

Structural Evaluation of AES and Chosen-Key Distinguisher of 9-round AES-128

Jérémy Jean

joint work with Pierre-Alain Fouque and Thomas Peyrin
(appeared at CRYPTO 2013)

École Normale Supérieure, France

Crypto Seminar in Luxembourg — December 17, 2013

<http://www.di.ens.fr/~jean/>



Outline

1. Motivations
2. Algorithms
3. Application to AES-128
 - Truncated differences
 - Actual differences
4. Distinguishing 9R AES-128
5. The End

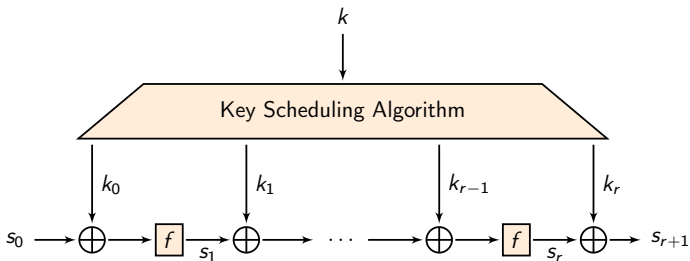
Outline

1. Motivations
2. Algorithms
3. Application to AES-128
 - Truncated differences
 - Actual differences
4. Distinguishing 9R AES-128
5. The End

Block Ciphers

Iterated SPN Block Ciphers

- ▶ Internal Permutation : f
- ▶ Number of Iterations : r
- ▶ SPN : $f = P \circ S$ applies Substitution (S) and Permutation (P).
- ▶ Secret Key : k
- ▶ Key Scheduling Algorithm : $k \rightarrow (k_0, \dots, k_r)$
- ▶ Ex : AES, PRESENT, SQUARE, Serpent, etc.

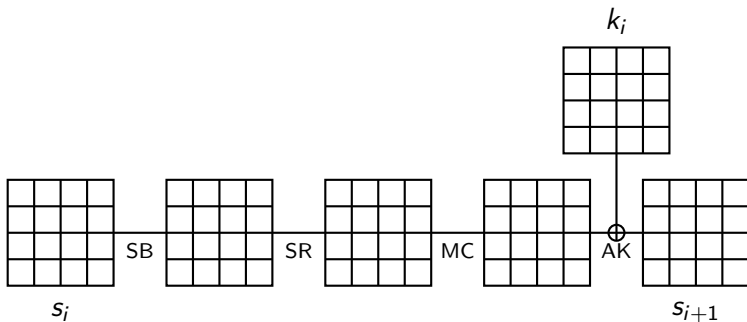


Advanced Encryption Standard

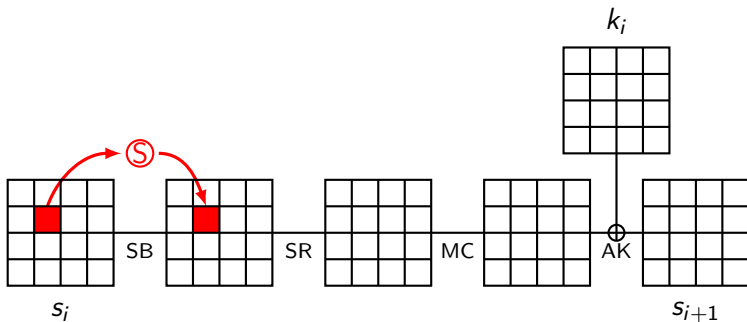
The AES Block Cipher (Rijndael)

- ▶ Designed by Joan Daemen and Vincent Rijmen
- ▶ Key-Alternating Cipher (round function : f)
- ▶ Block size : 128 bits — Key sizes : 128, 192 or 256 bits
- ▶ Number r of iterations : 10, 12 or 14
- ▶ Substitution-Permutation Network structure

