On Impossible Differential Cryptanalysis

María Naya-Plasencia
INRIA, France

Clervaux, ESC 2015

Outline

- Introduction and Survey
- Generic Complexities
- Generic Improvements
- Examples

Outline

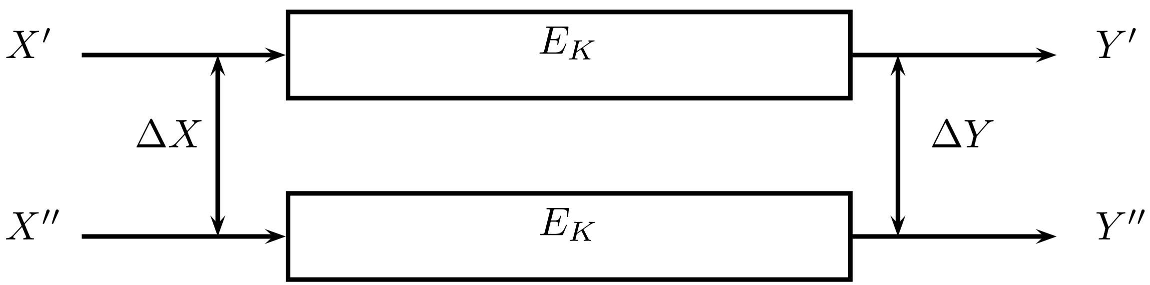
- "Scrutinizing and Improving Impossible Differential Attacks: Applications to CLEFIA, Camellia, LBlock and Simon", Asiacrypt 2014, joint work with Christina Boura and Valentin Suder.

- New Results on Impossible Differential Cryptanalysis of SPN Ciphers , joint work with Christina Boura, Virginie Lallemand and Valentin Suder.

Impossible Differential Attacks

Classical Differential Attacks [BS'90]

Given an input difference between two plaintexts, some output differences occur more often than others.

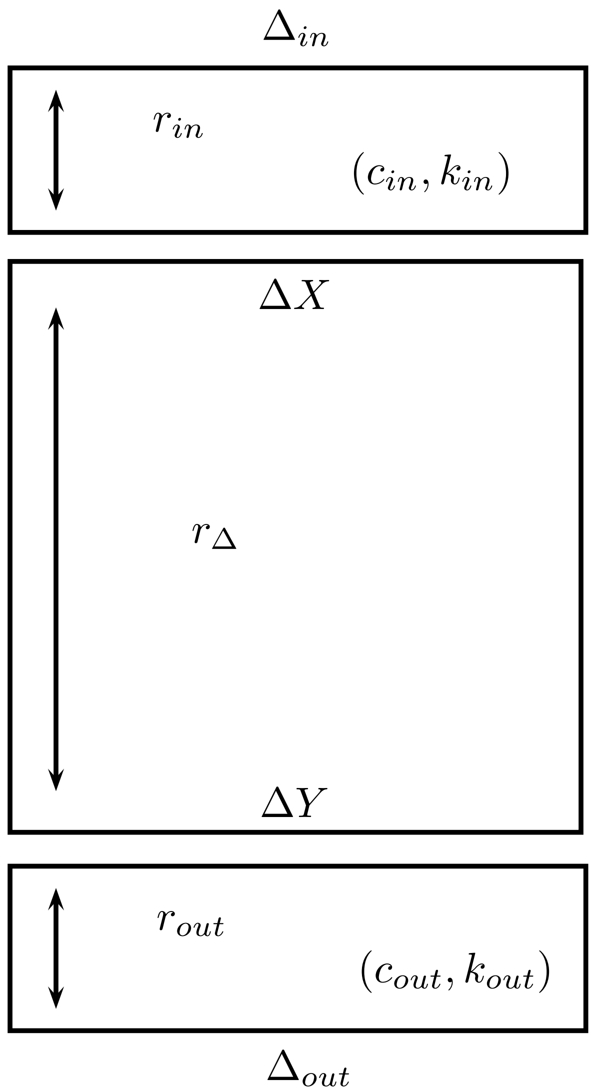


A differential is a pair (Δ_X, Δ_Y).

