Related Key Differential Cryptanalysis of Midori

Using constraint programming

David Gerault Pascal Lafourcade

LIMOS, University Clermont Auvergne

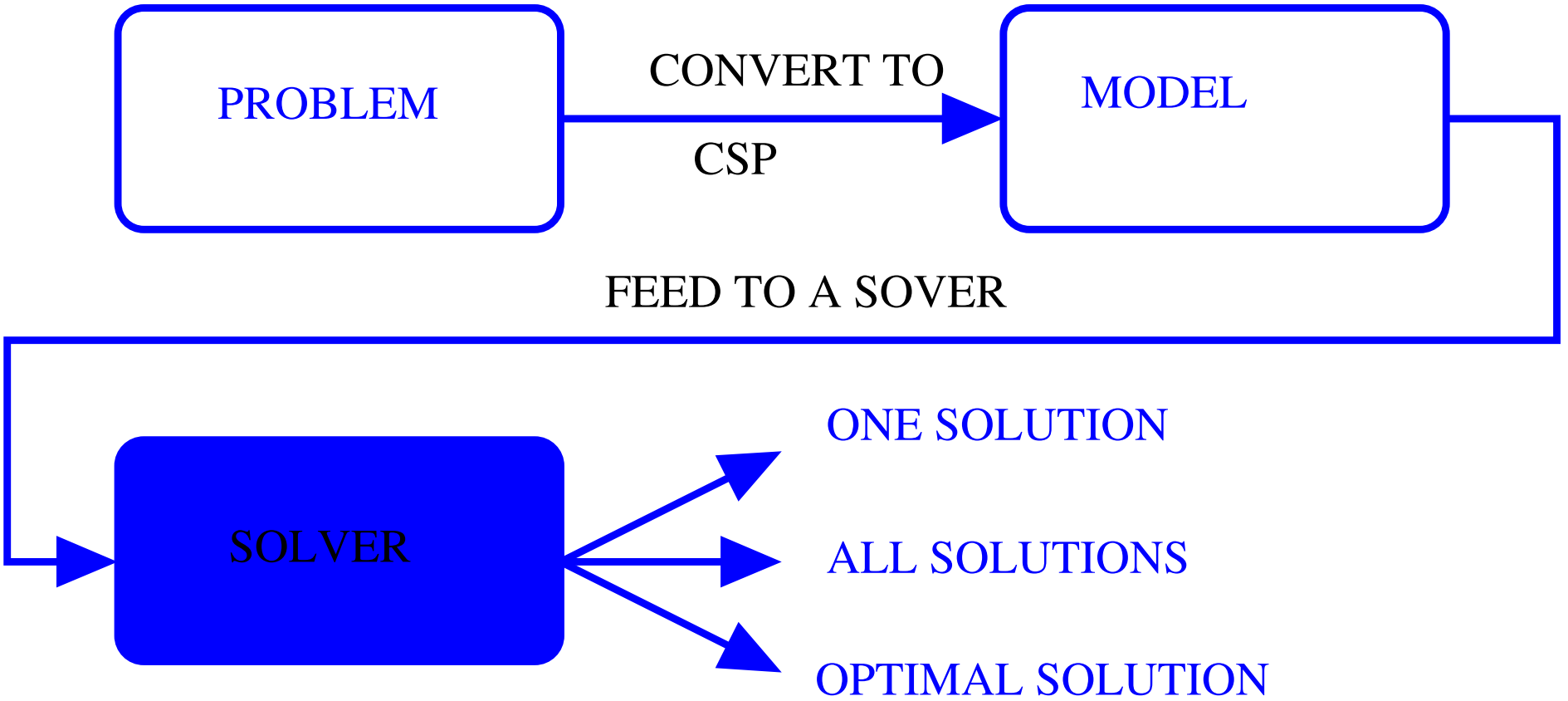
What we did, short version

In short : Automatic security evaluation of Midori in the related key model using constraint-programming.

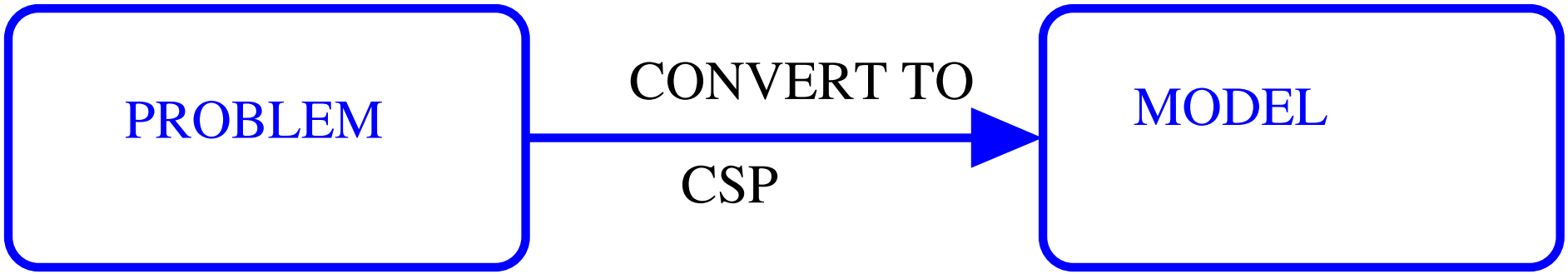
Constraint programming (CP)

Definition

"Constraint programming represents one of the closest approaches computer science has yet made to the holy grail of programming : the user states the problem, the computer solves it." (E. Freuder)

