Lightweight Cryptography

María Naya-Plasencia

Lightweight Cryptography

María Naya-Plasencia
Inria, France

Summer School on real-world crypto and privacy
Šibenik, Croatia - June 15 2018

Outline

- Symmetric lightweight primitives
- Most used cryptanalysis
  - *Impossible Differential* Attacks
  - *Meet-in-the-middle*
  - Dedicated attacks
- Conclusions and remarks

Symmetric Lightweight Primitives

Lightweight Primitives

- Lightweight primitives designed for constrained environments, like RFID tags, sensor networks.

- Real need $\Rightarrow$ an enormous amount of proposals in the last years (block and stream ciphers, hash functions):
PRESENT, LED, KATAN/KTANTAN, KLEIN, PRINCE, PRINTcipher, LBLOCK, TWINE, XTEA, mCrypton, Iceberg, HIGHT, Piccolo, SIMON, SPECK, SEA, DESL...

- NIST competition to start around december 2018, comments on call close the 28 June!

Draft: NIST competition

AEAD and hash functions. (Some) requirements:

- Efficient for short messages.
- Compact HW and embedded SW implementations with low RAM/ROM.
- Key preprocessing efficient.
- Different strategies: low energy/low power/low latency.
- Performant in different microcontroller architectures...

Better in constrained environments than existing standards.

Lightweight Primitives

- Any attack better than the generic one is considered a "break".
- Cryptanalysis of lightweight primitives: a fundamental task, responsibility of the community.
- Importance of cryptanalysis (especially on new proposals): the more a cipher is analyzed, the more confidence we can have in it...
- ...or know which algorithms are not secure to use.

Lightweight Primitives

- Lightweight: more 'risky' design, lower security margin, simpler components.

- Often innovative constructions: dedicated attacks

- Types of attacks: single-key/related-key, distinguisher/key-recovery, weak-keys,...
- Importance of attacks on reduced versions.
- High complexities: ugly properties or security margin determined.

Main Objectives of this talk

- Perform a (non-exhaustive) survey of proposals and their security status.
- Provide the intuition of the "most useful attacks" against LW ciphers.
- Conclusions and remarks (link with hash functions).

Survey of Proposals [1]

- *Feistel Networks - best external analysis*
  DESLX - none
  ITUbee - self-similarity (8/20r)
  LBlock - **imposs. diff.** (24/32r)
  SEA - none
  SIMON and SPECK - **imposs. diff.**, diff, 0-correl.
  XTEA - **mitm** (23/64r)
  CLEFIA - **imposs. diff.** (13/18r)
  HIGHT - 0-correlation (27/32r)
  TWINE - **mitm,imposs. diff.**,0-corr (25/36r)

[1] mainly from https://cryptolux.org/index.php/Lightweight_Block_Ciphers

Survey of Proposals

- *Substitution-Permutation Network*
  KLEIN - **dedicated attack** (full round)
  LED - EM generic attacks (8/12r, 128K)
  Zorro - diff. (full round)
  mCrypton - **mitm** (9/12r, 128K)
  PRESENT - mult. dim. lin. (27/31r)
  PRINTcipher - **invariant-wk** (full round)
  PRIDE - diff (18/20r)
  PRINCE - mult. diff (10/12r)
  Fantomas/Robin -none/**invariant-wk** (full round)

Survey of Proposals

- *FSR-based*
  KTANTAN/KATAN - **mitm** (153/254r)
  Grain - correl./ cube attacks (some full)
  Trivium - cube attacks (800/1152) -
  Sprout - guess-and-determine (full round)
  Quark -condit. diff (25%)
  Fruit - divide and conquer (full)
  Lizard - guess-and-det. (full)

Survey of Proposals

- *ARX*
  Chaskey - diff-lin (7/8r)
  Hight - 0-correl (27/32r)
  LEA - diff. (14/24r)
  RC5 - diff. (full round)
  Salsa20 - diff (8/20r)
  Sparx - **imposs. diff.** (15/24r)
  Speck - diff. (17/32r)

More Proposals

For more details, primitives, classifications, see:

State of the Art in Lightweight Symmetric Cryptography,
by Alex Biryukov and Leo Perrin
https://eprint.iacr.org/2017/511

Most Successful Attacks

Families of attacks

- Impossible differentials (Feistel)
- Mitm / guess and determine (SPN, FSR)
- Dedicated: (differential/linear...)

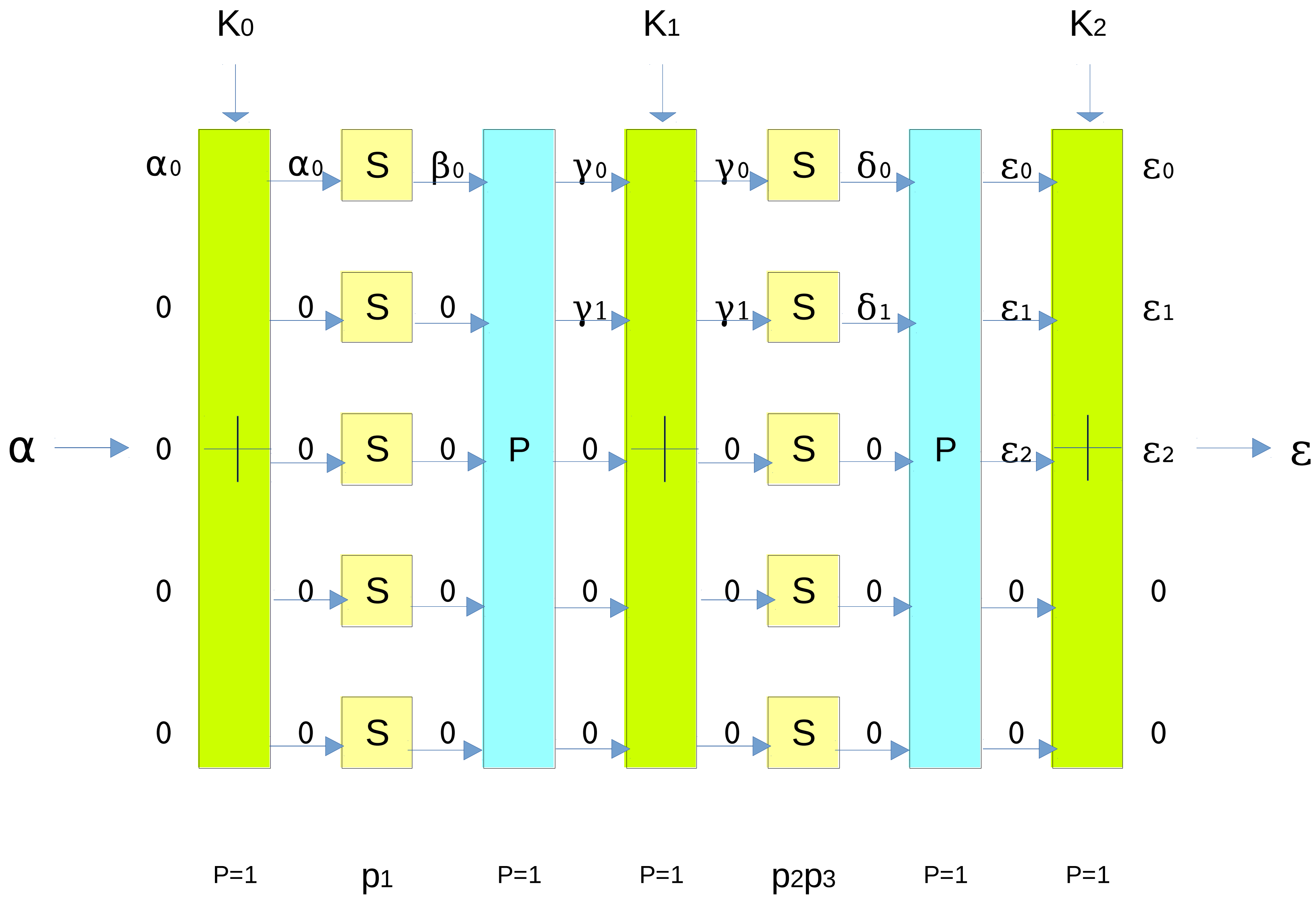
Impossible Differential Attacks

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).

Differential path: example

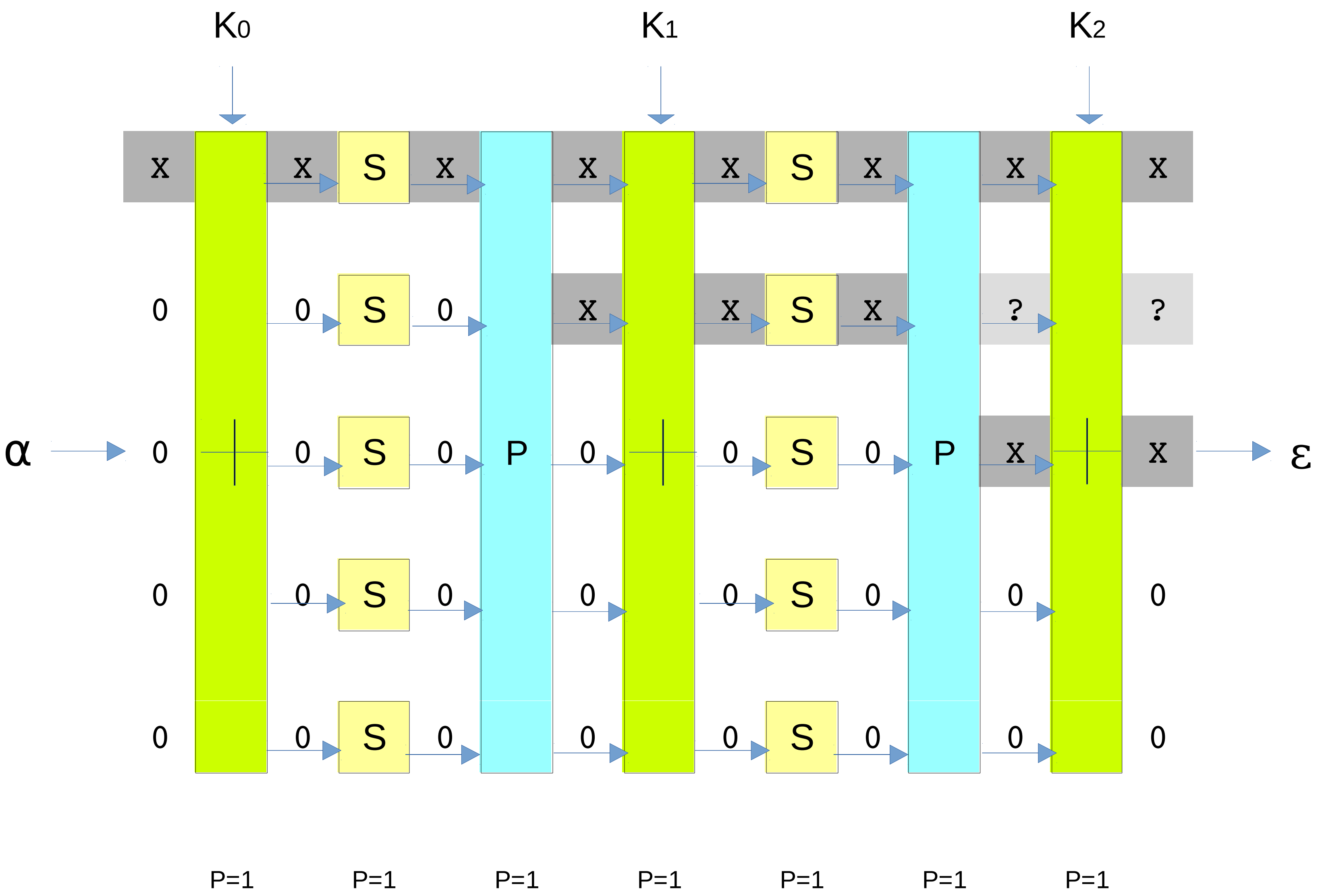


Truncated Differential Attacks [K 94]