Impossible Differential Attacks [K,BBS'98]

- Impossible differential attacks use a differential with probability 0.
- We can find the impossible differential using the Miss-in-the-middle [BBS'98] technique.
- Extend it backward and forward $\Rightarrow$ Active Sboxes transitions give information on the involved key bits.
- Generic framework,improvements[BNPS14,BLNPS15]

Impossible Differential Attack



Discarding Wrong Keys

- Given one pair of inputs with Δ_{in} that produces Δ_{out},

- all the (partial) keys that produce ΔX from Δ_{in} and ΔY from Δ_{out} differ from the correct one.

- If we consider N pairs verifying $(\Delta_{in}, \Delta_{out})$ the probability of NOT discarding a candidat key is

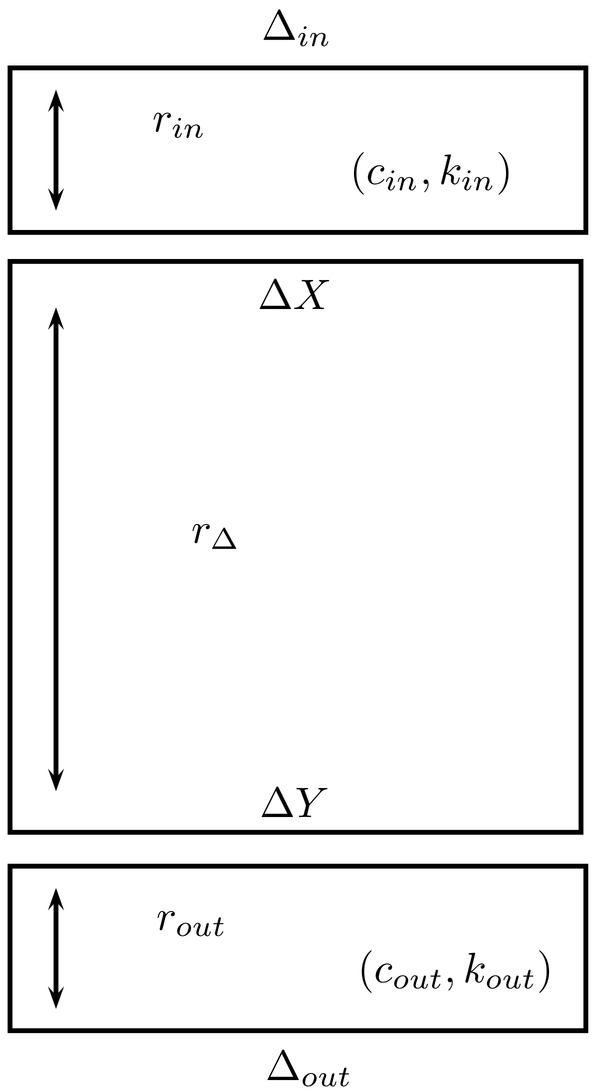
$$(1 - 2^{-c_{in} - c_{out}})^N$$

Some previous attacks

Algorithm	# rounds	Type of error	Gravity of error
CLEFIA-128 (without whit.)	14	data complexity higher than codebook	attack does not work
CLEFIA-128	13	cannot be verified without implementation	-
Camellia (without FL/FL^{-1})	12	big flaw in computation	attack does not work
Camellia-128	12	big flaw in computation	attack does not work
Camellia-128/192/256 (without FL/FL^{-1})	11/13/14	small complexity flaws	corrected attacks work
LBlock	22	small complexity flaw	corrected attack works
SIMON (all versions)	14/15/16/ 19/22	data complexity higher than codebook	attacks do not work
SIMON (all versions)	13/15/17 20/25	big flaw in computation	attacks do not work