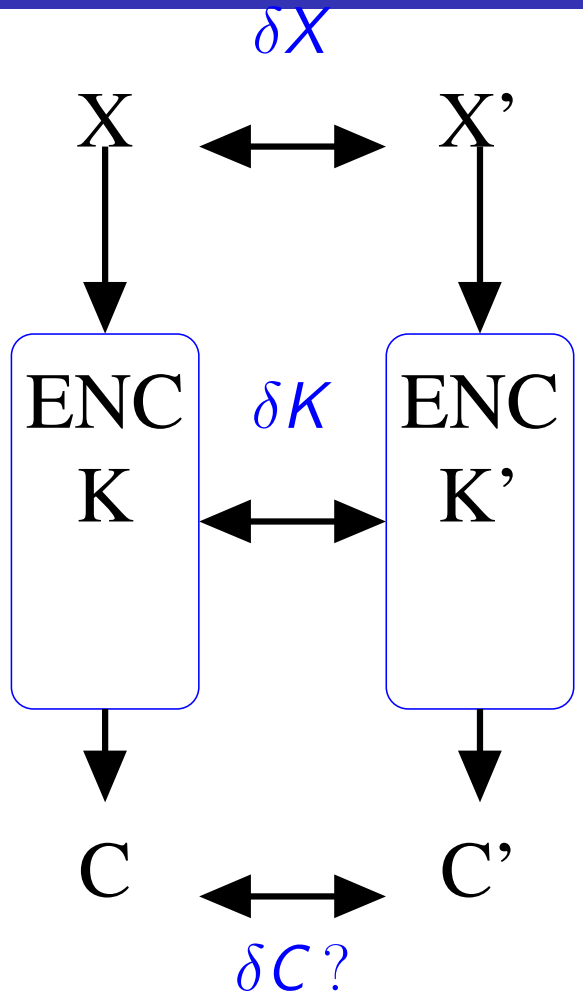


Modelling a problem as a CSP



- Define variables on given domains
  - [23..42] x
  - bool y
  - array [1..N,1..M] of floats δ ...
- Define *constraints*, *i.e.* relations between them
  - $x + y < 5$
  - $(a, b, c) \in \{(2, 3, 4), (1, 7, 2)\}$
  - Sums, products, alldifferent ...
- (optional) Define an objective function to optimize
  - Minimize(x+y)
  - Maximize(Sum(i in 1..N, j in 1..M) $\delta[i][j]$)
- Feed it to the solver, and let the magic happen...

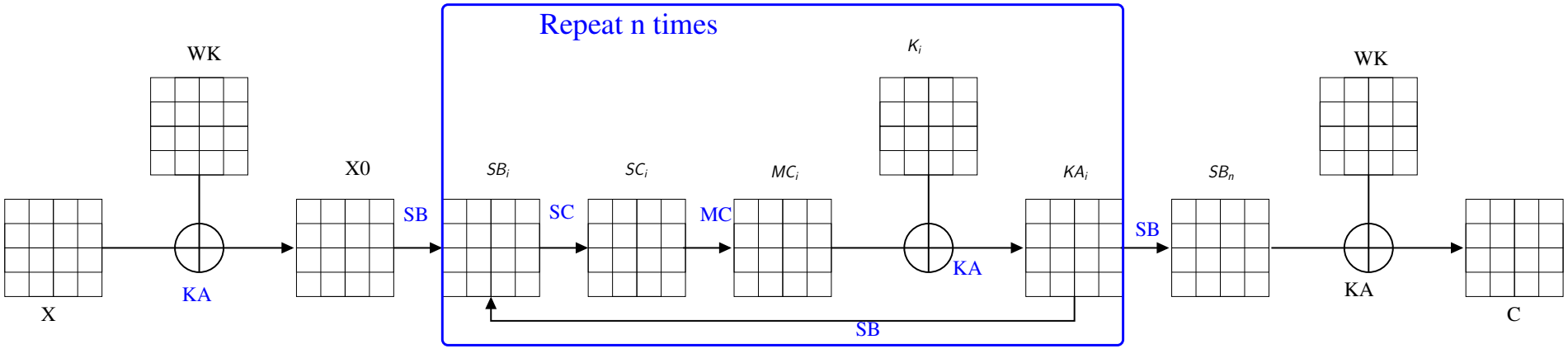
Related Key differential cryptanalysis



Aim

For given δX and δK, and random X and K, $Pr[(\delta X, \delta K) \rightarrow \delta C]$?
Related key differentials $\delta X, \delta K, \delta C$ such that $Pr[(\delta X, \delta K) \rightarrow \delta C]$ is maximal ?

Midori : a lightweight block cipher

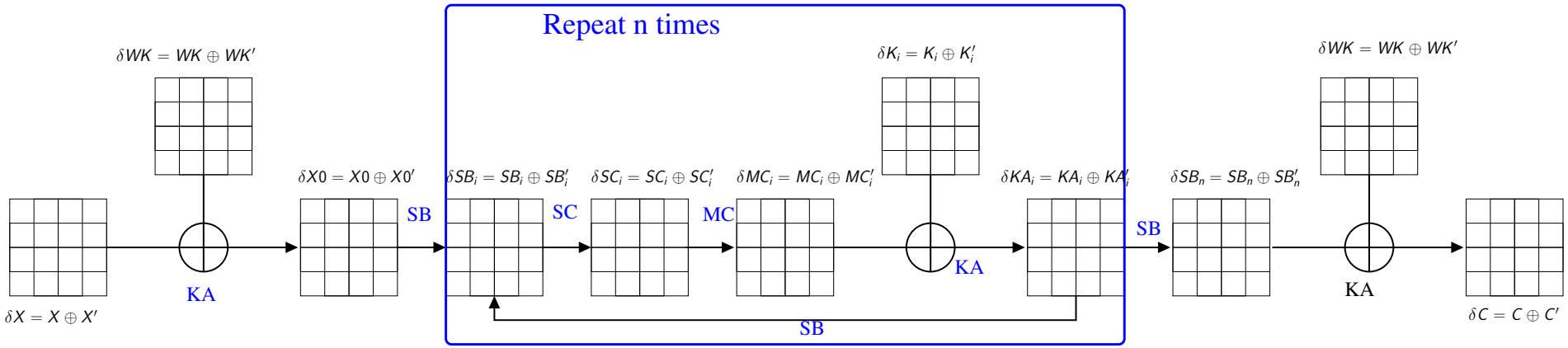


Midori (Banik et al., Asiacrypt'15)

128-bit key

Midori 64	Midori 128
X = 64 bits = 16 4-bit words	X = 128 bits = 16 bytes
$K = K0 \| \| K1$	$K_i = K \oplus cste_i$
$K_i = K(i \mod 2) \oplus cste_i$	WK=K
WK=$K0 \oplus K1$	20 rounds
16 rounds	