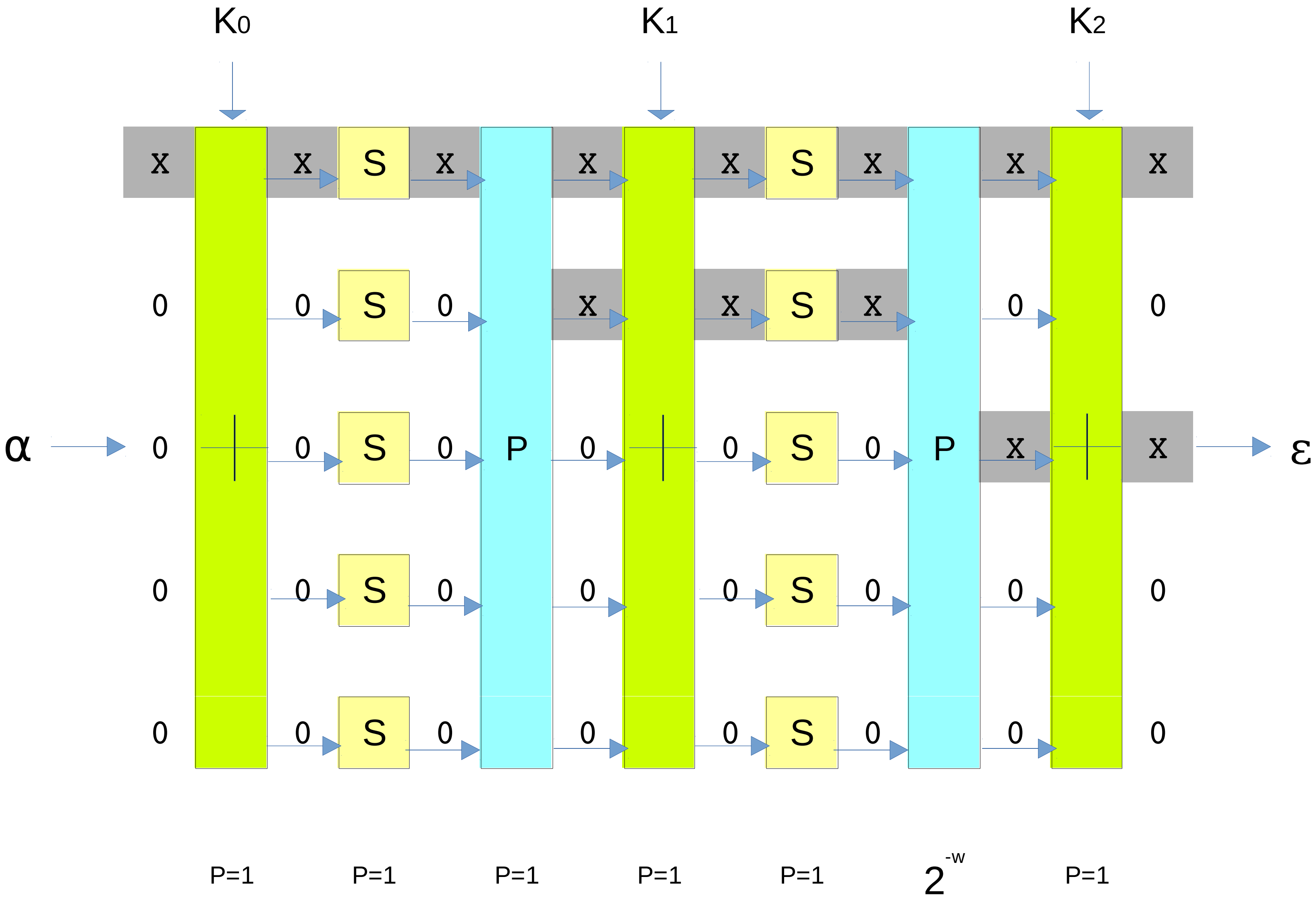
A truncated path predicts only parts of the differences.

Let's see a simple example:

Truncated path: example



Truncated path: example



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.

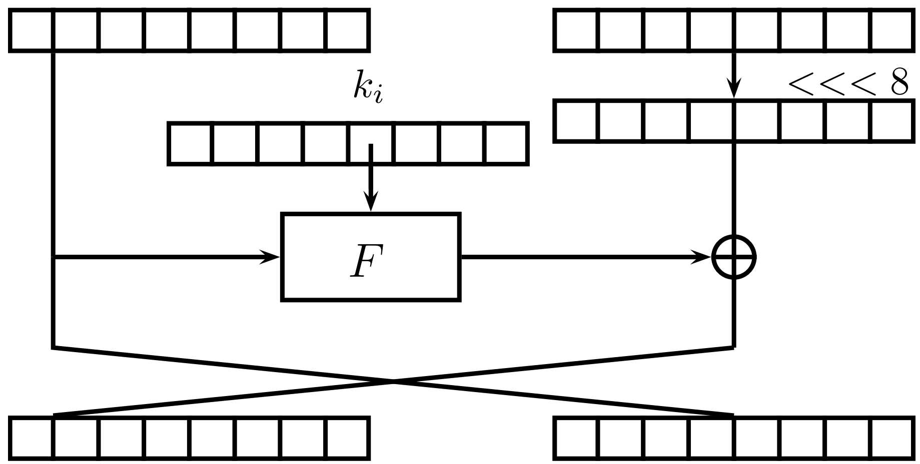
- Extend it backward and forward $\Rightarrow$ Active Sboxes transitions give information on the involved key bits.

- Generic framework and improvements [BNPS14,BLNPS17]

Example: LBlock

Designed by Wu and Zhang, (ACNS 2011).

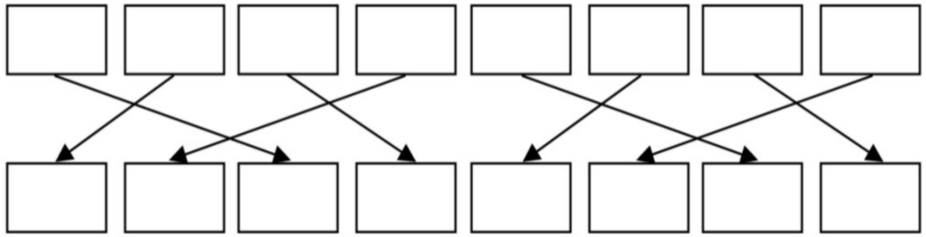
- 80-bit key and 64-bit state.
- 32 rounds.



Example: LBlock

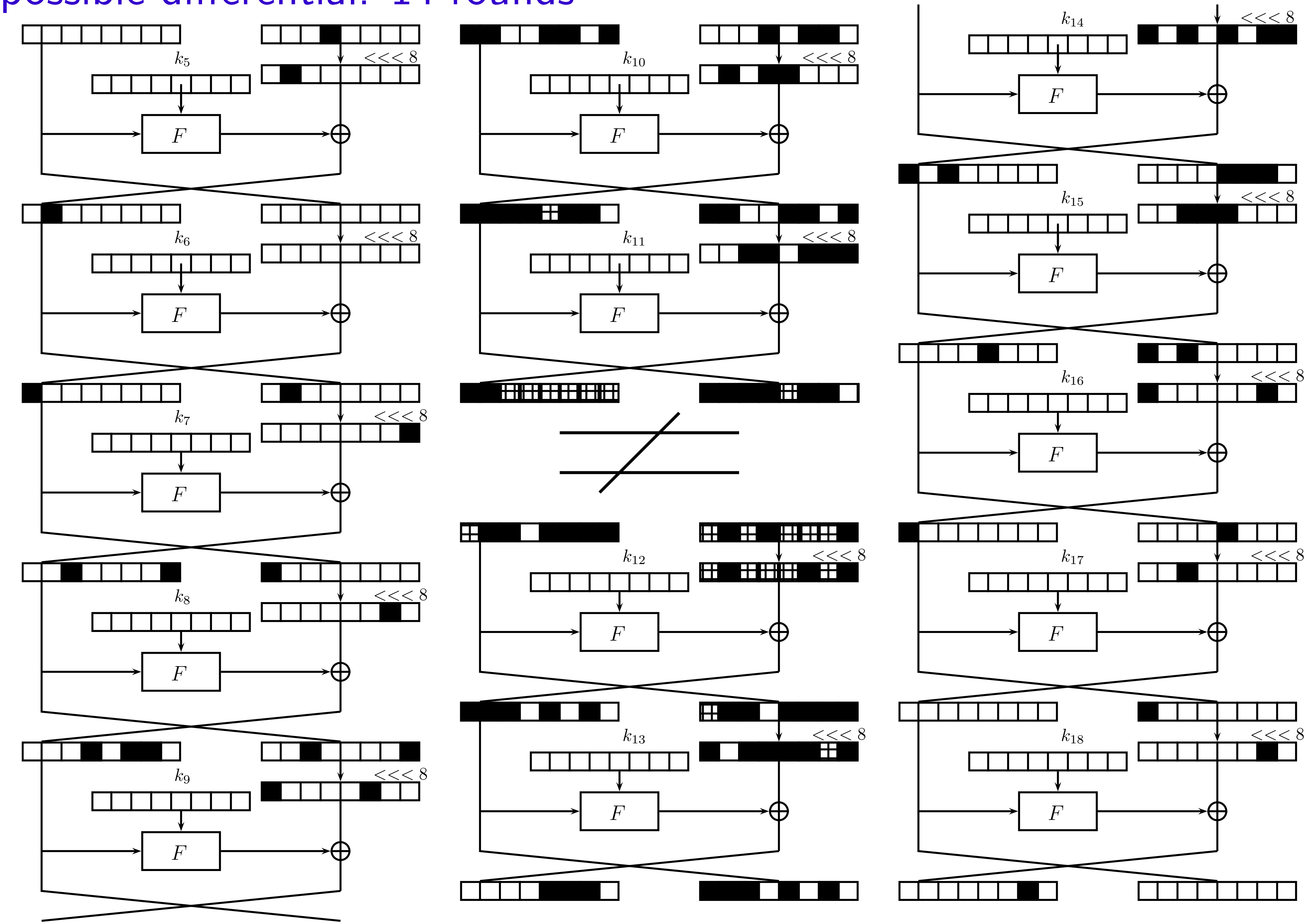
Inside the function F:

- add the subkey to the input.
- 8 different Sboxes 4×4.
- a nibble permutation P:

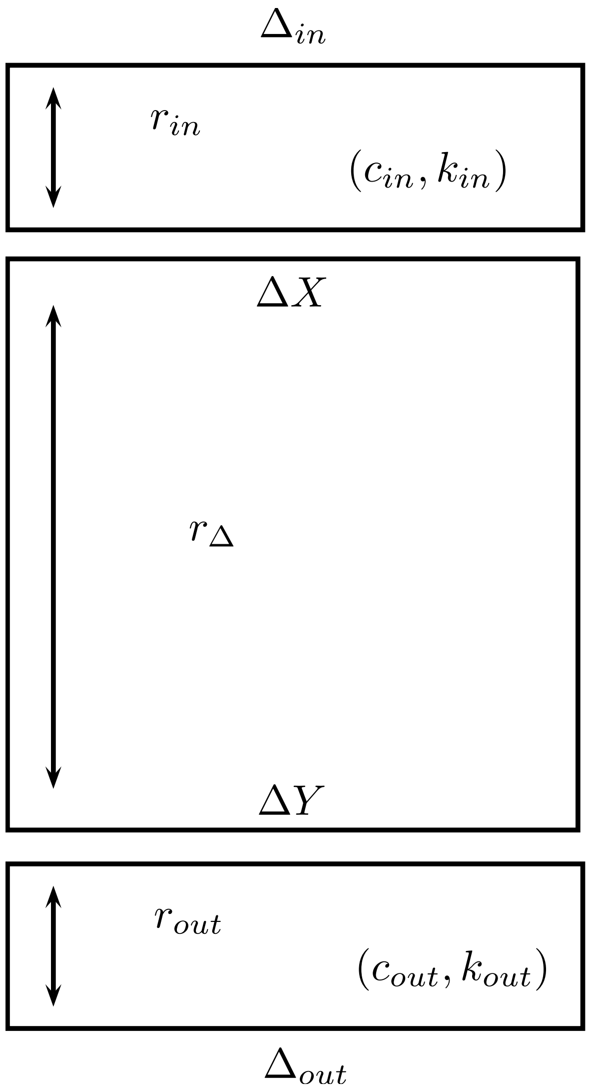


Best attack so far: Imp. Diff. on 23 rounds [CFMS'14,BMNPS'14] and RK on 24 rounds [SHS'15].