AES Round Function



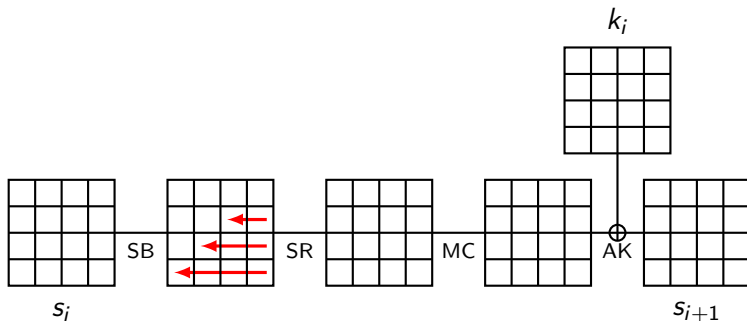
AES Round Function



One Step

- ▶ SubBytes (SB) layer : applies S-Box S to all bytes

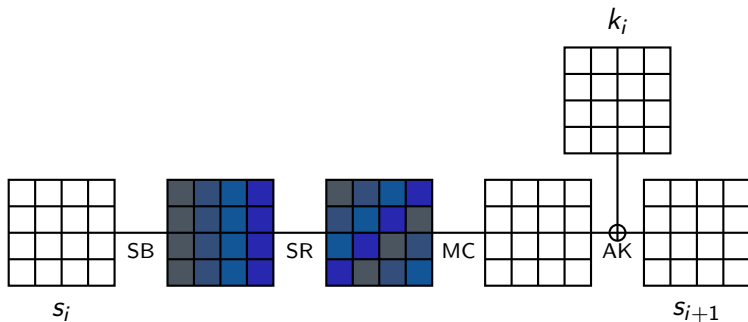
AES Round Function



One Step

- ▶ SubBytes (SB) layer : applies S-Box S to all bytes
- ▶ ShiftRows (SR) layer

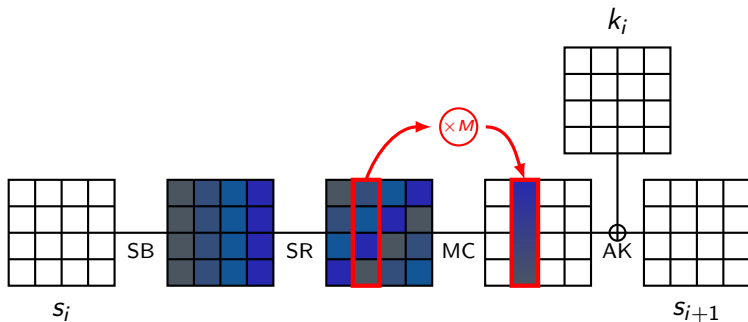
AES Round Function



One Step

- ▶ SubBytes (SB) layer : applies S-Box S to all bytes
- ▶ ShiftRows (SR) layer

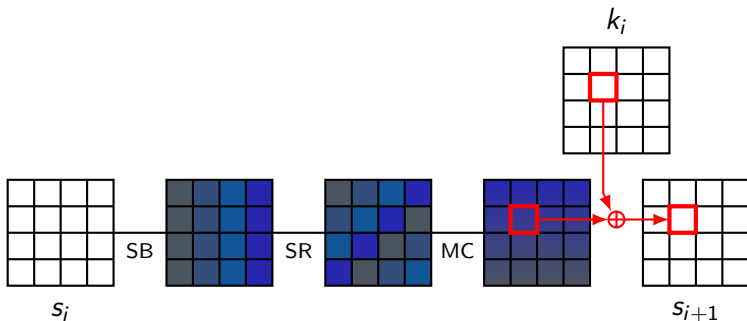
AES Round Function



One Step

- ▶ SubBytes (SB) layer : applies S-Box S to all bytes
- ▶ ShiftRows (SR) layer
- ▶ MixColumns (MC) layer : applies MDS matrix M to all columns

AES Round Function



One Step

- ▶ SubBytes (SB) layer : applies S-Box S to all bytes
- ▶ ShiftRows (SR) layer
- ▶ MixColumns (MC) layer : applies MDS matrix M to all columns
- ▶ AddRoundKey (AK) xors the subkey k_i to the state

Differentials and Differential Characteristics

Differential Characteristics

- ▶ Differential characteristics are easier to handle than differentials.
 ⇒ We usually **focus on characteristics**.
- ▶ Designers' goal : upper-bound the differential probability of characteristics.

Example : 4-round AES



- ▶ 4-round characteristic with 25 active S-Boxes (minimal).
- ▶ AES S-Box : $p_{max} = 2^{-6}$.
- ▶ Differential probability : $p \leq 2^{-6 \times 25} = 2^{-150}$.

AES

Design of the AES

- ▶ AES Permutation : **structurally bounded diffusion** for any rounds
- ▶ Provably resistant to Single-Key (SK) differential attacks
- ▶ Very easy to get the bounds by hand (just using the fact that the MixColumns matrix is MDS)

Minimal Number of Active S-Boxes for AES in the SK model

Rounds	1	2	3	4	5	6	7	8	9	10
min	1	5	9	25	26	30	34	50	51	55

Question

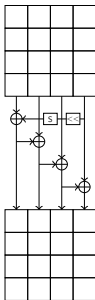
What would this table look like for the AES structure in the RK model ?

AES Key Schedule

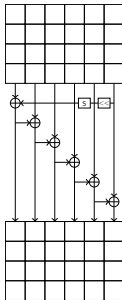
Design of the AES Key Schedule

- ▶ Ad-hoc key schedule

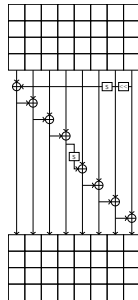
⇒ RK attacks for AES-192/256 [BKN-C09], [BK-A09], [BN-E10]



(a) AES-128.



