Exhausting Demirci-Selçuk Meet-in-the-Middle Attacks against Reduced-Round AES

Patrick Derbez[1] Pierre-Alain Fouque[1,2]

École Normale Supérieure, France

Université de Rennes 1, France

March 13, 2013

Outline

Outline for section 1

Advanced Encryption Standard

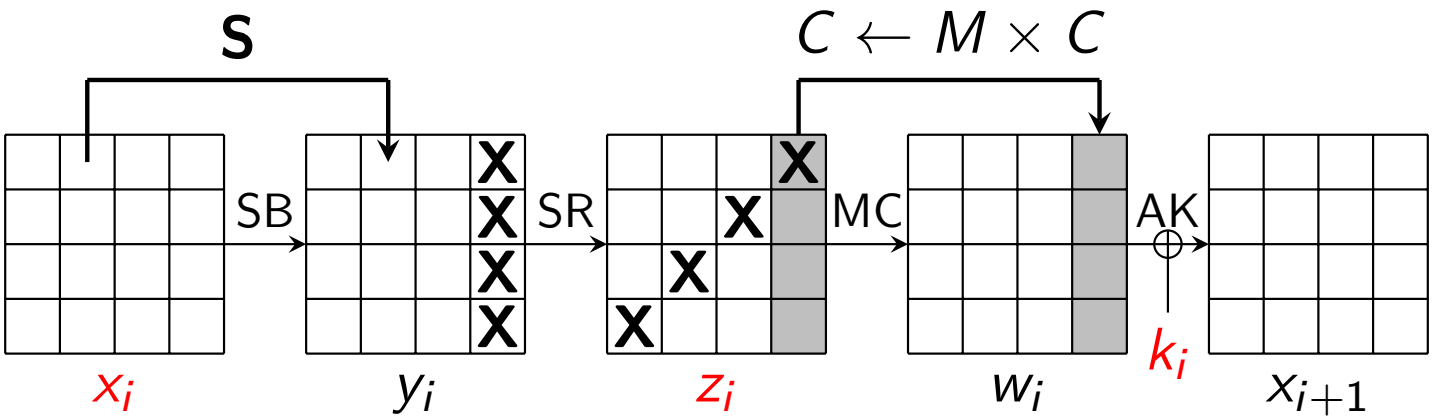
- Advanced Encryption Standard competition began in 1997
- Rijndael was selected to be the new AES in 2001

AES basic structures

- iterated block cipher
- substitution permutation network
- block size: 128 bits
- 3 different key lengths: 128, 192, 256 bits
- number of rounds depends on key lengths: 10, 12, 14 rounds

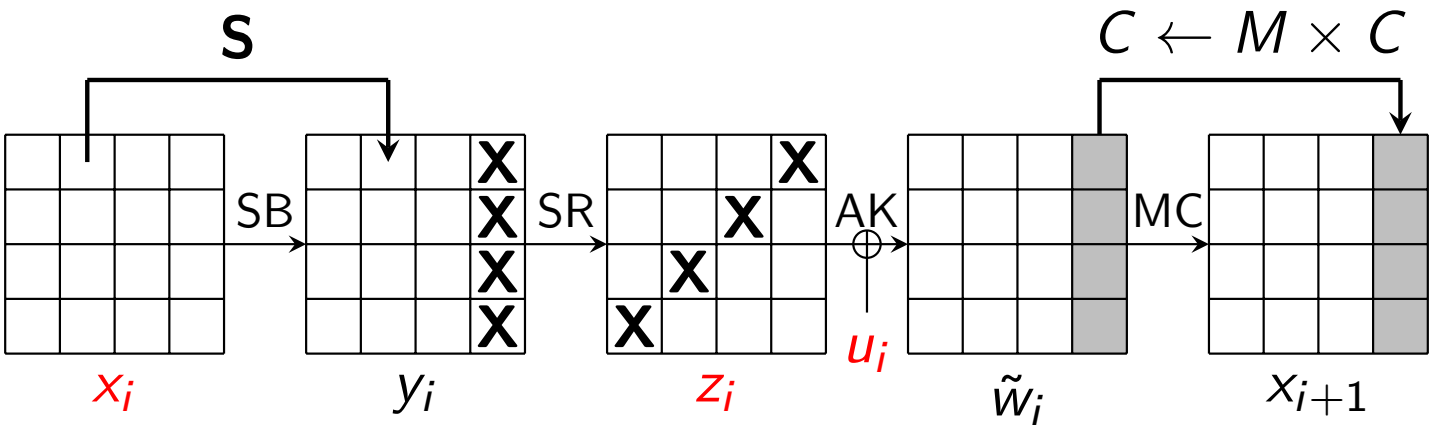
Description of the AES

- Each 16-byte block is represented as a 4×4 matrix of bytes
- Each byte representing an element from $\mathbb{F}_{256}$
- 4 simple operations on the state matrix every round (except the last round)



Description of the AES

- Each 16-byte block is represented as a 4×4 matrix of bytes
- Each byte representing an element from $\mathbb{F}_{256}$
- 4 simple operations on the state matrix every round (except the last round)



$$k_i = M \times u_i$$

AES and recent attacks

- Designed to be strong against Linear and Differential cryptanalysis.
- Fairly simple algebraic description...
- ... but attacks using SAT-solver or Gröbner basis algorithms never endanger it.
- Related-subkey attacks on the full AES-192/AES-256.
- Bicliques attacks on the full AES-128/AES-192/AES-256:

Version	Data	Time	Memory
128	2^{88}	$2^{126.2}$	2^{8}
192	2^{80}	$2^{189.4}$	2^{8}
256	2^{40}	$2^{254.4}$	2^{8}

Outline for section 2

Preliminary

Definition: δ-set

Set of 256 AES-states that are all different in one state byte and all equal in the other state bytes.

Preliminary

Definition: δ-set

Set of 256 AES-states that are all different in one state byte and all equal in the other state bytes.

- At FSE 2008, Demirci and Selçuk described a 4-round property for AES.

4-round property

Consider the encryption of a δ-set through four full AES rounds. For each of the 16 bytes of the state, the ordered sequence of 256 values of that byte in the corresponding ciphertexts is fully determined by just 25-byte parameters.

Preliminary

Definition: δ-set

Set of 256 AES-states that are all different in one state byte and all equal in the other state bytes.

- At FSE 2008, Demirci and Selçuk described a 4-round property for AES.

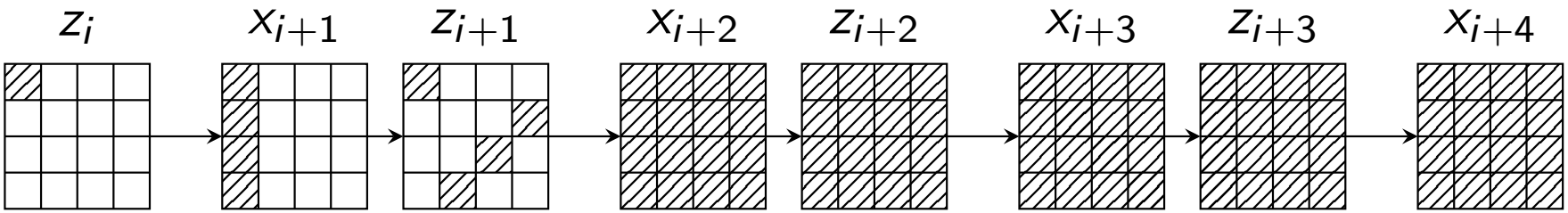
4-round property

Consider the encryption of a δ-set through four full AES rounds. For each of the 16 bytes of the state, the ordered sequence of 256 values of that byte in the corresponding ciphertexts is fully determined by just 25-byte parameters.

- At most $2^{8\times 25} = 2^{200}$ possible sequences out of the $2^{8\times 256} = 2^{2048}$ theoretically possible.

Proof of the 4-round property

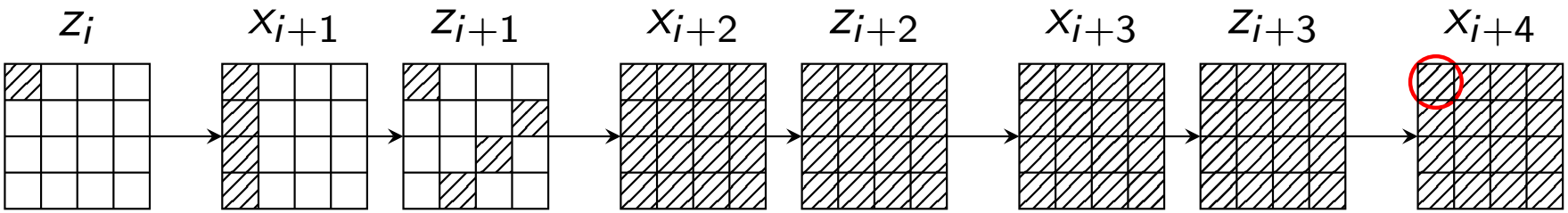
- Let consider the encryption of a δ-set through four full AES rounds:



Reminder: $z_j = SR \circ SB(x_j)$ and $x_{j+1} = AK \circ MC(z_j)$.