Propagation of XOR differences

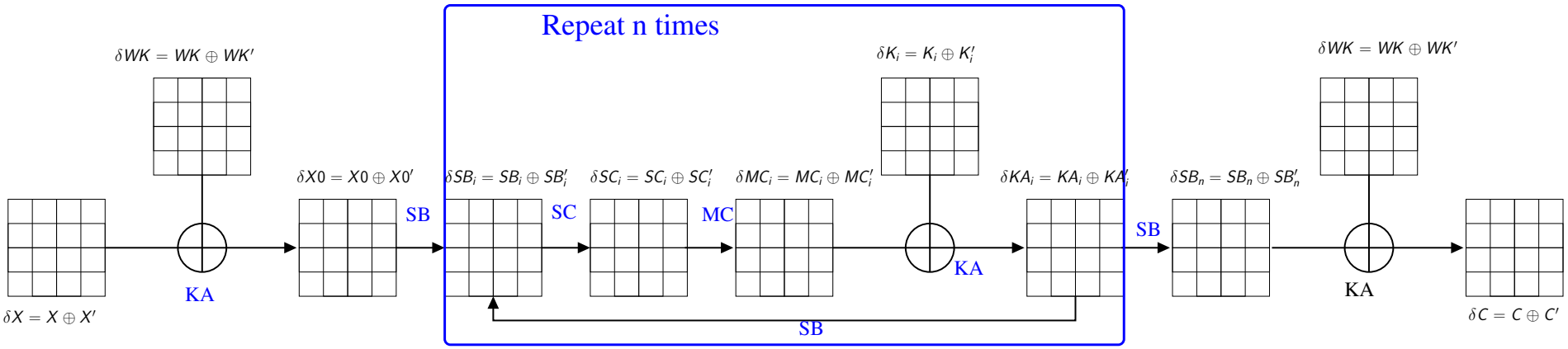


$$Op(a) \oplus Op(b) \ ?$$

- Linear Operators $L(\cdot)$
  - $L(a) \oplus L(b) = L(a \oplus b)$ holds with probability 1
- Non-linear operator : SB
  - $SB(a) \oplus SB(b) \neq SB(a \oplus b)$
  - Difference propagation depends on the values of a and b
  - Probabilistic propagation : $Pr[SB(a) \oplus SB(b) = \delta_{out} | a \oplus b = \delta_{in}]$ (easy to compute)
  - However, $a \oplus b = 0 \equiv SB(a) \oplus SB(b) = 0$
  - Similarly, $a \oplus b \neq 0 \equiv SB(a) \oplus SB(b) \neq 0$

We want to minimize the number of *active* Sboxes

2-Step Solving process



Step 1

- Abstract words to booleans
- $\delta X[j][k] = 0$
  $\Rightarrow \Delta X[j][k] = 0$ (false)
- $\delta X[j][k] \in [1, 255]$
  $\Rightarrow \Delta X[j][k] = 1$ (true)
- Some solutions are not consistent

Step 2

- Concretize booleans to words
- $\Delta X[j][k] = 0 \Rightarrow \delta X[j][k] = 0$
- $\Delta X[j][k] = 1 \Rightarrow$ Find
  $\delta X[j][k] \in [1, W]$

Related Work & Contributions : Automatic search

Automatic Related-Key security analysis

Searching for optimal related key differential characteristics for word oriented block ciphers

Previous Work

- Specialized algorithm : Biryukov et al., EUROCRYPT 2010 Step 1
- MILP : Mouha et al., ISC 2012 Step 1
- CP : Gerault et al., CP 2016 Steps 1 and 2

Related Work & Contributions : Automatic search

Automatic Related-Key security analysis

Searching for optimal related key differential characteristics for word oriented block ciphers

Previous Work

- Specialized algorithm : Biryukov et al., EUROCRYPT 2010 Step 1
- MILP : Mouha et al., ISC 2012 Step 1
- CP : Gerault et al., CP 2016 Steps 1 and 2

Our contribution

- Models for Midori 64 and 128
- Step 1 in MiniZinc
- Step 2 in Choco
- All optimal related key differential characteristics obtained within 10 hours !

Related Work & Contributions : Cryptanalysis of Midori

Cryptanalysis

Finding attacks on Midori

Type	Rounds	Data	Time	Reference
Midori64				
Impossible differential	10	$2^{62,4}$	$2^{80,81}$	Chen et al., 2016
Meet-in-the-middle	12	$2^{55.5}$	$2^{125,5}$	Lin et al., 2015
Invariant subspace (for one key in 2^{96})	full(16)	2	2^{16}	Guo et al., 2015
Related-key differential	14	2^{59}	2^{116}	Dong, 2016
Related-key differential	**full(16)**	$2^{23.75}$	$2^{35.8}$	This work
Midori128				
Related-key differential	**full(20)**	$2^{47.7}$	$2^{47.7}$	This work

Related Work & Contributions : Cryptanalysis of Midori

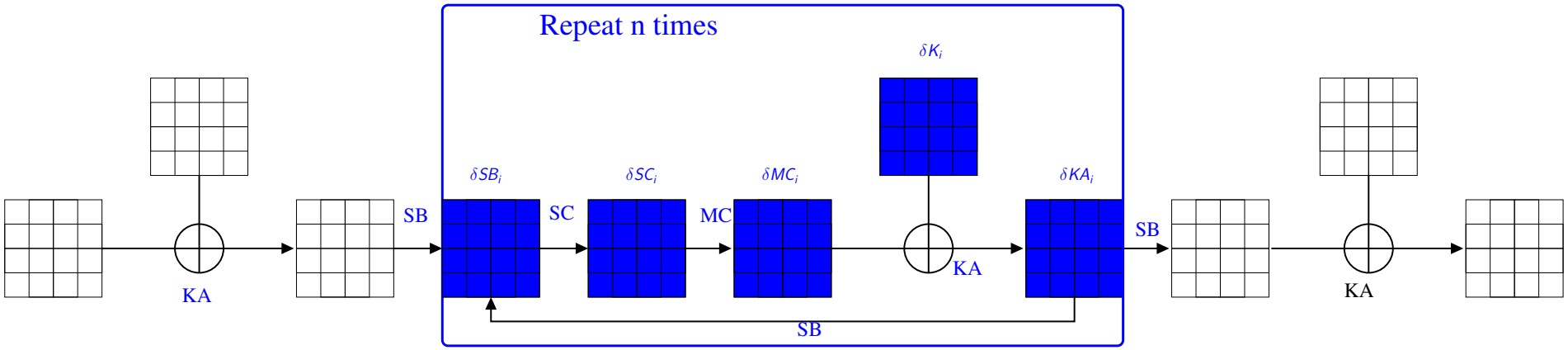
Cryptanalysis

Finding attacks on Midori

Type	Rounds	Data	Time	Reference
Midori64				
Impossible differential	10	$2^{62,4}$	$2^{80,81}$	Chen et al., 2016
Meet-in-the-middle	12	$2^{55.5}$	$2^{125,5}$	Lin et al., 2015
Invariant subspace (for one key in 2^{96})	full(16)	2	2^{16}	Guo et al., 2015
Related-key differential	14	2^{59}	2^{116}	Dong, 2016
Related-key differential	**full(16)**	$2^{23.75}$	$2^{35.8}$	This work
Midori128				
Related-key differential	**full(20)**	$2^{47.7}$	$2^{47.7}$	This work

New practical attacks on both versions of Midori !