Impossible differential: 14 rounds



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

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

For the Attacks to Work

We need, for a state size s and a key size $|K|$:

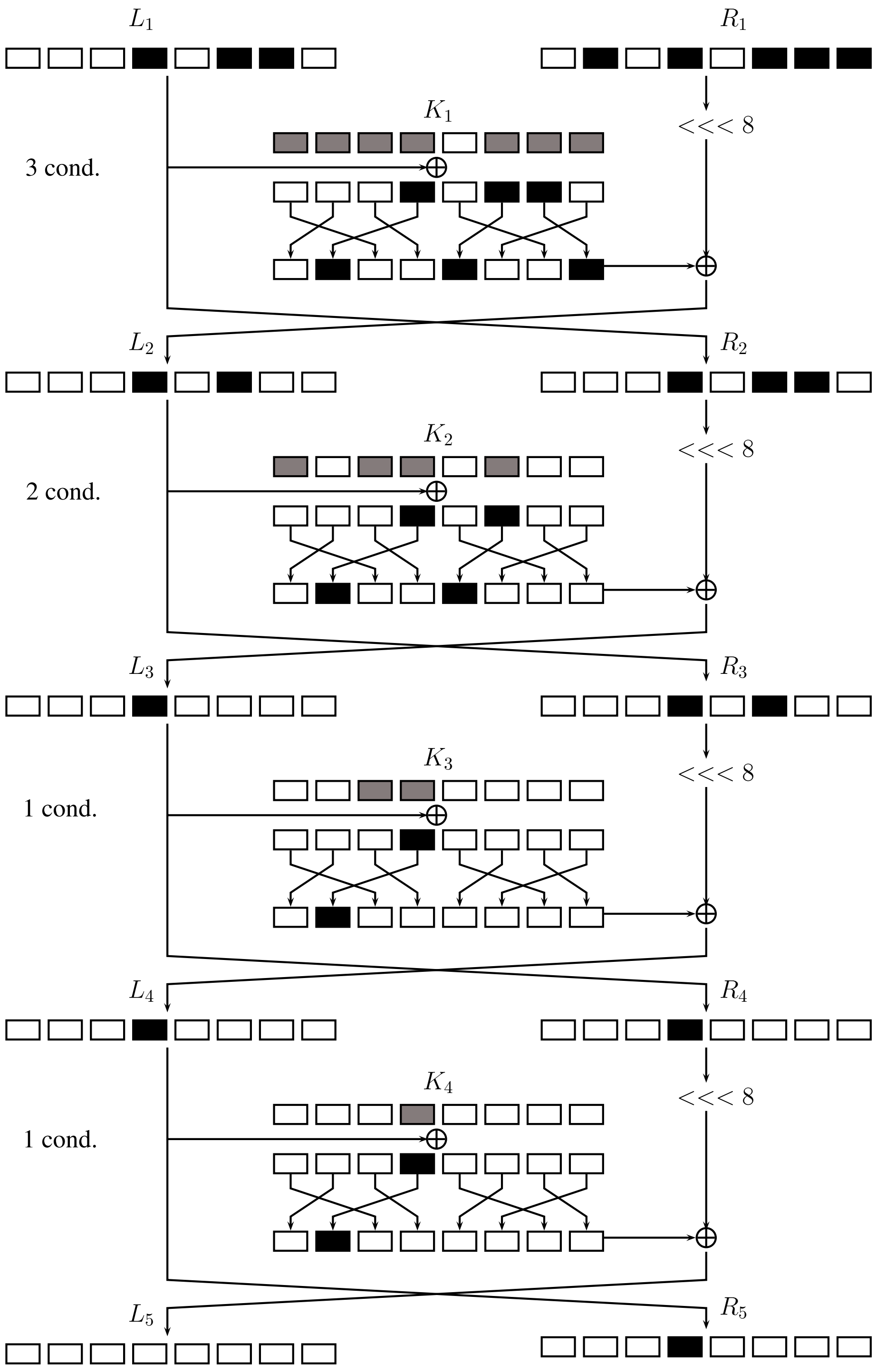
$$C_{data} < 2^s$$

and

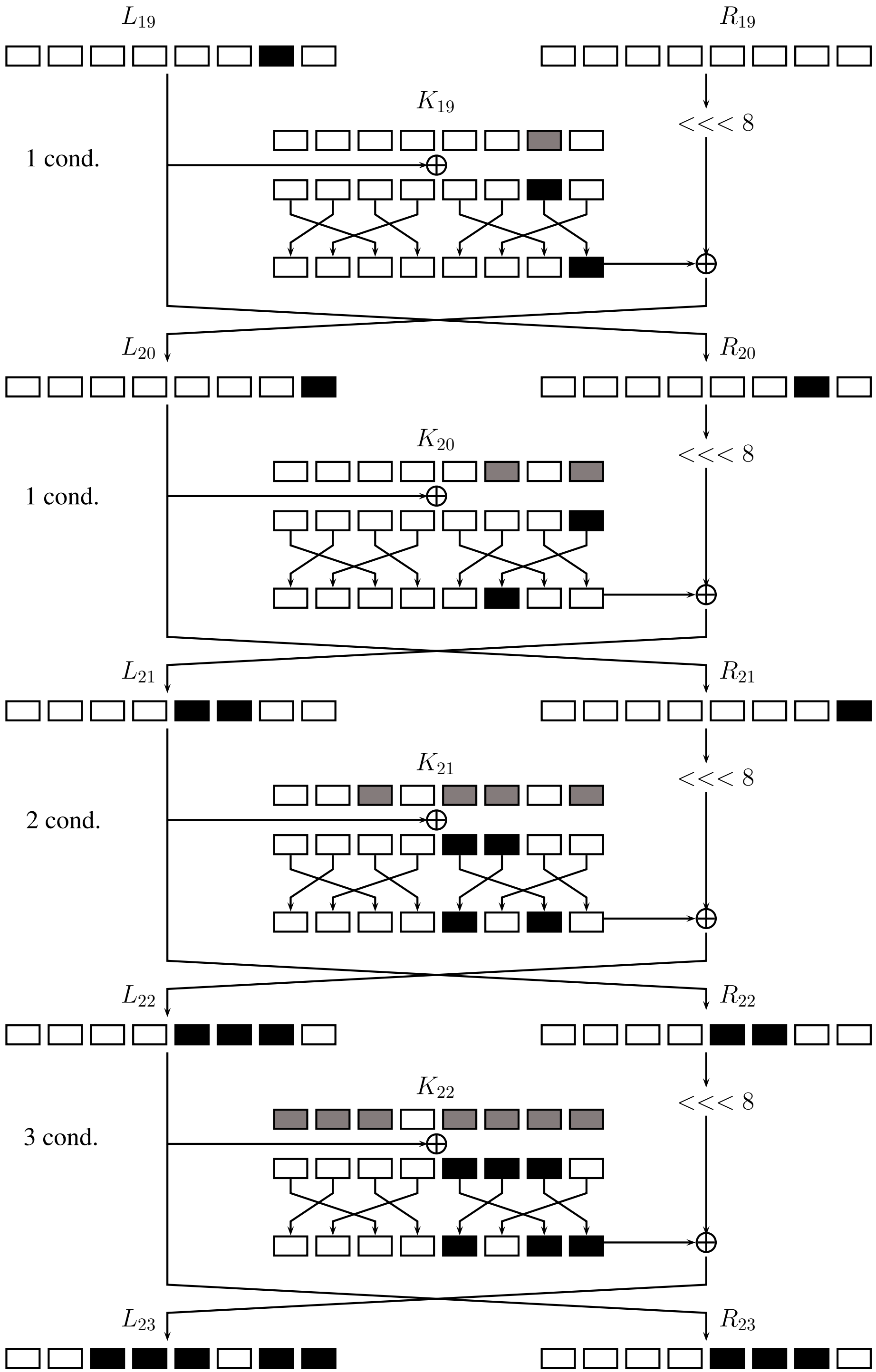
$$C_{data} + 2^{|k_{in} \cup k_{out}|} C_N + 2^{|K| - |k_{in} \cup k_{out}|} P 2^{|k_{in} \cup k_{out}|} < 2^{|K|}$$

where C_{data} is the data needed for obtaining N pairs $(\Delta_{in}, \Delta_{out})$, C_N 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.

First Rounds



Last Rounds



L19
R19
K19
<<< 8
1 cond.
L20
R20
K20
<<< 8
1 cond.
L21
R21
K21
<<< 8
2 cond.
L22
R22
K22
<<< 8
3 cond.
L23
R23

Impossible Differential on LBlock

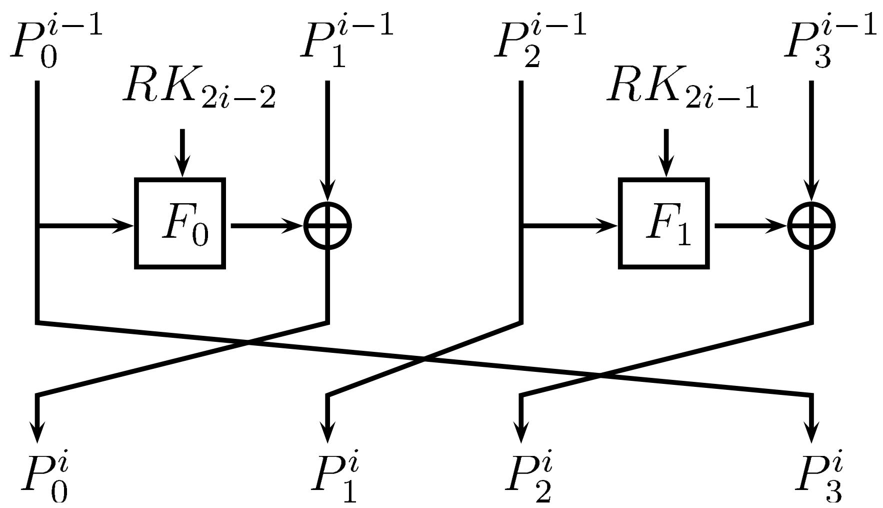
- For 21 rounds a complexity of $2^{69.5}$ in time with 2^{63} data, for 22: $2^{71.53}$ time and 2^{60} data, for 23: $2^{75.36}$ time and 2^{59} data.
- Feistel constructions in general are good targets

Improvements [BN-PS14,BLN-PS17,B18]

- Multiple impossible differentials (related to [JN-PP13])
- Correctly choosing Δ_{in} and Δ_{out} (related to [MRST09])
- State-test technique (related to [MRST09])
- More accurate estimate of the pairs [B18]