Proof of the 4-round property

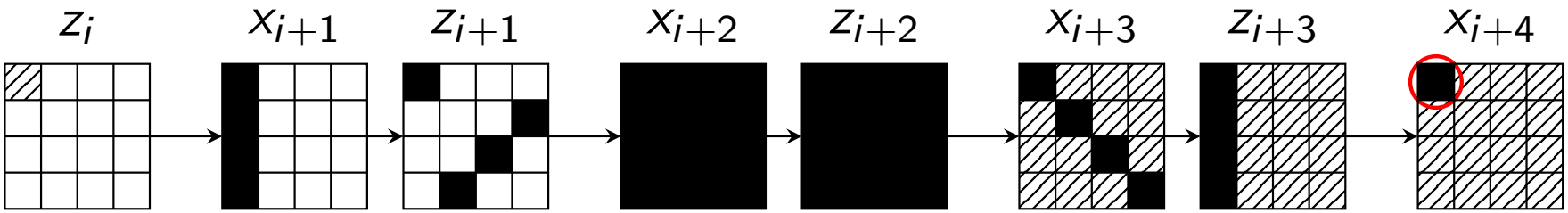
- Let consider the encryption of a δ-set through four full AES rounds:
  - To build the 256 values of the circled byte...



Reminder: $z_j = SR \circ SB(x_j)$ and $x_{j+1} = AK \circ MC(z_j)$.

Proof of the 4-round property

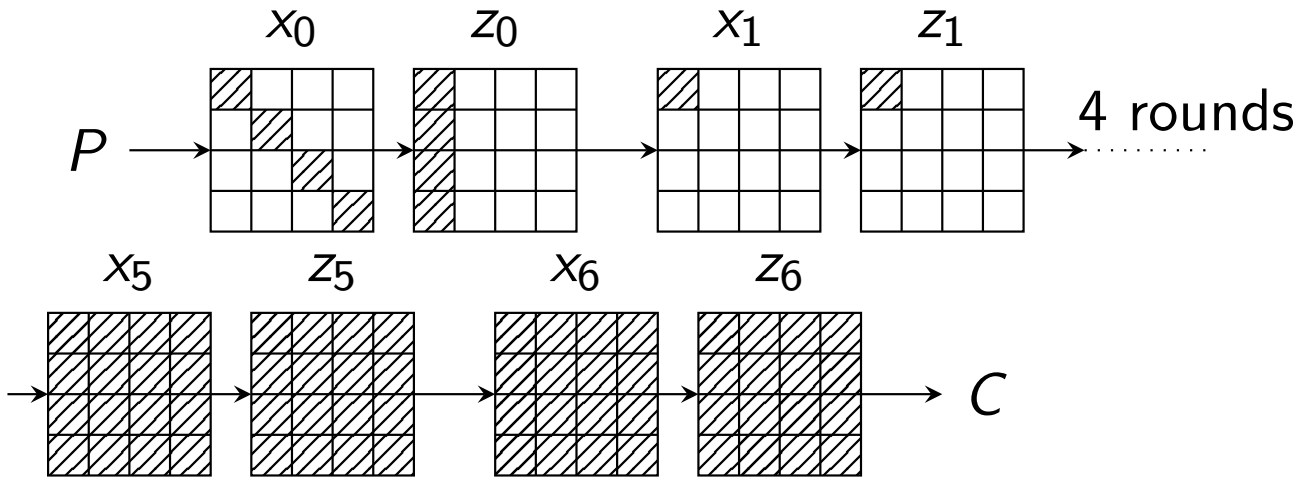
- Let consider the encryption of a δ-set through four full AES rounds:
  - To build the 256 values of the circled byte...
  - ...guess the black bytes for one message and propagate the differences.



Reminder: $z_j = SR \circ SB(x_j)$ and $x_{j+1} = AK \circ MC(z_j)$.

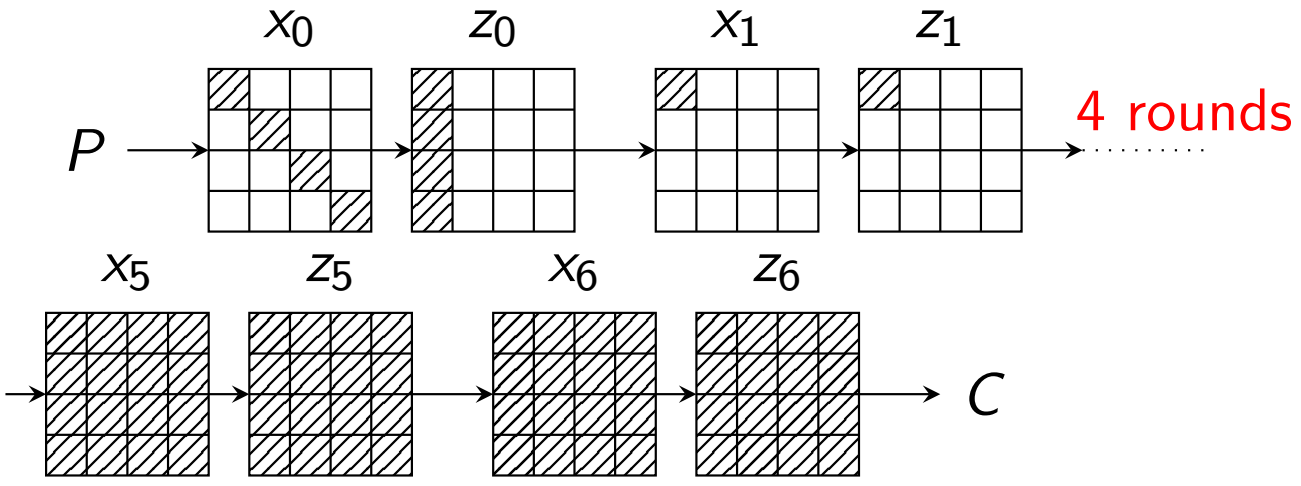
Basic attack

- They first use the property to mount an attack on 7 rounds of AES-256.



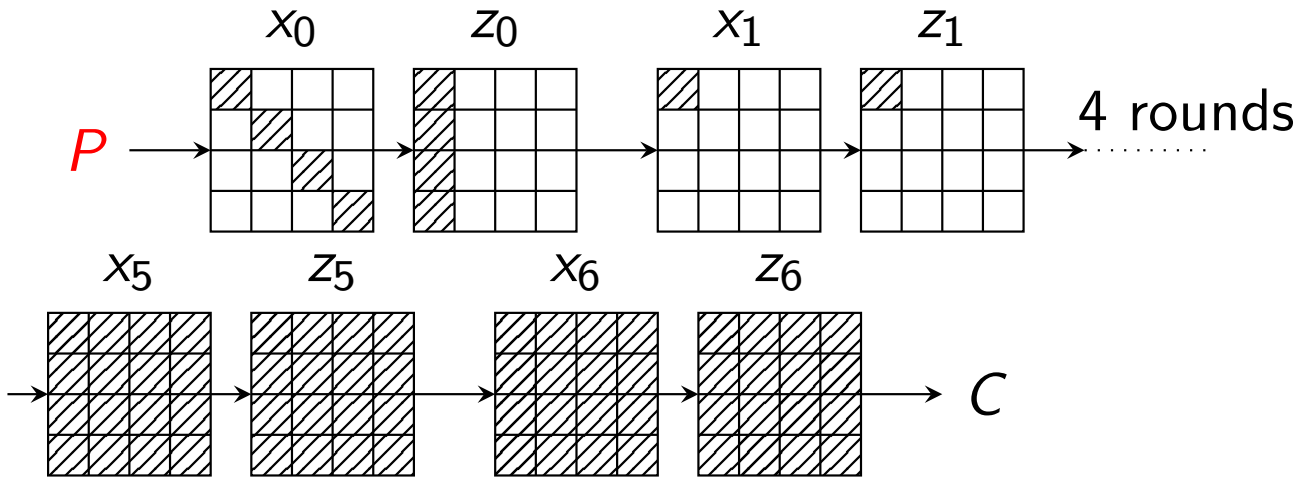
Basic attack

- They first use the property to mount an attack on 7 rounds of AES-256.
  1. Compute the 2^{200} possible sequences and store them in a hash table.



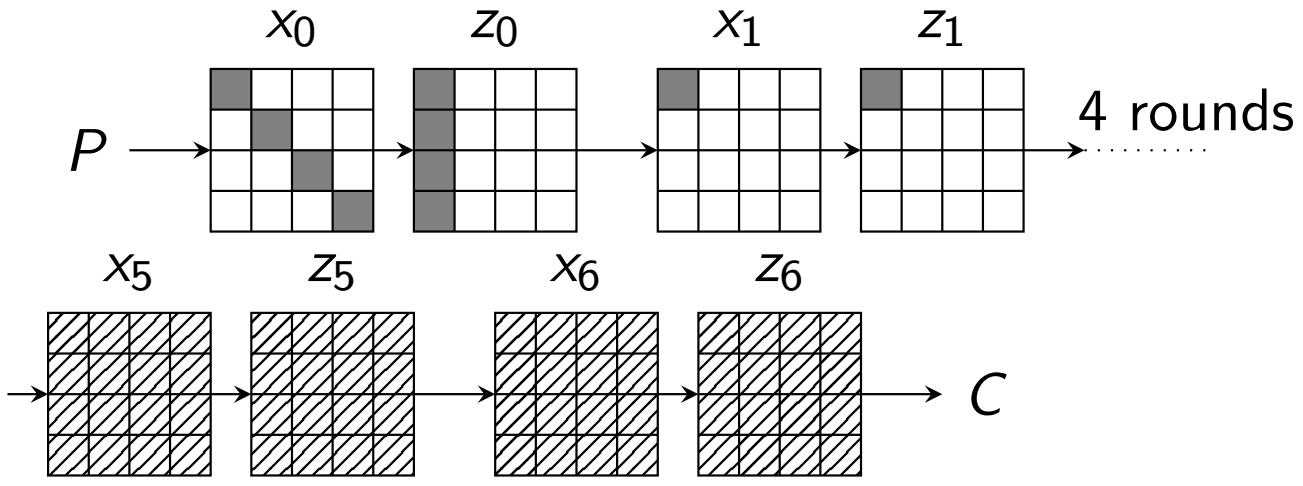
Basic attack

- They first use the property to mount an attack on 7 rounds of AES-256.
  1. Compute the 2^{200} possible sequences and store them in a hash table.
  2. Ask for a structure of 2^{32} plaintexts and choose one of them.