CP Model



Variables

Step 1 : One boolean Δ for each word of the state

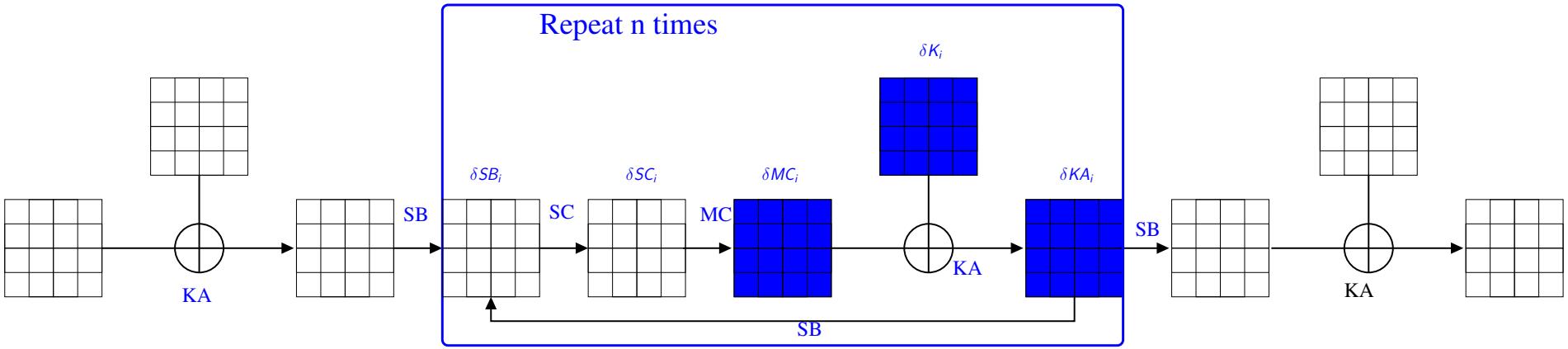
Step 2 : One word δ for each word of the state, a probability P for each SB, and the output from Step 1

Objective function

Step 1 : Minimize $\sum_{i=1}^{n} \sum_{j,k=0}^{3} \Delta SB_i[j][k]$

Step2 : Maximize $\sum_{i=1}^{n} \sum_{j,k=0}^{3} P_i[j][k]$

KA Constraint




```
constraint forall(r in 0..n-1,j in 0..3, i in 0..3) (
        XOR(DMC[r,i,j], DK[r mod 2,i,j], DKA[r,i,j])
);
```

Definition of the KA constraint

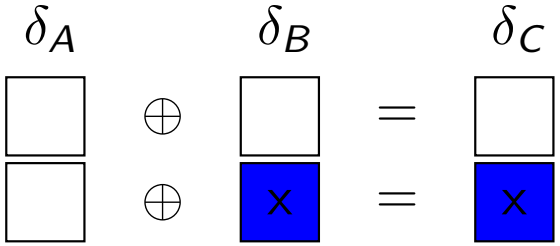
Step 1 : $XOR(\Delta MC_i[j][k], \Delta K_i[j][k], \Delta KA_i[j][k])$

Step 2 : $XOR(\delta MC_i[j][k], \delta K_i[j][k], \delta KA_i[j][k])$

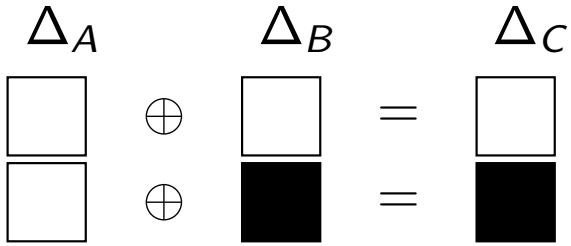
XOR Constraint : Step 1

(white $= 0$, colored $\neq 0$)

Word values



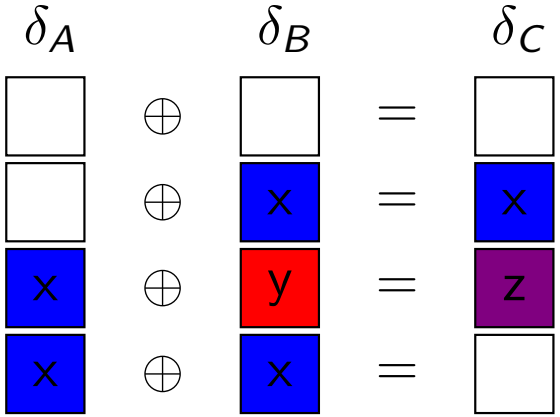
Boolean abstraction



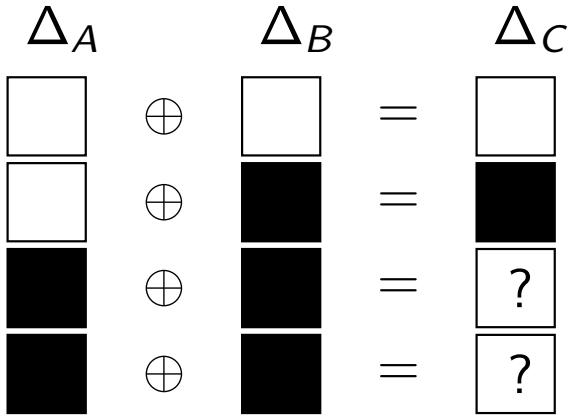
XOR Constraint : Step 1

(white $= 0$, colored $\neq 0$)