Example: CLEFIA-128

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



Multiple Impossible Differentials

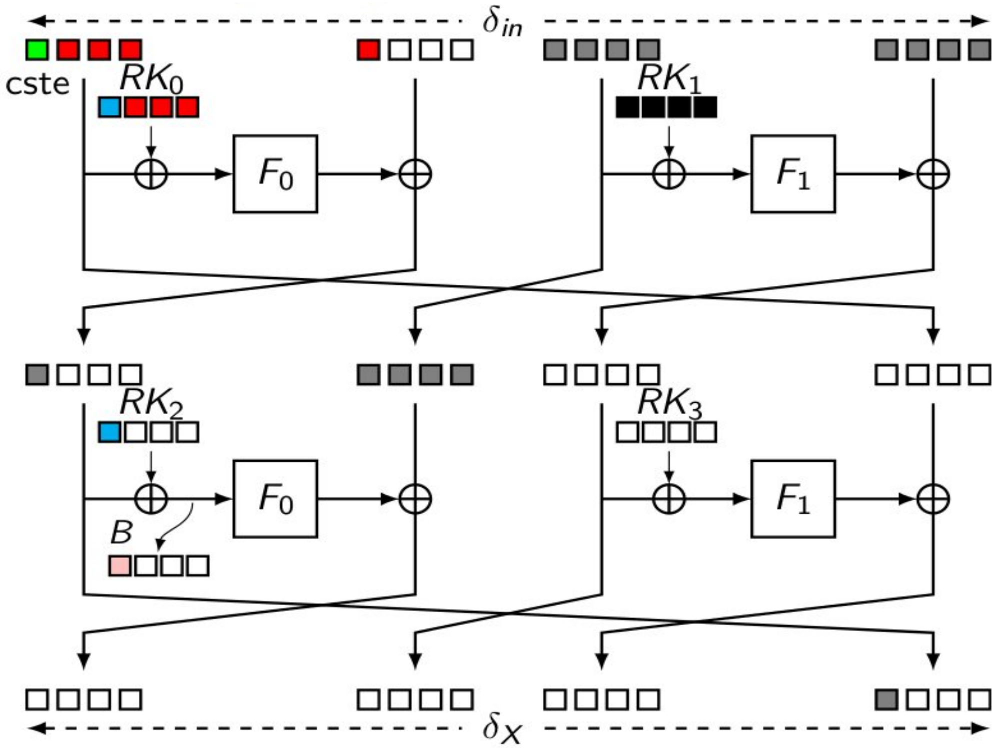
Formalize the idea of [Tsunoo et al. 08]:
CLEFIA has two 9-round impossible differentials $((0,0,0,A) \not\to (0,0,0,B))$ and $((0,A,0,0) \not\to (0,B,0,0))$ when A and B verify:

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: $C_{data} = 2^{113}$ becomes $C_{data} = 2^{113}/24$

State Test Technique

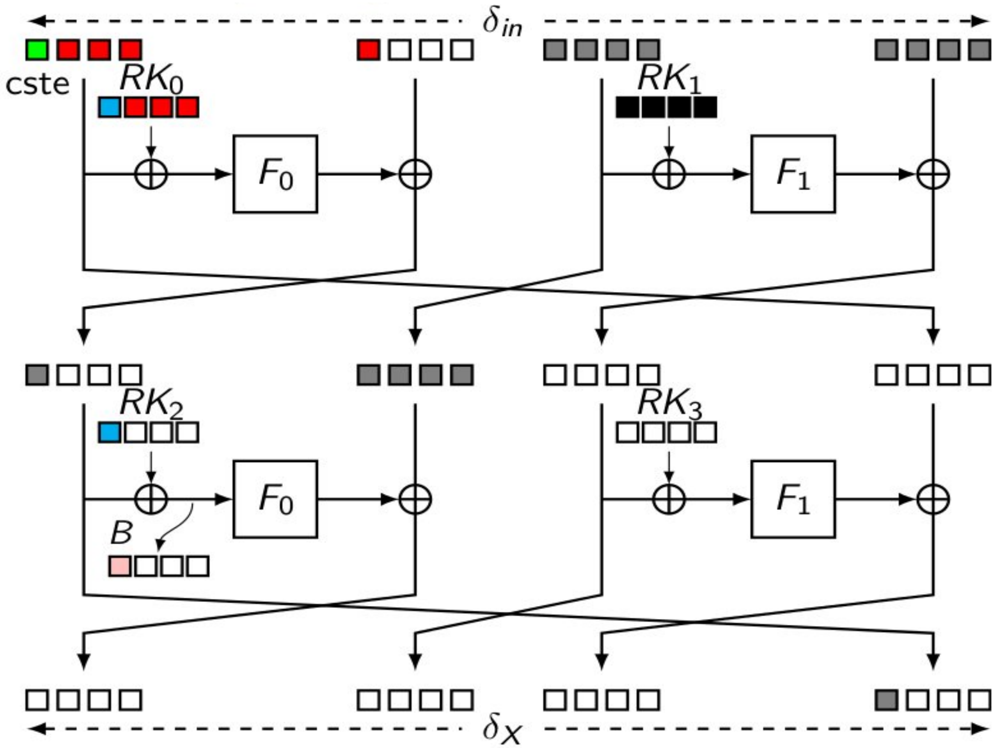
Reduce the number of key bits involved.



$$B = \blacksquare \oplus S_0(\blacksquare \oplus \blacksquare) \oplus \blacksquare$$

State Test Technique

Reduce the number of key bits involved.



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

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

Applications of Improved Impossible Diff

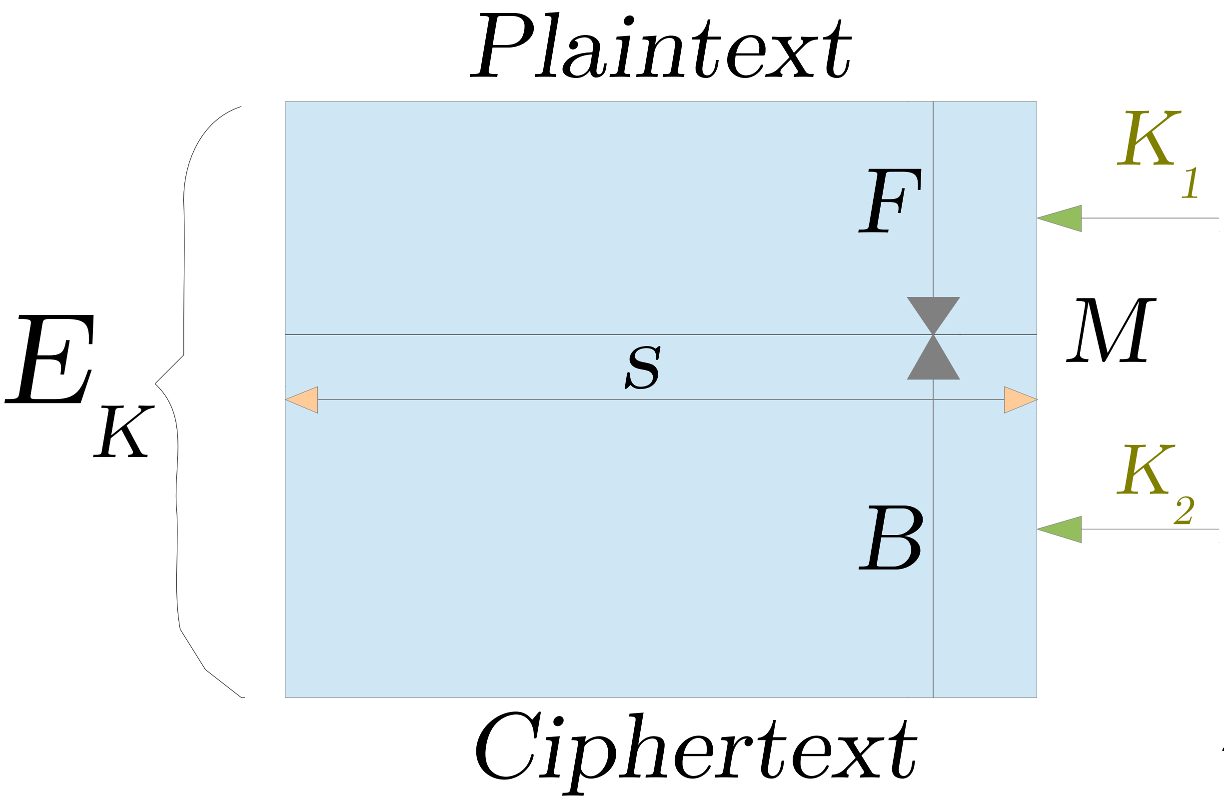
- CLEFIA: best attack on CLEFIA (13 rounds).
- Camellia: Improved best attacks for Camellia.
- AES: attacks comparable with best mitm ones (7 rounds).
- LBlock: best attack (on 24 rounds).

Meet-in-the-middle attacks

Meet-in-the-Middle Attacks

- Introduced by Diffie and Hellman in 1977.
- Largely applied tool.
- Few data needed.
- Many improvements: partial matching, bicliques, sieve-in-the-middle...

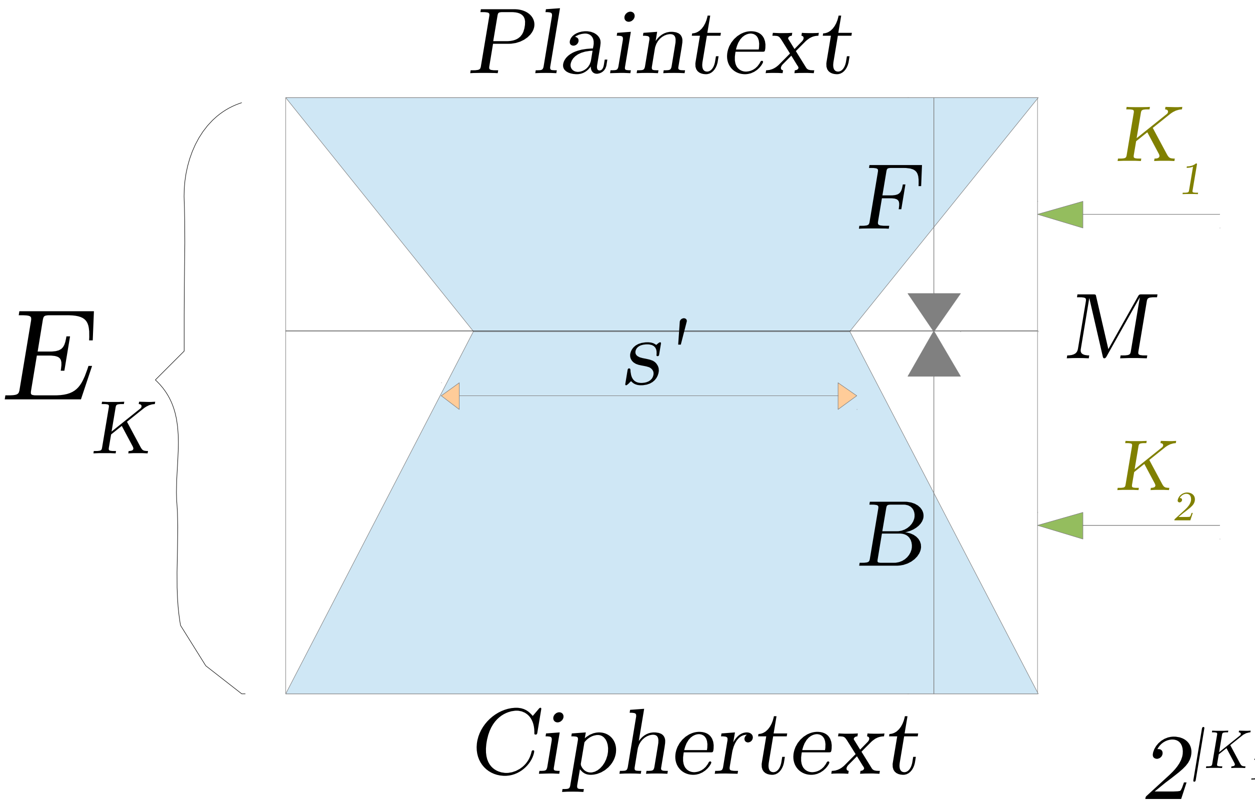
Meet-in-the-Middle Attacks [Diffie Hellman 77]