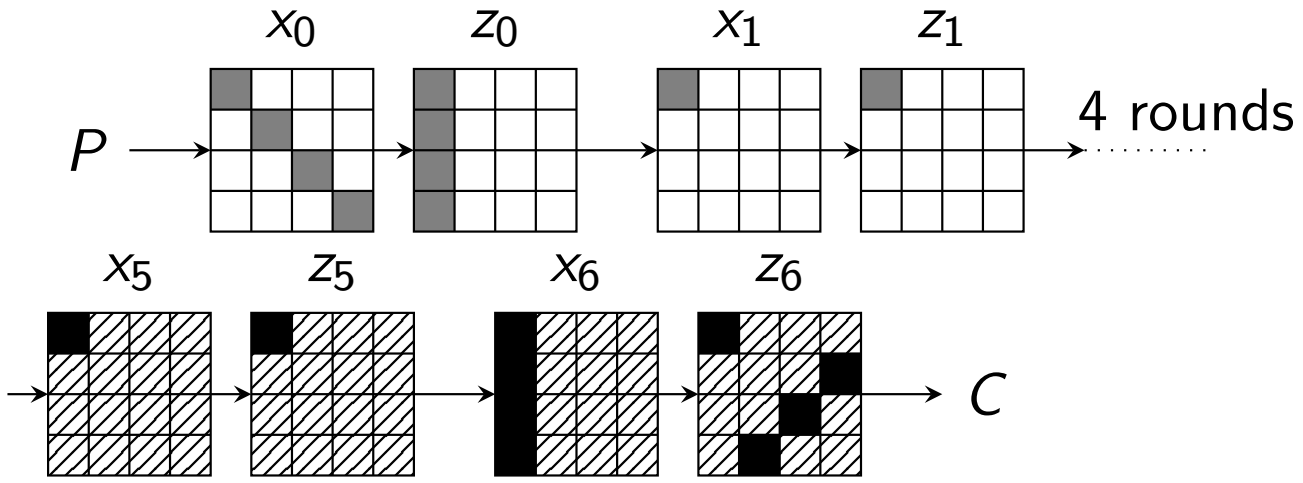
Basic attack

- They first use the property to mount an attack on 7 rounds of AES-256.
  1. Compute the 2^{200} possible sequences and store them in a hash table.
  2. Ask for a structure of 2^{32} plaintexts and choose one of them.
  3. Guess gray bytes to identify a δ-set and sort it.



Basic attack

- They first use the property to mount an attack on 7 rounds of AES-256.
  1. Compute the 2^{200} possible sequences and store them in a hash table.
  2. Ask for a structure of 2^{32} plaintexts and choose one of them.
  3. Guess gray bytes to identify a δ-set and sort it.
  4. Guess black bytes to compute the sequence and check if it belongs to the table.



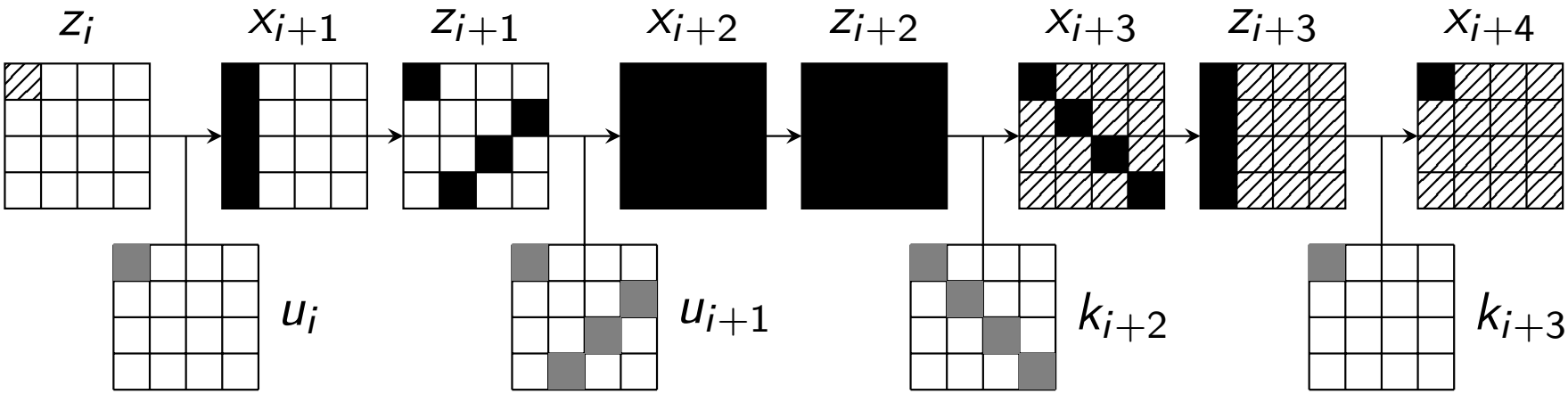
Comments

- Let $\mathcal{B}_{on}$ (resp. $\mathcal{B}_{off}$) be the state bytes needed in the online (resp. offline) phase.
- *A priori*, the time complexity of the online phase is $2^{8\times|\mathcal{B}_{on}|} \times 2^8$ partial encryptions/decryptions and the memory requirement is $2^{8\times|\mathcal{B}_{off}|}$ 256-byte sequences.
- In our case $|\mathcal{B}_{on}| = 10$ and $|\mathcal{B}_{off}| = 25$.
- The memory complexity of this attack is too high to apply it on the 128 and 192-bit versions.
- But its time complexity is low enough to mount an attack from it on 8 rounds AES-256.

Comments (cont.)

- Bytes of $\mathcal{B}_{off}$ (resp. $\mathcal{B}_{on}$) are related by the AES equations

$\Longrightarrow$ they may assume less values than expected.



- Let $\mathcal{K}_{off}$ be the vector space generated by these subkey bytes.
- In a similar way, we define $\mathcal{K}_{on}$ from $\mathcal{B}_{on}$.

Previous Improvements

- Difference instead of Value: Store sequences of differences to remove the byte of x_5 from $\mathcal{B}_{off}$ or from $\mathcal{B}_{on}$.

Previous Improvements

- Difference instead of Value: Store sequences of differences to remove the byte of x_5 from $\mathcal{B}_{off}$ or from $\mathcal{B}_{on}$.
- Multiset: Store unordered sequences to slightly reduces the memory requirement and, as the S-box is a bijection, to remove the byte of x_1 from $\mathcal{B}_{on}$.

Previous Improvements

- Difference instead of Value: Store sequences of differences to remove the byte of x_5 from $\mathcal{B}_{off}$ or from $\mathcal{B}_{on}$.
- Multiset: Store unordered sequences to slightly reduces the memory requirement and, as the S-box is a bijection, to remove the byte of x_1 from $\mathcal{B}_{on}$.
- Data/Time/Memory Trade-Off: Store only a fraction ε of the possible sequences. In exchange, data and time complexities are increased by a factor $1/\varepsilon$.

Previous Improvements

- Difference instead of Value: Store sequences of differences to remove the byte of x_5 from $\mathcal{B}_{off}$ or from $\mathcal{B}_{on}$.
- Multiset: Store unordered sequences to slightly reduces the memory requirement and, as the S-box is a bijection, to remove the byte of x_1 from $\mathcal{B}_{on}$.
- Data/Time/Memory Trade-Off: Store only a fraction ε of the possible sequences. In exchange, data and time complexities are increased by a factor $1/\varepsilon$.
- Data Recycling: The structure used in the attack contains 2^{24} δ-sets. Thus the data may be reused 2^{24} times in the Data/Time/Memory Trade-Off.

Summary

- The basic attack of Demirci and Selçuk requires a huge memory and a relatively small time complexity.
- The classical data/time/memory trade-off allows to *balance* these complexities.
- But it increases the data complexity and randomizes the attack.
- On seven rounds, the amount of data needed is approximately 2^{70} chosen plaintexts.

Summary

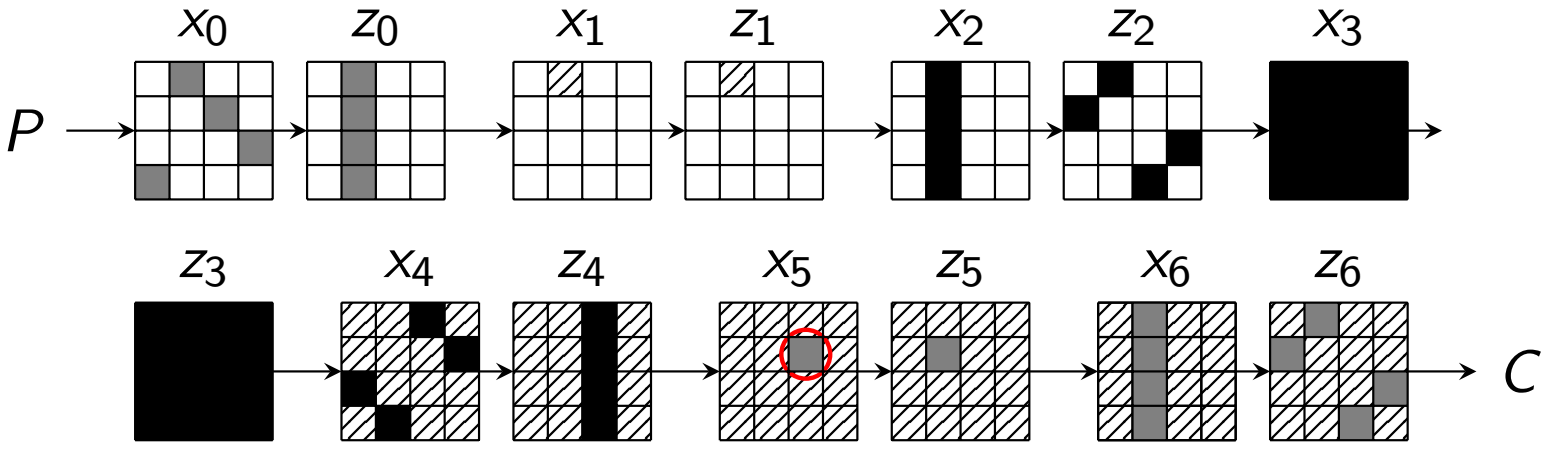
- The basic attack of Demirci and Selçuk requires a huge memory and a relatively small time complexity.
- The classical data/time/memory trade-off allows to *balance* these complexities.
- But it increases the data complexity and randomizes the attack.
- On seven rounds, the amount of data needed is approximately 2^{70} chosen plaintexts.