Word values



Boolean abstraction



Δ_A	Δ_B	Δ_C
0	0	0
0	1	1
1	0	1
1	1	?

```
predicate XOR(var 0..1: a,var 0..1: b,var 0..1: res) =
      a+b+res!=1
;

predicate XOR3(var 0..1: a,var 0..1: b,var 0..1: c, var 0..1: res) =
      a+b+c+res!=1
```

XOR constraint : Step 2

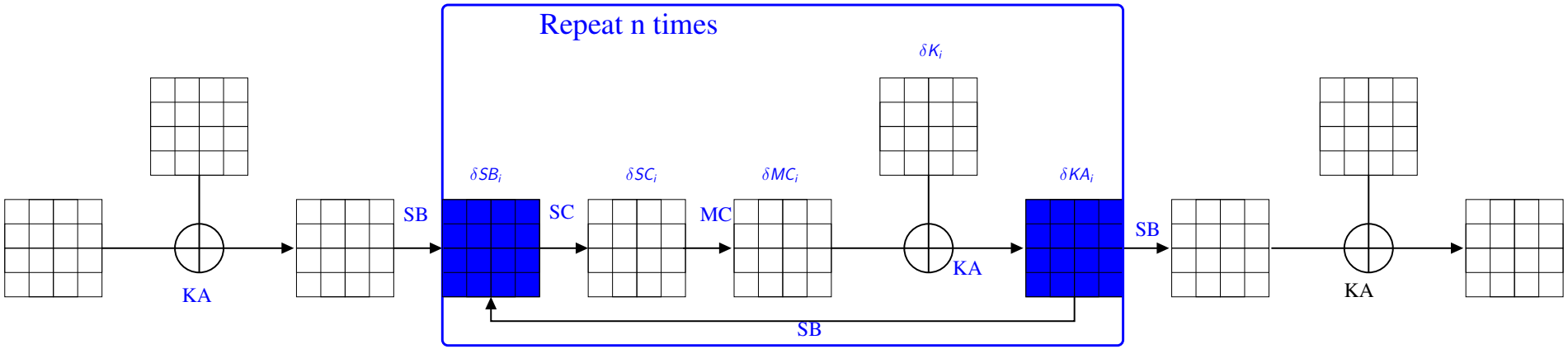
$\mathcal{XOR}$ Table

A	**B**	$A \oplus B$
0	0	0
0	1	1
0	2	2
	...	
	...	
	...	
255	253	2
255	254	1
255	255	0

Definition of the XOR constraint

$(\delta A, \delta B, \delta C) \in \mathcal{XOR}$

S-Box : Step 1

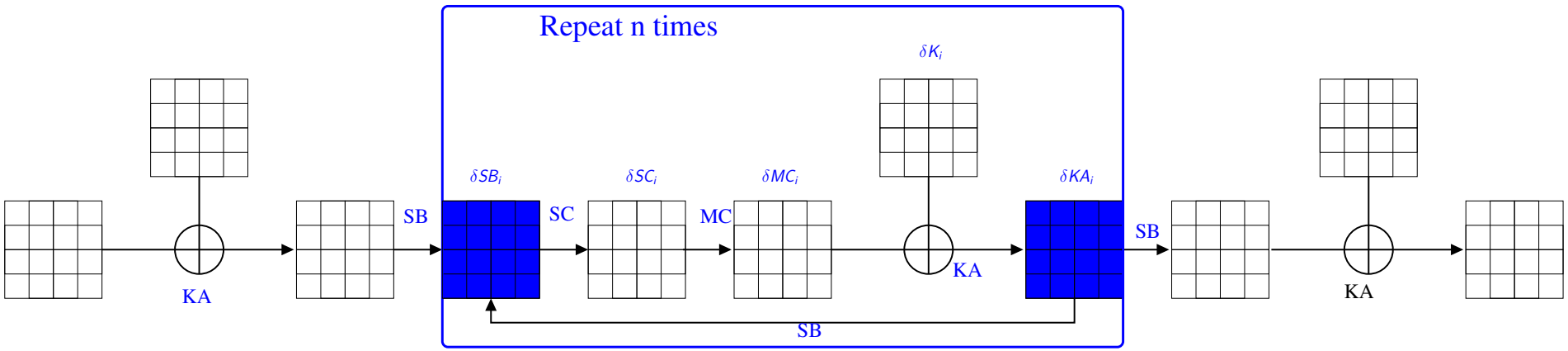


A	B	$\Delta_{A,B} = A \oplus B$	$S(A) == S(B)?$	$\Delta_{SB(A),SB(B)} = SB(A) \oplus SB(B)$
x	x	0	true	0
x	y	1	false	1

Good news !

No effect ! Bijective S-Boxes do not introduce nor remove differences.

S-box constraint : Step 2



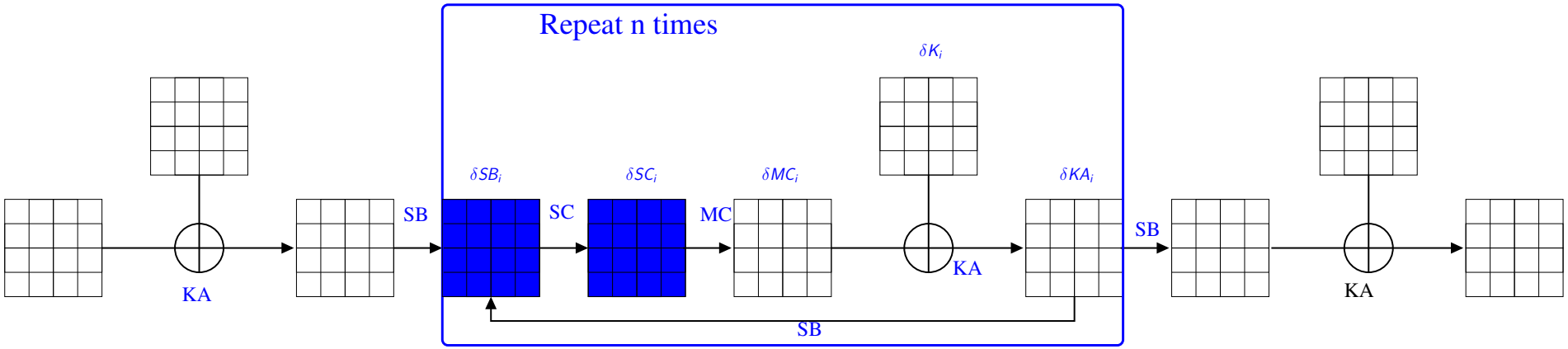
$\mathcal{DDT}$ Table

δ_{IN}	δ_{OUT}	$\Pr[\delta_{IN} \rightarrow \delta_{OUT}]$
0	0	1
1	1	0.125
1	2	0.25
	...	
f	f	0.125