(b) AES-192.



(c) AES-256.

Our Contributions

Main contribution

We propose an **algorithm** finding all the “smallest” RK characteristics :

- ▶ runs in time **linear** in the number of rounds, exponential in the state size (previous algorithms are exponential in both)
- ▶ for AES-128, requires a few hours on a single PC instead of several days previously
- ▶ for AES-128, depending on the output required, memory usually ranges from 0.5GB to 60GB

Side results for AES-128

- ▶ we provide the first chosen-key distinguisher for **9-round** AES-128
- ▶ AES-128 can not be proven secure against RK attacks with structural arguments only
- ▶ best RK characteristic for 5 rounds AES-128 has probability 2^{-105}

Outline

1. Motivations
2. Algorithms
3. Application to AES-128
 - Truncated differences
 - Actual differences
4. Distinguishing 9R AES-128
5. The End

Outline

1. Motivations
2. Algorithms
3. Application to AES-128
 - Truncated differences
 - Actual differences
4. Distinguishing 9R AES-128
5. The End

Existing Algorithms (1/2)

Matsui's Algorithm (e.g. DES)

- ▶ Works by **induction** :
derive best n -round char. from best
chars. on $1, \dots, n-1$ rounds
- ▶ Compute best char. for 1R
- ▶ Traverse a **tree** of depth 2 for 2R
- ▶ Pruning possible (A^* optim.)

Tree Example

$$p_i^j \stackrel{\text{def}}{=} \mathbb{P}(\Delta_i \rightarrow \Delta_j)$$

Δ_1

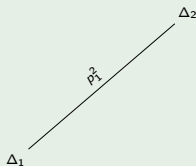
Existing Algorithms (1/2)

Matsui's Algorithm (e.g. DES)

- ▶ Works by **induction** :
derive best n -round char. from best
chars. on $1, \dots, n-1$ rounds
- ▶ Compute best char. for 1R
- ▶ Traverse a **tree** of depth 2 for 2R
- ▶ Pruning possible (A^* optim.)

Tree Example

$$p_i^j \stackrel{\text{def}}{=} \mathbb{P}(\Delta_i \rightarrow \Delta_j)$$



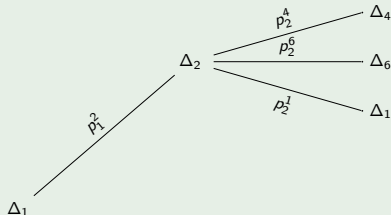
Existing Algorithms (1/2)

Matsui's Algorithm (e.g. DES)

- ▶ Works by **induction** :
derive best n -round char. from best
chars. on $1, \dots, n-1$ rounds
- ▶ Compute best char. for 1R
- ▶ Traverse a **tree** of depth 2 for 2R
- ▶ Pruning possible (A^* optim.)

Tree Example

$$p_i^j \stackrel{\text{def}}{=} \mathbb{P}(\Delta_i \rightarrow \Delta_j)$$



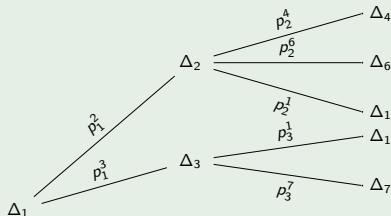
Existing Algorithms (1/2)

Matsui's Algorithm (e.g. DES)

- ▶ Works by **induction** :
derive best n -round char. from best
chars. on $1, \dots, n-1$ rounds
- ▶ Compute best char. for 1R
- ▶ Traverse a **tree** of depth 2 for 2R
- ▶ Pruning possible (A^* optim.)

Tree Example

$$p_i^j \stackrel{\text{def}}{=} \mathbb{P}(\Delta_i \rightarrow \Delta_j)$$



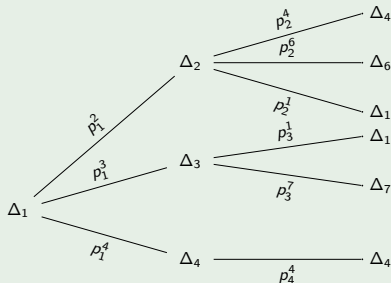
Existing Algorithms (1/2)

Matsui's Algorithm (e.g. DES)

- ▶ Works by **induction** :
derive best n -round char. from best
chars. on $1, \dots, n-1$ rounds
- ▶ Compute best char. for 1R
- ▶ Traverse a **tree** of depth 2 for 2R
- ▶ Pruning possible (A^* optim.)