$K = K_1 \cup K_2$

$$2^{|K_1|} + 2^{|K_2|} + 2^{|K|-s}$$

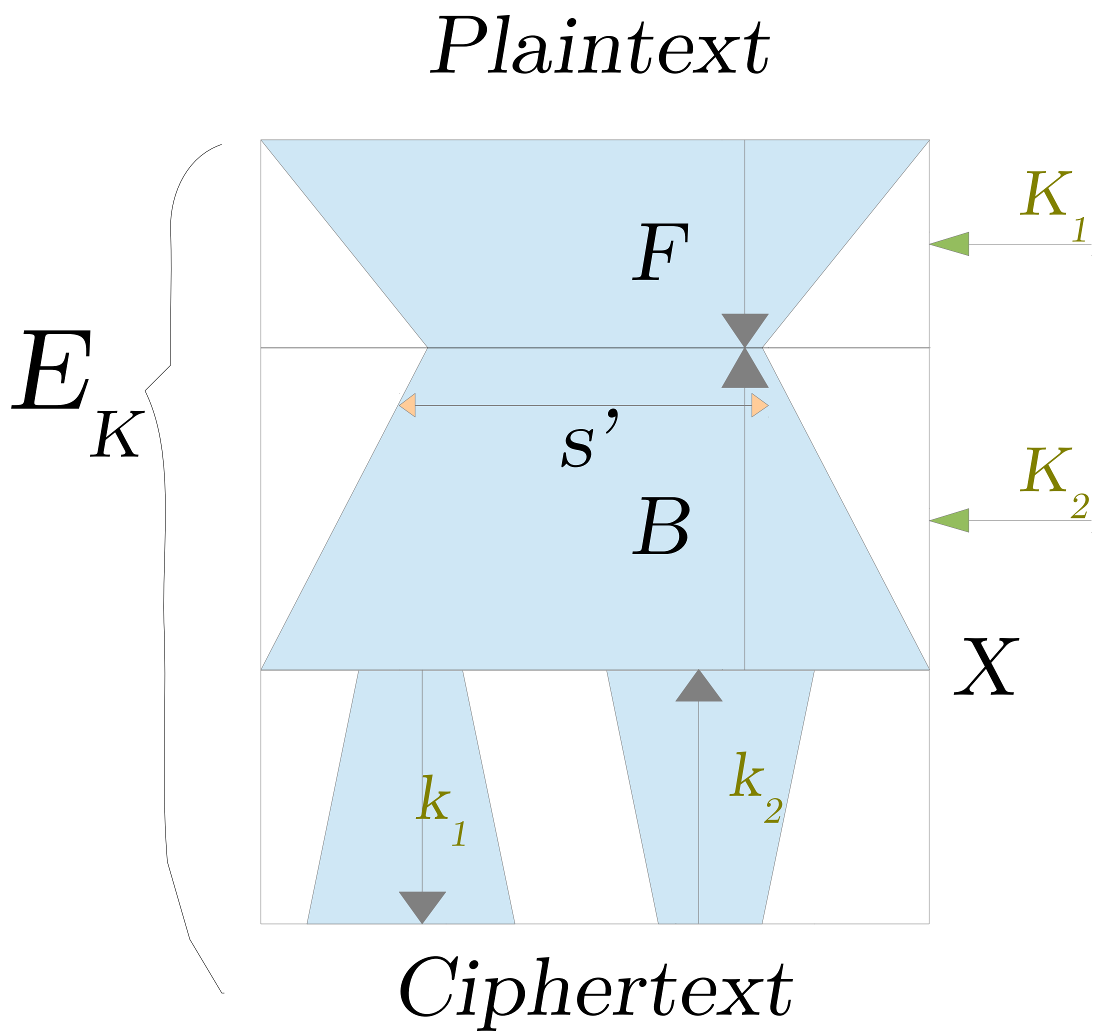
With Partial Matching [AS'08]



$K = K_1 \cup K_2$

$2^{|K_1|} + 2^{|K_2|} + 2^{|K| - s'}$

With Bicliques [KRS'11]



$K = K_1 \cup K_2$

$$2^{|k_1|} + 2^{|k_2|} +$$

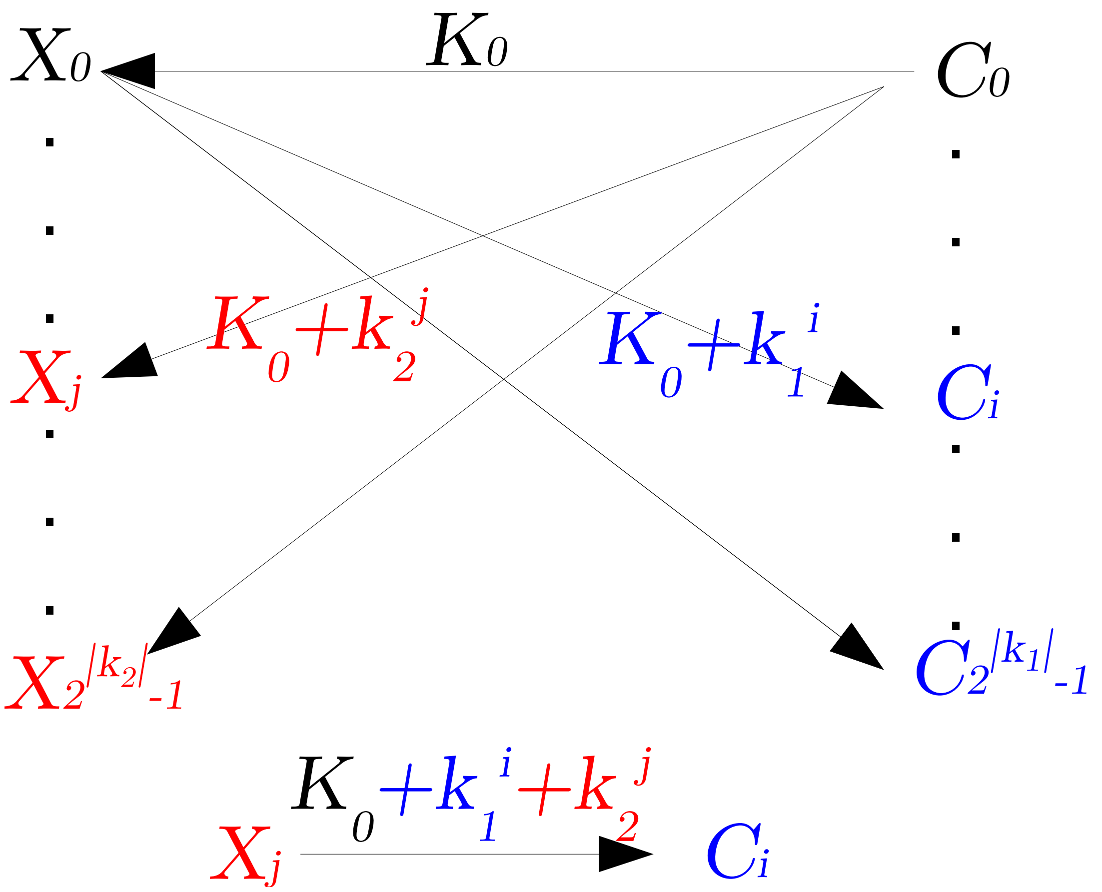
$$2^{|K_1|} + 2^{|K_2|} + 2^{|K|-s'}$$

Bicliques

- Improvement of MITM attacks, but also...
- It can always be applied to reduce the total number of computations (at least the precomputed part)
  $\Rightarrow$ acceleration of exhaustive search [BKR'11] [2]
- Many other accelerated exhaustive search on LW block ciphers: PRESENT, LED, KLEIN, HIGHT, Piccolo, TWINE, LBlock ... (less than 2 bits of gain).
- Is everything broken? No.

[2] Most important application: best key-recovery on AES-128 in $2^{126.1}$ instead of the naive 2^{128}.

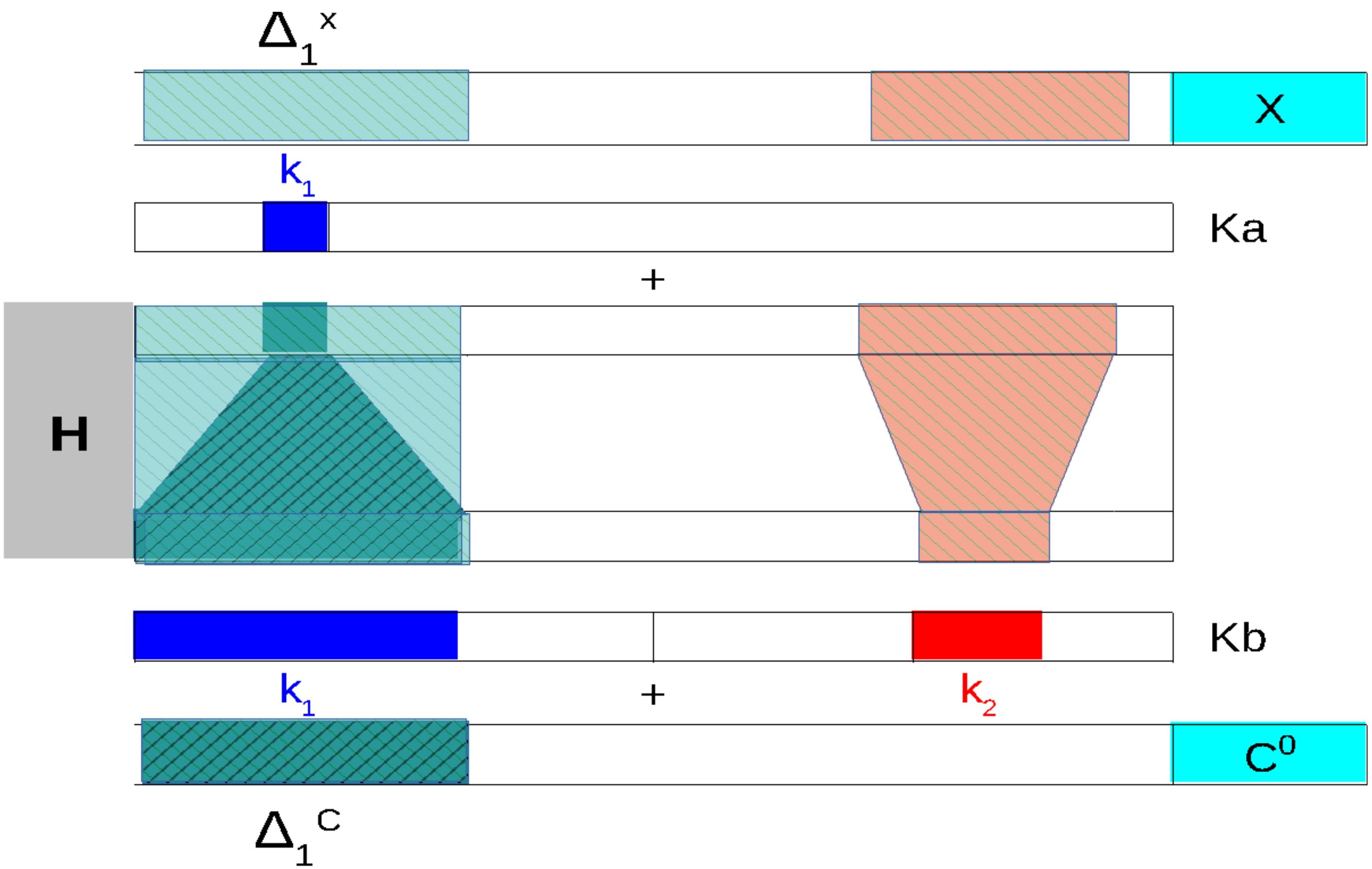
Bicliques



With $2^{|k_1|}+2^{|k_2|}$ computations, $2^{|k_1+k_2|}$ Transitions.