$$\Longrightarrow \text{How to reduce it?}$$

First improvement

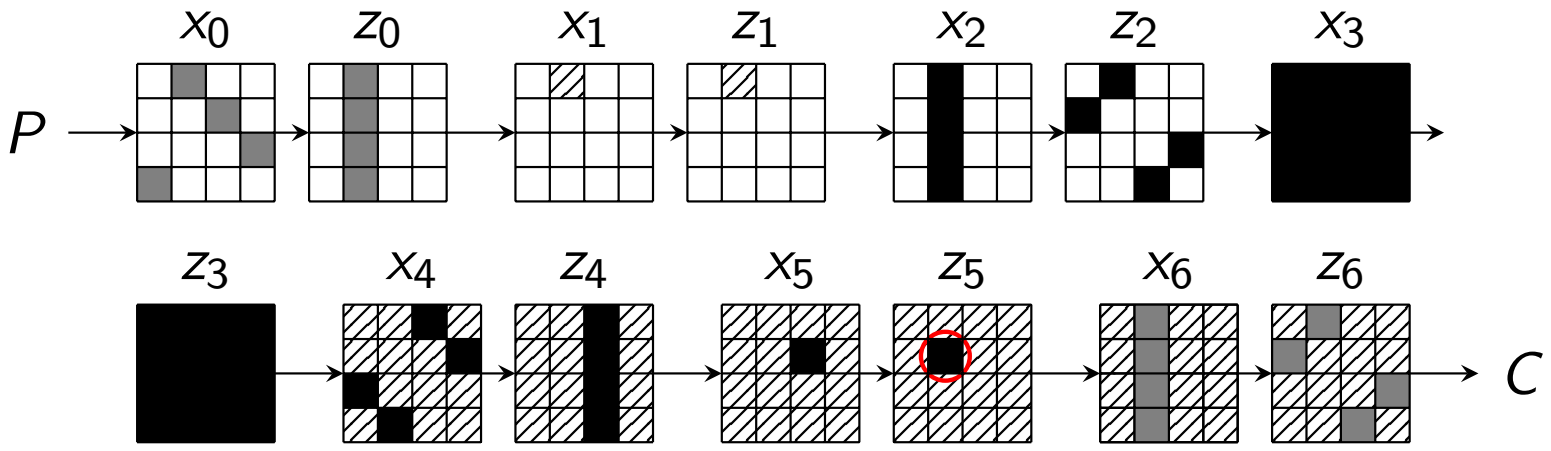
- Demirci and Selçuk sort a δ-set according to the value of the active byte of z_1.
- We propose to sort it according to the difference in that byte.
- As a consequence, the byte of u_i is removed from the generators of $\mathcal{K}_{off}$.
- In an other hand, we can reuse a δ-set 256 times in the data/time/memory trade-off.

Second improvement



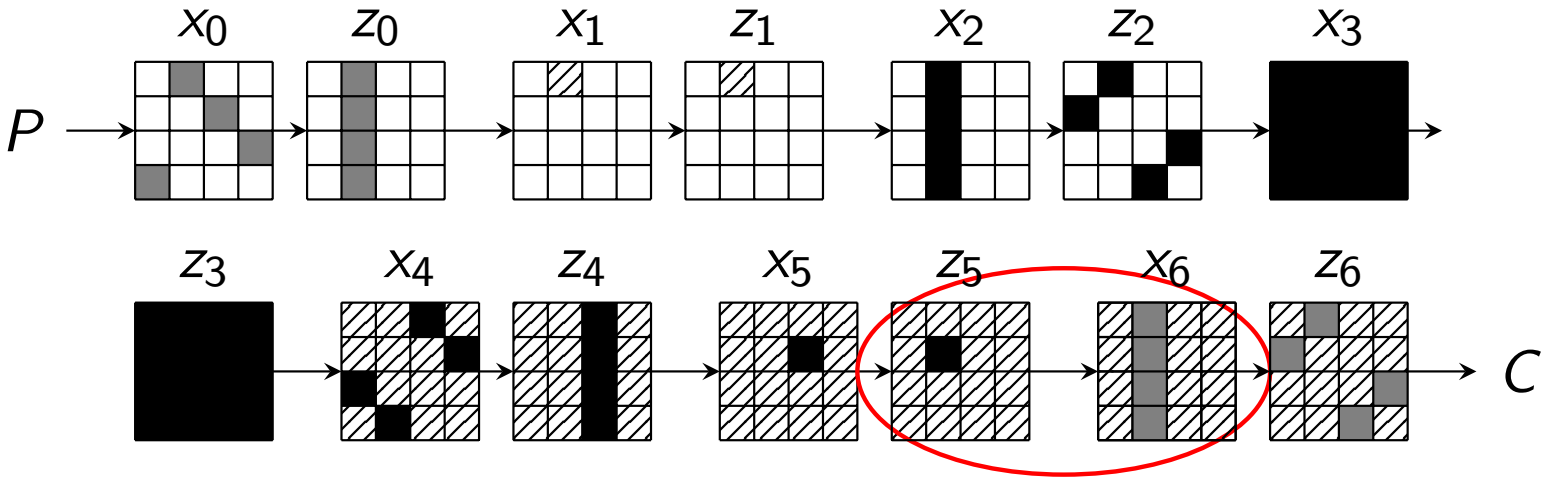
- Demirci and Selçuk consider simple cases.

Second improvement



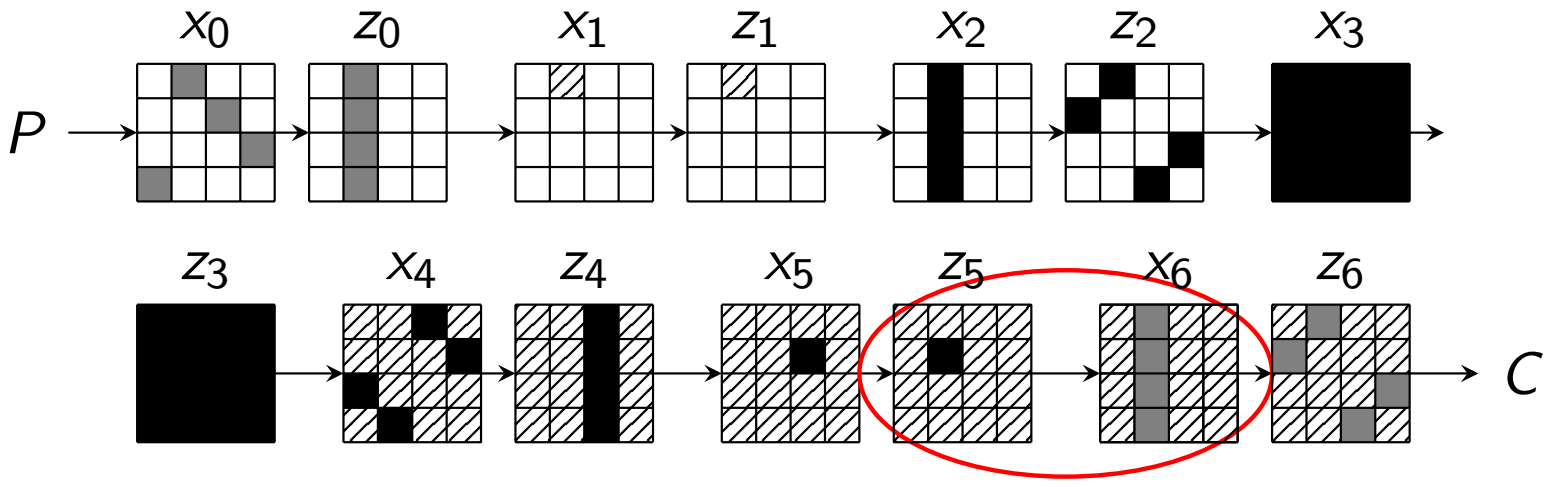
- Demirci and Selçuk consider simple cases.