Tree Example

$$p_i^j \stackrel{\text{def}}{=} \mathbb{P}(\Delta_i \rightarrow \Delta_j)$$



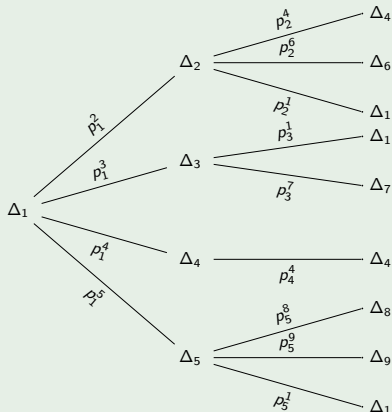
Existing Algorithms (1/2)

Matsui's Algorithm (e.g. DES)

- ▶ Works by **induction** :
derive best n -round char. from best
chars. on $1, \dots, n-1$ rounds
- ▶ Compute best char. for 1R
- ▶ Traverse a **tree** of depth 2 for 2R
- ▶ Pruning possible (A^* optim.)

Tree Example

$$p_i^j \stackrel{\text{def}}{=} \mathbb{P}(\Delta_i \rightarrow \Delta_j)$$



Existing Algorithms (1/2)

Matsui's Algorithm (e.g. DES)

- ▶ Works by **induction** :
derive best n -round char. from best
chars. on $1, \dots, n-1$ rounds
- ▶ Compute best char. for 1R
- ▶ Traverse a **tree** of depth 2 for 2R
- ▶ Pruning possible (A^* optim.)

Pros

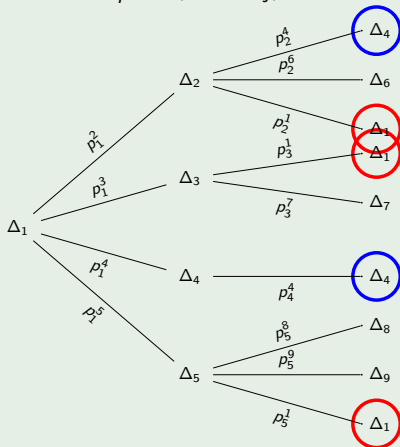
- ▶ works on DES in single-key

Drawbacks

- ▶ Rely on non-equivalent differential probabilities : needs dominant characteristic(s)
- ▶ Poor performances for AES
- ▶ Differences visited several times

Tree Example

$$p_i^j \stackrel{\text{def}}{=} \mathbb{P}(\Delta_i \rightarrow \Delta_j)$$



Existing Algorithms (2/2)

Biryukov-Nikolic [BN-E10]

- ▶ Adapt Matsui's algorithm
- ▶ Different algos for several KS

Pros

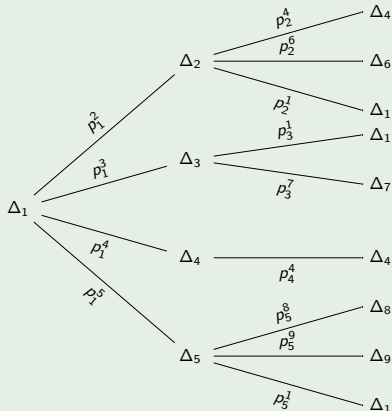
- ▶ Switch to truncated differences
⇒ less edges
- ▶ Representation of trunc. differences
⇒ handle branching in the KS
- ▶ Works on AES

Cons

- ▶ Not that fast because AES-128 has no predominant characteristic
- ▶ Differences visited several times
- ▶ Nodes visited exponential in the number of rounds

Tree Example

$$p_i^j \stackrel{\text{def}}{=} \mathbb{P}(\Delta_i \rightarrow \Delta_j)$$

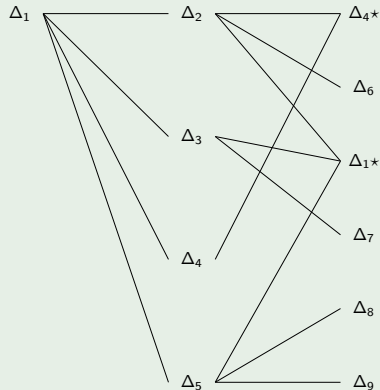


Our Algorithm

Algorithm

- ▶ Switch to a graph representation
- ▶ Merge equal diff. of the same round
- ▶ Graph traversal similar as Dijkstra
- ▶ Path search seen as **Markov process**

Graph Example

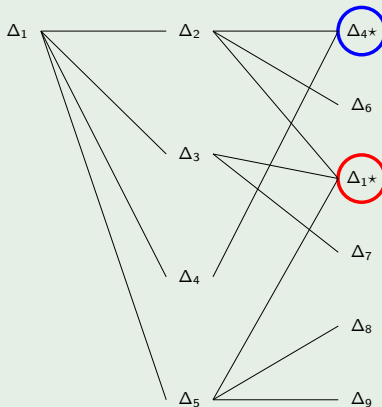


Our Algorithm

Algorithm

- ▶ Switch to a graph representation
- ▶ Merge equal diff. of the same round
- ▶ Graph traversal similar as Dijkstra
- ▶ Path search seen as Markov process