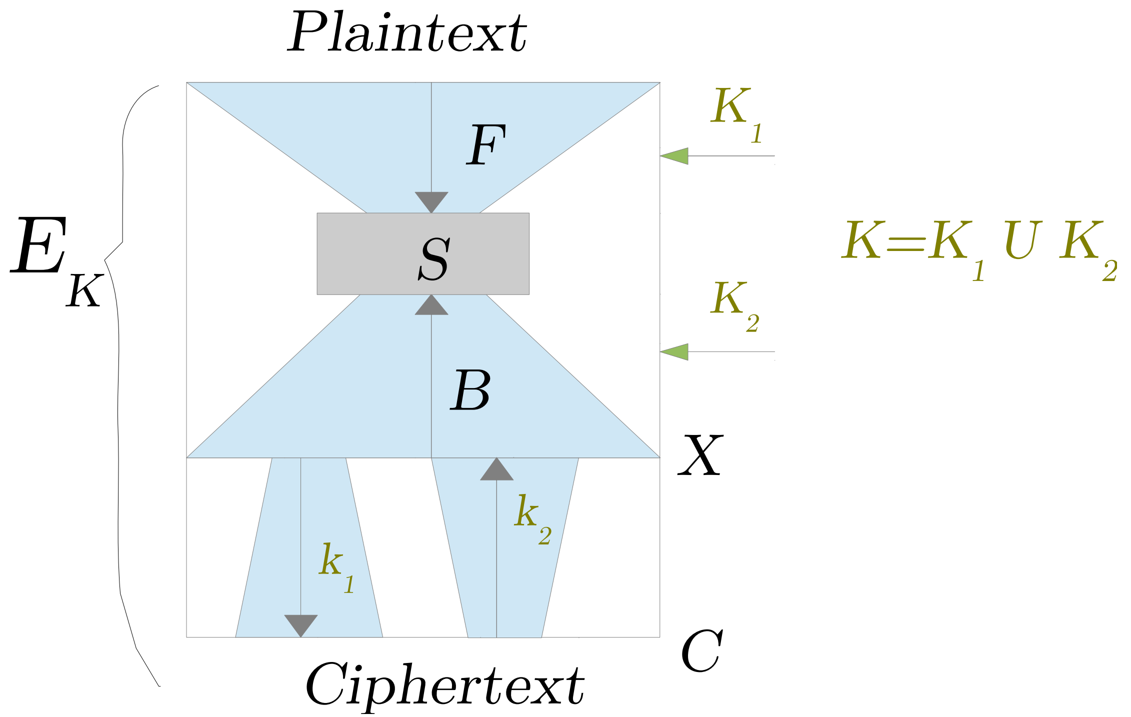
Improved Bicliques [CN-PV 13]

Can we build bicliques with only one pair of P-C?

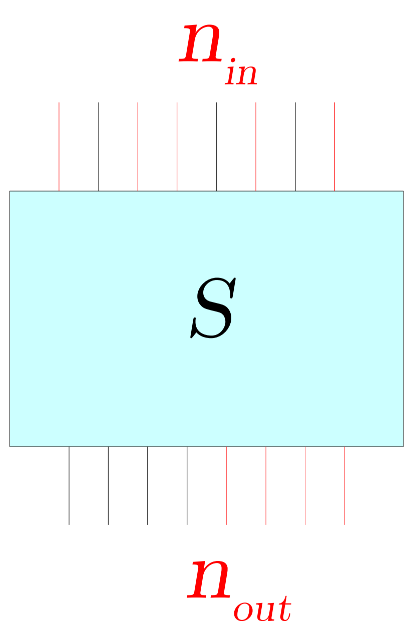


Sieve-in-the-Middle [CN-PV'13]

- Compute partial inputs and outputs of S
  $\Rightarrow$ sieving with transitions instead of collisions.



When can we sieve?



- n_{in} known bits out of m: at most $2^{m-n_{in}}$ values for the n_{out} output bits.
- A transition exists with probability p.
- Sieve when $n_{in} + n_{out} > m \Rightarrow p < 1$

How do we sieve?

- We obtain a list L_A of partial inputs u and a list L_B of partial outputs v $\Rightarrow$ merge L_A and L_B with the condition (u, v) is a valid transition though S.

- Naive way costs $|L_A| \times |L_B| = 2^{|K_1|+|K_2|}$:
no gain with respect to exhaustive search.

- We need an efficient procedure.
Often S is a concatenation of S-boxes.

Merging the lists

Merging the lists with respect to R

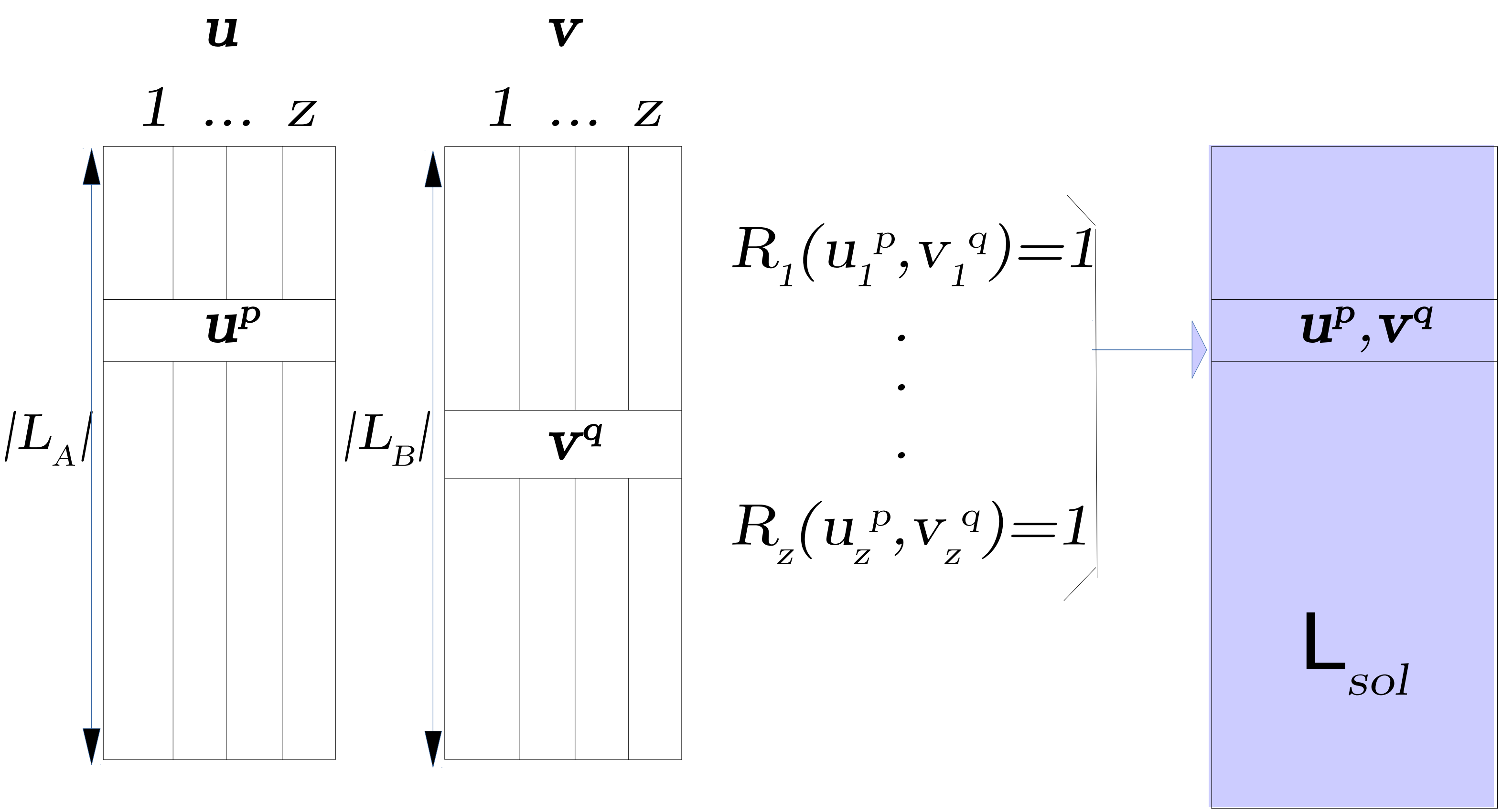
- R is group-wise, *i.e.* for z groups

$$R(u, v) = \Pi_{i=1}^{z} R_i(u_i, v_i)$$

  Find all $u \in L_A$ and $v \in L_B$ such that $R(u, v) = 1$.

- Subcase of the first problem in [N-P 11].
  First studied for rebound attacks.

Group-wise relation



Merging Algorithms

- Problem also appears in divide-and-conquer attacks (and rebound attacks).
- Solutions from list merging algorithms [N-P-11] and dissection algorithms [DDKS 12]
- Many applications: ARMADILLO2 [ABN-PVZ 11], ECHO256 [JN-PS 11], JH42 [N-PTV 11], Grøstl [JN-PP 12], Klein [LN-P 14], AES-like [JN-PP 14], Sprout [LN-P 15], Ketje [FN-PR 18]...

Some Applications SITM

- Reduced-round: PRESENT, DES, PRINCE, AES-biclique [Canteaut N-P Vayssieres 13]
- Reduced-round LBlock [Altawy Youssef 14]
- Best reduced-round KATAN [Fuhr Minaud 14]
- Reduced-round Simon [Song et al 14]
- Low-data AES [Bogdanov et.al 15] [Tao et al 15]
- MIBS80/PRESENT80 [Faghihi et al 16]

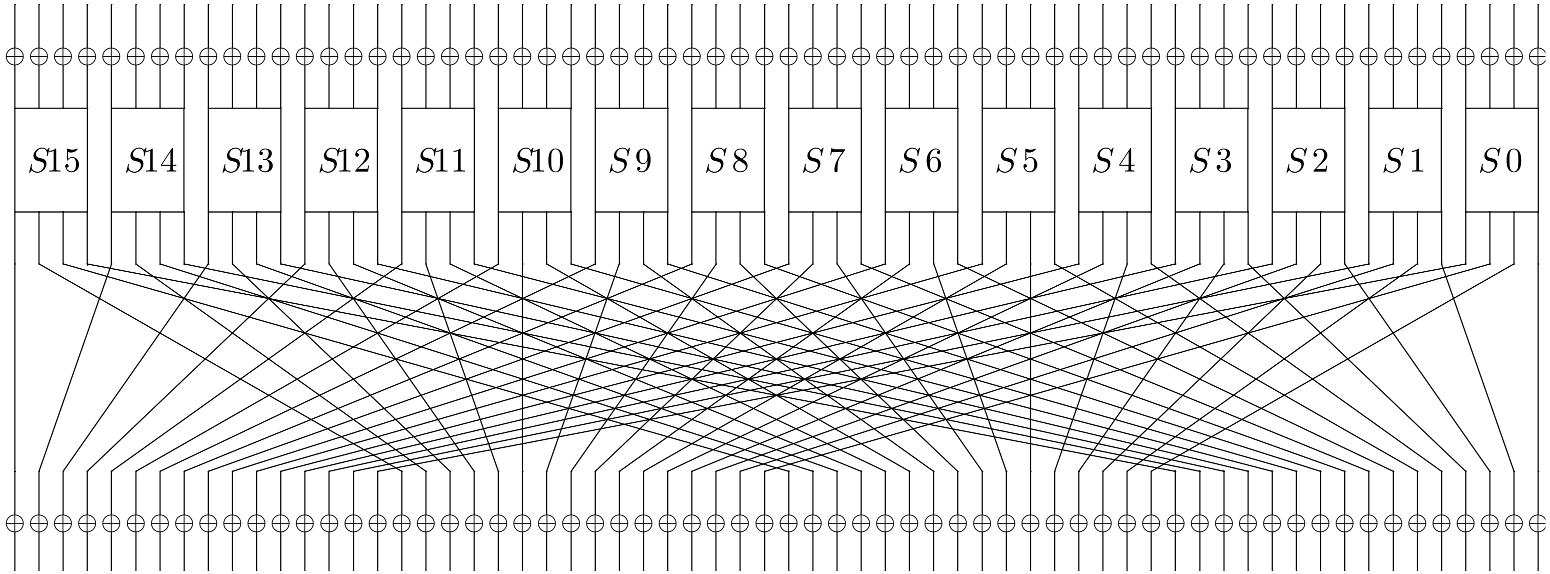
- Interesting for low data attacks...