Definition of the SB constraint

$(\delta KA_i[j][k], \delta SB_{i+1}[j][k], P_i[j][k]) \in \mathcal{DDT}$

SC



$$\begin{pmatrix} s0 & s4 & s8 & s12 \\ s1 & s5 & s9 & s13 \\ s2 & s6 & s10 & s14 \\ s3 & s7 & s11 & s15 \end{pmatrix} \rightarrow \begin{pmatrix} s0 & s5 & s15 & s10 \\ s7 & s2 & s8 & s13 \\ s14 & s11 & s1 & s4 \\ s9 & s12 & s6 & s3 \end{pmatrix}$$

Definition of the SC constraint

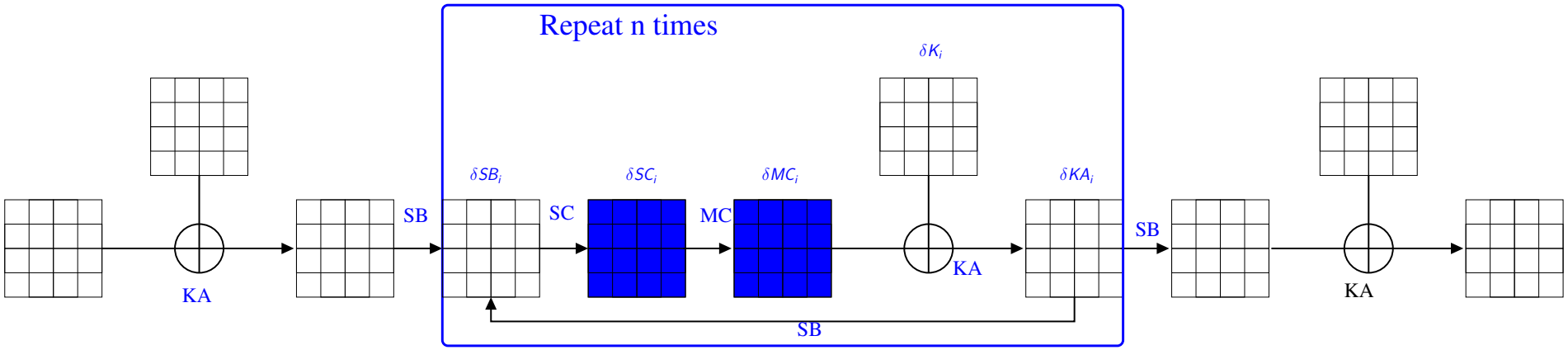
Step 1 :

$\Delta SC_i[0][0] = \Delta SB_i[0][0], \ldots, \Delta SC_i[3][3] = \Delta SB_i[3][0]$

Step 2 :

$\delta SC_i[0][0] = \delta SB_i[0][0], \ldots, \delta SC_i[3][3] = \delta SB_i[3][0]$

MC : Step 1

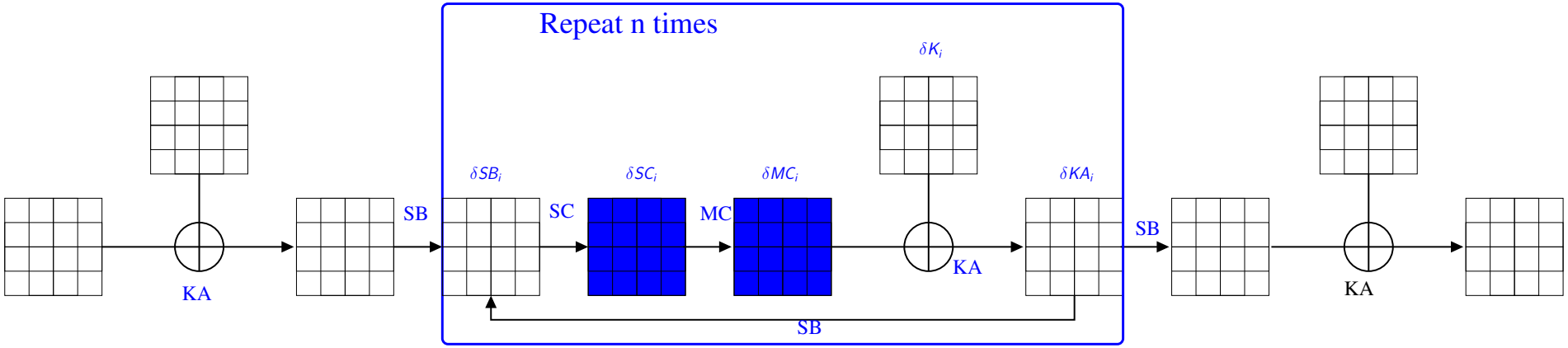


$$\mathrm{SC} \cdot \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} = \mathrm{MC}$$

```
constraint forall(r in 0..n-1, j in 0..3) (
      XOR3(DSC[r,1,j],DSC[r,2,j],DSC[r,3,j],DMC[r,0,j]) /\
      XOR3(DSC[r,0,j],DSC[r,2,j],DSC[r,3,j],DMC[r,1,j]) /\
      XOR3(DSC[r,0,j],DSC[r,1,j],DSC[r,3,j],DMC[r,2,j]) /\
      XOR3(DSC[r,0,j],DSC[r,1,j],DSC[r,2,j],DMC[r,3,j])
);

constraint forall(r in 0..n-1, j in 0..3) (
  sum(i in 0..3)(DSC[r,i,j]+DMC[r,i,j]) in {0,4,5,6,7,8}
```