Graph Example

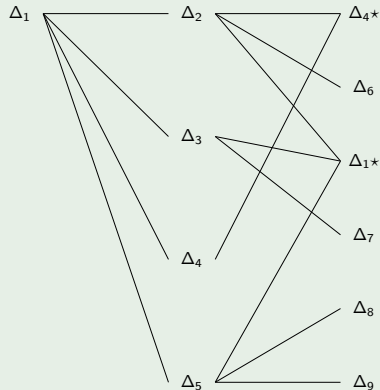


Our Algorithm

Algorithm

- ▶ Switch to a graph representation
- ▶ Merge equal diff. of the same round
- ▶ Graph traversal similar as Dijkstra
- ▶ Path search seen as **Markov process**

Graph Example



Our Algorithm

Algorithm

- ▶ Switch to a graph representation
- ▶ Merge equal diff. of the same round
- ▶ Graph traversal similar as Dijkstra
- ▶ Path search seen as **Markov process**

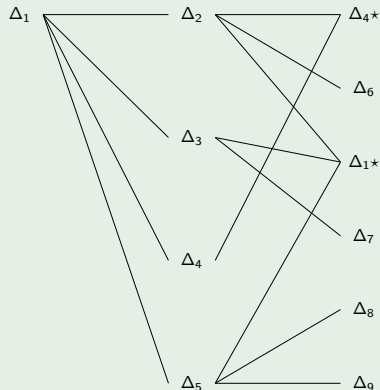
Pros

- ▶ Each difference in each round is visited **only once**
- ▶ Numbers of nodes and edges are **linear** in the number of rounds
- ▶ **A* optimization** still applies

Notes

- ▶ Only partial information propagated
- ▶ Need to adapt the Markov process

Graph Example



Outline

1. Motivations
2. Algorithms
3. Application to AES-128
 - Truncated differences
 - Actual differences
4. Distinguishing 9R AES-128
5. The End

Outline

1. Motivations
2. Algorithms
3. Application to AES-128
 - Truncated differences
 - Actual differences
4. Distinguishing 9R AES-128
5. The End

Different Levels of Analysis

Truncated Differences

- ▶ Basic Markov process
- ▶ Apply to any SPN cipher : we focus on AES-like ciphers
- ▶ Provide a **structural evaluation** of the cipher in regard to RK attacks
- ▶ For AES, similar results as the seminal work [DR-02] (for SK)

Actual Differences

- ▶ Enhanced Markov process :
 - ▶ More complete representation of differences
 - ▶ Add information for local system resolutions
- ▶ Need to be adapted to a particular cipher
- ▶ For AES, recover all the truncated results from [BN-E10]
- ▶ Full instantiation of characteristics while maximizing its probability
- ▶ Running time linear in the number of rounds

In reality : **Mixing** the two concepts

Outline

1. Motivations
2. Algorithms
3. Application to AES-128
 - Truncated differences
 - Actual differences
4. Distinguishing 9R AES-128
5. The End

Application to the Structure of AES-128

Structural Analysis

- ▶ We ignore the semantic definition of the S-Box and the MDS matrix
- ▶ We count the number of active S-Boxes (truncated differences)
- ▶ Do not apply to AES-128 with the instantiated S and P
- ▶ Give an estimation of the structural quality of the AES family

Related-Key Model (XOR difference of the keys)

Rounds	1	2	3	4	5	6	7	8	9	10
min	0	1	3	9	11	13	15	21	23	25

Hash Function Setting (KS considered independently)

Rounds	1	2	3	4	5	6	7	8	9	10
minmax	0	1	3	6	7	9	11	14	15	17

Examples of best truncated differential characteristics

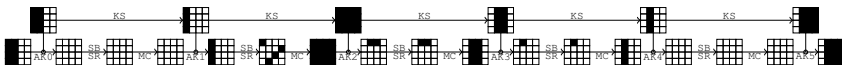


Figure : Best truncated differential characteristics for AES-128 when $r = 5$ rounds with 11 active S-Boxes.

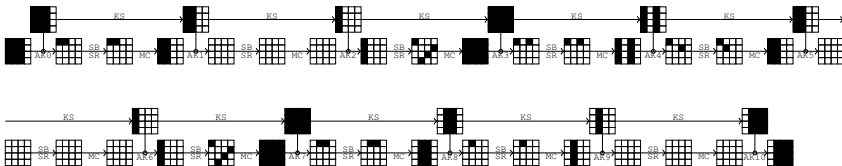


Figure : Best truncated differential characteristics for AES-128 when $r = 10$ rounds with 25 active S-Boxes.