PRESENT [BKLPPRSV'07]

- One of the most popular ciphers, proposed in 2007, and now ISO/IEC standard.
- Very large number of analysis published (20+).
- Best attacks so far: multiple linear attacks (27r/31r).

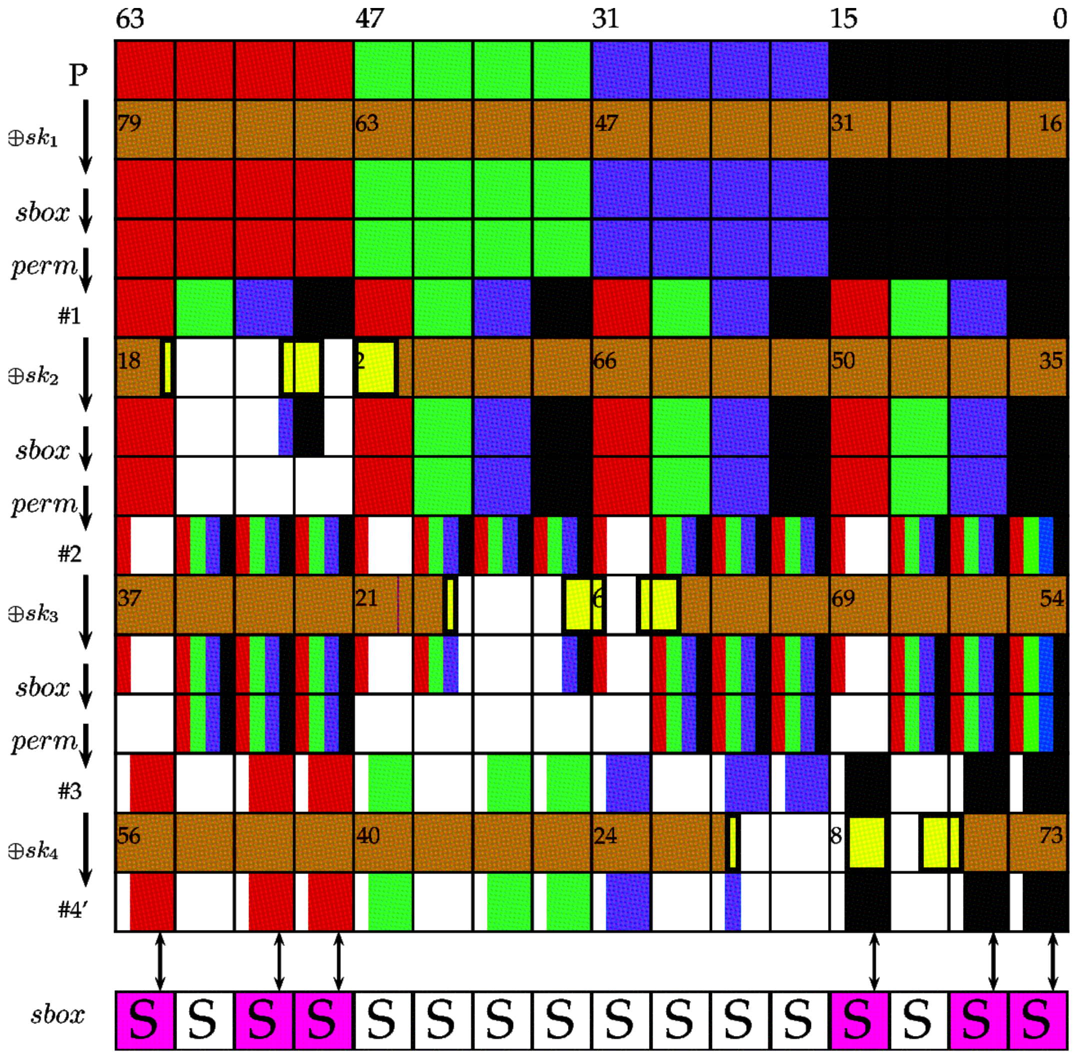
PRESENT

Block $n = 64$ bits, key 80 or 128 bits.

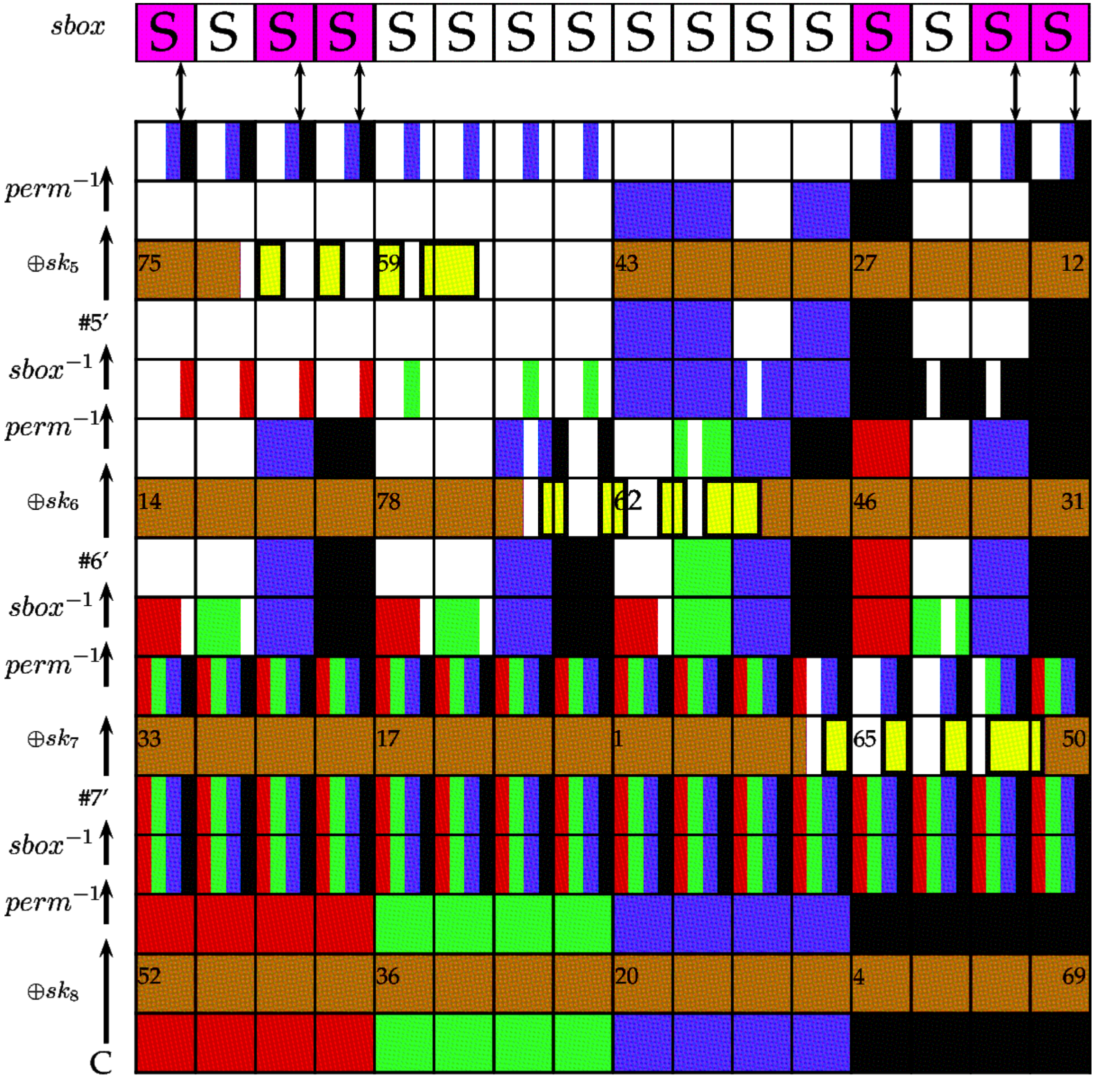


31 rounds + 1 key addition.

Forward Computation



Backward Computation



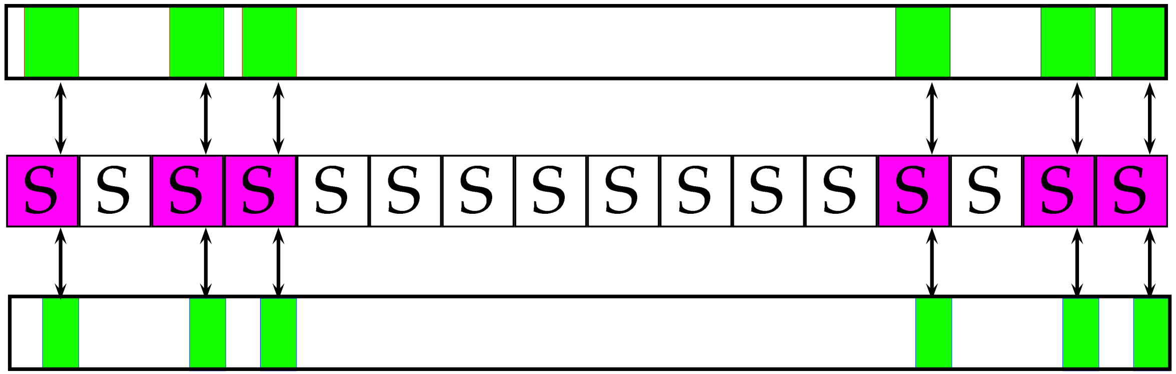
Sieving through the Sboxes: 1 Sbox

$x_3x_2x_1x_0$	$S(x)_3S(x)_2S(x)_1S(x)_0$
0000	1100
0001	0101
0010	0110
0011	1011
0100	1001
0101	0000
0110	1010
0111	1101
1000	0011
1001	1110
1010	1111
1011	1000
1100	0100
1101	0111
1110	0001
1111	0010

$x_2x_1x_0 \rightarrow_S y_1y_0$
000→00
000→11
001→01
001→10
010→10
010→11
011→00
011→11
100→00
100→01
101→00
101→11
110→01
110→10
111→01
111→10

16 values of x_2, x_1, x_0, y_1, y_0, out of 32, correspond to a valid transition.

Sieving through the Sboxes



- Probability for 1 Sbox $p = 16/32 = 1/2$
- Probability for the 6 Sboxes: $\frac{1}{2^6}$
- We only try $2^{80-6} = 2^{74}$ potential key candidates.
- 7 rounds (+1 bicliques).