Second improvement



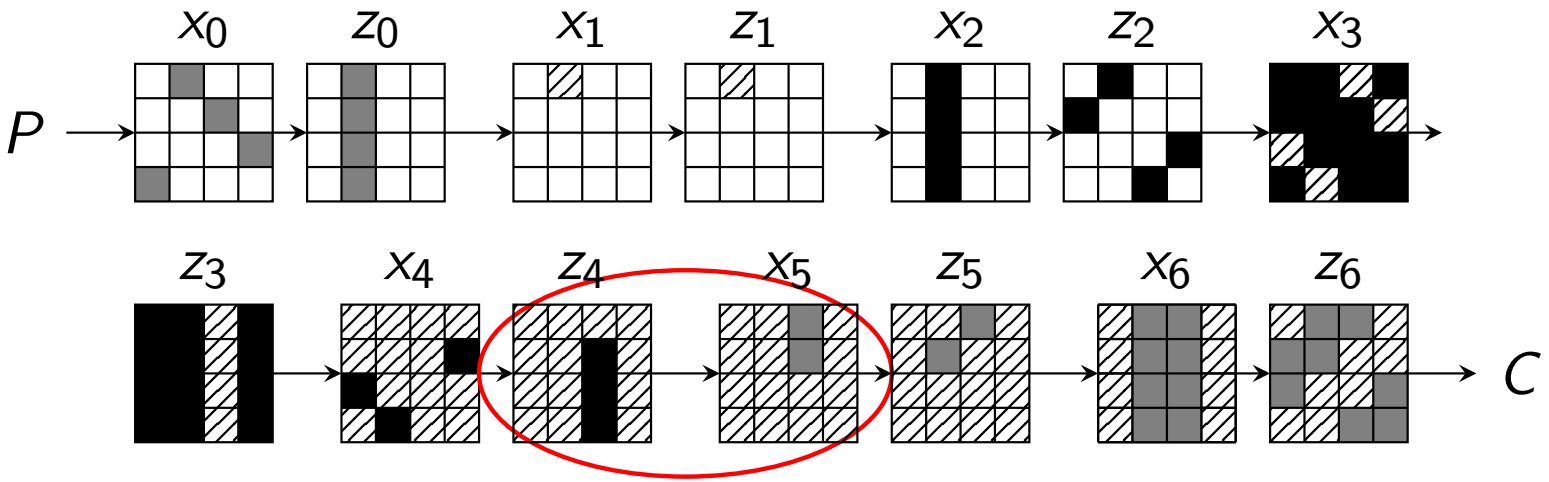
- Demirci and Selçuk consider simple cases.

Second improvement



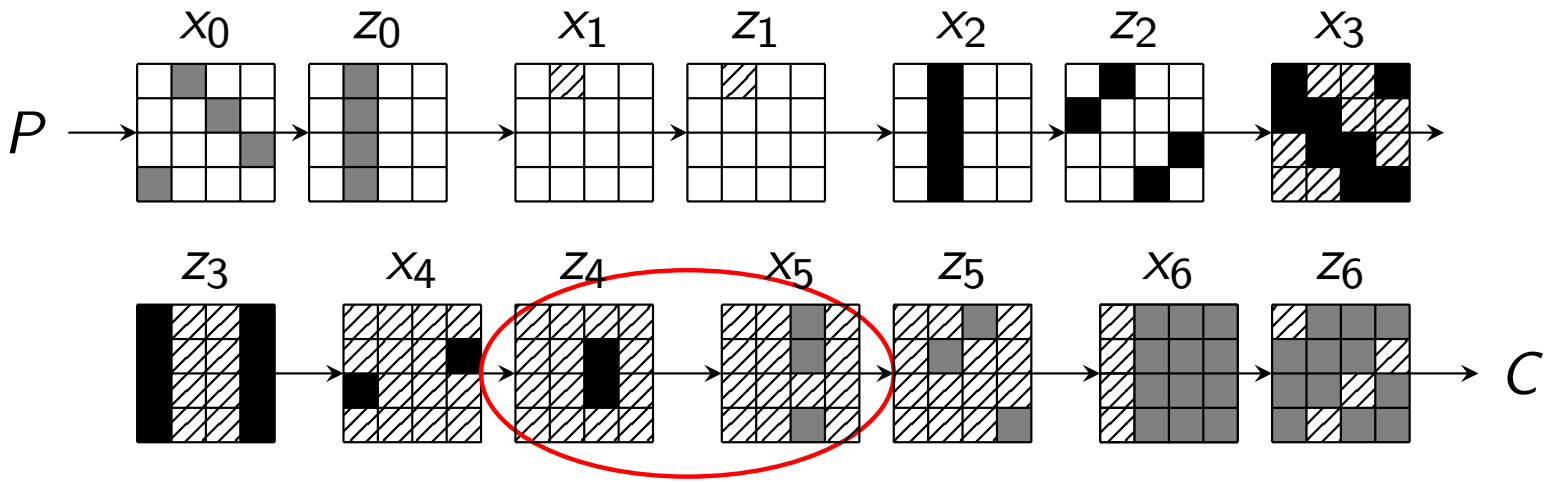
- Demirci and Selçuk consider simple cases.
- The matrix used in the MixColumns operation is MDS.

Second improvement



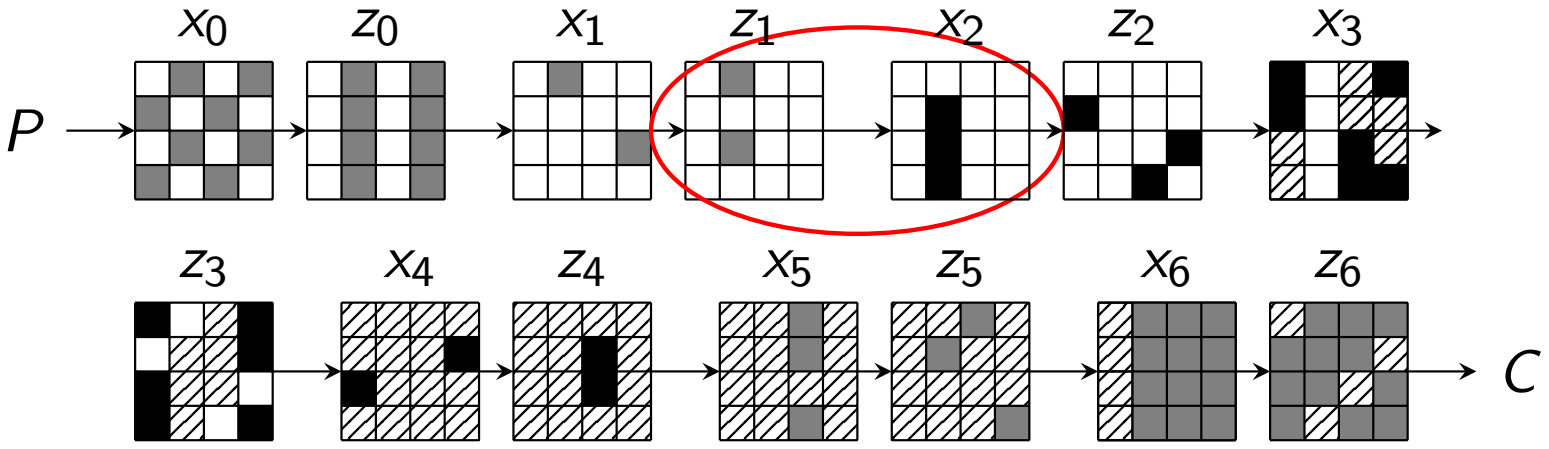
- Demirci and Selçuk consider simple cases.
- The matrix used in the MixColumns operation is MDS.

Second improvement



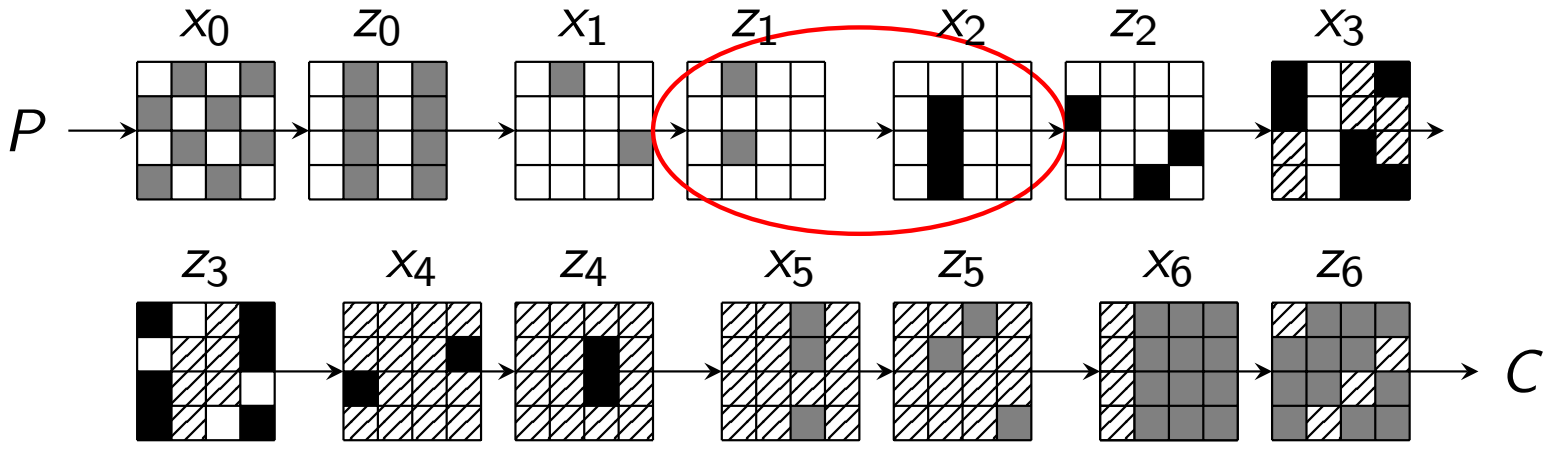
- Demirci and Selçuk consider simple cases.
- The matrix used in the MixColumns operation is MDS.

Second improvement



- Demirci and Selçuk consider simple cases.
- The matrix used in the MixColumns operation is MDS.
- The same idea may be applied to the δ-set.

Second improvement



- Demirci and Selçuk consider simple cases.
- The matrix used in the MixColumns operation is MDS.
- The same idea may be applied to the δ-set.