Impossibility Results for the Structure of AES-128 (1/2)

There exists a characteristic on 10 rounds with only **25** active S-Boxes
 \implies best RK differential attack in p_{max}^{-25} computations.

Result 1

It is impossible to prove the security of the full AES-128 against **related-key differential attacks** without considering the differential property of the S-Box.

Notes

- ▶ With a random S-Box, p_{max}^{-25} might be smaller than 2^{128}
 \implies when $p_{max} \geq 2^{-5}$
- ▶ **AES structure on its own not enough for RK security**
- ▶ For a specified S-Box with bounded $p_{max} \leq 2^{-6}$
 \implies security against RK attacks

Impossibility Results for the Structure of AES-128 (2/2)

There exists a characteristic on 8 rounds with only **21** active S-Boxes
 \implies best RK differential attack in p_{max}^{-21} computations.

Result 2

It is impossible to prove the security of 8-round AES-128 against **related-key differential attacks** without considering both the differential property of the S-Box and the P layer.

Notes

- ▶ With a random S-Box, same reason as before
- ▶ For a specified S-Box with bounded $p_{max} \leq 2^{-6}$:
 - ▶ Best attack might be $2^{6 \times 21} = 2^{126} \leq 2^{128}$
 - ▶ For AES, we have exhausted all the possible attacks, no valid one
 - ▶ P layer and KS introduce **linear dependencies** in the characteristic
 - ▶ P can be chosen such that there is/isn't solutions

Outline

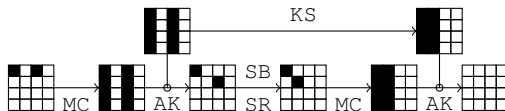
1. Motivations
2. Algorithms
3. Application to AES-128
 - Truncated differences
 - Actual differences**
4. Distinguishing 9R AES-128
5. The End

Markov Process and Filtering

Example of linear incompatibility in the case of AES-128

The linearity of the KS imposes all the active columns $[a, b, c, d]^T$ to be equal, which contradicts the first key addition (AK)

$$\mathbf{M} \cdot [x, 0, 0, 0]^T \oplus [x', 0, 0, 0]^T = \mathbf{M} \cdot [y, 0, 0, 0]^T \oplus [0, y', 0, 0]^T .$$



Post-filtering

- ▶ The problem with Markov process is that we lose all information from the past (how did I get to this difference?)... which is exactly what we need to detect the incompatibilities.
- ▶ We can still apply a filter on the output of the diff. characteristic search algorithm : test all the paths one by one and try to instantiate them.

State Compression

State Compression

Example of compressed truncated state and semi-compressed truncated state from a truncated state



(a) Truncated state. (b) Semi-compressed state. (c) Compressed state.

Dilemma

- ▶ if we compress the state too much, there will be too many inconsistent path, the filtering process will be too long
- ▶ if we don't compress enough, the differential characteristic search will be too long (or require too much memory)

Related-Key Attacks on AES-128

RK attacks against AES-128

- ▶ After **6 rounds**, there is no RK characteristic for AES-128 with a probability greater than 2^{-128} .
- ▶ For $1, \dots, 5$ rounds, our algorithm has found the best characteristics
- ▶ Same truncated characteristics as [\[BN-E10\]](#)
- ▶ Best instantiations of differences : **maximal probabilities**.

Best bounds on RK attacks for AES-128

Rounds	1	2	3	4	5
#S-Boxes	0	1	5	13	17
[BN-E10]	0	-6	-30	-78	-102
$\max \log_2(p)$	0	-6	-31	-81	-105

Outline

1. Motivations
2. Algorithms
3. Application to AES-128
 - Truncated differences
 - Actual differences
4. Distinguishing 9R AES-128
5. The End

Outline

1. Motivations
2. Algorithms
3. Application to AES-128
 - Truncated differences
 - Actual differences
4. Distinguishing 9R AES-128
5. The End

Distinguishing Model [KR-A07, BKN-C09]

Solve Open-Problem

We can use the best 5-round characteristic to construct a chosen-key distinguisher for **9-round AES-128**.

Let E_k be the 9-round AES-128 block cipher using key k .