Importance of Dedicated Cryptanalysis

Lightweight Dedicated Analysis

- Few cases broken by well known attacks (ex. Puffin or Puffin2 - multiple differentials)
- Happily, this is rare. Most of the times, new families or new ideas on known attacks exploiting the new properties are needed.
- Lightweight: more 'risky' design, lower security margin, simpler components.
- Often innovative constructions: dedicated attacks

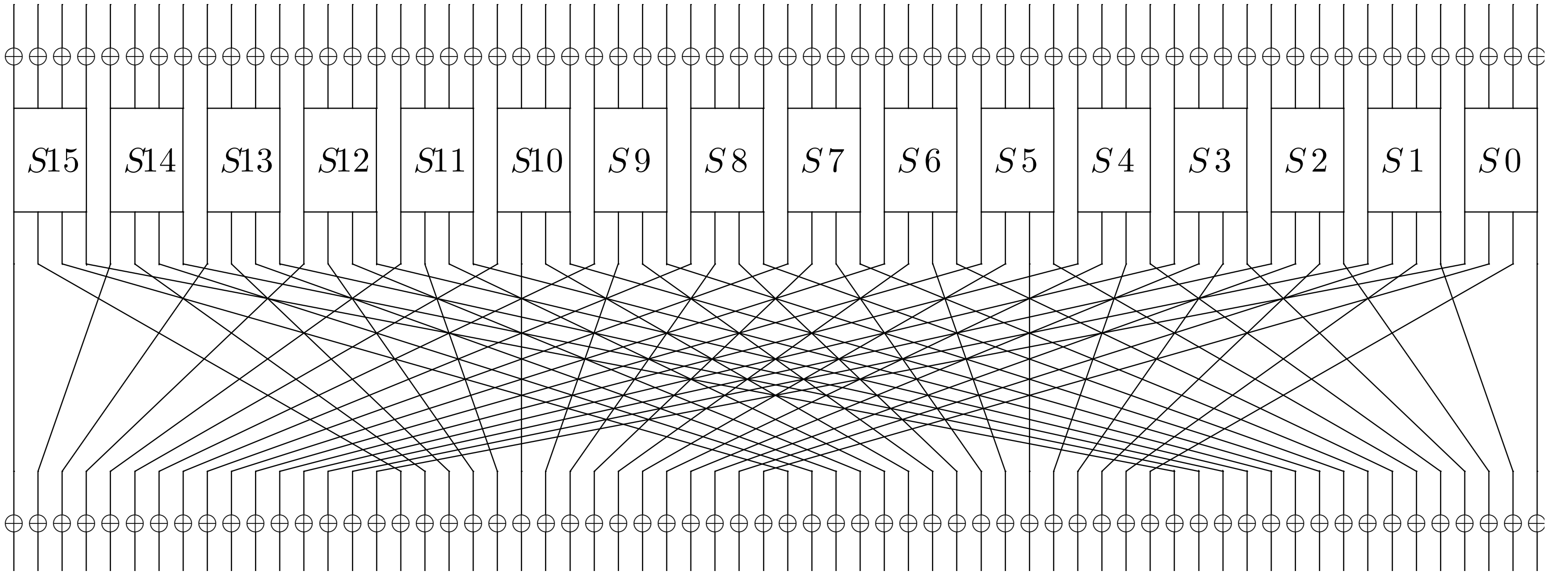
Ex: PRESENT and PRINTcipher

PRESENT [BKLPPRSV'07]

- One of the most popular ciphers, proposed in 2007, and now ISO/IEC standard.
- Very large number of analysis published (20+).
- Best attacks so far: multiple linear attacks (27r/31r).

PRESENT

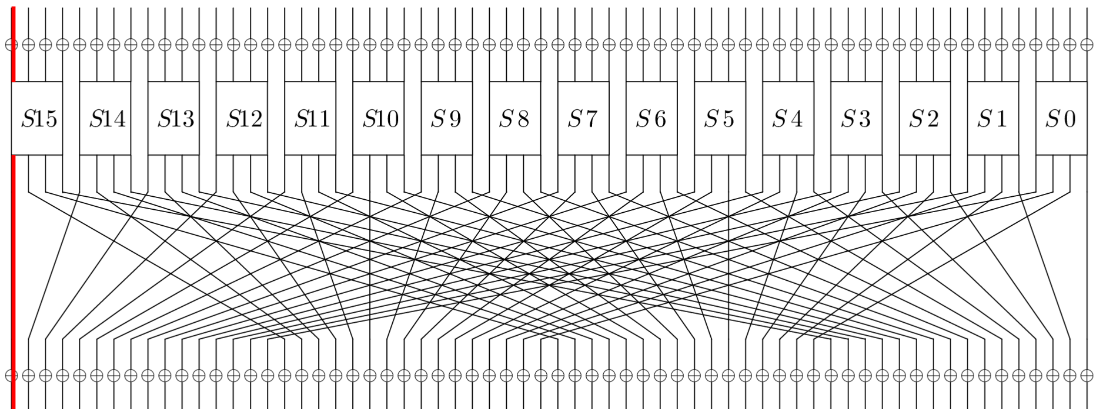
Block $n = 64$ bits, key 80 or 128 bits.



31 rounds $+$ 1 key addition.

PRESENT

Linear cyptanalysis: because of the Sbox, a linear approximation 1 to 1 with bias 2^{-3} per round [O-09].

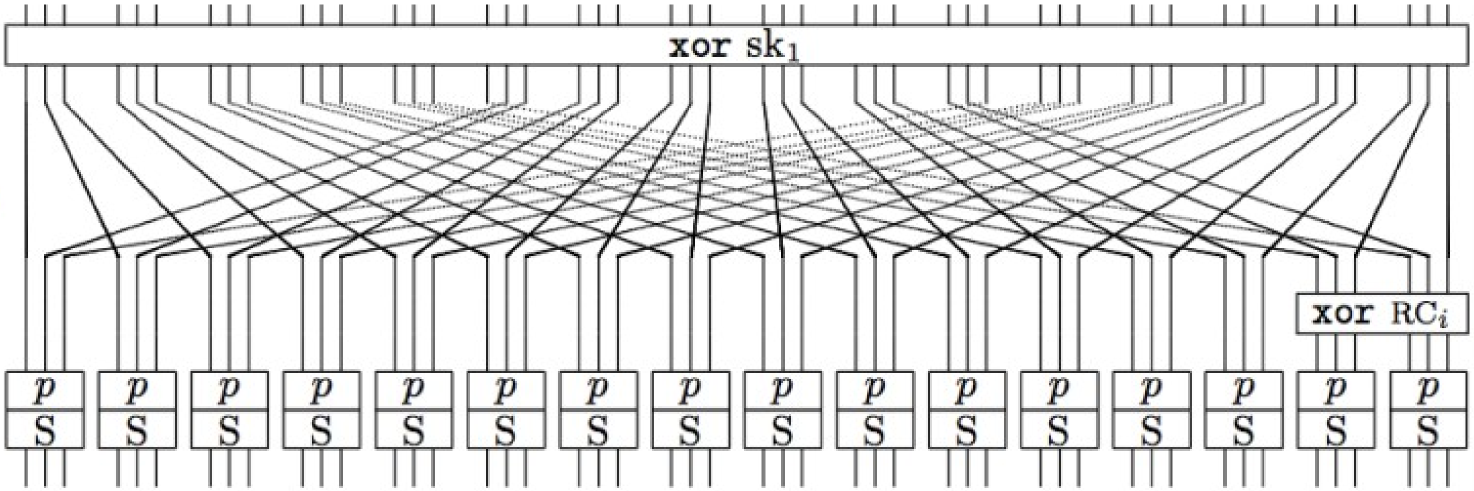


- Multiple linear attacks: consider several possible approxs simultaneously $\Rightarrow$ up to 27 rounds out of 31 [BN-14].

PRINTcipher

- Many PRESENT-like ciphers proposed, like Puffin, PRINTcipher
- Usually, weaker than the original.
- PRINTcipher[KLPR'10]: first cryptanalysis: invariant subspace attack[LAAZ'11].

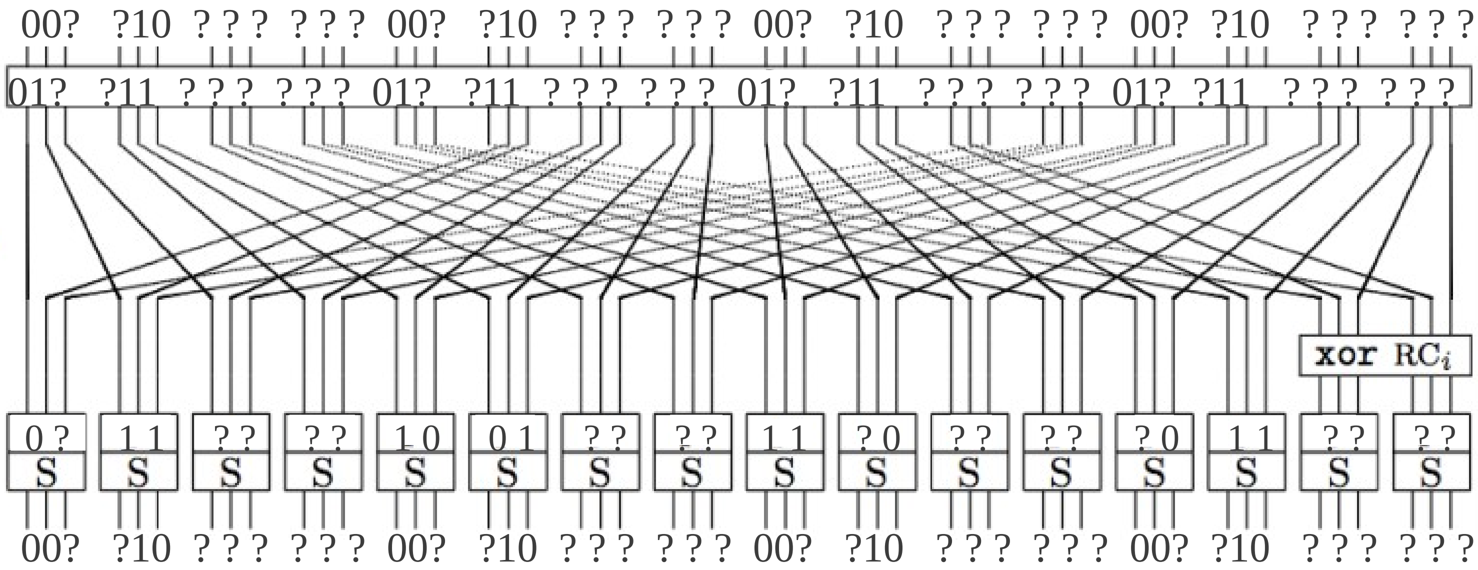
PRINTcipher



48 rounds.

The Invariant Subspace Attack [LAAZ'11]

With probability 1:



- Weak key attack, but a very bad property for 2^{51} keys...

The Invariant Subspace Attack

- More applications afterwards:
  iScream, Robin, Zorro, Midori.

- Importance of generalizing/understanding
  dedicated attacks:
  new families/techniques might appear.

Final remarks

Zorro - Hash Functions links

- Lightweight block cipher proposed [GGN-PS13] for easy masking.
- A modified AES with only four sboxes per round (SPN with partial non-linear layer).
- Bounds on number of active Sboxes? Computed using freedom degrees.
- Many analyses published. Problem: MC property
  $\Rightarrow$ devastating attack [BDDLT13, RASA13]

LED - Hash Functions links

- Lightweight block cipher proposed in [GPPR12].
- AES-like with simpler key-schedule and more rounds. Nice simple design.
- Analysis provided with respect to known key distinguishers (rebound-like). Seems like a lot of SHA-3 knowledge put into this design.

Hash functions links - Sum up

- Mitm, bicliques/initial structures:
  used for both scenarios
- Early abort $\leftarrow$ message modification techniques
- State-test tech. & choosing $\Delta_{in,out} \leftarrow$ Rebound attacks
- Mult. impos. diff. $\leftarrow$ mult. limited birthday distinguishers
- Using freedom degrees for bounds?... be careful!!
- Merging lists from rebounds/sieve in the middle
  $\rightarrow$ many applications
- *Other ex: AES distinguishers inspired on rebound attacks.*

Conclusion

To Sum Up

- Classical attacks, but also new dedicated ones exploiting the originality of the designs.
- Importance on generalizing: improvements, and dedicated might become well stablished techniques.
- Importance of reduced-round analysis to re-think security margin, or as first steps of further analysis.
- New ideas inspired by SHA-3: might help improving attacks further!
- Better identifying composite problems/ list merging situations might provide improved results.

To Sum Up[3]

A lot of ciphers to analyze/ a lot of work to do!

[3] Thank you to Christina Boura and Leo Perrin for their help with the figures and the slides.