Generic Complexities

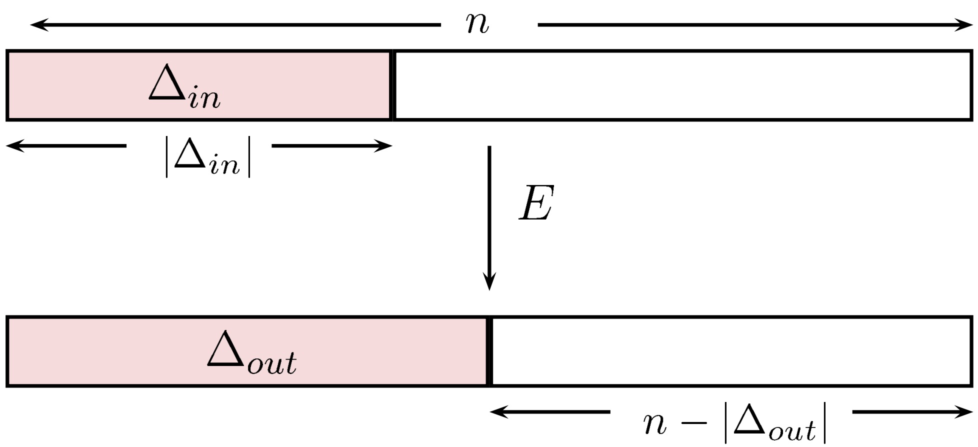
Impossible Differential Attack



Data Complexity

$P = (1 - 2^{-(c_{in}+c_{out})})^N < \frac{1}{2}$ (and not $\frac{1}{2^{|k_{in} \cup k_{out}|}}$).

- Finding N pairs verifying a given truncated differential.



Data Complexity

The limited birthday problem [GP10] solves it for 1 pair.

- For obtaining N pairs:

$$C_N = \max\left\{ \min_{\Delta \in \{\Delta_{in}, \Delta_{out}\}} \left\{ \sqrt{N 2^{n+1-|\Delta|}} \right\}, N 2^{n+1-|\Delta_{in}|-|\Delta_{out}|} \right\}.$$

For the attack to work, we need that $C_N < 2^n$, where n is the size of the state.

Time Complexity

$$C_N C_E + 2^{|k_{in} \cup k_{out}|} \frac{N}{2^{c_{in}+c_{out}}} C'_E + P 2^{|K|} C_E < 2^{|K|} C_E$$

where C_N is the data needed for obtaining N pairs $(\Delta_{in}, \Delta_{out})$, $\frac{N}{2^{c_{in}+c_{out}}} C'_E$ is the average cost of testing the pairs per candidate key (early abort technique [LKKD08]) and P is the probability of not discarding a candidate key.

Revisiting Time complexity

- Considering the key-schedule as a black box, we have to add one term:

$$\min(2^{K-k_{in}}, 2^{K-k_{out}}) \cdot P \cdot 2^{|k_{in} \cup k_{out}|} \cdot C_{KS},$$

- As now $|k_{in} \cup k_{out}| = k_{in} + k_{out}$, we have

$$\min(2^{K+k_{out}}, 2^{K+k_{in}}) \cdot P \cdot C_{KS}.$$

- Multiplying by $\max(2^{-k_{out}}, 2^{-k_{in}})$, we recover the $P2^K$ keys to test (3rd term).

Memory complexity

- The memory complexity is $\min\{N, 2^{|K_{in} \cup K_{out}|}\}$.
- Typically, N is smaller.