$\Longrightarrow$ New variants of the original attack

Finding best attacks

Once the cipher split in three parts:

- Number of variants:

$$\left(4 \times \binom{8}{5}\right)^2 \approx 2^{15.6}$$

- Number of sets $\mathcal{B}_{on}$ (resp. $\mathcal{B}_{off}$):

$$\left(4 \times \left(\binom{4}{1} + \binom{4}{2} + \binom{4}{3} + \binom{4}{4}\right)\right)^2 \approx 2^{11.8}$$

- For each of them we have to answer to the two following questions:
  - How many values can assume those state bytes?
  - How fast can we enumerate them?

Finding best attacks (cont.)

- *A priori*, not an easy task because S-boxes are involved in the keyschedules.
- We used the tool developed by Bouillaguet *et al.* and presented at CRYPTO'11.

OriginalTool

Input: System of equations E in variables X involving some S-boxes.
Output: An *optimal* algorithm to enumerate all the solutions of E with predictable time and memory complexities.

- The problem we seek to solve is very close to the problem solved by this tool but is still different.

Tweaked tool

- We have slightly tweaked the original tool.

TweakedTool

Input: System of equations E in variables X involving some S-boxes and a subset $Y \subseteq X$.
Output: A list of *optimal* algorithms to enumerate all the possible values of Y according to the system of equations E with predictable time and memory complexities.

- The output is a list because the number of enumerated values is not constant.
- The complexity is exponential in the number of involved S-boxes

$$\Longrightarrow \text{ apply it on } \mathcal{K} \text{ instead of } \mathcal{B}.$$

Results

- All attacks exhausted for the three key lengths.
- Results on 7-rounds AES-192 (last MixColumns performed):

number of guess in the offline phase

number of guess in the online phase	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
10																	4
11																4	
12															4		
13													4				
14												4					3
15											4					3	
16										4							
17									4					3			
18								4					3				2
19							4					3					1
20						4					3		1				
21					4			3				1					
22		4					3	1									
23	4				3		1										

- Best attacks require only 2^{32} chosen plaintexts.

Outline for section 3

Differential Enumeration Technique

- The idea of Dunkelman *et al.* is to store in the hash table only the multisets built from a δ-set containing a message m that belongs to a pair (m, m') following a well-chosen differential path.
- In a recent eprint paper, Derbez *et al.* used this idea to obtain the best known attacks on 7, 8 and 9 rounds:

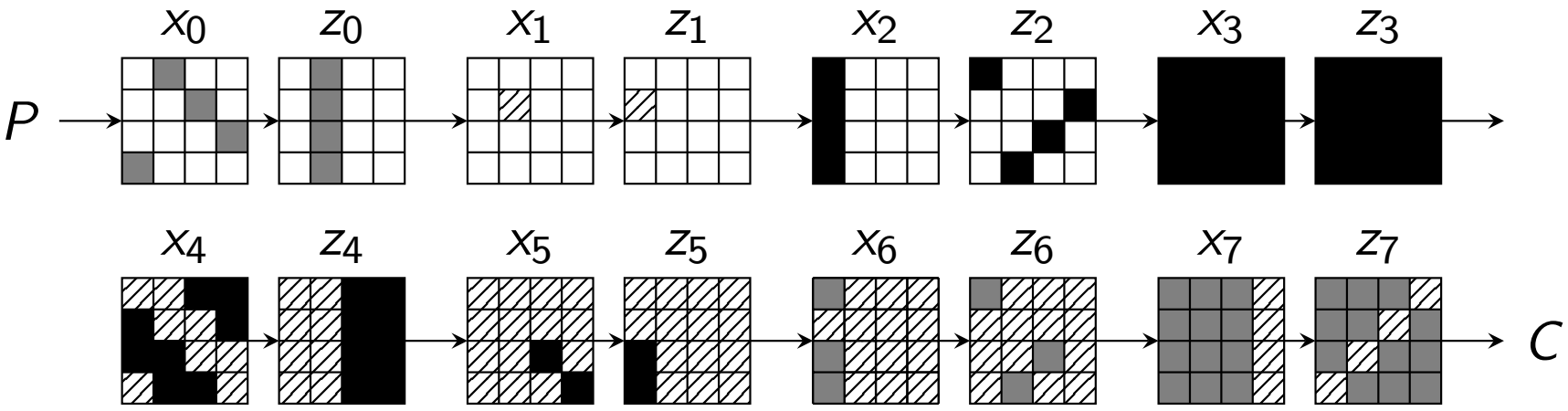
Version	Rounds	Data	Time	Memory
All	7	2^{97}	2^{99}	2^{98}
192	8	2^{113}	2^{172}	2^{82}
192	8	2^{107}	2^{172}	2^{96}
256	8	2^{113}	2^{196}	2^{82}
256	8	2^{107}	2^{196}	2^{96}
256	9	2^{120}	2^{203}	2^{203}

Differential Enumeration Technique

- The idea of Dunkelman *et al.* is to store in the hash table only the multisets built from a δ-set containing a message m that belongs to a pair (m, m') following a well-chosen differential path.
- In a recent eprint paper, Derbez *et al.* used this idea to obtain the best known attacks on 7, 8 and 9 rounds:

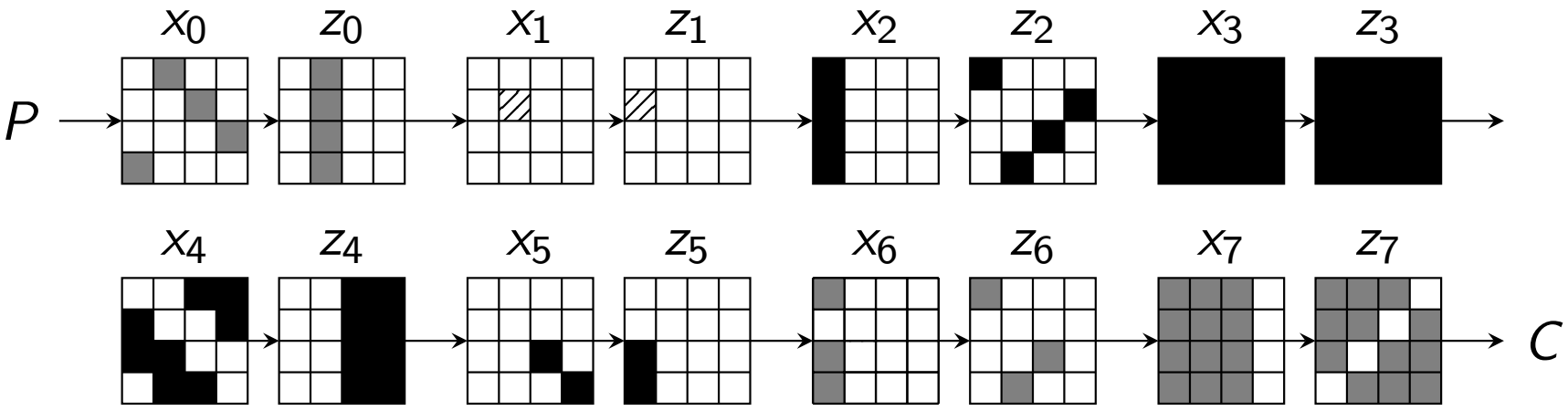
Version	Rounds	Data	Time	Memory
All	7	2^{97}	2^{99}	2^{98}
192	8	2^{113}	2^{172}	2^{82}
192	8	2^{107}	2^{172}	2^{96}
256	8	2^{113}	2^{196}	2^{82}
256	8	2^{107}	2^{196}	2^{96}
256	9	2^{120}	2^{203}	2^{203}

New attack on 8 rounds



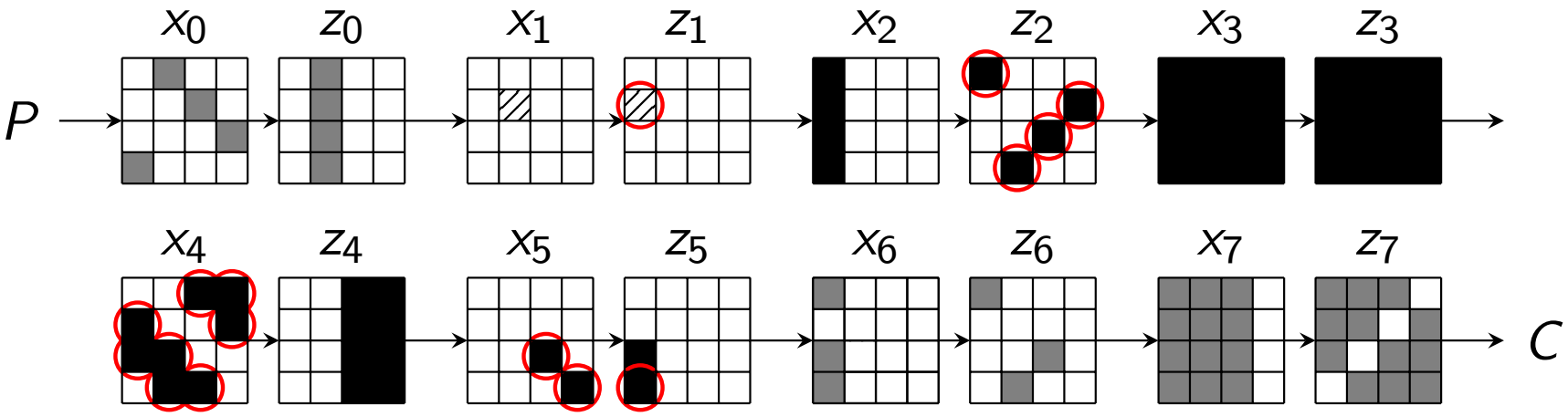
- Bytes of $\mathcal{B}_{on}$ are in gray.
- Bytes of $\mathcal{B}_{off}$ are in black.

