 15$ ciphertexts lie in the same coset on average

Cryptanalysis Summary

- Probabilities for the modified cipher
  - S_0, S_1, S_2: $944/1024$, S_3: $925/1024$
  - Round function: $(944/1024)^6 \times (925/1024)^2 \approx 2^{-1}$
  - Full cipher: $(2^{-1})^{11} = 2^{-11}$
  - If 30 000 plaintexts lie in the same coset, $30\,000 \times 2^{-11} \approx 15$ ciphertexts lie in the same coset on average
- Complexity of the cryptanalysis
  - Data: 30 000 plaintext/ciphertext pairs (2×300 Kb)

Cryptanalysis Summary

- Probabilities for the modified cipher
  - S_0, S_1, S_2: 944/1024, S_3: 925/1024
  - Round function: $(944/1024)^6 \times (925/1024)^2 \approx 2^{-1}$
  - Full cipher: $(2^{-1})^{11} = 2^{-11}$
  - If 30 000 plaintexts lie in the same coset, $30\,000 \times 2^{-11} \approx 15$ ciphertexts lie in the same coset on average
- Complexity of the cryptanalysis
  - Data: 30 000 plaintext/ciphertext pairs (2×300 Kb)
  - Time: $\approx$ 10s on a laptop (Core i7, 4 cores, 2.50GHz)

Cryptanalysis Summary

- Probabilities for the modified cipher
  - S_0, S_1, S_2: 944/1024, S_3: 925/1024
  - Round function: $(944/1024)^6 \times (925/1024)^2 \approx 2^{-1}$
  - Full cipher: $(2^{-1})^{11} = 2^{-11}$
  - If 30 000 plaintexts lie in the same coset, $30\,000 \times 2^{-11} \approx 15$ ciphertexts lie in the same coset on average
- Complexity of the cryptanalysis
  - Data: 30 000 plaintext/ciphertext pairs (2×300 Kb)
  - Time: $\approx$ 10s on a laptop (Core i7, 4 cores, 2.50GHz)
  - Probability of success $> 95\%$

Summary of the talk

Conclusion

- Proposition of an AES-like backdoored algorithm (80-bit block, 120-bit key, 11 rounds)
  - The backdoor is at the design level
  - Resistant to most known cryptanalyses
  - But absolutely unsuitable for actual security
  - Illustrates the issue of using foreign encryption algorithms which might be backdoored

Conclusion

- Proposition of an AES-like backdoored algorithm (80-bit block, 120-bit key, 11 rounds)
  - The backdoor is at the design level
  - Resistant to most known cryptanalyses
  - But absolutely unsuitable for actual security
  - Illustrates the issue of using foreign encryption algorithms which might be backdoored
- Future work
  - First step in a larger research work
  - Use of more sophisticated combinatorial structures
  - Considering key space partionning
  - Other backdoored algorithms to be published. Use of zero-knowledge cryptanalysis proof

Thank you for your attention
Questions & Answers