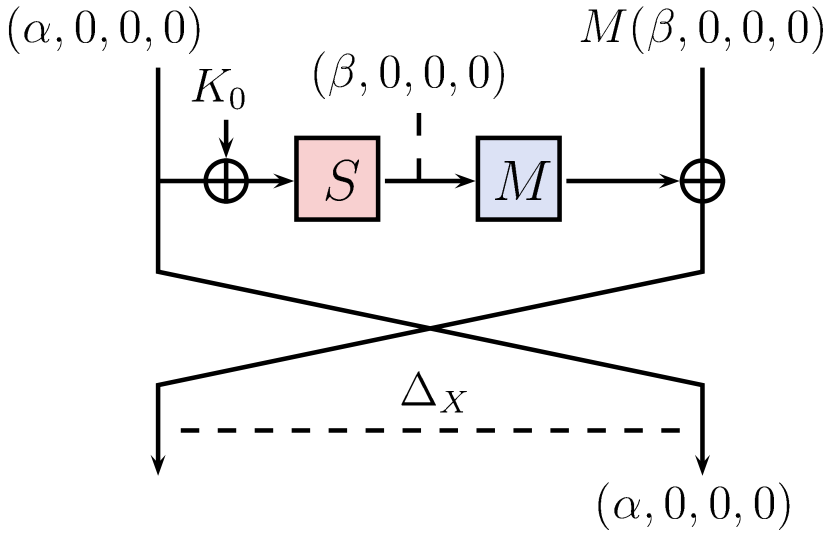
Generic Improvements

Improvements from [BN-PS14]

- Correctly choosing Δ_{in} and Δ_{out} (related to [MRST09])
- Multiple impossible differentials (related to [JN-PP13])
- State-test technique (related to [MRST09])

Choosing Δ_{in} and Δ_{out}

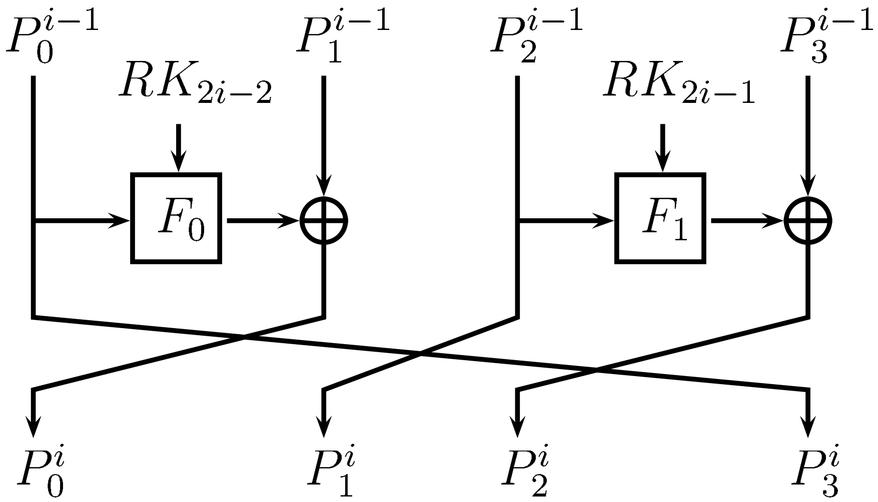
or how to match the time complexity estimation:



$|\Delta_{in}| = 8$ and $c_{in} = 4$ OR $|\Delta_{in}| = 7$ and $c_{in} = 3$

Example: CLEFIA-128

- block size: $4 \times 32 = 128$ bits
- key size: 128 bits
- $\#$ of rounds: 18



Multiple Impossible Differentials

Formalize the idea of [Tsunoo et al. 08]:

CLEFIA has two 9-round impossible differentials $((0,0,0,A) \not\rightarrow (0,0,0,B))$ and $((0,A,0,0) \not\rightarrow (0,B,0,0))$ when A and B verify:

A	B				
$(0,0,0,\alpha)$	$(0,0,\beta,0)$	or	$(0,\beta,0,0)$	or	$(\beta,0,0,0)$
$(0,0,\alpha,0)$	$(0,0,0,\beta)$	or	$(0,\beta,0,0)$	or	$(\beta,0,0,0)$
$(0,\alpha,0,0)$	$(0,0,0,\beta)$	or	$(0,0,\beta,0)$	or	$(\beta,0,0,0)$
$(\alpha,0,0,0)$	$(0,0,0,\beta)$	or	$(0,0,\beta,0)$	or	$(0,\beta,0,0)$

24 in total: $C_N = 2^{113}$ becomes $C_N = 2^{113}/24$

Revisiting Multiple Impossible Differentials

- It seemed difficult to combine with other techniques. New vision:

Could be seen as the application in parallel of independent attacks with the same parametters but (possibly) different involved keybits. When considering multiples, we need less data!