New attack on 8 rounds



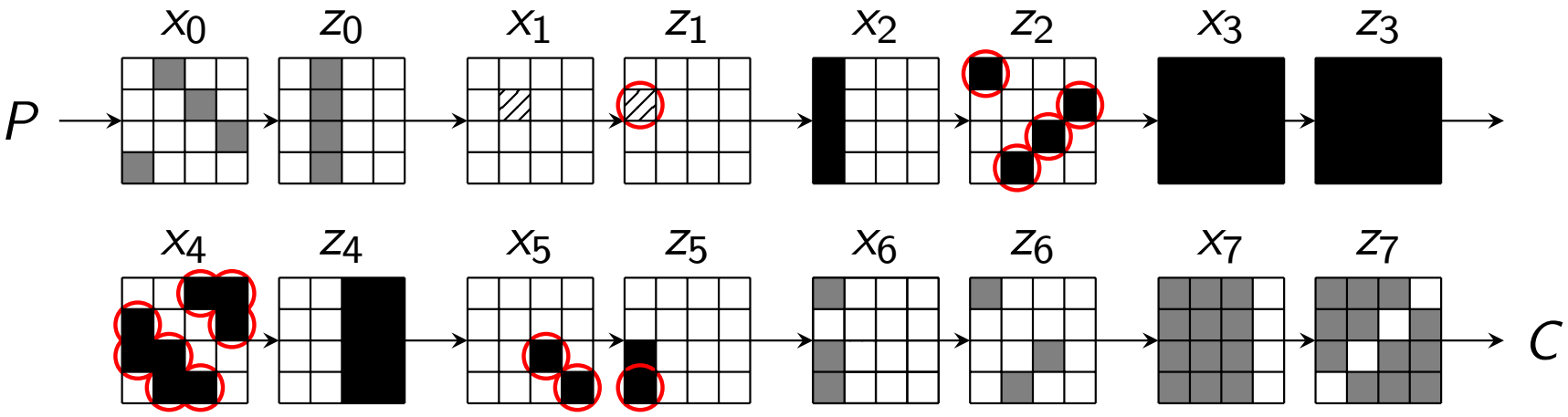
- Consider a pair that follows the differential.

New attack on 8 rounds



- Guess differences in circled bytes to deduce black bytes.

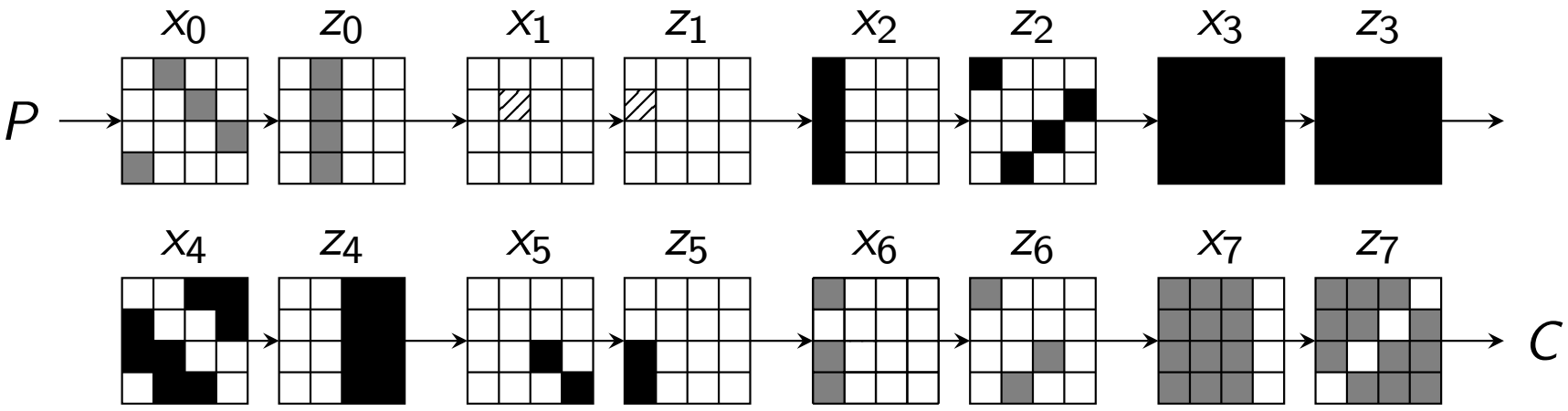
New attack on 8 rounds



- Guess differences in circled bytes to deduce black bytes.

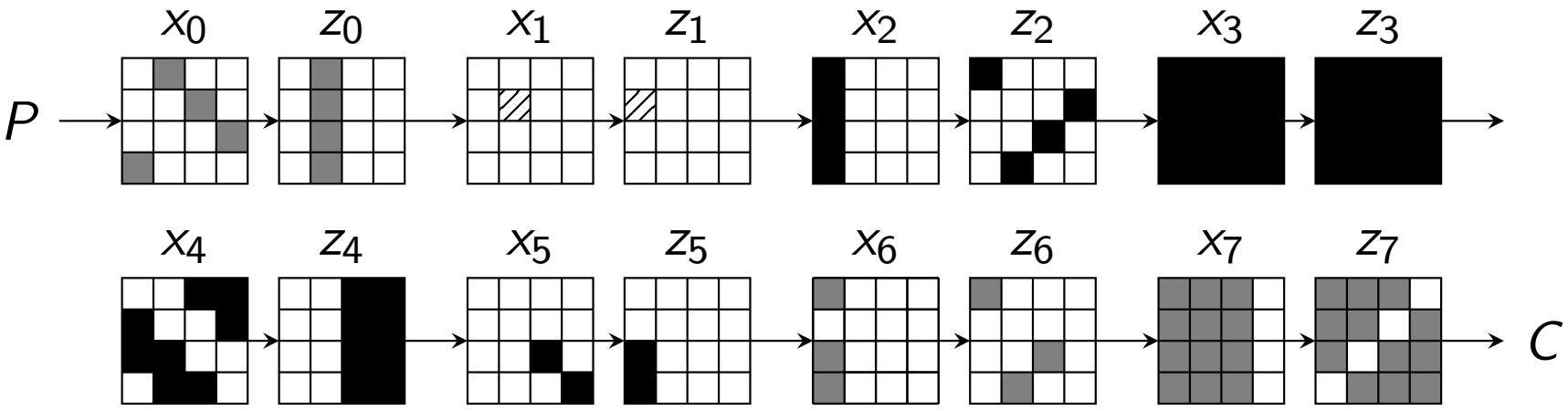
$\Longrightarrow$ Bytes of $\mathcal{B}_{off}$ can assume only 2^{128} values.
(instead of 2^{240})

New attack on 8 rounds



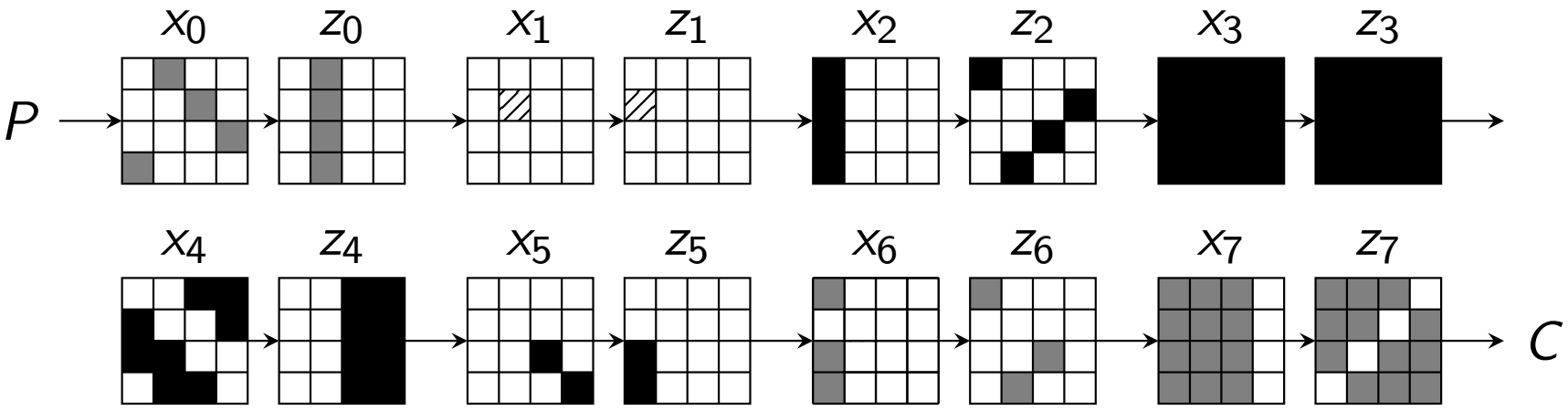
- In the online phase we now need to focus on finding such a pair.