Limited Birthday Problem [GP-FSE10]

Given

- ▶ a **fully** instantiated difference δ in the key,
- ▶ a **partially** instantiated difference Δ_{IN} in the plaintext,
- ▶ a **partially** instantiated difference Δ_{OUT} in the ciphertext,

find

- ▶ a key k ,
- ▶ a pair of messages (m, m') ,

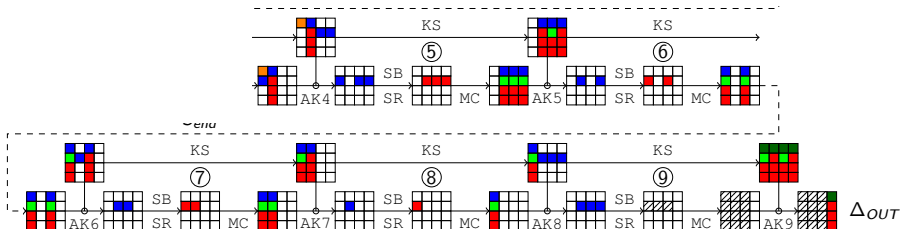
such that :

$$m \oplus m' \in \Delta_{IN}$$
$$\text{and : } E_k(m) \oplus E_{k \oplus \delta}(m') \in \Delta_{OUT}.$$

9-Round characteristic for AES-128

Construction of the characteristic

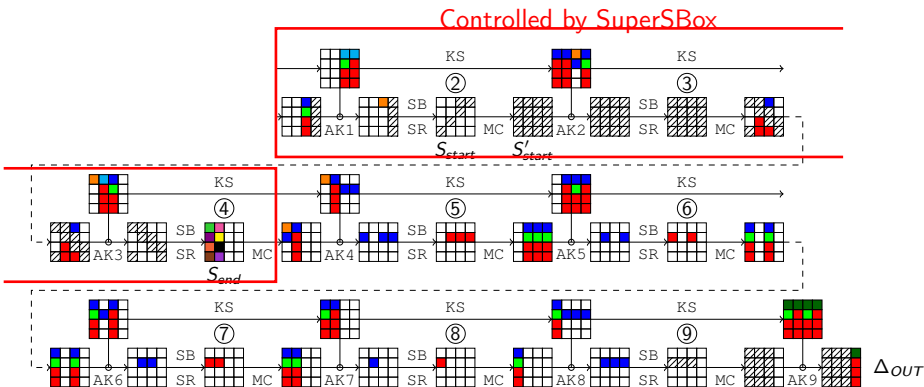
Take the best 5-round characteristic for AES-128 we have found.



9-Round characteristic for AES-128

Construction of the characteristic

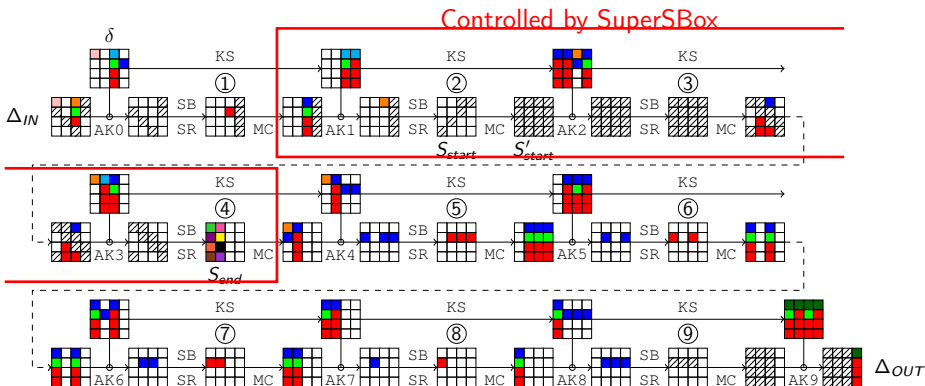
Prepend three rounds to be controlled by the SuperSBox technique.



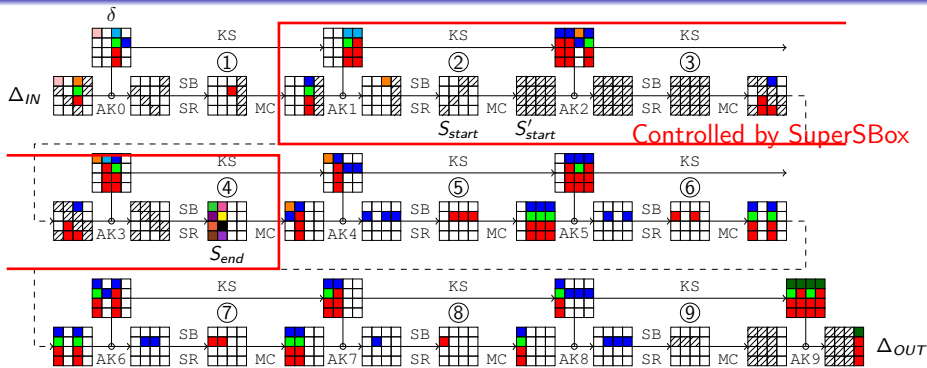
9-Round characteristic for AES-128

Construction of the characteristic

Prepend one other round, as inactive as possible.