New Idea: Multiple Differentials

We now consider not only multiple impossible differentials, but also multiple differentials:

- for a fixed pair (Δ_X, Δ_Y), different transitions to m_{in} Δ_{in} and m_{out} Δ_{out}.

$$C'_N = \frac{C_N}{n_{in}n_{out}m_{in}m_{out}} = \frac{C_N}{M}.$$

State Test Technique

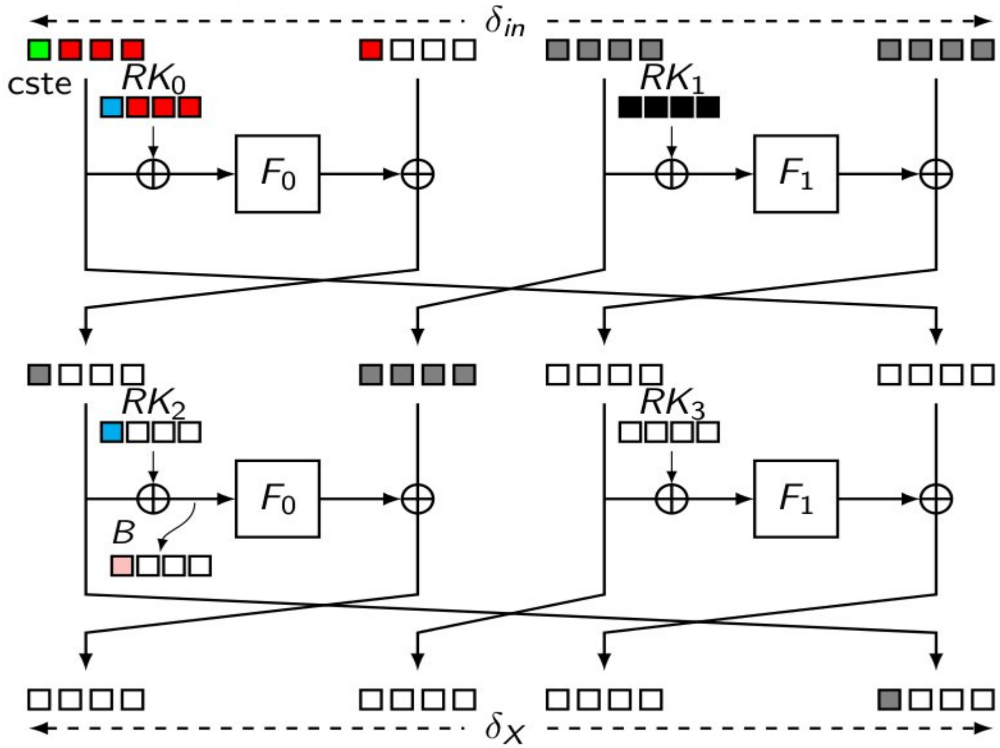
- Aims at reducing the number of key bits involved.

- In the end, for a partial key candidate, we obtained a list of possible partial states. If all the values appear, the partial key can be eliminated.

- Complex to apply and to combine.