New attack on 8 rounds



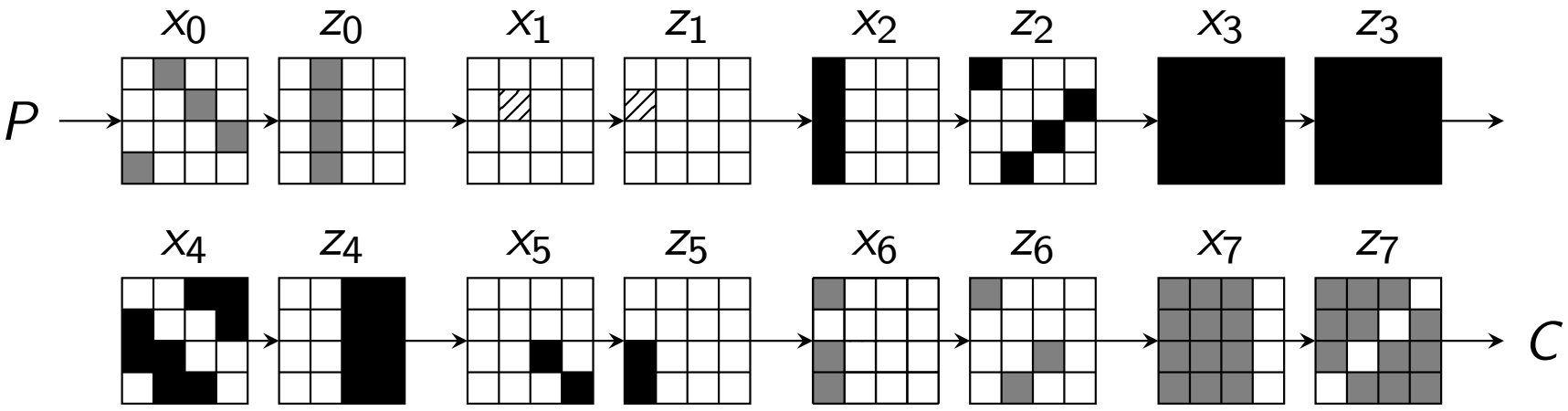
- Start by asking for a structure of 2^{32} plaintexts.

New attack on 8 rounds



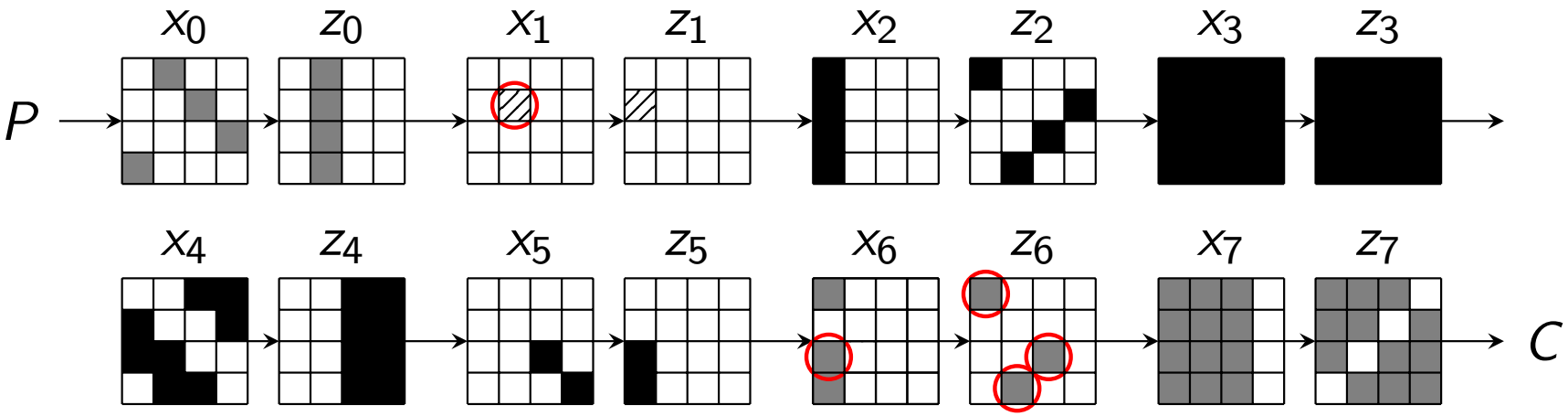
- Store the ciphertexts in a hash table in order to identify the pairs that have a non-zero probability to follow the differential path.

New attack on 8 rounds



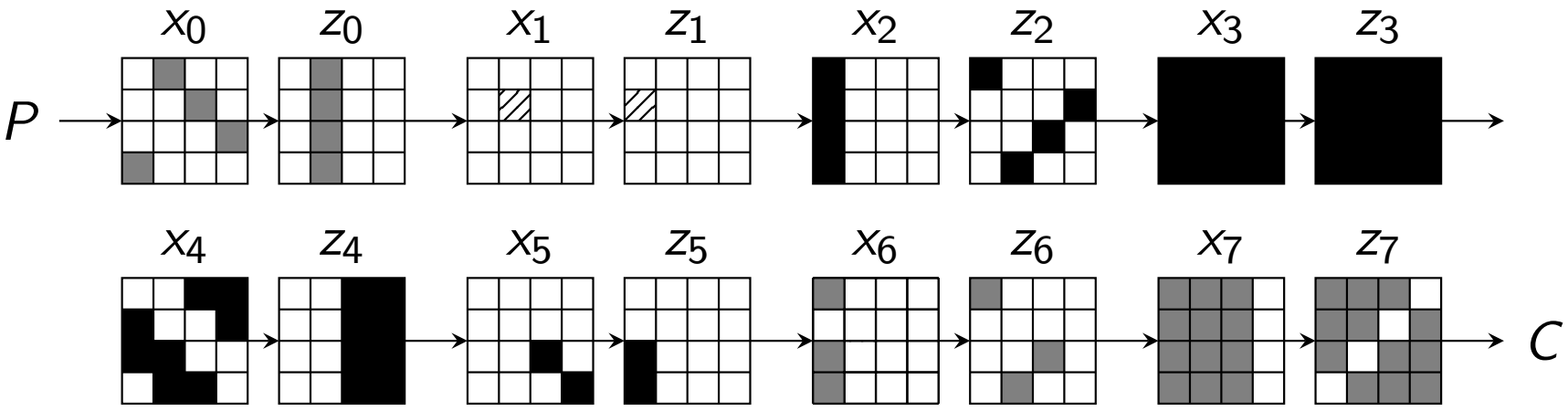
- Finds possible values of $\mathcal{B}_{on}$ for each of these pairs.

New attack on 8 rounds



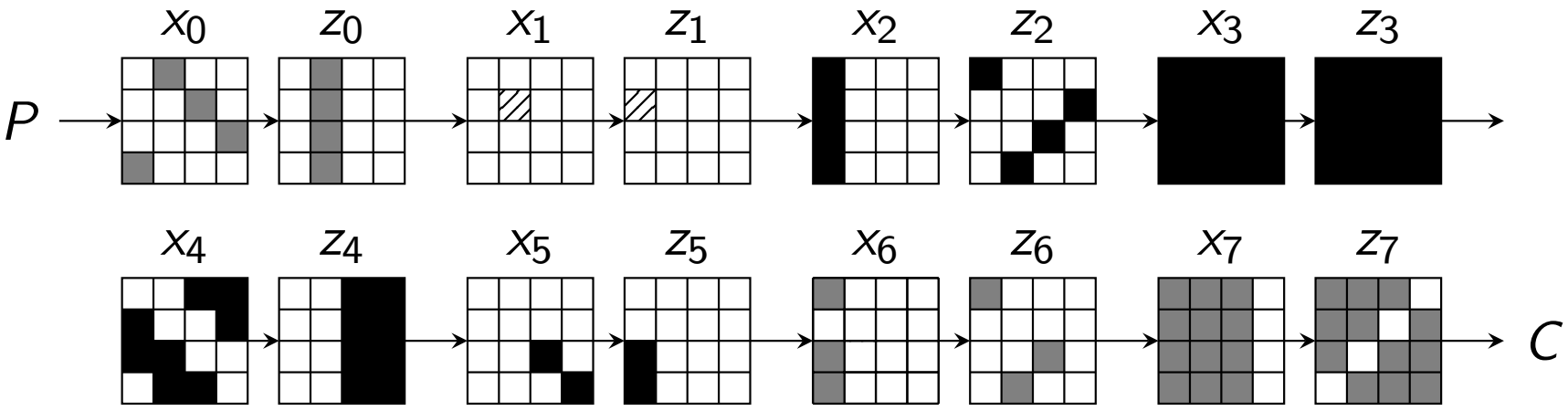
- Finds possible values of $\mathcal{B}_{on}$ for each of these pairs.
- Essentially by guessing the differences in circled bytes.

New attack on 8 rounds



- Finally identify the δ-set, compute the multiset and check if it belongs to the table.

New attack on 8 rounds



- Restart with a new structure until a match occurs.

Results

- New attacks on 8 rounds:

Version	Rounds	Data	Time	Memory
192	8	2^{113}	2^{140}	2^{130}
256	8	2^{113}	2^{156}	2^{130}

Results

- New attacks on 8 rounds:

Version	Rounds	Data	Time	Memory
192	8	2^{113}	2^{140}	2^{130}
256	8	2^{113}	2^{156}	2^{130}

- It is possible to perform many attacks in parallel to reduce the data complexity:

Version	Rounds	Data	Time	Memory
192	8	$2^{104.83}$	2^{140}	$2^{138.17}$
256	8	$2^{102.83}$	2^{156}	$2^{140.17}$

Results

- New attacks on 8 rounds:

Version	Rounds	Data	Time	Memory
192	8	2^{113}	2^{140}	2^{130}
256	8	2^{113}	2^{156}	2^{130}

- It is possible to perform many attacks in parallel to reduce the data complexity:

Version	Rounds	Data	Time	Memory
192	8	$2^{104.83}$	2^{140}	$2^{138.17}$
256	8	$2^{102.83}$	2^{156}	$2^{140.17}$

- Limitation: We only tried cases where bytes of $\mathcal{B}_{on}$ and $\mathcal{B}_{off}$ and active bytes of the differentials are *synchronized*.

Outline for section 4

Conclusion

- Generalization of Demirci-Selçuk attack.
- News attacks requiring at most 2^{32} chosen plaintexts.
- Best known attacks on 8 rounds for AES-192/AES-256.
- Results found in an automatic way.

Thanks

Thank you for your attention!