MC : Step 2

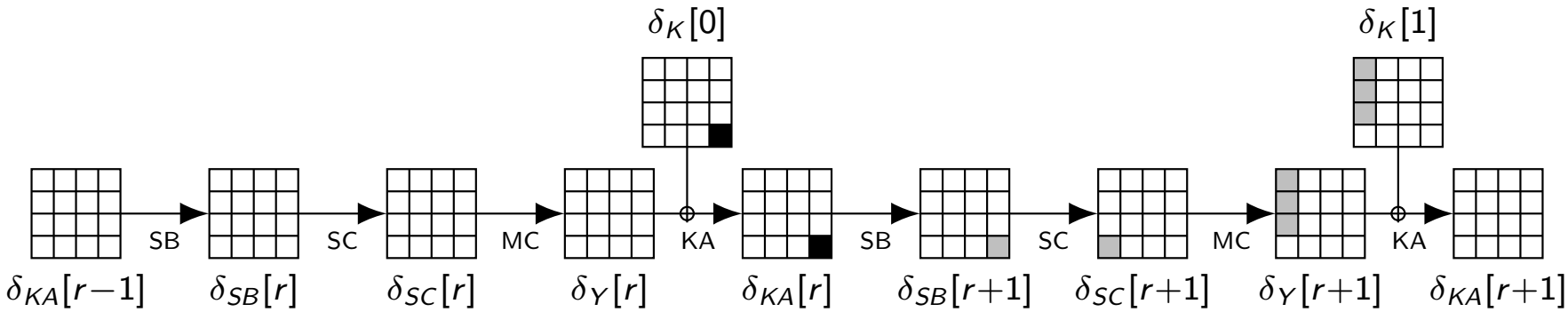


$$\text{SC} \cdot \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} = \text{MC}$$

Definition of the MC constraint

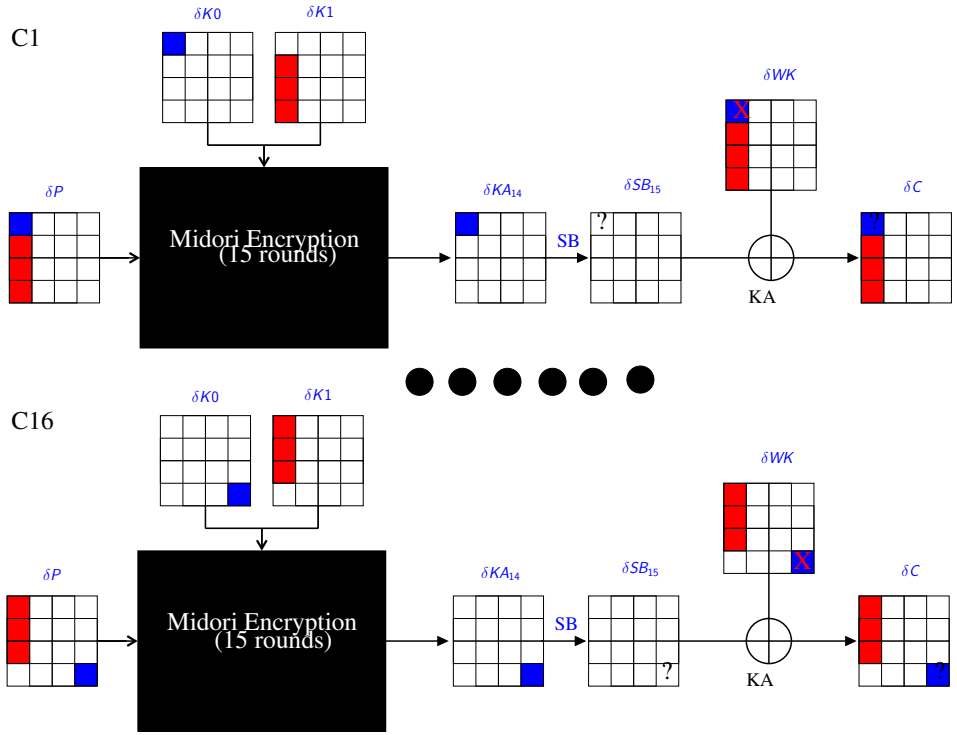
$\text{XOR}(\delta SC_i[1,k], \delta SC_i[2,k], \delta SC_i[3,k], \delta MC_i[0,k])$
$\text{XOR}(\delta SC_i[0,k], \delta SC_i[2,k], \delta SC_i[3,k], \delta MC_i[1,k])$
$\text{XOR}(\delta SC_i[0,k], \delta SC_i[1,k], \delta SC_i[3,k], \delta MC_i[2,k])$
$\text{XOR}(\delta SC_i[0,k], \delta SC_i[1,k], \delta SC_i[2,k], \delta MC_i[3,k])$

Results : Midori64



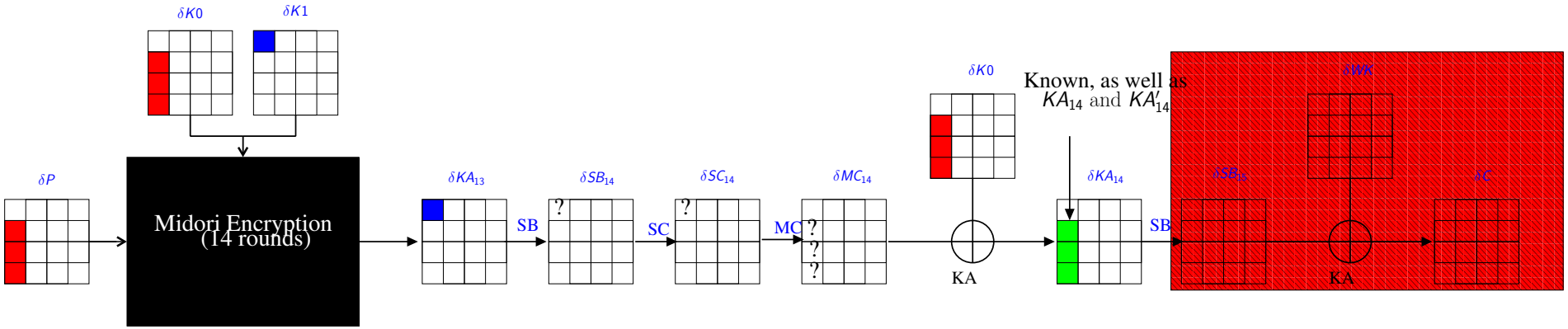
- Full round distinguisher (16)
  - Step 1 : 1 active Sbox per 2 rounds -> 8 total
  - Step 2 : 2^{-2} for each Sbox -> 2^{-16} total
- 15 rounds
  - Step 1 : 1 active Sbox per 2 rounds -> 7 total
  - Step 2 : 2^{-2} for each Sbox -> 2^{-14} total