Revisiting the State Test Technique

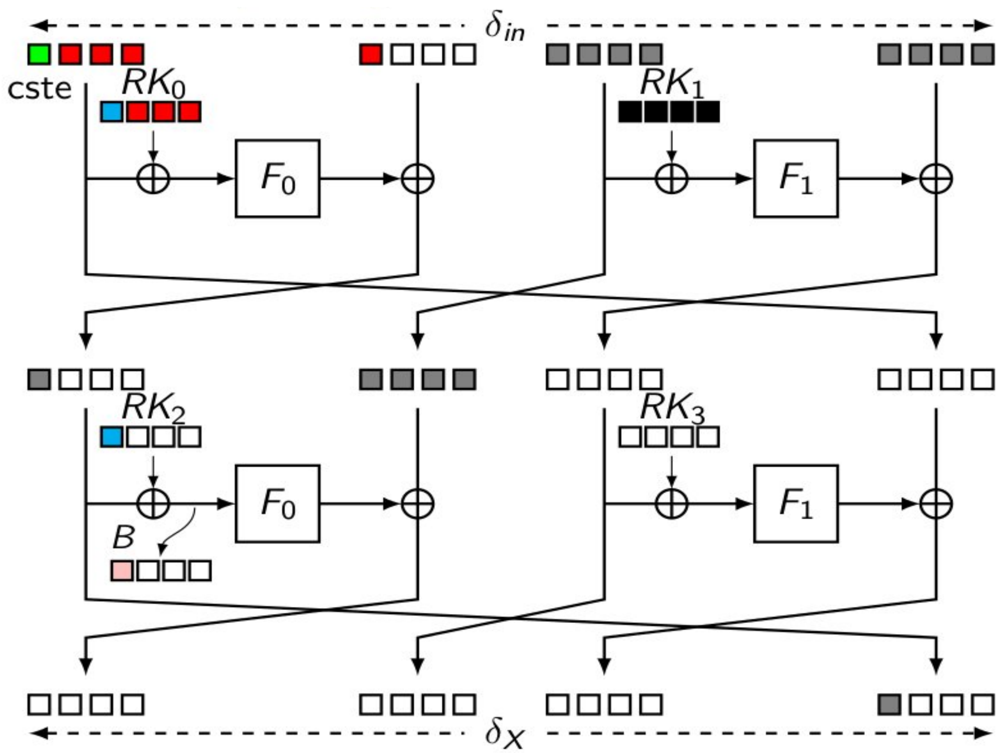
Reduce the number of key bits involved.



$$B = \quad {\color{cyan}\blacksquare} \oplus S_0({\color{cyan}\blacksquare} \oplus {\color{green}\blacksquare}) \oplus {\color{red}\blacksquare}$$

Revisiting the State Test Technique

Reduce the number of key bits involved.



$B' = \quad {\color{cyan}\blacksquare} \oplus S_0({\color{cyan}\blacksquare} \oplus {\color{green}\blacksquare})$ (with $B' = B \bigoplus {\color{red}\blacksquare}$)

$|k_{in} \cup k_{out}| = 122$ bits $\quad \Rightarrow \quad |k_{in} \cup k_{out}| = 122 {\color{cyan}-16} + \underbrace{8}_{B'}$ bits

Combination of previous ideas

- Black box key schedule and multiples:

$$2^{K-k_{out}^{inv}} \cdot (P^{1/M} \cdot 2^{|k_{in} \cup k_{out}|})^M \cdot 2^{-k_{out}^{int}} \cdot 2^{-k_{in}^{int}} \cdot C_{KS}$$

Giving
$$2^K \cdot P \cdot 2^{k_{in}^{inv}} \cdot C_{KS},$$
that multiplied by $2^{-k_{in}^{inv}} \cdot \frac{1}{C_{KS}}$, gives the number of candidate keys to test (and so the last term remains the same).

- Multiples and state-test straight forward now.

Applications of Improved Impossible Diff

Feistel constructions:

- CLEFIA: best attack on CLEFIA (13 rounds).
  State-test and multiples
- Camellia: Improved best attacks for Camellia.
- Simon: previous best attacks on versions 32 and 48.
  Works thanks to multiples!! (now, see [WLVSRT14])
- LBlock: best attack (on 23 rounds).

Applications of Improved Impossible Diff

SPN constructions:

- AES: new trade-offs on 7 rounds (best memory).
- Crypton: best attack on 128b.
- Aria: best impossible differential attack.