9-Round CK Distinguisher for AES-128



Distinguishing algorithm

- ▶ Generate 2^{15} valid pairs of keys (about 2^{27} of them exist, since $\mathbb{P}_{KS} = 2^{-101}$)
 - ▶ Store the i th SuperSBox from S'_{start} to S_{end} in T_i (costs 2^{32})
 - ▶ For all 5 differences at S_{start} (costs 2^{40}), check the tables and :
 - ▶ Check backward direction : $p = 2^{-7}$ (a single S-Box)
 - ▶ Check forward direction : $p = 2^{-6 \times 8} = 2^{-48}$ (8 S-Boxes)

Time complexity

Complexity of the distinguishing algorithm

- ▶ Check probability : $2^{-7-48} = 2^{-55}$
- ▶ Time complexity :

$$2^{15} \times (2^{32} + 2^{40}) \approx 2^{55} \text{ computations}$$

- ▶ For 2^{15} different pairs of keys :
 - ▶ Construct the SuperSBoxes in 2^{32} operations
 - ▶ Try all values for the 5 byte-differences in 2^{40} operations

Generic time complexity

- ▶ Limited-Birthday Problem [GP-FSE10]
- ▶ Input space (Δ_{IN}) of size $4 \times 8 + 7 = 39$ bits
- ▶ Output space (Δ_{OUT}) of size $3 \times 7 = 21$ bits
- ▶ Time complexity : 2^{68} encryptions

Outline

1. Motivations
2. Algorithms
3. Application to AES-128
 - Truncated differences
 - Actual differences
4. Distinguishing 9R AES-128
5. The End

Outline

1. Motivations
2. Algorithms
3. Application to AES-128
 - Truncated differences
 - Actual differences
4. Distinguishing 9R AES-128
5. The End

Conclusion

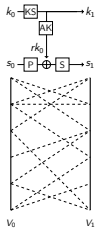
- ▶ New differential characteristics finding algorithm for SPN ciphers
 - ▶ **Graph-based** approach : Dijkstra and A^* optimization
 - ▶ Search the best truncated differential characteristics
 - ▶ Time complexity **linear** in the number of rounds considered
- ▶ Applications to the **structure** of AES-128 :
 - ▶ Impossibility results for related-key attacks
 - ▶ Impossibility results for the hash function setting
 - ▶ Exact probabilities for the best differential characteristics
(eg. 2^{-105} for 5 rounds)
- ▶ **Chosen-key distinguisher for 9-round AES-128**
 - ▶ Solve open problem
 - ▶ Time Complexity : 2^{55} encryptions
 - ▶ Generic Complexity : 2^{68} encryptions
- ▶ More details in the paper and its extended version (ePrint/2013/366)

Conclusion

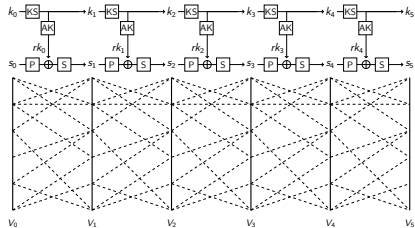
Thank you for your attention !

Questions ?

The graph G

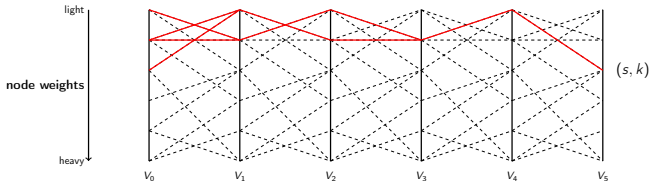


(d) Graph G .



(e) Graph G_5 .

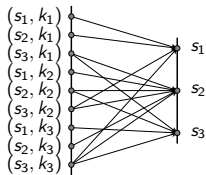
G is a bipartite directed acyclic graph, with the weight on the nodes.



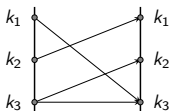
Implementation Tricks

Implementation Tricks

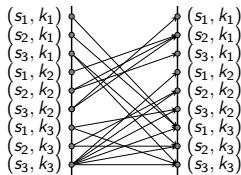
- ▶ we store only the graph G for one round, the entire graph is obtained by repeating G .
- ▶ instead of storing a huge graph G of all the best differential transitions for one round, we store separate graphs G_{BC} and G_{KS} . Then, G can be obtained by making the “product” of G_{BC} and G_{KS} .



(f) Graph G_{BC} .



(g) Graph G_{KS} .



(h) Graph G .

$$(s_i, k_j) \rightarrow (s_{i'}, k_{j'}) \in G \iff k_j \rightarrow k_{j'} \in G_{KS} \text{ and } (s_i, k_j) \rightarrow s_{i'} \in G_{BC}.$$