Key recovery : Midori64



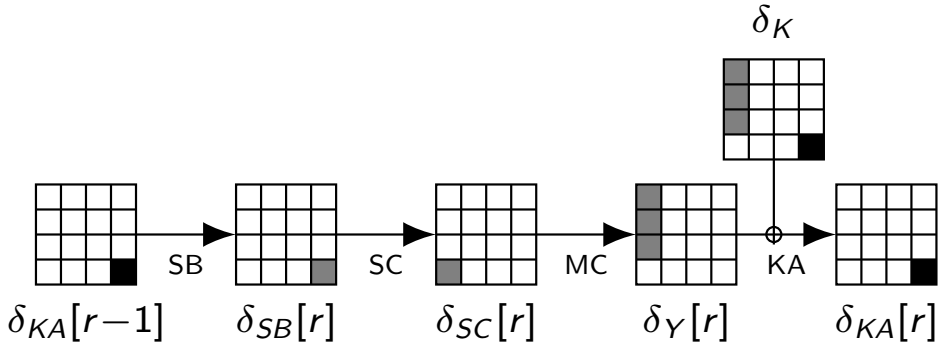
- 1 WK word per 15-round RK differential characteristic
- Recovery of one WK word in $\approx 2^{19}$ operations
- Recovery of WK in $\approx 2^{23}$ operations
- But WK alone is useless...

Key recovery : Midori64, part 2



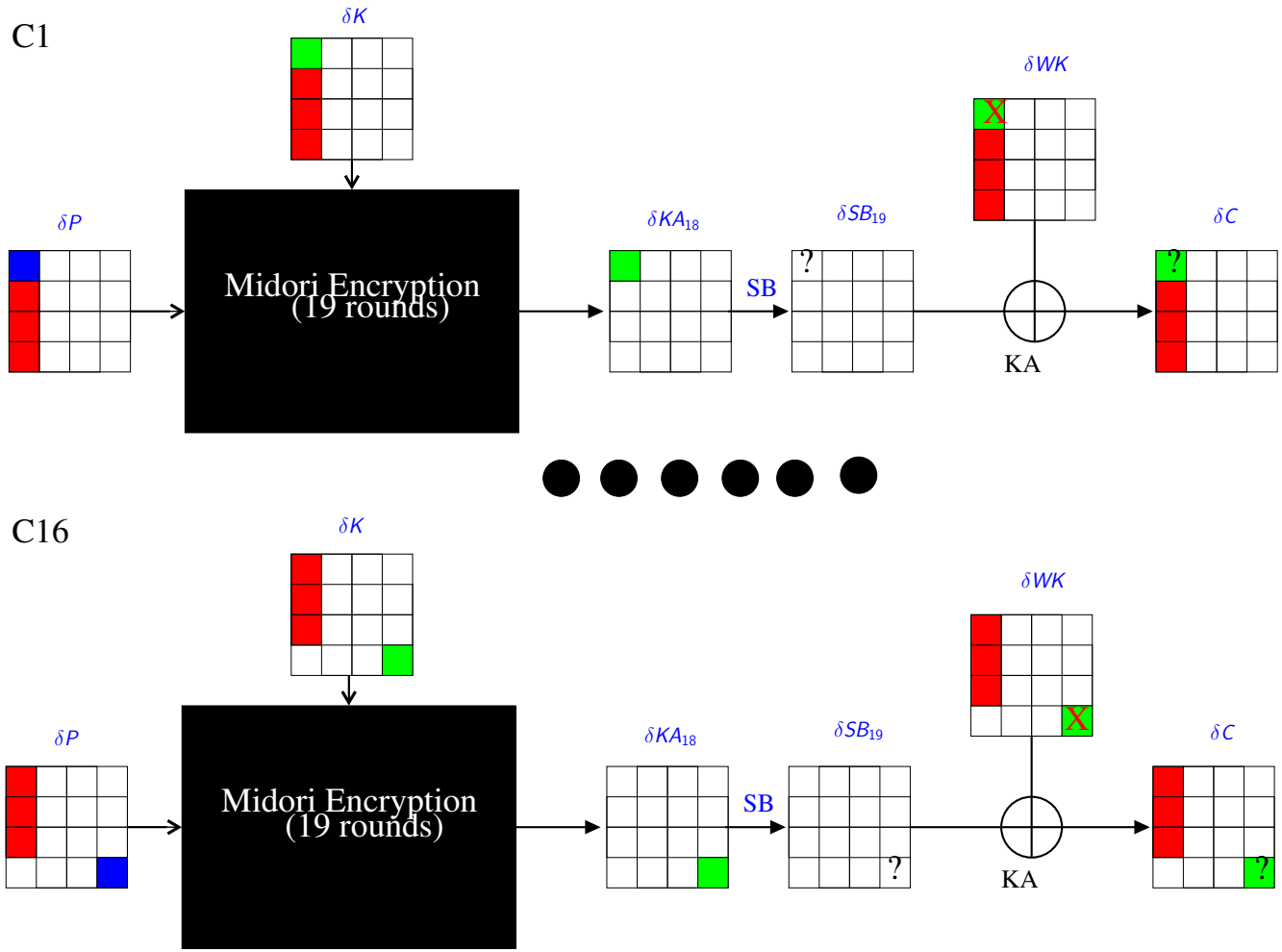
- Decipher the last round with WK
- Use a 14-round RK differential characteristic to get candidates for a word of SB_{14}
- Guess the other 3 words of the corresponding column of SC_{13}
- Obtain a candidate for a column of MC_{13}
- Recover candidates for a column of $K0$
- Reapeat for each column (with a different differential)

Results : Midori128



- Full round distinguisher (20)
  - Step 1 : 1 active Sbox per round -> 20 total
  - Step 2 : 2^{-2} for each Sbox -> 2^{-40} total
- 19 rounds
  - Step 1 : 1 active Sbox per round -> 19 total
  - Step 2 : 2^{-2} for each Sbox -> 2^{-38} total

Key recovery : Midori128



- 1 WK word per RK differential characteristic
- Recovery of one WK word in $\approx 2^{43}$ operations
- Recovery of WK in $\approx 2^{47}$ operations
- Here, K = WK => we are done !

Conclusion and future work

Conclusion

- CP is useful
- Midori should be used with care

Future work

- Apply the same method to other ciphers
- Find better attacks in the single key setting
- Relate with invariant subspace attacks

Thanks your attention !

Questions ?