Examples

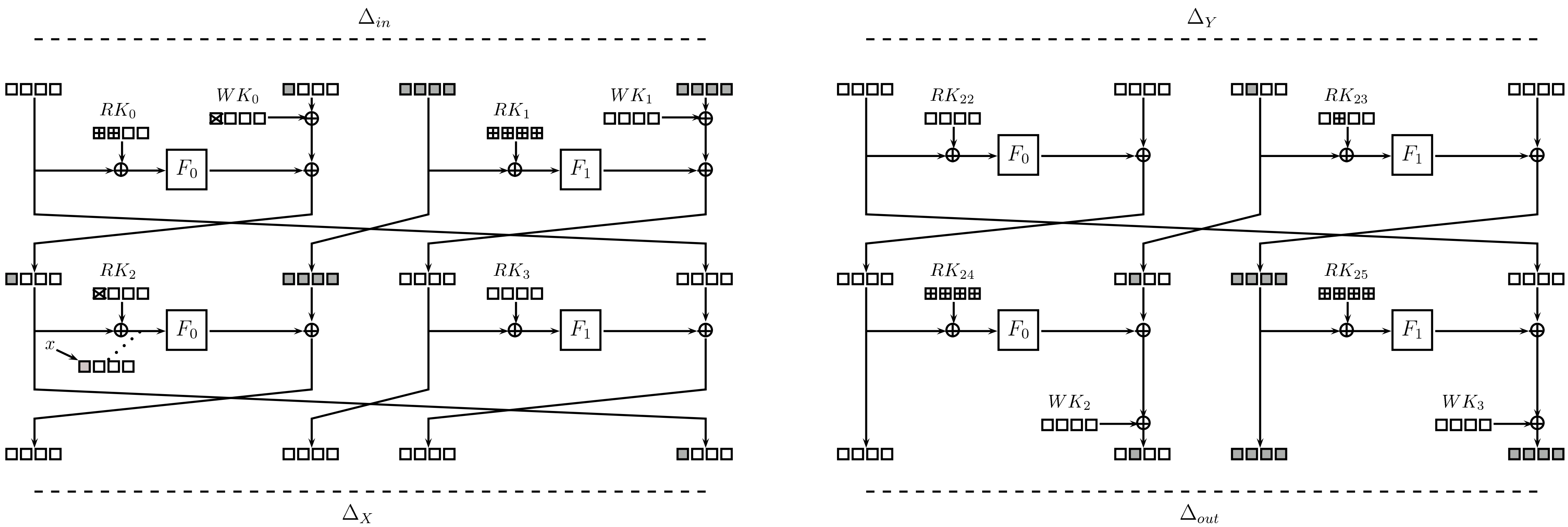
CLEFIA-128

For CLEFIA, we can combine state-test and multiples and provide the best known attack:

Previous best: $2^{121.2}$ time, $2^{117.8}$ data and $2^{86.8}$ memory

Now (improving Asiacrypt paper!!)$2^{114.4}$ time, $2^{114.4}$ data and 2^{80} memory

CLEFIA-128



$$|\Delta_{in}| = |\Delta_{out}| = 48,\ c_{in} = c_{out} = 40$$

$$r_\Delta = 9,\ N_{min} = 2^{80},\ N = N_{min}2^{\varepsilon},\ C_N = 2^{113+\varepsilon}$$

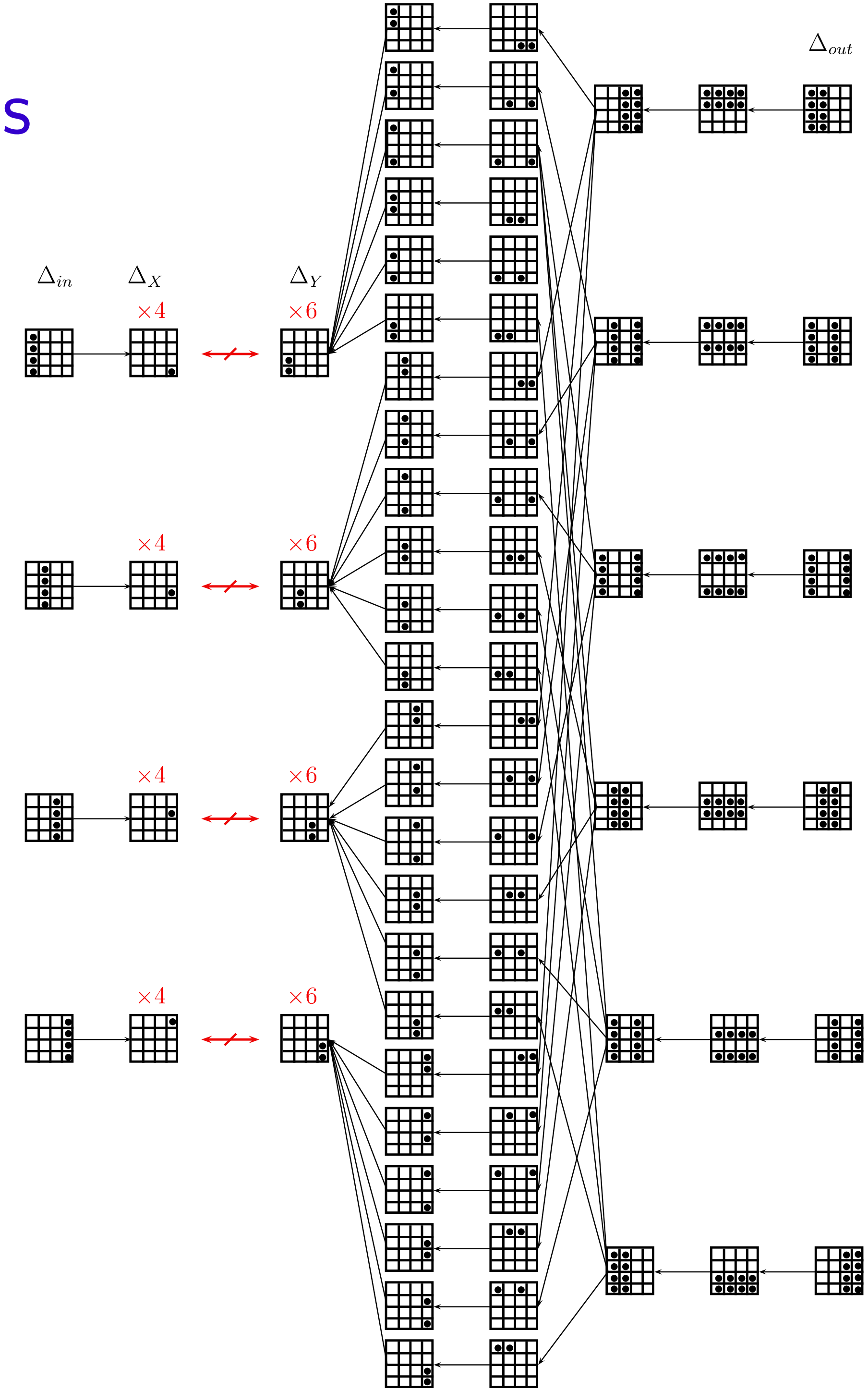
Crypton

Example of how multiple impossible differentials and multiple differentials can considerably improve the complexities.

From 2^{121} data, $2^{116.2}$ time and 2^{112} memory

to $2^{114.92}$ data, $2^{113.7}$ time and $2^{88.5}$memory.

Multiple multiples



Conclusion

To Sum Up

- Impossible differential attacks, important family that provides best results on several ciphers.

- We have now a generic approach easy to apply. When mounting an attack, check the exact complexity by determining the procedure (but normally corresponds to the theoretical one).

- New improvements?